Generating Risky Samples with Conformity Constraints via Diffusion Models

Han Yu[1], Hao Zou[1], Xingxuan Zhang[1], Zhengyi Wang[1], Yue He[2], Kehan Li[1], Peng Cui[1]*

[1]Tsinghua University
[2]Renmin University of China
yuh21@mails.tsinghua.edu.cn, ahio@163.com, xingxuanzhang@hotmail.com, wang-zy21@mails.tsinghua.edu.cn
hy865865@gmail.com, lkh20@mails.tsinghua.edu.cn, cuip@tsinghua.edu.cn

Abstract

Although neural networks achieve promising performance in many tasks, they may still fail when encountering some examples and bring about risks to applications. To discover risky samples, previous literature attempts to search for patterns of risky samples within existing datasets or inject perturbation into them. Yet in this way the diversity of risky samples is limited by the coverage of existing datasets. To overcome this limitation, recent works adopt diffusion models to produce new risky samples beyond the coverage of existing datasets. However, these methods struggle in the conformity between generated samples and expected categories, which could introduce label noise and severely limit their effectiveness in applications. To address this issue, we propose *RiskyDiff* that incorporates the embeddings of both texts and images as implicit constraints of category conformity. We also design a conformity score to further explicitly strengthen the category conformity, as well as introduce the mechanisms of embedding screening and risky gradient guidance to boost the risk of generated samples. Extensive experiments reveal that RiskyDiff greatly outperforms existing methods in terms of the degree of risk, generation quality, and conformity with conditioned categories. We also empirically show the generalization ability of the models can be enhanced by augmenting training data with generated samples of high conformity.

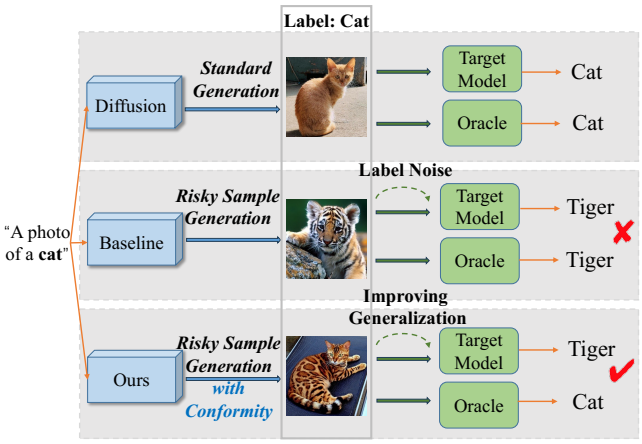

Figure 1: Illustration of risky sample generation with conformity constraints. We require generated samples to deceive the target model but conform to the conditioned category, so that generated samples could improve generalization of the target model after being added to training data. Otherwise, there could be severe label noise in generated samples.

Introduction

Deep learning have exhibited impressive power and achieved promising performance (Dosovitskiy 2020; Tian, Ye, and Doermann 2025) across various applications, where computer vision is a representative. However, models are still susceptible to some risky samples when encountering high-stake applications, such as health care (Irvin et al. 2019; Feuerriegel et al. 2024) and autonomous driving (Lang et al. 2019; Li et al. 2022). Thus discovering these risky samples is crucial to unveil the vulnerability of models and enhance their performance by augmenting the datasets.

Previously, a series of works named error slice discovery (d'Eon et al. 2022; Eyuboglu et al. 2022) propose to find coherent patterns of risky samples from an existing dataset. Besides, adversarial attack (Szegedy et al. 2014; Fletcher 2000; Goodfellow, Shlens, and Szegedy 2015; Madry 2017) aims to inject perturbations into existing data to fool the models. Nevertheless, these solutions are only capable of finding risky samples within or close to the coverage of existing data, which severely limits the diversity of risky samples. It is of urgent need to directly generate risky samples beyond the finite empirical data.

Recent development of diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2021; Dhariwal and Nichol 2021; Ho and Salimans 2022) presents an opportunity to achieve this target via image generation. Some works (Chen et al. 2023; Dai, Liang, and Xiao 2024) try to take advantage of diffusion models to directly generate risky samples beyond the coverage of existing reference datasets, but the ground-truth category labels are not guaranteed to keep unchanged during the generation process. In the standard conditional generation process via advanced diffusion models, the conformity between the generated samples and their expected category labels could mostly be guaranteed for common objects. However, when attempting to generate risky samples, the image generation process is guided to-

*Corresponding author

wards difficult or rare cases, where diffusion models might not be guaranteed to keep the expected category label unchanged. This could lead to failure of making generated samples conform to the conditioned category. Such nonconformity introduces label noise into generated samples, which could severely limit their application in customized generation demand and hinder their usefulness in augmenting original datasets to improve model performance.

To mitigate the above issues, we leverage diffusion models as the generative backbone and propose **RiskyDiff** that utilizes both image and text embeddings as implicit constraints of category conformity. We also introduce guidance of an extra conformity score that serves as an explicit constraint of conformity. Meanwhile, we introduce mechanisms of embedding screening and risky gradient guidance to increase the risk of generated samples, i.e. the proportion of generated samples that the classification model makes mistakes on. Extensive experiments on various image classification datasets validate the effectiveness of our method. The results confirm that RiskyDiff outperforms existing baselines in terms of the degree of risk, generation quality, and conformity to the conditioned categories. We also demonstrate that the generalization performance of target models can be improved by augmenting original training datasets with generated risky samples of high conformity. Our contributions are summarized below:

- We incorporate both text and image embeddings as input conditions, as well as a conformity score, to encourage category conformity when generating risky samples via diffusion models.
- We introduce the mechanisms of embedding screening and risky gradient guidance to promote the risk of the generated samples with respect to target models.
- Experiments show that our method contributes to higher conformity to expected categories, along with higher risk and generation quality. By adding generated high conformity samples to the training datasets, the target models accomplish improved performance of generalization.

Related Work

Error Slice Discovery Error slice discovery aims to find the interpretable subset of validation data where a model underperforms. They emphasize the coherence of the discovered risky samples. Spotlight (d'Eon et al. 2022) aims to learn a sphere as the risky region. Domino (Eyuboglu et al. 2022) adapts the Gaussian mixture algorithm to an error-aware version. InfEmbed (Wang et al. 2023) uses influence functions as representations of test samples before clustering. PlaneSpot (Plumb et al. 2023) adopts a combination of the dimension-reduced representation and the prediction probability before clustering.

Adversarial attack Adversarial attack (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015; Miyato et al. 2018; Kurakin, Goodfellow, and Bengio 2016) adds a perturbation with a small norm to the image and deceive deep learning models. When model parameters and gradients are not available, black-box attacks (Dong et al. 2021; Cheng et al. 2024; Ilyas, Engstrom, and Madry 2019) are investigated. To generate adversarial examples beyond the ℓ_p norm constraints, unrestricted adversarial attacks typically operate on the semantic content of the image, such as color, shape, and texture (Alaifari, Alberti, and Gauksson 2019; Bhattad et al. 2020; Shamsabadi, Sanchez-Matilla, and Cavallaro 2020). However, overdependence on reference images limits the coverage of potential adversarial examples.

Diffusion models Ho, Jain, and Abbeel (2020) firstly demonstrate the impressive capability of diffusion models (Sohl-Dickstein et al. 2015) in generating images of high quality. It reverses a diffusion process to recover images by denoising noisy data. To accelerate the generation process, denoising diffusion implicit models (DDIM) (Song, Meng, and Ermon 2021) is proposed to reduce the required inference steps and improve sampling efficiency. Dhariwal and Nichol (2021) modify the model architecture to improve generation quality and introduce gradients of extra classifiers into the sampling process to achieve conditional generation. To eliminate the requirement on extra classifiers, classifier-free guidance (Ho and Salimans 2022) is proposed. To strengthen the flexibility in controlling the content, Stable Diffusion (Rombach et al. 2022) and Stable-unCLIP (Ramesh et al. 2022) are trained to produce images with user-specified text prompts.

Method

Preliminaries

Throughout the paper we use $[N]$ to represent $\{1, 2, ..., N\}$. Let $f : \mathcal{X} \mapsto \mathcal{Y}$ be an image classification model, where $\mathcal{Y}$ is the category label space.

Diffusion models The forward process of diffusion is a Markov chain with the transition probability function as $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$. A nice property of it is $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t = \Pi_{i=1}^{t}\alpha_i$. The objective for a diffusion generative model is to simulate the backward diffusion process, i.e. sampling $\boldsymbol{x}_{t-1}$ based on $\boldsymbol{x}_t$. Thus a noise predictor is trained by minimizing the loss $\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),t\in[T]}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2$ (Ho, Jain, and Abbeel 2020). Given the trained noise predictor, the sampling process of DDIM (Song, Meng, and Ermon 2021) is

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \tag{1}$$

For conditional diffusion models, the most common practice is classifier-free guidance (Ho and Salimans 2021), where the noise predictor is equipped and trained with an additional input condition $\boldsymbol{c}$, i.e. $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})$. Here $\boldsymbol{c}$ can be a text embedding for Stable Diffusion (Rombach et al. 2022) or both text and image embeddings for Stable-unCLIP (Ramesh et al. 2022). In recent years, training and sampling of diffusion models are conducted in the latent representation space (Rombach et al. 2022) instead of pixel space via a pair of pretrained encoder and decoder. Thus we use $\boldsymbol{z} \in \mathcal{Z}$ for latent codes and $\boldsymbol{x} \in \mathcal{X}$ for raw images.

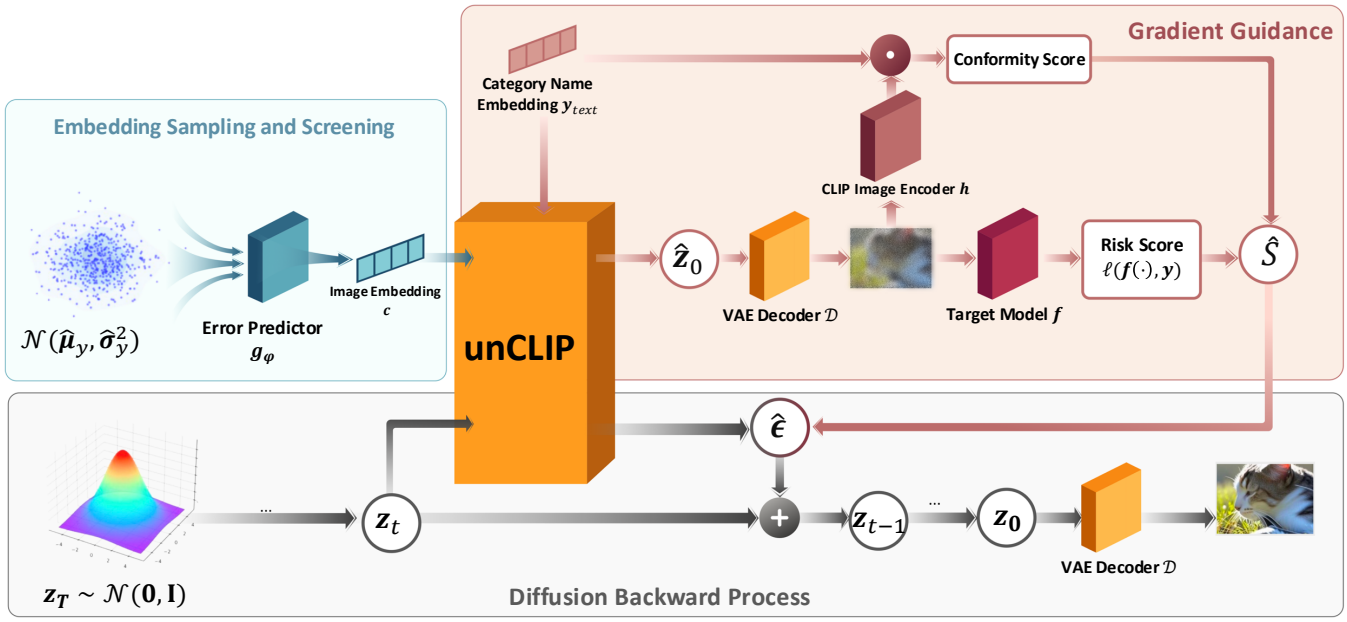

Figure 2: Overall framework of RiskyDiff. The majority of this figure shows a single backward step of the sampling process.

RiskyDiff

Given the advanced generative capability of diffusion models, we seek to leverage them in risky sample generation. Although there are preliminary attempts to employ diffusion models to generate risky samples (Chen et al. 2023; Dai, Liang, and Xiao 2024), their methods lack attention to the conformity between generated samples and the desired category labels. When fixing a specific category y and attempting to generate risky examples, we expect to generate samples that deceive a target model f but still conform to the category y from the view of humans. Previous methods adopt conditional diffusion models $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, y)$ with the desired category label y (category index) as the input condition of the noise predictor, which is the only implicit supervision of conformity. The neglect of such conformity in the design of previous algorithms makes them fail to keep the ground-truth category label unchanged when achieving a high degree of risk for generated samples.

To mitigate this issue and further increase the degree of risk of generated samples, we propose **RiskyDiff**. We adopt Stable-unCLIP $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{y}_{text})$ (Ramesh et al. 2022) that takes not only the desired category text embedding $\boldsymbol{y}_{text}$ but also a proper image embedding $\boldsymbol{c}$ as input conditions of the diffusion noise predictor, serving as implicit supervisions of conformity, and we add an explicit conformity constraint to keep the ground-truth label unchanged. We also design the mechanisms of embedding screening and risky gradient guidance to generate samples with higher risks. The following subsections are detailed descriptions of our algorithm, with a full procedure depicted in Figure 2 and Algorithm 1.

Embedding Sampling and Screening Given a desired category y, for $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{y}_{text})$, in each step of the diffusion sampling process, CLIP (Radford et al. 2021) text embedding of the category name $\boldsymbol{y}_{text}$ is the input condition of the noise predictor, serving as an implicit constraint of conformity. For the other input condition, i.e. CLIP image embedding $\boldsymbol{c}$, we expect it to correspond to an image of the category y so that it serves as another implicit constraint of conformity. We implement this via sampling from an estimated Gaussian distribution of the category. For the target model f of a specific image classification task, its validation data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is often available. Thus we leverage these data to calculate the mean $\hat{\boldsymbol{\mu}}_y$ and dimension-wise variance $\hat{\boldsymbol{\sigma}}_y^2$ of their CLIP embeddings, and use them as the mean and diagonal elements of the covariance matrix for the Gaussian distribution. Since we want to generate risky samples, we design a simple embedding screening procedure to filter for the candidate embeddings that are more likely to generate samples on which f makes mistakes. We fit an MLP as the error predictor $g_{\boldsymbol{\varphi}}$ using CLIP image embeddings of validation data and $e_i = \mathbb{I}(f(\boldsymbol{x}_i) \neq y)$, i.e. whether f makes a mistake on $\boldsymbol{x}_i$. We do not require this error predictor to be very precise since it only serves as screening. Then we repeatedly sample $\boldsymbol{c} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_y, \mathrm{diag}(\hat{\boldsymbol{\sigma}}_y^2))$ until we have collected enough embeddings that are predicted by $g_{\boldsymbol{\varphi}}$ as 1.

Risky Gradient Guidance To fool the classifier f so that it does not predict the generated sample as y, inspired by classifier-guided conditional generation (Dhariwal and Nichol 2021), we incorporate the gradient of f into the sampling process of Stable-unCLIP. Specifically, for step t, we adopt the rough estimate $\hat{\boldsymbol{z}}_0 = \frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{y}_{text})}{\sqrt{\bar{\alpha}_t}}$ and decode it as $\hat{\boldsymbol{x}}$. Then we calculate the risk score $\hat{S} = \ell(f(\hat{\boldsymbol{x}}), y)$ as the loss between the output of f and y. For DDIM, we incorporate the gradient of the risk score into noise prediction following Dhariwal and Nichol (2021):

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{y}_{text}) - s\sqrt{1-\bar{\alpha}_t}\frac{\nabla_{\boldsymbol{z}_t}\hat{S}}{||\nabla_{\boldsymbol{z}_t}\hat{S}||} \quad (2)$$

Algorithm 1: RiskyDiff

Input: Tareget model f, unCLIP $\boldsymbol{\epsilon_\theta}$ along with its image embedder h and image decoder $\mathcal{D}$, validation dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, category name embedding $\boldsymbol{y}_{text}$ and category label y, gradient scale s, conformity coefficient λ.
Calculate image embeddings: $\boldsymbol{c}_i = h(\boldsymbol{x}_i),\ \forall i \in [n]$.
Calculate model errors: $e_i = \mathbb{I}(f(\boldsymbol{x}_i) \neq y_i),\ \forall i \in [n]$.
Fit an error predictor $g_{\boldsymbol{\varphi}}$ with $\{(\boldsymbol{c}_i, e_i)\}_{i=1}^n$.
Calculate $\hat{\boldsymbol{\mu}}_y$ as the mean of $\{\boldsymbol{c}_i | y_i = y, i \in [n]\}$.
Calculate $\hat{\boldsymbol{\sigma}}_y^2$ whose d-th dimension is the variance of $\{c_{i,d} | y_i = y, i \in [n]\}$.
Repeat sampling $\boldsymbol{c} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_y, \mathrm{diag}(\hat{\boldsymbol{\sigma}}_y^2))$ till $g_{\boldsymbol{\varphi}}(\boldsymbol{c}) = 1$.
Sample $\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T$ to 1 **do**
  $\hat{\boldsymbol{x}} = \mathcal{D}(\hat{\boldsymbol{z}}_0) = \mathcal{D}(\frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{y}_{text})}{\sqrt{\bar{\alpha}_t}})$.
  $\hat{S} = \ell(f(\hat{\boldsymbol{x}}), y) + \lambda h(\hat{\boldsymbol{x}}) \cdot \boldsymbol{y}_{text}$.
  $\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{y}_{text}) - s\sqrt{1-\bar{\alpha}_t}\frac{\nabla_{\boldsymbol{z}_t}\hat{S}}{||\nabla_{\boldsymbol{z}_t}\hat{S}||}$.
  $\boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}(\frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}}{\sqrt{\bar{\alpha}_t}}) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}$.
end for
return: $\mathcal{D}(\boldsymbol{z}_0)$.

where s is the scale of gradient guidance. This could guide the sampling process of the diffusion model towards the direction where $\hat{S}$ is larger, i.e. $f(\hat{\boldsymbol{x}})$ is less likely to be y.

Conformity Gradient Guidance To further guarantee the conformity between the generated sample and the conditioned category y, we introduce an explicit constraint into the sampling process of diffusion. In addition to $\ell(f(\hat{\boldsymbol{x}}), y)$, we add the inner product of CLIP image embedding of $\hat{\boldsymbol{x}}$ and the CLIP text embedding of the category name, i.e. $h(\hat{\boldsymbol{x}}) \cdot \boldsymbol{y}_{text}$, to the guidance score. This enhances their conformity in the representation space of CLIP. Thus the score could be rewritten as:

$$\hat{S} = \ell(f(\hat{\boldsymbol{x}}), y) + \lambda h(\hat{\boldsymbol{x}}) \cdot \boldsymbol{y}_{text} \tag{3}$$

where λ is conformity coefficient, h is CLIP image encoder, $\boldsymbol{y}_{text}$ is CLIP text embedding of the category name. Note that since CLIP image and text encoders are both already available during training and sampling of Stable-unCLIP, our adoption of them does not require extra resources.

Experiments

Experimental Settings

Here we provide key details of experimental settings. More details can be found in Appendix.

Datasets We conduct our experiments on four image classification datasets: CIFAR-100 (Krizhevsky, Hinton et al. 2009), ImageNet (Deng et al. 2009), PACS (Li et al. 2017), and NICO++ (Zhang et al. 2023). The former two are well-known datasets in standard image classification, while the latter two are widely used image classification datasets in domain generalization, each with multiple domains or environments, so that we could evaluate both in-distribution (ID) generalization and out-of-distribution (OOD) generalization performance when utilizing generated risky samples to augment training data.

Target models For each dataset, we generate risky examples for four target models. For CIFAR-100, we employ DenseNet-121 (Huang et al. 2017), ResNet-50 (He et al. 2016), WideResNet (Zagoruyko 2016), and GoogleNet (Szegedy et al. 2015), trained using code from a popular github repository[1]. For ImageNet, we directly use DenseNet-121, ResNet-50, EfficientNet-B2 (Tan and Le 2019), and ViT-B/16 (Dosovitskiy 2020) trained on ImageNet provided by torchvision. For PACS and NICO++, we train models of the same backbones as those of ImageNet using code from the domain generalization benchmark DomainBed (Gulrajani and Lopez-Paz 2021). More specifically, for PACS, we train on the domain of art and leave domains of photo, cartoon, and sketch as OOD domains to evaluate OOD generalization performance after augmenting training data with generated risky samples. For NICO++, we train on the domains of autumn, rock, dim, and grass, and leave outdoor and water as OOD domains.

Baselines In our experiments of risky sample generation, we mainly compare with AdvDiffuser (Chen et al. 2023) and AdvDiff (Dai, Liang, and Xiao 2024), which also adopt diffusion models to generate risky samples. For the experiments of generalization enhancement via augmenting training data with generated risky samples, we also compare with the classic data augmentation technique Mixup (Zhang 2017) and straightforward utilization of Stable-unCLIP (Ramesh et al. 2022) as a data augmenter.

Other details For PACS, we set the gradient scale $s = 20$. For the other three datasets, we set $s = 10$. We set the conformity coefficient $\lambda = 1e-4$. For detailed hyperparameter analyses, please refer to Appendix. For hyperparameters of baselines, we directly follow their own settings. As for the number of generated images, for ImageNet we generate 20 images for each category, resulting in a total of 20,000 samples for each method. For CIFAR-100, PACS, and NICO++, we generate 120 images for each category, resulting in 12,000, 840, and 7,200 samples for each method.

Experimental Results

We conduct experiments from multiple perspectives to comprehensively illustrate the superiority of our method.

Degree of risk We compare the degree of risk, measured by the error rate of generated samples, of our method with baselines on four target models for four datasets. From Table 1, we can see that our method significantly increases the degree of risk of generated samples over previous methods across various target models and various datasets. In ImageNet, PACS, and NICO++, our method increases the error rate by a large margin. This provides strong evidence of our method's superior capability in generating risky samples that

[1] https://github.com/weiaicunzai/pytorch-cifar100

Dataset	CIFAR-100				ImageNet			
Target Model	DenseNet-121	ResNet-50	WideResNet	GoogleNet	DenseNet-121	ResNet-50	EfficientNet-B2	ViT-B/16
AdvDiffuser	83.1	77.3	77.5	81.8	51.2	65.1	53.5	68.9
AdvDiff	44.5	44.1	47.2	46.1	15.0	12.5	23.2	11.4
Ours	**83.5**	**78.5**	**85.5**	**83.2**	**68.2**	**71.5**	**70.8**	**71.4**
Dataset	PACS				NICO++			
Target Model	DenseNet-121	ResNet-50	EfficientNet-B2	ViT-B/16	DenseNet-121	ResNet-50	EfficientNet-B2	ViT-B/16
AdvDiffuser	15.1	31.2	15.6	23.4	35.0	38.6	28.0	35.2
AdvDiff	17.3	16.4	14.5	33.0	19.2	16.5	13.9	12.8
Ours	**19.3**	**56.9**	**19.8**	**34.6**	**39.6**	**40.2**	**41.3**	**39.2**

Table 1: Error rate (%) of generated samples on CIFAR-100, ImageNet, PACS, and NICO++ with respect to various target models. Higher is better. We can see that our method RiskyDiff consistently achieves the highest error rate in all settings.

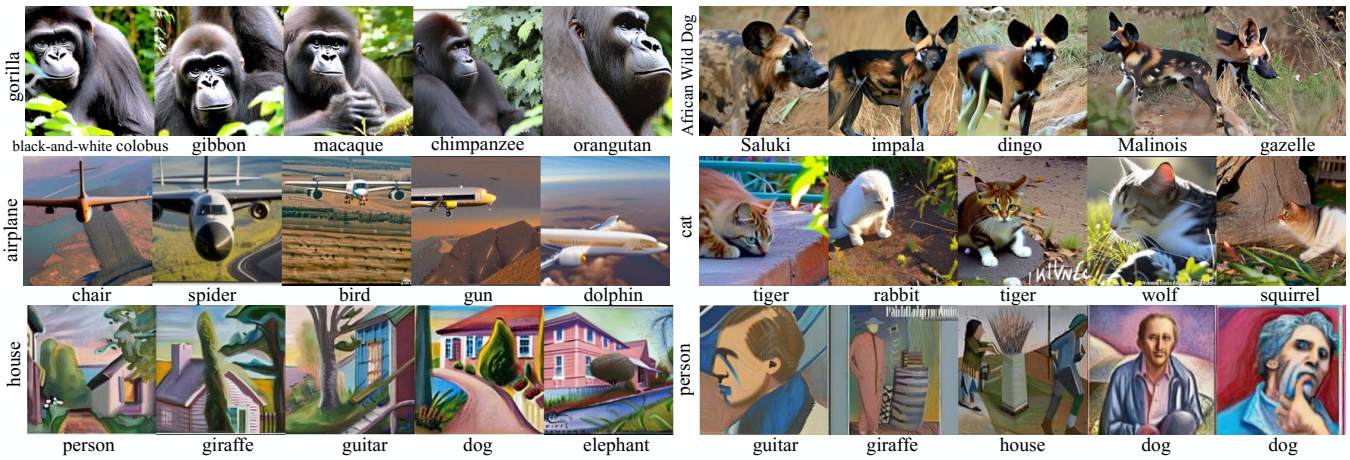

Figure 3: Risky samples generated by RiskyDiff. Three rows correspond to ImageNet, NICO++, and PACS, each including two categories. The left caption implies ground truth category. The lower caption imply prediction of the target model (ResNet-50).

could deceive the given model. We provide examples generated by our method in Figure 3. We can see that many of our generated risky samples are semantically interpretable. E.g., for the category "gorilla", the lighting and shadow in the face of the gorilla in the first image bear resemblance to the black and white furs of "black-and-white colobus". For the category "cat", the color and stripes of the cat resemble those of a tiger, which could be the reason for misclassification. Thus our method could also help explore and interpret possible mistakes that a model might make in prediction. For more examples, please refer to Appendix.

Generation quality Another important aspect of risky sample generation is the quality of generated images. In previous literature (Chen et al. 2023; Dai, Liang, and Xiao 2024), they directly employ FID scores as the evaluation metric that is also commonly adopted in image generation tasks. However, when the distribution of generated images is closer to the original data distribution, a lower FID could also be achieved, while we want the distribution of generated images to be closer to that of the data on which the target model predicts falsely. Therefore, we calculate the FID between generated images and the original error samples. We adopt the implementation of clean-FID (Parmar, Zhang, and Zhu 2022) to calculate FID scores. From Table 3, we can see that our method consistently achieves a lower FID score than baselines in every setting. This indicates that our method is capable of generating images with higher quality.

Dataset	CIFAR-100		
Target Model	DenseNet-121	WideResNet	GoogleNet
AdvDiffuser	44.7	43.9	44.8
AdvDiff	41.1	41.6	42.4
RiskyDiff (Ours)	**62.5**	**62.2**	**61.0**
Dataset	ImageNet		
Target Model	DenseNet-121	EfficientNet-B2	ViT-B/16
AdvDiffuser	32.9	30.6	29.7
AdvDiff	8.1	7.4	6.1
RiskyDiff (Ours)	**42.6**	**40.2**	**39.0**

Table 2: Transferability of generated risky samples. The error rate (%) is measured on other models using samples generated for ResNet-50. Higher is better.

Transferability Although our generation process is conducted with respect to a specific target model, we conduct experiments to verify that our generated risky samples could transfer well to other models. In Table 2, we can see that samples generated by RiskyDiff for ResNet-50 still achieve

Dataset	CIFAR-100				ImageNet			
Target Model	DenseNet-121	ResNet-50	WideResNet	GoogleNet	DenseNet-121	ResNet-50	EfficientNet-B2	ViT-B/16
AdvDiffuser	45.2	45.0	45.2	45.1	26.4	27.1	25.4	23.1
AdvDiff	45.3	45.0	44.1	45.2	13.5	15.7	14.9	13.8
RiskyDiff (Ours)	**33.2**	**32.9**	**34.2**	**33.4**	**10.2**	**12.4**	**10.4**	**11.5**
Dataset	PACS				NICO++			
Target Model	DenseNet-121	ResNet-50	EfficientNet-B2	ViT-B/16	DenseNet-121	ResNet-50	EfficientNet-B2	ViT-B/16
AdvDiffuser	67.4	72.3	78.4	70.5	25.6	25.1	27.2	26.5
AdvDiff	67.8	72.6	78.9	75.4	26.1	25.9	26.1	26.4
RiskyDiff (Ours)	**47.9**	**67.0**	**54.6**	**46.1**	**17.3**	**17.5**	**23.2**	**18.4**

Table 3: FID scores between generated samples and original error samples on CIFAR-100, ImageNet, PACS, and NICO++. Lower is better. Our method consistently achieves the lowest FID in all settings, indicating the highest image generation quality.

a high error rate on other models and outperform those of other methods on CIFAR-100 and ImageNet. More results can be found in Appendix.

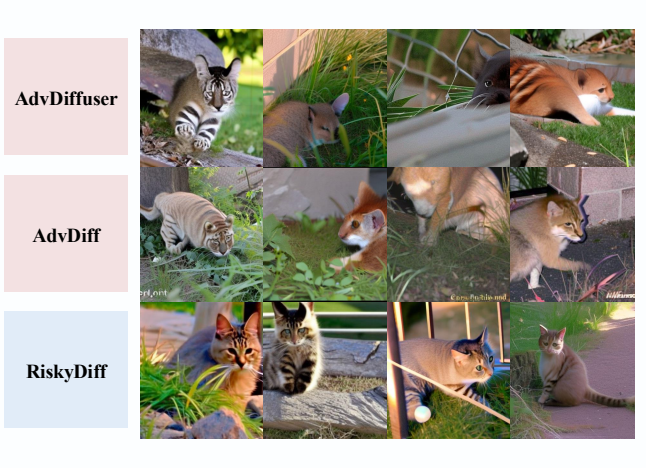

Figure 4: Images of cats generated by AdvDiffuser, AdvDiff, and our method with ResNet-50 as the target model on NICO++. It shows that AdvDiffuser and AdvDiff fail to preserve the true characteristics of cats while our method can.

Generation conformity A fundamental aspect of risky sample generation is the conformity between the generated samples and the expected category labels. In a standard generation process via an advanced diffusion model, the conformity between the generated samples and their expected category labels could mostly be guaranteed. However, when attempting to generate risky samples, the image generation process is guided towards difficult or rare cases, where the diffusion model might not be guaranteed to keep the expected category label unchanged. Thus it is necessary to conduct extra analyses on the generation conformity. From Figure 4, for the category "cat", we can see that some images generated by AdvDiffuser and AdvDiff are not really cats already. For example, the first image of AdvDiffuser looks more like a tiger instead of a cat. Meanwhile, the conformity of our generated samples could also be validated from experiments of generalization enhancement in the next paragraph.

Generalization enhancement A critical but less investigated aspect is how the generated or perturbed samples could be used to further help the target models correct their potential mistakes and enhance their generalization ability on natural examples. Therefore, we retrain the models on CIFAR-100, PACS, and NICO++ by adding the generated data to the original training data as augmentation and evaluate their test performance. Each retraining process is repeated three times with different random seeds. For CIFAR-100, we measure the accuracy on the test set. For domain generalization datasets PACS and NICO++, we measure the ID accuracy on a holdout validation set of the training domains and OOD accuracy on all the other domains. In addition to AdvDiffuser and AdvDiff, we also compare with the classic data augmentation technique Mixup (Zhang 2017) and direct utilization of images generated by Stable-unCLIP (Ramesh et al. 2022).

Model	DenseNet-121	ResNet-50	WideResNet	GoogleNet
Original	79.1	78.4	78.9	77.0
Mixup	$79.5_{\pm 0.2}$	$78.9_{\pm 0.1}$	$79.9_{\pm 0.1}$	$76.9_{\pm 0.1}$
unCLIP	$79.5_{\pm 0.3}$	$79.4_{\pm 0.4}$	$79.3_{\pm 0.2}$	$77.2_{\pm 0.2}$
AdvDiffuser	$79.0_{\pm 0.2}$	$78.8_{\pm 0.1}$	$79.1_{\pm 0.3}$	$76.3_{\pm 0.3}$
AdvDiff	$79.0_{\pm 0.2}$	$78.4_{\pm 0.0}$	$79.2_{\pm 0.3}$	$76.5_{\pm 0.1}$
RiskyDiff (Ours)	$\mathbf{79.8}_{\pm 0.1}$	$\mathbf{79.7}_{\pm 0.0}$	$\mathbf{80.4}_{\pm 0.2}$	$\mathbf{77.7}_{\pm 0.1}$

Table 4: Accuracy of various methods on CIFAR-100. For unCLIP, AdvDiffuser, AdvDiff, and RiskyDiff, we add their generated images to training data and retrain.

From Table 4, we can see that our generated images successfully improve the test accuracy of different models on CIFAR-100 compared with the original models, and our method outperforms baselines of risky sample generation and other data augmentation techniques in terms of the performance after retraining. We also notice that images generated by AdvDiffuser and AdvDiff could decrease the performance of the original model under some settings. This serves as additional evidence that the nonconformity between images generated by these methods and their expected category labels is more severe than ours, thus bringing stronger label noise and hurting the model performance. From Table 5 and 6, we can see that our generated images increase both ID and OOD generalization performance by a large margin compared with the original models, which demonstrates the enhancement of both ID and OOD generalization abilities brought by our generated images, and our

Model	DenseNet-121		ResNet-50		EfficientNet-B2		ViT-B/16	
Method	ID Acc.	OOD Acc.	ID Acc.	OOD Acc.	ID Acc.	OOD Acc.	ID Acc.	OOD Acc.
Original	94.7	73.6	93.7	77.7	96.0	76.3	94.4	72.3
Mixup	$94.6_{\pm 0.3}$	$71.1_{\pm 1.7}$	$95.2_{\pm 0.4}$	$72.4_{\pm 1.4}$	$96.1_{\pm 0.4}$	$73.8_{\pm 3.5}$	$96.5_{\pm 0.3}$	$74.2_{\pm 0.7}$
unCLIP	$94.8_{\pm 0.3}$	$73.6_{\pm 1.8}$	$94.3_{\pm 0.6}$	$74.4_{\pm 1.1}$	$95.6_{\pm 0.0}$	$78.3_{\pm 0.7}$	$95.8_{\pm 0.3}$	$73.6_{\pm 2.8}$
AdvDiffuser	$94.4_{\pm 0.2}$	$75.3_{\pm 2.9}$	$94.5_{\pm 0.6}$	$74.2_{\pm 2.2}$	$95.7_{\pm 0.3}$	$78.5_{\pm 1.1}$	$95.1_{\pm 0.4}$	$73.9_{\pm 3.8}$
AdvDiff	$93.8_{\pm 0.4}$	$70.8_{\pm 0.4}$	$94.5_{\pm 0.2}$	$74.4_{\pm 0.7}$	$95.8_{\pm 0.2}$	$78.3_{\pm 1.0}$	$95.7_{\pm 0.1}$	$75.0_{\pm 0.4}$
RiskyDiff (Ours)	$\mathbf{96.0}_{\pm 0.1}$	$\mathbf{77.8}_{\pm 0.8}$	$\mathbf{95.8}_{\pm 0.1}$	$\mathbf{81.4}_{\pm 1.0}$	$\mathbf{96.5}_{\pm 0.1}$	$\mathbf{82.8}_{\pm 0.5}$	$\mathbf{96.7}_{\pm 0.1}$	$\mathbf{79.1}_{\pm 0.1}$

Table 5: ID and OOD accuracy of various methods on PACS. For unCLIP, AdvDiffuser, AdvDiff, and RiskyDiff, it implies that we add their generated images to the training data and retrain.

Model	DenseNet-121		ResNet-50		EfficientNet-B2		ViT-B/16	
Method	ID Acc.	OOD Acc.	ID Acc.	OOD Acc.	ID Acc.	OOD Acc.	ID Acc.	OOD Acc.
Original	82.5	69.0	82.9	69.6	85.8	73.1	86.0	74.3
Mixup	$84.1_{\pm 0.4}$	$71.5_{\pm 0.4}$	$84.2_{\pm 0.1}$	$70.7_{\pm 0.2}$	$86.8_{\pm 0.1}$	$75.0_{\pm 0.2}$	$87.0_{\pm 0.1}$	$74.5_{\pm 0.7}$
unCLIP	$82.9_{\pm 0.2}$	$69.6_{\pm 0.2}$	$83.3_{\pm 0.1}$	$70.0_{\pm 0.2}$	$86.2_{\pm 0.1}$	$73.4_{\pm 0.1}$	$84.8_{\pm 2.0}$	$73.7_{\pm 0.5}$
AdvDiffuser	$81.4_{\pm 2.1}$	$69.1_{\pm 0.1}$	$81.5_{\pm 2.0}$	$69.4_{\pm 0.3}$	$81.9_{\pm 0.2}$	$73.5_{\pm 0.5}$	$82.3_{\pm 0.3}$	$74.9_{\pm 0.7}$
AdvDiff	$77.9_{\pm 0.2}$	$68.4_{\pm 0.6}$	$80.1_{\pm 2.1}$	$69.7_{\pm 0.4}$	$84.7_{\pm 2.0}$	$73.3_{\pm 0.3}$	$82.1_{\pm 0.1}$	$74.6_{\pm 0.3}$
RiskyDiff (Ours)	$\mathbf{85.9}_{\pm 0.2}$	$\mathbf{72.7}_{\pm 0.0}$	$\mathbf{85.1}_{\pm 0.0}$	$\mathbf{71.5}_{\pm 0.2}$	$\mathbf{88.2}_{\pm 0.0}$	$\mathbf{76.2}_{\pm 0.1}$	$\mathbf{87.8}_{\pm 0.1}$	$\mathbf{75.4}_{\pm 0.4}$

Table 6: ID and OOD accuracy of various methods on NICO++. For unCLIP, AdvDiffuser, AdvDiff, and RiskyDiff, it implies that we add their generated images to the training data and retrain.

method consistently outperforms other methods on various model backbones. Similar to CIFAR-100, we also observe a decrease of performance for AdvDiffuser and AdvDiff compared with the original models across many settings. This further indicates the possible label noise brought by these methods due to the nonconformity between generated examples and category labels, and signifies the higher conformity of our generated examples.

Val. Size	0.1	0.2	0.3	0.5	0.7	0.8	0.9	1
Error rate	39.4	39.6	39.6	39.8	40.2	40.6	41.0	41.2
FID	17.9	17.9	17.7	17.7	17.7	17.5	17.5	17.5

Table 7: Error rate and FID scores for ResNet-50 on NICO++ with varying size of validation data. "Val. Size" represents the used fraction of the original validation data.

Ablation Study To verify the usefulness of different components of our method, we conduct an ablation study of embedding screening and gradient guidance on NICO++ with different target models. Figure 5 shows that each of the individual components could increase the error rate, and the combination outperforms each individual one. Meanwhile, to understand the reliability of validation data, we conduct experiments by reducing the size of validation data gradually to $1/10$ of the original size. From Table 7, we can see that when decreasing validation data size, both error rate and FID change by only a small margin. This indicates that our method is effective even with a small size of validation data.

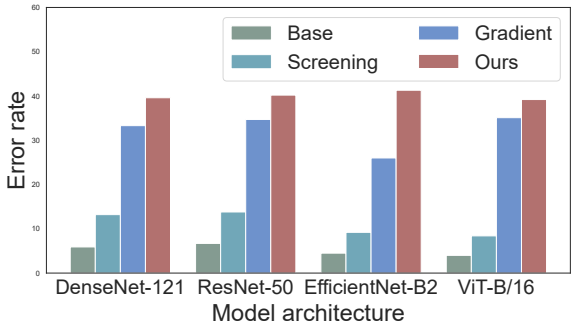

Figure 5: Ablation study of embedding screening and gradient guidance on NICO++. "Base" is direct generating images using Stable-unCLIP. "Screening" is embedding screening. "Gradient" is gradient guidance.

Conclusion

In this paper, to keep conformity between generated samples and expected category labels, we introduce RiskyDiff by integrating both image and text embeddings as implicit constraints of conformity, and propose an explicit conformity constraint via a conformity score. We also design the mechanisms of embedding screening and risky gradient guidance. Experiments demonstrate that RiskyDiff significantly outperforms baselines in terms of error rate, generation quality, and category conformity, and illustrate that the generated risky samples of high conformity could be utilized to enhance the generalization capability of the target models.

Acknowledgements

This work was supported by Tsinghua-Toyota Joint Research Fund, NSFC (No. 62425206, 62141607), and Beijing Municipal Science and Technology Project (No. Z241100004224009). Peng Cui is the corresponding author. All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Alaifari, R.; Alberti, G. S.; and Gauksson, T. 2019. ADef: an Iterative Algorithm to Construct Adversarial Deformations. In *International Conference on Learning Representations*.

Bhattad, A.; Chong, M. J.; Liang, K.; Li, B.; and Forsyth, D. 2020. Unrestricted Adversarial Examples via Semantic Manipulation. In *International Conference on Learning Representations*.

Chen, X.; Gao, X.; Zhao, J.; Ye, K.; and Xu, C.-Z. 2023. Advdiffuser: Natural adversarial example synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4562–4572.

Cheng, S.; Miao, Y.; Dong, Y.; Yang, X.; Gao, X.-S.; and Zhu, J. 2024. Efficient Black-box Adversarial Attacks via Bayesian Optimization Guided by a Function Prior. In *Forty-first International Conference on Machine Learning*.

Dai, X.; Liang, K.; and Xiao, B. 2024. Advdiff: Generating unrestricted adversarial examples using diffusion models. In *European Conference on Computer Vision*, 93–109. Springer.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

d'Eon, G.; d'Eon, J.; Wright, J. R.; and Leyton-Brown, K. 2022. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 1962–1981.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Dong, Y.; Cheng, S.; Pang, T.; Su, H.; and Zhu, J. 2021. Query-efficient black-box adversarial attacks guided by a transfer-based prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 9536–9548.

Dosovitskiy, A. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Eyuboglu, S.; Varma, M.; Saab, K. K.; Delbrouck, J.-B.; Lee-Messer, C.; Dunnmon, J.; Zou, J.; and Re, C. 2022. Domino: Discovering Systematic Errors with Cross-Modal Embeddings. In *International Conference on Learning Representations*.

Feuerriegel, S.; Frauen, D.; Melnychuk, V.; Schweisthal, J.; Hess, K.; Curth, A.; Bauer, S.; Kilbertus, N.; Kohane, I. S.; and van der Schaar, M. 2024. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 30(4): 958–968.

Fletcher, R. 2000. *Practical methods of optimization*. John Wiley & Sons.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Gulrajani, I.; and Lopez-Paz, D. 2021. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Ilyas, A.; Engstrom, L.; and Madry, A. 2019. Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors. In *International Conference on Learning Representations*.

Irvin, J.; Rajpurkar, P.; Ko, M.; Yu, Y.; Ciurea-Ilcus, S.; Chute, C.; Marklund, H.; Haghgoo, B.; Ball, R.; Shpanskaya, K.; et al. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 590–597.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Li, K.; Chen, K.; Wang, H.; Hong, L.; Ye, C.; Han, J.; Chen, Y.; Zhang, W.; Xu, C.; Yeung, D.-Y.; et al. 2022. Coda: A real-world road corner case dataset for object detection in autonomous driving. In *European Conference on Computer Vision*, 406–423. Springer.

Madry, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.

Parmar, G.; Zhang, R.; and Zhu, J.-Y. 2022. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11410–11420.

Plumb, G.; Johnson, N.; Cabrera, A.; and Talwalkar, A. 2023. Towards a More Rigorous Science of Blindspot Discovery in Image Classification Models. *Transactions on Machine Learning Research*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shamsabadi, A. S.; Sanchez-Matilla, R.; and Cavallaro, A. 2020. Colorfool: Semantic adversarial colorization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1151–1160.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Tian, Y.; Ye, Q.; and Doermann, D. 2025. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.

Wang, F.; Adebayo, J.; Tan, S.; Garcia-Olano, D.; and Kokhlikyan, N. 2023. Error Discovery by Clustering Influence Embeddings. *Advances in Neural Information Processing Systems*, 36.

Zagoruyko, S. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, H. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, X.; He, Y.; Xu, R.; Yu, H.; Shen, Z.; and Cui, P. 2023. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16036–16047.