CausalDiff: Causality-Inspired Disentanglement via Diffusion Model for Adversarial Defense

Mingkun Zhang
CAS Key Laboratory of AI Safety
Institute of Computing Technology, CAS
zhangmingkun20z@ict.ac.cn

Keping Bi
Key Laboratory of Network
Data Science and Technology
Institute of Computing Technology, CAS
bikeping@ict.ac.cn

Wei Chen *
CAS Key Laboratory of AI Safety
Institute of Computing Technology, CAS
chenwei2022@ict.ac.cn

Quanrun Chen
School of Statistics University
of International Business and Economics
qchen@uibe.edu.cn

Jiafeng Guo
Key Laboratory of Network
Data Science and Technology
Institute of Computing Technology, CAS
guojiafeng@ict.ac.cn

Xueqi Cheng
CAS Key Laboratory of AI Safety
Institute of Computing Technology, CAS
cxq@ict.ac.cn

Abstract

Despite ongoing efforts to defend neural classifiers from adversarial attacks, they remain vulnerable, especially to unseen attacks. In contrast, humans are difficult to be cheated by subtle manipulations, since we make judgments only based on essential factors. Inspired by this observation, we attempt to model label generation with essential label-causative factors and incorporate label-non-causative factors to assist data generation. For an adversarial example, we aim to discriminate the perturbations as non-causative factors and make predictions only based on the label-causative factors. Concretely, we propose a casual diffusion model (CausalDiff) that adapts diffusion models for conditional data generation and disentangles the two types of casual factors by learning towards a novel casual information bottleneck objective. Empirically, CausalDiff has significantly outperformed state-of-the-art defense methods on various unseen attacks, achieving an average robustness of 86.39% (+4.01%) on CIFAR-10, 56.25% (+3.13%) on CIFAR-100, and 82.62% (+4.93%) on GTSRB (German Traffic Sign Recognition Benchmark). The code is available at https://github.com/CAS-AISafetyBasicResearchGroup/CausalDiff

1 Introduction

Neural classifiers, despite their impressive performance in various applications, are susceptible to adversarial attacks [1, 2], which can deceive them into making erroneous judgments on subtly manipulated examples. Such vulnerabilities pose severe threats in safety-critical scenarios such as face recognition [3, 4] and autonomous driving [5]. There has been extensive work on defending against adversarial attacks such as certified defenses [6, 7], adversarial training [8, 9], and adversarial purification Samangouei et al. [10].

*Corresponding Author.
CAS stands for Chinese Academy of Sciences

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

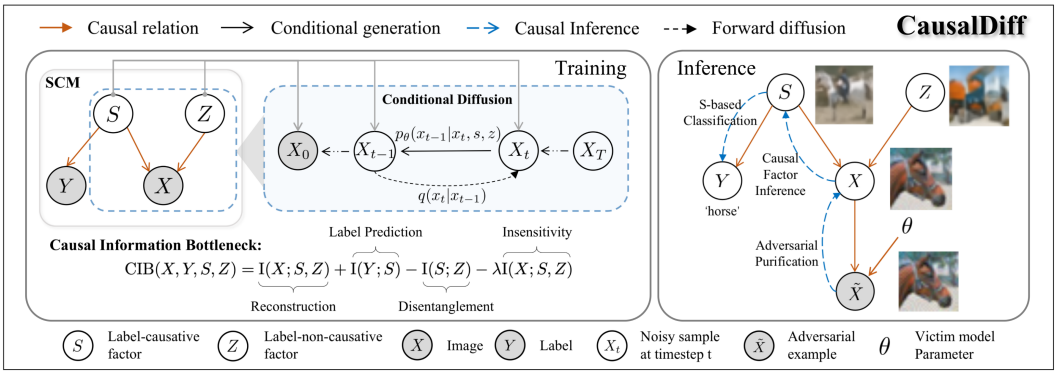


Figure 1: Illustration of training (Left) and inference (Right) processes of our proposed CausalDiff model. During training, the model constructs a structural causal model leveraging a conditional diffusion model, disentangling the (label) Y-causative feature S and the Y-non-causative feature Z through maximization of the Causal Information Bottleneck (CIB). In the inference stage, CausalDiff first purifies an adversarial example $\tilde{X}$, yielded by perturbing X according to the target victim model parameterized by θ, to obtain the benign counterpart X^*. Then, it infers the Y-causative feature S^* for label prediction. We visualize the vectors of S and Z inferred from a perturbed image of a horse using a diffusion model. We find that S captures the general concept of a horse, even when the input image only shows the head, while Z carries information about the horse's skin color.

Although effective, these methods have some limitations. Certified defense methods have limited practicality due to the small certified region that can theoretically guarantee robustness. Adversarial training approaches suffer from a significant decline in robustness against unseen attacks since they take effect by adding adversarial examples into the training set. Purification methods, not designed for specific attacks, struggle to determine the optimal denoising level for unforeseen attacks with differing degrees of perturbation. Consequently, they face challenges in effectively defending against unseen attacks.

In reality, attack behaviors are often unpredictable. Is there a way of strengthening a model to act like humans, i.e., be insensitive to subtle perturbations and robust against various unforeseen attacks? Given an image of an object, we typically identify the key visual features that are necessary to determine its category and disregard other factors such as styles, backgrounds, details, or perturbations. This allows us to make robust judgments. Inspired by this human decision-making process, we would like to learn a model that can disentangle the essential features for determining the category from other non-essential ones.

It is natural to treat the essential features as the causal factors of the label, and both essential and the other features as the causal factors of the entire image. Then, we can learn to disentangle them by modeling the process of data generation and label prediction with a structural causal model (SCM) [11, 12](shown in Fig. 1 (Left)). According to the theoretical results provided by Liu et al. [13], the identifiability of such SCMs can be guaranteed under mild conditions. Although similar SCMs have also been employed in modeling the generation of multi-domain data [14] and adversarial examples [15, 16], they either aim to enhance out-of-distribution robustness or protect the model from a certain type of attack. In contrast, our research focuses on modeling the generation of native in-domain data to enhance adversarial robustness against various unseen attacks.

Fig. 1 (Left) depicts our SCM, where Y denotes the category (e.g., horse) of an input image X; S denotes the essential semantic features of determining Y (i.e., Y-causative factors), such as the characteristics of eyes, ears, nose, mouth, etc. of horses; and Z represents the other features (i.e., Y-non-causative factors) that are not needed to predict Y but are important to generate X, such as the fur color and the image background. Given an adversarial example produced by an unknown attack, our model aims to disentangle its Y-causative features S from the spurious factors in Z and make a robust prediction. To successfully learn the disentanglement, our SCM is guided by the tasks of data generation and label prediction, where there are three major challenges:

1) How do we model the conditional generation of X given S and Z effectively? To this end, we employ a well-recognized diffusion model with state-of-the-art (SOTA) generative performance and

efficiency, i.e., the Denoising Diffusion Probabilistic Model (DDPM) [17], as the backbone. We further adapt it for conditional generation from latent variables rather than random noise. 2) What training objective should we use to effectively learn the disentanglement of S and Z? With this regard, we propose a Causal Information Bottleneck (CIB) optimization objective. CIB aligns the information in the latent variables (S, Z) with observed variables (X, Y) with a bottleneck set by the mutual information (MI) between S, Z and X. The derived function will minimize the MI between S and Z while learning the other causal relations, ensuring their disentanglement within the causal framework. 3) Given an adversarial example, what inference strategies shall we adopt to make robust predictions? As shown in Fig. 1 (Right), according to how the adversarial example $\tilde{X}$ is generated, we first purify it to yield a benign example X^* and then infer the Y-causative factor S based on X^* for final classification. We name the entire causal defense framework based on diffusion models as CausalDiff.

The experimental results on facing various unseen attacks, encompassing both black-box and white-box ones, show that CausalDiff has superior performance compared to representative adversarial defense baselines including state-of-the-art (SOTA) methods. Specifically, our CausalDiff achieves robustness of 86.39% (+4.01%) on CIFAR10, 56.25% (+3.13%) on CIFAR-100 [18], and 81.79% (+4.93%) on GTSRB [19].

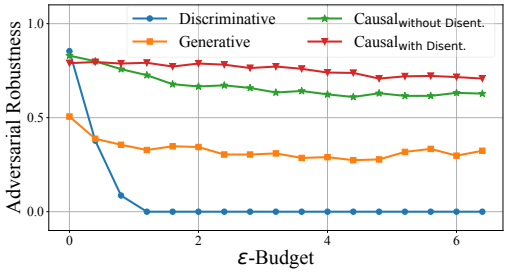


Figure 2: Adversarial robustness of four models against 100-step PGD attack under varying attack strength indicated by ϵ-budget.

Table 1: The experimental results of four models on toy data. The variation of latent v and logits $p(y|v)$ is measured between clean and adversarial examples. The model margin is estimated by the minimal adversarial perturbation required to flip the label, employing both ℓ_2 and ℓ_∞ norm.

Model	$\triangle\ v \downarrow$	$\triangle\ p(y\|v) \downarrow$	margin $\uparrow$ (ℓ_2 / ℓ_∞)
Discriminative	1.15	0.81	2.14 / 0.38
Generative	**0.06**	0.27	1.12 / 0.24
$\text{Causal}_{\text{w/o Disent.}}$	0.29	0.32	9.58 / 5.30
$\text{Causal}_{\text{w/ Disent.}}$	0.27	**0.22**	**10.64 / 6.28**

In summary, we highlight our contributions as follows: 1) We propose a novel causal diffusion framework (CausalDiff) to defend against unseen attacks by modeling the generation of native in-domain data with the category(Y)-causative factors and the other Y-non-causal factors; 2) We propose a Causal Information Bottleneck (CIB) objective to disentangle Y-causative from Y-non-causative factors during causal model training and an associated inference algorithm for adversarial defense; 3) CausalDiff significantly outperforms SOTA methods in defending against various unseen attacks.

2 Related Work

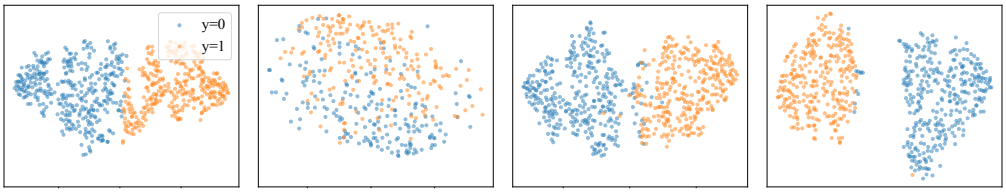
Adversarial Defense. Adversarial training primarily focuses on optimizing the training algorithm [8, 20, 21], incorporating data augmentation [22, 23, 9], and enhancing acceleration [24, 25]. Despite its effectiveness, the trained models could still be vulnerable to unseen attacks [26, 27]. Adversarial purification, orthogonal to our work, utilizes a generative model to purify adversarial noise from examples before classification. Leveraging diffusion models [28, 17, 29, 30], diffusion-based purification have shown to be effective [31–35, 27].

Causal Learning for Robustness. Causal representation learning [12, 36, 14, 37] focuses on discovering invariant mechanisms within structural causal models and has achieved remarkable performance in improving model transferability [14, 13, 38–40] and interpretability [41, 42]. Moreover, in terms of adversarial robustness, researchers [43, 15, 44, 45, 16, 46–48] have attempted to model the attack behaviors in causal structures to identify the adversarial factors. However, modeling particular attack types will limit the model robustness on other types of attacks [27].

Conditional Diffusion Model. Diffusion models [28, 17, 29, 30, 49] have achieved compelling image generation capabilities. To equip them with the ability of controllable generation, the sampling

process can be guided by 1) classifier confidence to generate images of a certain category [30, 50, 51], and 2) semantic embedding of text content or a certain style [52–55].

3 Pilot Study on Toy Data



(a) Discriminative (b) Generative (c) Causal$_{\text{without Disent.}}$ (d) Causal$_{\text{with Disent.}}$

Figure 3: Visualizations of feature space for the two categories on toy data by T-SNE for (a) discriminative model, (b) generative model, (c) causal model without disentanglement, and (d) causal model with disentanglement.

To compare our proposed causal model with other representative models in terms of adversarial robustness, we constructed a toy dataset according to the hypothesis that the essential factors are the basis of generating labels and they also determine the generation of data together with label-non-causative features. Pilot studies on this toy data will provide insights into whether our causal model will work and why it works.

3.1 Experimental Settings

Toy Data Construction. We constructed 2000 samples following the causal structure in Fig. 1 (Left). Specifically, for each data point, we randomly sample the vectors s and z from two different normal distributions, project s to a score y_s with random weights, obtain its label by comparing y_s with the medium score, and generate the representation x by projecting the concatenation $[s; z]$ with another random matrix. Please refer to Appendix B.3 for detailed information.

Models for Comparisons. We investigated four representative models: 1) a *Discriminative* model for classification, 2) a *Generative* model that predicts the label of an adversarial example with $\max_y p(x|y)$ [27], 3) a causal model without disentanglement (*Causal*$_{without\ Disent.}$) that models the generation of both x and y with the same latent factor v, 4) our causal model with disentanglement (*Causal*$_{with\ Disent.}$) that is illustrated in Fig. 1. The concrete structures of the four models and further details are presented in Appendix B.3.

3.2 Experimental Observations

Adversarial Robustness. We evaluate the model's robustness against a 100-step PGD attack with varying ϵ-budgets under the ℓ_∞ bound [8]. Model performance in terms of both clean accuracy (when $\epsilon = 0$) and robust accuracy is examined.

As shown in Fig. 2: 1) The *Discriminative* model exhibits the highest clean accuracy but suffers a rapid decline in robustness, dropping to 0% when the attack budget ϵ reaches 1.2., which highlights its vulnerability. 2) The *Generative* model has the lowest clean accuracy while its robustness does not dramatically regress with larger attack budgets. The low clean accuracy may be due to the inconsistency between the generation process it modeled and the way this toy dataset is constructed. The small gap between clean and robust accuracy indicates its decent effectiveness in defending against adversarial attacks. 3) *Causal*$_{with\ Disent.}$ obtains the second-best clean accuracy, yet its robustness gradually declines with increasing attack strength, maintaining 61.4% robustness at $\epsilon = 6.4$. 4) Our model, *Causal*$_{with\ Disent.}$, has slightly lower clean accuracy than Causal without Disent. but the best robustness among the four. As the attack strength increases, it maintains at least 71.8% robust accuracy at $\epsilon = 6.4$, indicating that it is promising to enhance model robustness by causal modeling with disentanglement of the label-causative factors from the non-causative ones.

Sensitivity to Perturbations. To delve further into how the model behavior changes when defending against adversarial perturbations, we measure the variation between latent variables of clean and adversarial examples, denoted as $\triangle\ v = 1 - cosine(v, v_{\text{adv}})$, where v and v_{adv} represents the latent vector of clean and adversarial example, respectively ($v = s$ for Causal$_{with\ Disent.}$). We also compute the change of predicted logits, $\triangle\ p(y|v) = p(y|v) - p(y|v_{\text{adv}})$, with y being the true class label. Note that a larger variation in the latent factor for prediction, or the predicted logits, results in increased insensitivity to perturbations.

The experimental results, as shown in Tab. 4, indicate that Causal$_{with\ Disent.}$, compared to Causal$_{without\ Disent.}$, exhibits less sensitivity in both latent variables and prediction outcomes. This suggests that by disentangling the label-causative factor s, it becomes more challenging for attackers to perturb the model and alter its prediction. The small variation of v in the Generative model is probably attributed to the lack of discriminability in v.

Prediction Margins. To intuitively explain the sensitivity of each model to perturbations, we estimate the minimal adversarial perturbation $\|\delta\|$ under PGD required to flip the correct model prediction y of a sample x. It can be interpreted as the prediction margins in terms of perturbation, denoted as $\text{margin}(x, y) = \min \|\delta\|$, subject to $p(y|x + \delta) < p(\bar{y}|x + \delta)$ [56]. We measure the $\|\delta\|$ under ℓ_2 and ℓ_∞ norms. Additionally, we visualized the latent vector space of each model to intuitively observe the margin of the classification boundary.

As shown in Tab. 4, Causal$_{with\ Disent.}$ has the largest prediction margin. This implies that an attacker would need to add significantly more perturbation to successfully cheat our model. We can draw similar conclusions from Fig. 3, which illustrates the distribution of predictive features extracted by each model for correctly classified clean samples across categories. Again, Causal$_{with\ Disent.}$ has the largest margin between the classes, indicating that it has high confidence in its prediction and increases the cost and difficulty for an attacker to succeed.

4 Causal Diffusion Model

To enhance model robustness on real-world data, in this section, we propose a *Causal Diffusion (CausalDiff)* Modeling approach, that couples our previously studied SCM (shown in Fig. 6(d)) with diffusion models. We will introduce the three major components in CausalDiff: conditional diffusion generation, causal information bottleneck optimization, and adversarial example inference. We take the Denoising Diffusion Probabilistic Model (DDPM) [17] as an instance for illustration and it can be easily adapted to other diffusion models.

4.1 Conditional Diffusion Generation

Standard diffusion models generate images based on random noise which do not apply to the conditional generation of X based on S and Z in our SCM. In standard diffusion models, at each time step t during denoising, a UNet $\epsilon_\theta(x_t, t)$ is employed to decode an image given input x_t. While diffusion models are highly effective in image generation, they lack an explicit decoder component for generating images from latent variables. To handle this, we develop a conditional DDPM using latent variables S and Z controlling the generation process. This approach draws inspiration from both the class-conditional diffusion model [51] as well as the style control mechanism introduced in DiffAE [52]. The output h^t_{out} at each layer of the UNet depends on t and x_t, i.e.,

$$h^t_{\text{out}} = t_{\text{s}} \cdot \text{GroupNorm}(h) + t_{\text{b}}, \tag{1}$$

where h is the feature map of x_t, t_{s} and t_{b} are the scale and bias of timestep t.

Inspired by the class-conditional diffusion model [51] and the style control mechanism in DiffAE [52], we adapt the standard diffusion generation to be conditioned on S and Z in addition to timestep t. Specifically, the UNet becomes $\epsilon_\theta(x_t, t, s, z)$, the final output $h^{t,s,z}_{\text{out}}$ is calculated based on the original h^t_{out}, the hidden state s and z (representing causal factor S and Z) encoded from input x:

$$h^{t,s,z}_{\text{out}} = z_{\text{s}} \cdot h^t_{\text{out}} + s_{\text{b}}, \tag{2}$$

where z_s and s_b are produced from the affine projections of z and s respectively, i.e., $z_{\text{s}} = \text{Affine}_z(z)$ and $s_{\text{b}} = \text{Affine}_s(s)$. As such, the label-causative factor S acts as a bias that can affect the direction of the latent vector and change its semantics, while the label-non-causative factor Z can only scale the latent vector in a similar way to style control.

4.2 Causal Information Bottleneck Optimization

To learn the causal factors S, Z in our SCM and disentangle them, we propose a Causal Information Bottleneck (CIB) optimization objective. It maximizes the mutual information between the latent factors S, Z and the observed data sample (X, Y) with an information bottleneck that constrains the information retained in S, Z with respect to X.

Specifically, to align the information captured in the latent factors with the observed data (X, Y), we maximize the mutual information between them, denoted as $\mathrm{I(X, Y; S, Z)}$, which can be derived as:

$$\mathrm{I}(X, Y; S, Z) = \mathrm{I}(X; S, Z) + \mathrm{I}(Y; S) - \mathrm{I}(S; Z) - \mathrm{I}(X; Y). \tag{3}$$

A detailed proof is presented in Appendix A.1. Among the resultant terms, $\mathrm{I}(X; Y)$ is solely dependent on the observed data, independent of latent variables or the causal model, and thus can be ignored in the learning process. Maximizing $\mathrm{I}(X; S, Z)$ will urge S and Z to capture ample information about X. $\mathrm{I}(S; Y)$ indicates that the Y-causative factor S should be correlated with Y. The term, $-\mathrm{I}(S; Z)$ ensures S and Z to be effectively disentangled.

Existing work on optimizing similar SCMs adapts the Evidence Lower BOund (ELBO) from Variational Autoencoders (VAE) to formulate the causal ELBO objectives [14]. This objective only maximizes the likelihood of (X, Y), and does not consider the latent factors. Consequently, the final optimization goal of causal ELBO differs from $I(X, Y; S, Z)$ in that it does not have $-\mathrm{I}(S; Z)$ in Eq. (3), which is crucial for disentanglement. Further details are discussed in Appendix A.4.

To avoid X contain too many unimportant details, we constrain the mutual information between X and the latent factors S, Z with an information bottleneck I_c. Then, the updated objective becomes:

$$\max \mathrm{I}(X, Y; S, Z), \quad s.t. \mathrm{I}(X; S, Z) \leq \mathrm{I}_c. \tag{4}$$

Employing Lagrange multiplier $\lambda \geq 0$, we formulate our objective as $\max \mathrm{I}(X,Y;S,Z) - \lambda(\mathrm{I}(X; S, Z) - \mathrm{I}_c)$. Since I_c is a constant, it is equal to maximize the **Causal Information Bottleneck** (CIB):

$$\mathrm{CIB}(X, Y, S, Z) = \mathrm{I}(X; S, Z) + \mathrm{I}(Y; S) - \mathrm{I}(S; Z) - \lambda \mathrm{I}(X; S, Z). \tag{5}$$

$I(X; S, Z)$ and $-\lambda I(X; S, Z)$ indicate two opposing optimization directions. Because it is unclear whether $(1 - \lambda)$ should be positive or negative, we approximate these terms via two separate lower bounds instead of combining them.

To maximize the Causal Information Bottleneck (CIB) in Eq. (5), we derive its lower bound as the concrete training loss function. When using the diffusion model ϵ_θ, classifier $f_y(s; \theta)$ (to estimate $p_\theta(y|s)$), and the encoder $f_{s,z}(x; \theta)$ (to estimate $p_\theta(s, z|x)$), the lower bound of CIB is:

$$\mathbb{E}_{p(x,s,z)}[\log p_\theta(x|s, z)] + \mathbb{E}_{p(y,s)}[\log p_\theta(y|s)] - \mathrm{I}^\theta_{\mathrm{CLUB}}(S; Z) - \lambda \mathbb{E}_{p(x)}[\mathcal{D}_{\mathrm{KL}}(p_\theta(s, z|x) \| q(s, z))], \tag{6}$$

where $p_\theta(s|z)$ is a variational distribution to estimate $p(s|z)$ and $\mathrm{I}^\theta_{\mathrm{CLUB}}(S; Z) = \mathbb{E}_{p(s,z)}[\log p_\theta(s|z)] - \mathbb{E}_{p(z)}\mathbb{E}_{p(s)}[\log p_\theta(s|z)]$ represents the Contrastive Log-Ratio Upper Bound (CLUB) of mutual information proposed by Cheng et al. [57]. $p_\theta(x|s, z)$ represents the likelihood estimated by the conditional diffusion model. $\mathcal{D}_{\mathrm{KL}}$ refers to the Kullback-Leibler (KL) divergence [58]; and $q(\cdot)$ denotes the prior distribution of the latent variable, e.g., a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. A detailed proof can be found in Appendix A.2.

Loss Function. Thus, maximizing the lower bound of CIB is equal to minimizing the loss function:

$$\begin{aligned} \mathcal{L}(x, y, s, z; \theta) = \; & \alpha \mathbb{E}_{\epsilon,t} \|\epsilon_\theta(x_t, t, s, z) - \epsilon_t\|_2^2 + \gamma \mathcal{L}_{\mathrm{CE}}(s, y; \theta) \\ & + \eta \mathrm{I}^\theta_{\mathrm{CLUB}}(S; Z) + \lambda \mathcal{D}_{\mathrm{KL}}(p_\theta(s, z|x) \| q(s, z)), \end{aligned} \tag{7}$$

where $\alpha, \lambda, \gamma, \eta$ determine the weighting of each term in the optimization process. A detailed derivation can be found in Appendix A.3.

Algorithm. We pretrain the model with the data reconstruction loss (the first term in Eq. (7)) alone before training the model with the entire loss, so that the model can learn the causal factors, disentanglement, and classification from a decent starting point. For space concern, we illustrate the training process in Appendix B.2 involves the training algorithm (Algorithm 1) and the pretraining algorithm (Algorithm 2).

Table 2: Clean accuracy and adversarial robustness on CIFAR-10 against **StAdv** under ℓ_∞ ($\epsilon = 0.05$) norm bound and **AutoAttack (AA)** under ℓ_2 ($\epsilon = 0.5$) as well as ℓ_∞ ($\epsilon = 8/255$) bound. We calculate the average robustness across three attack methods to evaluate the model's robustness against unseen attacks. We use underlining to highlight the best robustness for each attack method within each defense category, and bold font to denote the state-of-the-art (SOTA) across all methods.

	METHOD	BACKBONE	CLEAN ACC (%)	ROBUST ACC(%) AA l_∞	AA ℓ_2	STADV	AVG
ADV. TRAIN	AT-l_∞ [23]	DDPM	88.87	63.28	64.65	4.88	44.27
	AT-l_2 [23]	DDPM	93.16	49.41	81.05	5.27	45.24
	AT-l_∞ [9]	EDM	93.36	<u>70.90</u>	69.73	2.93	47.85
	AT-l_2 [9]	EDM	95.90	53.32	<u>84.77</u>	5.08	47.72
	CAUSALADV-T [15]	WRN-76-10	83.71	8.76	21.95	75.60	35.44
	CAUSALADV-M [15]	WRN-76-10	70.22	24.36	49.10	48.60	40.69
	DICE [16]	WRN-34-10	82.85	37.51	41.58	<u>82.46</u>	<u>53.85</u>
PURIFY	DIFFPURE [33]	SCORE SDE	87.50	53.12	<u>75.59</u>	12.89	47.20
	LM-DDPM[27]	DDPM	80.47	53.32	63.09	74.22	63.54
	LM-EDM[27]	EDM	87.89	<u>71.68</u>	75.00	<u>87.50</u>	<u>78.06</u>
OTHERS	SBGC [59]	SCORE SDE	95.04	0.00	0.00	0.00	0.00
	CAMA [43]	VAE	32.19	3.38	5.53	27.54	12.15
	RDC [27]	EDM	89.85	<u>75.67</u>	<u>82.03</u>	<u>89.45</u>	<u>82.38</u>
OURS	**CAUSALDIFF**	DDPM	90.23	**83.01**	**86.33**	89.84	**86.39**
	CAUSALDIFF W/O CFI	DDPM	83.20	74.61	75.59	82.23	77.48
	CAUSALDIFF W/O AP	DDPM	91.21	69.14	84.96	**91.21**	81.77

4.3 Adversarial Example Inference

Guided by the causal generation of an adversarial example $\tilde{X}$ according to Fig. 1 (Right), we illustrate the process for robust classification. Following a typical attack paradigm, $\tilde{X}$ is produced by adding an adversarial perturbation to a target clean example X when attacking a model θ. To make a robust prediction on $\tilde{X}$, our robust inference process comprises three steps: 1) purifying $\tilde{X}$ to benign X by the unconditional diffusion model $\epsilon_\theta(x_t, t)$, 2) inferring S and Z from X utilizing the causal model $\epsilon_\theta(x_t, t, s, z)$, and 3) predicting Y based on S using a classifier $f_y(s; \theta)$.

Adversarial Purification (AP). We follow the concept of Likelihood Maximization (LM) [33, 27] to purify the adversarial example $\tilde{X}$ to a benign X^* by maximizing the data log-likelihood $\log p_\theta(x)$:

$$x^* = \arg\max_x \log p_\theta(\tilde{x}). \tag{8}$$

Concretely, we maximize its lower bound utilizing the unconditional diffusion model $\epsilon_\theta(x_t, t)$ trained according to Section C.2. Chen et al. [27] suggest using one random timestep t during each purification iteration while we believe that smaller timesteps should be more effective since they retain more information from the original example. Thus, we limit the random selection to within the first 50 timesteps. This way significantly boosts adversarial robustness, which will be discussed in Section 5.3.

Causal Factor Inference (CFI). In order to infer the causal and non-causal factors S and Z, which can reconstruct the original image X, we optimize the latent variables by maximizing the conditional likelihood $p_\theta(x|s, z)$ employing the trained conditional diffusion model $\epsilon_\theta(x_t, t, s, z)$:

$$s^*, z^* = \arg\max_{s,z} \log p_\theta(x^*|s, z). \tag{9}$$

Similarly to purification, we obtain s^* and z^* by maximizing the lower bound $-\mathbb{E}_{\epsilon,t}[w_t \|\epsilon_\theta(x_t, t, s, z) - \epsilon\|]$ using the conditional diffusion model $\epsilon_\theta(x_t, t, s, z)$. For efficiency concerns, instead of using all the timesteps for estimation as in Chen et al. [27], we sample $N_{\text{purify}} = 5$ timesteps at the same intervals across the entire timesteps.

Latent-S-Based Classification (LSBC). After obtaining s^* according to Eq. (9), we use the trained classifier $f_y(s; \theta)$ to predict label Y:

$$y^* = \arg\max_y \log p_\theta(y|s^*). \tag{10}$$

Table 3: Clean accuracy and adversarial robustness against **AutoAttack (AA)** on **GTSRB (Left)** and **CIFAR-100 (Right)** dataset. We use $\epsilon = 8/255$ as ℓ_∞ and $\epsilon = 0.5$ as ℓ_2 norm bound.

Method	Clean Acc	Robust Acc			
		AA l_∞	AA ℓ_2	Fog	Avg
DOA [61]	76.56	31.25	36.72	68.36	45.44
GTSRB-CNN [3]	93.95	62.30	74.80	65.43	67.51
AT-4 [8]	92.58	74.78	80.47	78.13	77.69
AT-8 [8]	91.21	74.02	79.10	73.44	75.52
AT-16 [8]	89.65	73.24	75.59	69.92	72.92
CausalDiff	97.85	**80.86**	**80.86**	**86.13**	**82.62**

Method	Clean Acc	Robust Acc
WRN40-2	78.13	0.00
AT-DDPM [23]	63.56	34.64
AT-EDM [9]	75.22	42.67
DiffPure [33]	39.06	7.81
DC [27]	79.69	39.06
RDC [27]	80.47	53.12
CausalDiff	65.62	**56.25**

Within our inference pipeline, both adversarial purification and causal factor inference leverage the diffusion model learned toward the CIB optimization objective while they take effect independently. When we combine these two approaches, adversarial robustness could be enhanced further. The concrete inference algorithm in Algorithm 3 is presented as Appendix B.2.

4.4 Comparison with Adversarial Purification

First, **our CausalDiff can be viewed as semantic-level purification**. Instead of pixel-level denoising, CausalDiff purifies an image in the latent space, trying to remove the effect of perturbation by putting it to the label-non-causative features. Second, conventional purification inevitably loses information essential for classification during denoising. By disentangling label-causative features from label-non-causative features, CausalDiff can retain essential information in the label-causative features to a large extent. Third, unlike pixel-level purification which does not know the optimal denoising level for various attacks, CausalDiff acts adaptively on different attacks by the causal inference of S and Z. Fourth, they can be combined to further enhance adversarial robustness.

5 Experiments

In this section, we introduce the experimental settings in Section 5.1. Section 5.2 presents the main results defending against unforeseen attacks on CIFAR-10, CIFAR-100, and GTSRB datasets. We then evaluate the effectiveness of individual components of CausalDiff in Section 5.3. Due to space constraints, we provide analyses of core components during training and inference in Appendix C.1 and Appendix C.2. Additionally, we showcase examples in Appendix C.3.

5.1 Experimental Settings

Datasets and Model Architecture. Our experiments utilize the CIFAR-10, CIFAR-100 [18] and GTSRB [19] datasets. CIFAR-10 and CIFAR-100 each consists of 50,000 training images, categorized into 10 and 100 classes, respectively. GTSRB comprises 39,209 training images (each histogram equalized and resized to $3 \times 32 \times 32$) of German traffic signs, categorized into 43 classes. We use DDPM [17] as the diffusion model. Further details are available in Appendix B.2.

Attack Evaluation Method. For the CIFAR-10 dataset, we utilize **seven types of attack strategies** with both ℓ_∞ and ℓ_2 norm bounds for evaluation. These strategies include StAdv attack [60], BPDA+EOT, and AutoAttack [2] (AA), which comprises white-box attacks such as APGD-ce, APGD-t, FAB-t, and a black-box Square Attack. For CIFAR-100, we follow the setting of Chen et al. [27], evaluating against the ℓ_∞ threat model with $\epsilon = 8/255$. For the GTSRB dataset, we utilize **four types of attacks as well as fog corruptions**. Attack methods include AutoAttack, which comprises APGD-ce, APGD-t, FAB-t, and Square Attack.

5.2 Comparisons on Unseen Attacks

CIFAR-10. From the experimental results for CIFAR-10 presented in Table 2, we have several observations: 1) The adversarial training methods perform robustly on the same type of attacks they use for training but poorly on other unseen attacks. The only exception is that when the models employ

the most effective attack (i.e., AT-ℓ_∞) for adversarial training, the robustness regarding ℓ_2 is not hurt much. 2) In contrast, purification methods, especially the ones based on more powerful diffusion models (DDPM and EDM), have decent robustness on each unseen attack and much better average robustness. This is reasonable since they learn to purify the adversarial samples without targeting any specific attacks. 3) Our CausalDiff performs the best not only regarding the average robustness but also on each type of attack. Remarkably, the average robust accuracy (86.39%) is only 3.84% lower than the clean accuracy (90.23%) and it boosts the robustness on the most challenging attack (ℓ_∞) to 83.01%. Notably, the reported CausalDiff is based on DDPM, which acts less effectively than EDM, which can be seen from AT and LM purification. Grounded on a stronger backbone, CausalDiff could achieve even better performance. Table 4 shows the performance under the BPDA + EOT (EOT = 20) attack. Under this attack, models whose gradients are not accessible and also be compared (e.g., ADP [31]). The results again confirm the superior performance of CausalDiff across different attack types.

CIFAR-100. The experimental results on CIFAR-100 in Table 3 (Right) indicate that CausalDiff achieves the highest robustness among all, even with a much lower clean accuracy. The low clean accuracy is likely due to the insufficient training samples to learn S, Z, and its weaker backbone - DDPM. We believe more augmented data and a stronger diffusion backbone like EDM (used by RDC [27]) could further enhance the performance.

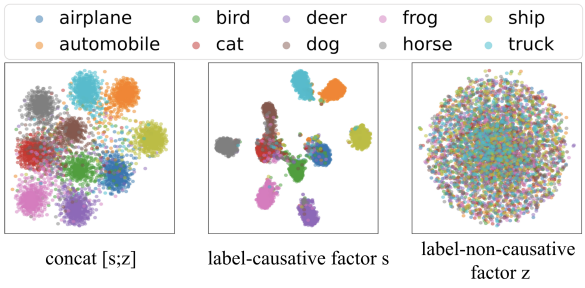


Figure 4: Visualization by T-SNE of the feature space, inferred by our CausalDiff, of the label-causative factor s, label-non-causative factor z, and their concatenation.

Table 4: Clean and robust accuracy on CIFAR-10 against **BPDA + EOT** against ℓ_∞ ($\epsilon = 8/255$) threat model.

Method	Clean Acc (%)	Robust Acc (%)
Purify - EBM [62]	84.12	54.90
LM - DDPM [27]	83.20	69.73
ADP [31]	86.14	70.01
RDC [27]	89.85	75.67
GDMP [35]	93.50	76.22
DiffPure [33]	89.02	81.40
CausalDiff	90.23	**88.48**

GTSRB. The left part of Table 3 show that CausalDiff also has compelling robustness on traffic sign classification, in terms of not only unforeseen adversarial attacks but also natural corruptions like fog. Evaluations based on different types of tasks, and different numbers of classes (10, 100, and 43) have all shown the efficacy of CausalDiff.

5.3 Ablation Study

As mentioned in Section 4.3, our CausalDiff contains Adversarial Purification (AP), Causal Factor Inference (CFI), and Latent-S-Based Classification (LSBC). The last block of Table 2 shows the individual effect of the AP and CFI in CausalDiff. We can see: 1) Our DDPM-based purification (CausalDiff w/o CFI, achieved by AP plus a standard classifier) performs similarly to LM-EDM and is significantly better than LM-DDPM, showing that the strategy of sampling within small timesteps for purification is much more effective than entire timesteps. (We show more analysis on this in Appendix C.2.) 2) The core component of CasualDiff - causal disentanglement (CausalDiff w/o AP) outperforms all the baselines in terms of average robustness except RDC which incorporates an extra LM purification step. It shows that modeling the generative of native in-domain data can enhance the model's inherent robustness and thus effectively defend against various types of attacks.

5.4 Visualization of Latent Factors

To understand how the latent causal factors S and Z in CausalDiff take effect during adversarial classification, we visualize S, Z, and their concatenation using t-SNE in Fig. 4. We randomly sampled 5000 correctly classified test samples from CIFAR-10 for visualization. We find that the Y-causative factor S of samples in each category are located in the same cluster, with clear margins between different clusters except the categories - dog, cat, and bird. These three categories share more commonalities than the others and are not surprising to have blurred boundaries. Additionally,

the S vectors of airplanes (dark blue) are near those of birds (green), and trucks have S vectors near automobiles. In contrast to the S vectors, the vectors of Z do not exhibit correlations with the categories. This also aligns with our objective to extract the Y-non-causative factors to Z. The vectors of their concatenation, i.e., $[s; z]$, also display in clusters but with much more blurred boundaries. These observations are consistent with our commonsense knowledge, showing that CausalDiff has learned reasonable Y-causative factors by S and Y-non-causative factors by Z.

6 Conclusion

We develop a causal model based on diffusion model to improve adversarial robustness. A pilot study on toy data suggests that the model defends against adversarial attacks by leveraging label-causative features to resist perturbations and expand the model's margin. Moreover, on the CIFAR-10, CIFAR-100 and GTSRB datasets, our model appears to capture semantic features consistent with the human decision-making process and surpass all baseline models, achieving state-of-the-art performance in adversarial robustness, particularly against unseen attacks.

Acknowledgments and Disclosure of Funding

This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDB0680101, CAS Project for Young Scientists in Basic Research under Grant No. YSBR-034, the Innovation Project of ICT CAS under Grant No. E261090, the National Natural Science Foundation of China (NSFC) under Grants No. 62302486, the Innovation Project of ICT CAS under Grants No. E361140, the CAS Special Research Assistant Funding Project, the project under Grants No. JCKY2022130C039, and the Strategic Priority Research Program of the CAS under Grants No. XDB0680102.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[2] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*. PMLR, 2020, pp. 2206–2216.

[3] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[4] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016, pp. 1528–1540.

[5] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE intelligent vehicles symposium (IV)*. IEEE, 2011, pp. 163–168.

[6] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, "Provably minimally-distorted adversarial examples," *arXiv preprint arXiv:1709.10207*, 2017.

[7] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," *Advances in neural information processing systems*, vol. 30, 2017.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[9] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan, "Better diffusion models further improve adversarial training," *arXiv preprint arXiv:2302.04638*, 2023.

[10] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018.

[11] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic books, 2018.

[12] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[13] C. Liu, X. Sun, J. Wang, H. Tang, T. Li, T. Qin, W. Chen, and T.-Y. Liu, "Learning causal semantic representation for out-of-distribution prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6155–6170, 2021.

[14] X. Sun, B. Wu, X. Zheng, C. Liu, W. Chen, T. Qin, and T.-Y. Liu, "Recovering latent causal factor for generalization to distributional shifts," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 846–16 859, 2021.

[15] Y. Zhang, M. Gong, T. Liu, G. Niu, X. Tian, B. Han, B. Schölkopf, and K. Zhang, "Causaladv: Adversarial robustness through the lens of causality," *arXiv preprint arXiv:2106.06196*, 2021.

[16] Q. Ren, Y. Chen, Y. Mo, Q. Wu, and J. Yan, "Dice: Domain-attack invariant causal learning for improved data privacy protection and adversarial robustness," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1483–1492.

[17] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[18] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[19] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[20] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.

[21] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli, "Adversarial robustness through local linearization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. A. Mann, "Improving robustness using generated data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4218–4233, 2021.

[23] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann, "Fixing data augmentation to improve adversarial robustness," *arXiv preprint arXiv:2103.01946*, 2021.

[24] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[25] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, "Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization," *arXiv preprint arXiv:1911.03437*, 2019.

[26] C. Laidlaw, S. Singla, and S. Feizi, "Perceptual adversarial robustness: Defense against unseen threat models," *arXiv preprint arXiv:2006.12655*, 2020.

[27] H. Chen, Y. Dong, Z. Wang, X. Yang, C. Duan, H. Su, and J. Zhu, "Robust classification via a single diffusion model," *arXiv preprint arXiv:2305.15241*, 2023.

[28] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.

[29] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[30] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[31] J. Yoon, S. J. Hwang, and J. Lee, "Adversarial purification with score-based generative models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 062–12 072.

[32] C. Xiao, Z. Chen, K. Jin, J. Wang, W. Nie, M. Liu, A. Anandkumar, B. Li, and D. Song, "Densepure: Understanding diffusion models towards adversarial robustness," *arXiv preprint arXiv:2211.00322*, 2022.

[33] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," *arXiv preprint arXiv:2205.07460*, 2022.

[34] Q. Wu, H. Ye, and Y. Gu, "Guided diffusion model for adversarial purification from random noise," *arXiv preprint arXiv:2206.10875*, 2022.

[35] J. Wang, Z. Lyu, D. Lin, B. Dai, and H. Fu, "Guided diffusion model for adversarial purification," *arXiv preprint arXiv:2205.14969*, 2022.

[36] J. Mitrovic, B. McWilliams, J. Walker, L. Buesing, and C. Blundell, "Representation learning via invariant causal mechanisms," *arXiv preprint arXiv:2010.07922*, 2020.

[37] B. Schölkopf, "Causality for machine learning," in *Probabilistic and Causal Inference: The Works of Judea Pearl*, 2022, pp. 765–804.

[38] Y. Chen, Y. Zhang, Y. Bian, H. Yang, M. Kaili, B. Xie, T. Liu, B. Han, and J. Cheng, "Learning causally invariant representations for out-of-distribution generalization on graphs," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 131–22 148, 2022.

[39] W. Wang, X. Lin, F. Feng, X. He, M. Lin, and T.-S. Chua, "Causal representation learning for out-of-distribution recommendation," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 3562–3571.

[40] R. Wang, M. Yi, Z. Chen, and S. Zhu, "Out-of-distribution generalization with causal invariant transformations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 375–385.

[41] R. Moraffah, M. Karami, R. Guo, A. Raglin, and H. Liu, "Causal interpretability for machine learning-problems, methods and evaluation," *ACM SIGKDD Explorations Newsletter*, vol. 22, no. 1, pp. 18–33, 2020.

[42] G. Xu, T. D. Duong, Q. Li, S. Liu, and X. Wang, "Causality learning: A new perspective for interpretable machine learning," *arXiv preprint arXiv:2006.16789*, 2020.

[43] C. Zhang, K. Zhang, and Y. Li, "A causal view on robustness of neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 289–301, 2020.

[44] K. Tang, M. Tao, and H. Zhang, "Adversarial visual robustness by causal intervention," *arXiv preprint arXiv:2106.09534*, 2021.

[45] S. Yang, T. Guo, Y. Wang, and C. Xu, "Adversarial robustness through disentangled representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3145–3153.

[46] B.-K. Lee, J. Kim, and Y. M. Ro, "Mitigating adversarial vulnerability through causal parameter estimation by adversarial double machine learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4499–4509.

[47] J. Kim, B.-K. Lee, and Y. M. Ro, "Demystifying causal features on adversarial examples and causal inoculation for robust network by adversarial instrumental variable regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 302–12 312.

[48] H. Hua, J. Yan, X. Fang, W. Huang, H. Yin, and W. Ge, "Causal information bottleneck boosts adversarial robustness of deep neural network," *arXiv preprint arXiv:2210.14229*, 2022.

[49] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 565–26 577, 2022.

[50] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[51] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[52] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, "Diffusion autoencoders: Toward a meaningful and decodable representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 619–10 629.

[53] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[54] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[55] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[56] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang, "Mma training: Direct input space margin maximization through adversarial training," *arXiv preprint arXiv:1812.02637*, 2018.

[57] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *International conference on machine learning*. PMLR, 2020, pp. 1779–1788.

[58] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[59] R. S. Zimmermann, L. Schott, Y. Song, B. A. Dunn, and D. A. Klindt, "Score-based generative classifiers," *arXiv preprint arXiv:2110.00473*, 2021.

[60] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," *arXiv preprint arXiv:1801.02612*, 2018.

[61] T. Wu, L. Tong, and Y. Vorobeychik, "Defending against physically realizable attacks on image classification," *arXiv preprint arXiv:1909.09552*, 2019.

[62] M. Hill, J. Mitchell, and S.-C. Zhu, "Stochastic security: Adversarial defense using long-run dynamics of energy-based models," *arXiv preprint arXiv:2005.13525*, 2020.

[63] M. Boudiaf, J. Rony, I. M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed, "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in *European conference on computer vision*. Springer, 2020, pp. 548–564.

A Proof of Propositions

In this section, we will present the detailed proof for the theoretical results mentioned in the main paper.

A.1 Proof of Causal Information Bottleneck (CIB)

According to the Structural Causal Model (SCM), we have $p(x, y, s, z) = p(s)p(z)p(x|s, z)p(y|s)$. Thus, the Causal Information Bottleneck (CIB) can be represented as:

$$
\begin{aligned}
&\mathrm{I}(X, Y; S, Z) = \mathbb{E}_{p(x,y,s,z)} \log \frac{p(x, y, s, z)}{p(x, y)p(s, z)} \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(s)p(z)p(x|s, z)p(y|s)}{p(x, y)p(s, z)} \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(x|s, z)p(y|s)}{p(x, y)} + \mathbb{E}_{p(x,y,s,z)} \log \frac{p(s)p(z)}{p(s, z)} \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(x|s, z)p(y|s)}{p(x, y)} - \mathrm{I}(S; Z) \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(x|s, z)p(y|s)p(s, z)}{p(y|x)p(x)p(s, z)} - \mathrm{I}(S; Z) \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(y|s)}{p(y|x)} + \mathbb{E}_{p(x,y,s,z)} \log \frac{p(x|s, z)p(s, z)}{p(x)p(s, z)} - \mathrm{I}(S; Z) \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(y|s)}{p(y|x)} + \mathrm{I}(X; S, Z) - \mathrm{I}(S; Z) \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(y|s)p(s)p(y)}{p(y|x)p(s)p(y)} + \mathrm{I}(X; S, Z) - \mathrm{I}(S; Z) \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(y)}{p(y|x)} + \mathbb{E}_{p(x,y,s,z)} \log \frac{p(y|s)p(s)}{p(s)p(y)} + \mathrm{I}(X; S, Z) - \mathrm{I}(S; Z) \\
=&\mathbb{E}_{p(x,y,s,z)} \log \frac{p(y)p(x)}{p(y|x)p(x)} + \mathrm{I}(Y; S) + \mathrm{I}(X; S, Z) - \mathrm{I}(S; Z) \\
=&\mathrm{I}(X; Y) + \mathrm{I}(Y; S) + \mathrm{I}(X; S, Z) - \mathrm{I}(S; Z)
\end{aligned}
\tag{11}
$$

Therefore, we have proved the result in Eq. (5)

A.2 Proof of the Lower Bound of CIB

According to the result in Eq. (11), we further prove its lower bound shown in Eq. (6) in this section.

As for the lower bound of the recontruction term $\mathrm{I}(X; S, Z)$, we have:

$$
\begin{aligned}
&\mathrm{I}(X; S, Z) = \mathbb{E}_{p(x,s,z)}[\log \frac{p(x|s, z)}{p(x)}] \\
&= \mathbb{E}_{p(x,s,z)}[\log \frac{p(x|s, z)p_\theta(x|s, z)}{p(x)p_\theta(x|s, z)}], \quad \text{when using a variational distribution } p_\theta(x|s, z) \text{ to approximate } p(x|s, z), \\
&= \mathbb{E}_{p(x,s,z)}[\log \frac{p_\theta(x|s, z)}{p(x)}] + \mathbb{E}_{p(s,z)}\mathbb{E}_{p(x|s,z)}[\log \frac{p(x|s, z)}{p_\theta(x|s, z)}] \\
&= \mathbb{E}_{p(x,s,z)}[\log \frac{p_\theta(x|s, z)}{p(x)}] + \mathbb{E}_{p(s,z)}[\mathcal{D}_{\mathrm{KL}} \frac{p(x|s, z)}{p_\theta(x|s, z)}], \text{ where } \mathcal{D}_{\mathrm{KL}}(\cdot) \text{ represents KL-divergence,} \\
&\geq \mathbb{E}_{p(x,s,z)}[\log \frac{p_\theta(x|s, z)}{p(x)}] \\
&= \mathbb{E}_{p(x,s,z)}[\log p_\theta(x|s, z)] + \mathbb{E}_{p(x,s,z)}[\log \frac{1}{p(x)}] \\
&= \mathbb{E}_{p(x,s,z)}[\log p_\theta(x|s, z)] + \mathcal{H}(x), \text{ where } \mathcal{H}(x) \text{ indicates entropy of } x.
\end{aligned}
\tag{12}
$$

As for the upper bound of $\mathrm{I}(X;S,Z)$, we have:

$$\begin{aligned}
&\mathrm{I}(X;S,Z) = \mathbb{E}_{p(x,s,z)}[\log \frac{p(x|s,z)}{p(x)}] \\
&= \mathbb{E}_{p(x,s,z)}[\log \frac{p(x|s,z)q(s,z)}{p(x)q(s,z)}], \text{ when using a prior distribution } q(s,z) \text{ to estimate } p(s,z), \\
&= \mathbb{E}_{p(x,s,z)}[\log \frac{p(x|s,z)}{q(s,z)}] - \mathcal{D}_{\mathrm{KL}}(p(s,z)||q(s,z)), \text{ where } \mathcal{D}_{\mathrm{KL}}(\cdot) \text{ represents KL-divergence,} \\
&\leq \mathbb{E}_{p(x,s,z)}[\log \frac{p(x|s,z)}{q(s,z)}] \\
&= \mathbb{E}_{p(x)}\mathbb{E}_{p(s,z|x)}[\log \frac{p(s,z|x)}{q(s,z)}] \\
&= \mathbb{E}_{p(x)}[\mathcal{D}_{\mathrm{KL}}(p(s,z|x)||q(s,z))].
\end{aligned} \tag{13}$$

Regarding the label prediction term $\mathrm{I}(Y;S)$, we can maximize the mutual information between factor S and label Y by maximize $\mathbb{E}_{p(y,s)}[\log p_\theta(y|s)]$ as well as employing a cross-entropy loss function, according to Boudiaf et al. [63].

As for the disentangle term $\mathrm{I}(S;Z)$, according to the Contrastive Log-Ratio Upper Bound (CLUB) of mutual information proposed by Cheng et al. [57], we have

$$\mathrm{I}(S;Z) \leq \mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z) = \mathbb{E}_{p(s,z)}[\log p_\theta(s|z)] - \mathbb{E}_{p(z)}\mathbb{E}_{p(s)}[\log p_\theta(s|z)], \tag{14}$$

where $p_\theta(s|z)$ is a variational distribution to estimate $p(s|z)$.

Thus, we have proved the results in Eq. (6) that

$$\begin{aligned}
&\mathrm{I}(X;S,Z) + \mathrm{I}(Y;S) - \mathrm{I}(S;Z) - \lambda \mathrm{I}(X;S,Z) \\
&\geq \mathbb{E}_{p(x,s,z)}[\log p_\theta(x|s,z)] + \mathbb{E}_{p(y,s)}[\log p_\theta(y|s)] - \mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z) - \lambda\, \mathbb{E}_{p(x)}[\mathcal{D}_{\mathrm{KL}}(p_\theta(s,z|x)\|q(s,z))]. \\
&= \mathbb{E}_{p(x,s,z)}[\log p_\theta(x|s,z)] + \mathbb{E}_{p(y,s)}[\log p_\theta(y|s)] - \mathbb{E}_{p(s,z)}[\log p_\theta(s|z)] \\
&\quad + \mathbb{E}_{p(z)}\mathbb{E}_{p(s)}[\log p_\theta(s|z)] - \lambda\, \mathbb{E}_{p(x)}[\mathcal{D}_{\mathrm{KL}}(p_\theta(s,z|x)\|q(s,z))].
\end{aligned} \tag{15}$$

A.3 Detailed Derivation of Loss Function

Based on the result in Eq. (15), we can optimize the Causal Information Bottleneck (CIB) $\mathrm{I}(X,Y;S,Z)$ by maximizing its theoratically lower bound. In this part, we discuss the detailed derivation for the lower bound of CIB in term of designing the loss function proposed in Eq. (7).

Specifically, for the reconstruction term $\mathbb{E}_{p(x,s,z)}[\log p_\theta(x|s,z)]$, we can maximize the log-likelihood estimated by our conditional diffusion model:

$$\log p_\theta(x|s,z) \geq -\mathbb{E}_{\epsilon,t}[w_t \|\epsilon_\theta(x_t,t,s,z) - \epsilon\|] + C, \tag{16}$$

where $\epsilon_\theta(x_t,t,s,z)$ denotes our conditional diffusion model, and constant C is negligible [17].

Regarding the label prediction term $\mathbb{E}_{p(y,s)}[\log p_\theta(y|s)]$, following the results proposed by Boudiaf et al. [63], we can leveraging a cross-entropy loss function to maximize the mutual information between factor S and label Y.

As for the disentanglement term $\mathrm{I}(S;Z)$, we following the optimization strategies proposed by Cheng et al. [57], leveraging a predictor p_θ to learn the relationship between S and Z so as to estimate $\mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z)$.

Overall, we have proved the loss function of our Causal Information Bottleneck (CIB) proposed in Eq. (7):

$$\begin{aligned}
\mathcal{L}(x,y,s,z;\theta) = \; &\alpha\, \mathbb{E}_{\epsilon,t}\left\|\epsilon_\theta\left(x_t,t,s,z\right) - \epsilon_t\right\|_2^2 + \gamma \mathcal{L}_{\mathrm{CE}}(s,y;\theta) \\
&+ \eta\{\mathbb{E}_{p(s,z)}[\log p_\theta(s|z)] - \mathbb{E}_{p(z)}\mathbb{E}_{p(s)}[\log p_\theta(s|z)]\} + \lambda\, \mathcal{D}_{\mathrm{KL}}(p_\theta(s,z|x)\|q(s,z)),
\end{aligned} \tag{17}$$

where $\mathbb{E}_{p(s,z)}[\log p_\theta(s|z)] - \mathbb{E}_{p(z)}\mathbb{E}_{p(s)}[\log p_\theta(s|z)]$ is the estimation of $\mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z)$.

Algorithm 1 CausalDiff Algorithm

Require: Dataset $\mathcal{D}$; The CausalDiff parameterized by θ pretrained by Algorithm 2 involves an UNet $\epsilon_\theta(x_t, t, s, z)$ with diffusion timestep T, an encoder $f_{(s,z)}(x;\theta)$, a classifier $f_y(s;\theta)$, and an MI estimater $f_{\mathrm{CLUB}}(s, z;\theta)$ for the CLUB loss; an optimizer $\mathrm{optim}(\cdot)$, dropout probability p_{drop} of s, z for simultaneously training an unconditional generation, hyperparameters $\alpha, \lambda, \gamma, \eta$.

Initialize:
  Load $f_{s,z}(x;\theta)$ and ϵ_θ from the pretrained model θ.
  Initialize $f_y(s;\theta)$ and $f_{\mathrm{CLUB}}(s, z;\theta)$ randomly.
for ep=1 **to** N_2 **do**
  Get mini-batch $(x, y) \sim \mathcal{D}$
  $s, z = f_{(s,z)}(x;\theta)$
  $t \sim \mathrm{Uniform}(\{1, 2, ..., T\})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  if $rand(0, 1) \leq p_{\mathrm{drop}}$ **then**
    Compute loss $\mathcal{L}(x;\theta) = \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$
    Update ϵ_θ with $\mathrm{optim}(\theta, \mathcal{L}(x;\theta))$
  else
    Compute loss $\mathcal{L}(x, y, s, z;\theta)$ according to Eq. (7)
    Update $\epsilon_\theta(x_t, t, s, z)$, $f_{(s,z)}(x;\theta)$,
      and $f_y(s;\theta)$ by $\mathrm{optim}(\theta, \mathcal{L}(x, y, s, z;\theta))$
  end if
  Update $f_{\mathrm{CLUB}}(s, z;\theta)$ according to the CLUB algorithm [57]
end for

A.4 ELBO (Evidence Lower BOund) V.S. MI (Mutual Information)

the Causal Evidence Lower BOund (ELBO) for multi-domain datasets as proposed by Sun et al. [14], we directly formulate the Causal ELBO within our causal structure, as depicted in Fig. 1 (Left). Given that $p(x, y, s, z) = p(s)p(z)p(x|s, z)p(y|s)$, we can derive the Causal ELBO in single domain as follows:

$$\begin{aligned}\mathrm{ELBO} &= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\mathbb{E}_{q_\psi(s,z|x,y)} \log \frac{p_\theta(x, y, s, z)}{q_\psi(s, z|x, y)}] \\ &= \mathbb{E}_{p(x,y)}\Big\{\mathbb{E}_{q_\psi(s,z|x,y)}\Big[\log p_\theta(x \mid s, z) + \log \frac{p_\theta(s, z)}{q_\psi(s, z \mid x)} + \log p_\theta(y \mid s) + \log \frac{q_\psi(y \mid x)}{q_\psi(y \mid s)}\Big]\Big\},\end{aligned} \tag{18}$$

where p_θ is to learn the ground-truth p and q_ψ is variational distribution to mimic p_θ.

Specifically, the causal ELBO, compared to our Causal Information Bottleneck (CIB), incorporates the reconstruction term $\log p_\theta(x \mid s, z)$, the insensitivity term $\log \frac{p_\theta(s,z)}{q_\psi(s,z|x)}$, and the label prediction term $\log p_\theta(y \mid s)$. However, it overlooks the disentanglement of latent factors, a critical aspect for effectively learning the causal model. Additionally, we have empirically assessed the robustness of models trained with either CIB or causal ELBO (equivalent to $\eta = 0$) in section C.1. This evaluation aims to investigate the efficacy of the disentanglement term $\mathrm{I}(S; Z)$.

B More Implementation Details

B.1 Baselines

We include representative defense methods of adversarial training (AT), purification, and other types (e.g., generative-model-based approach) as baselines. Specifically, we compare with the AT methods [9, 23] that use DDPM or the Elucidating Diffusion Model (EDM) [49] for data augmentation and a theoretical framework TRADES [20] for adversarial training. We also include causality-based AT baselines such as CausalAdv [15] and DICE [16]. Purification baselines include DiffPure [33] (grounded on Score SDE [30]) and diffusion-based Likelihood Maximization (LM)[27] with EDM (as in the original paper) and DDPM (we reproduced). Other defense baselines comprise generative classifiers such as Score-Based Generative Classifier (SBGC) [59], [27], and the causality-based

Algorithm 2 CausalDiff Pretrain Algorithm

Require: Dataset $\mathcal{D}$; a diffusion model ϵ_θ, an encoder $f_{(s,z)}(x;\theta)$, a classifier $f_y(s;\theta)$; an optimizer $\mathrm{optim}(\cdot)$, probability p_{drop} of training samples for unconditional diffusion, diffusion training epoch N_1.

Initialize: randomly initialize parameter θ.
for ep=1 **to** N_1 **do**
  $(x, y) \sim \mathcal{D}$
  $s, z = f_{(s,z)}(x;\theta)$
  $s, z \leftarrow \phi$ with probability p_{drop} ▷ randomly mask the latent factors with probability p_{drop}
  $t \sim \mathrm{Uniform}(\{1, 2, ..., T\})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathcal{L}(x, s, z;\theta) = \|\epsilon_\theta(x_t, t, s, z) - \epsilon\|_2^2$
  Update ϵ_θ and $f_{(s,z)}(x;\theta)$ by $\mathrm{optim}(\theta, \mathcal{L}(x, s, z;\theta))$
end for

Algorithm 3 Adversarially Robust Inference Algorithm

Require: A fully-trained causal model involves diffusion model $\epsilon_\theta(x_t, t, s, z)$, encoder $f_{(s,z)}(x;\theta)$, and classifier $f_y(s;\theta)$; test image x, purification optimization steps N_{purify}, causal factor inference steps N_{infer}, number of sampling steps N_t for inference,optimizer $\mathrm{optim}_{\text{purify}}$ for purification, optimizer $\mathrm{optim}_{\text{infer}}$ for causal factor inference.

Initialize: purified image $x^* = x$
Stage 1: Adversarial Purification
for iter=1 **to** N_{purify} **do**
  $t \sim \mathrm{Uniform}(\{1, 2, ..., 50\})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathcal{L}(x^*, \theta) = \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$
  Update x^* by $\mathrm{optim}_{\text{purify}}(x^*; \mathcal{L}(x^*, \theta))$
end for

Stage 2: Causal Factor Inference
Initial $s^*, z^* = f_{(s,z)}(x^*;\theta)$
for iter=1 **to** N_{infer} **do**
  Sample N_t timesteps t at equal intervals from 1 to T
  Sample N_t $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ corresponds to each t
  Compute $\mathcal{L}(x^*, s^*, z^*;\theta) = \mathbb{E}_{\epsilon,t}\|\epsilon_\theta(x_t^*, t, s, z) - \epsilon\|_2^2$
  Update s^*, z^* by $\mathrm{optim}_{\text{infer}}(s^*, z^*; \mathcal{L}(x^*, s^*, z^*;\theta))$
end for

Stage 3: S-based Classification
$y = f_y(s^*;\theta)$
return y

generative model CAMA [43]. Notably, we also include the SOTA method on unseen attacks - Robust Diffusion Classifier (RDC), which incorporates purification and conditional generation based on labels for defense.

B.2 Implementation Details

We use DDPM [17] as our generative model. For the encoder $f_{s,z}(x;\theta)$ and the classification model $f_s(y;\theta)$, we employ WideResNet-70-16 (WRN-70-16) as the backbone.

Training Strategy. Considering the different complexities in learning $f_x(s, z;\theta)$, $f_{(s,z)}(x;\theta)$ and $f_y(s;\theta)$, attributed to the differing difficulty in generation and discrimination task, we segment the training of the entire causal model into two distinct phases. As outlined in Section 4.2, we employ a two-stage training process for our CausalDiff.

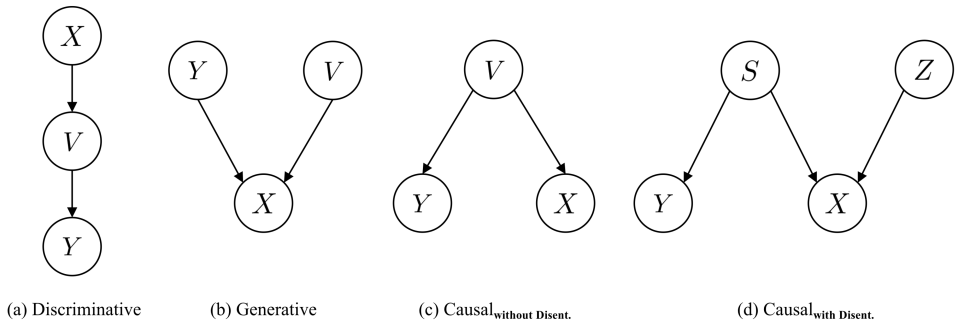


Figure 5: SCM of models for pilot study including (a) discriminative model, (b) generative model, (c) causal model without disentanglement, and (d) causal model with disentanglement.

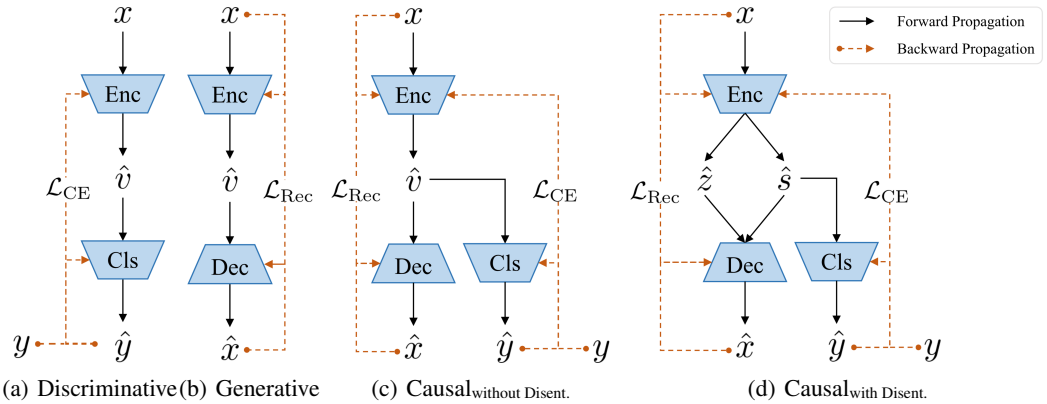


Figure 6: Architecture of models for pilot study including (a) discriminative model, (b) generative model, (c) causal model without disentanglement, and (d) causal model with disentanglement.

In the pretrain phase on CIFAR-10 and GTSRB datasets, focused on generation, primarily trains the conditional diffusion model $f_x(s, z; \theta)$ along with its corresponding encoder $f_{(s,z)}(x; \theta)$ according to Algorithm 2, setting $N_1 = 1440$. For CIFAR-100, we added 10,000 images generated by EDM [49] to our training set. Considering the computational cost, we only pretrain for $N_1 = 500$ epochs. Note that the augmented data is used only during the pretraining phase while not in the joint training phase.

Subsequently, we conduct joint training of the whole CausalDiff model for $N_2 = 560$, amounting to a total of 2000 epochs. The second phase, targeting discrimination and leveraging label information to guide disentanglement, involves the joint training of the entire causal model. Note that, in order to simultaneously train an unconditional diffusion model for adversarial purification, we follow Ho and Salimans [51] to mask the condition s and z with probability $p_{\text{drop}} = 0.1$. Thus, a single shared model is used for both adversarial purification and causal factor inference.

Both the pretraining and joint training phases utilize a learning rate of $1e^{-4}$ and a batch size of 128. For simplicity, we follow the setting of $w_t = 1$ [17]. We set $\alpha = 1.$, $\gamma = 1e^{-2}$, $\eta = 1e^{-5}$, $\lambda = 1e^{-2}$ as the weights for the loss function in Eq. (7).

Since we need a standard diffusion model $\epsilon_\theta(x_t, t)$ for purification during adversarial inference, we apply dropout of s and z with a ratio of $p_{\text{drop}} = 0.1$ for conditional diffusion generation as in Ho et al. [17] during both pretraining and training. Thus, the unconditional probability of generating x can also be estimated using the same model by masking s and z.

Inference Strategy. Leveraging the trained CausalDiff, we can infer its label from an adversarial example in accordance with Algorithm 3, following the inference pipeline outlined in Section 4.3. Specifically, we set $N_{\text{purify}} = 5$ and use momentum-SGD as our $\text{optim}_{\text{purify}}$, with a learning rate of

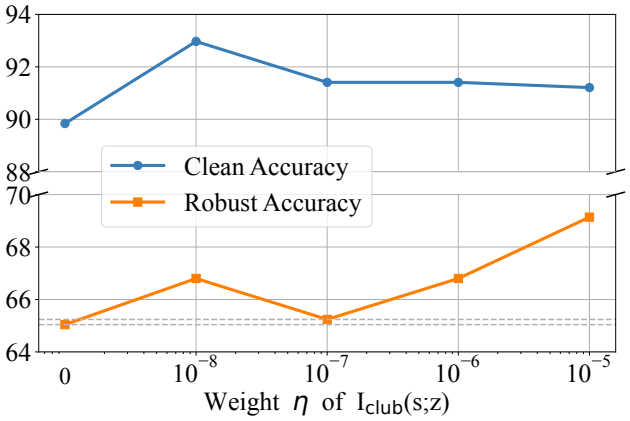


Figure 7: Impact of η for disentanglement term in loss function on clean accuracy and robust accuracy.

0.1. For causal factor inference, we choose $N_t = 10$ to sample 10 timesteps per iteration and adopt $N_{\text{infer}} = 10$ with momentum-SGD as $\text{optim}_{\text{infer}}$, setting the learning rate to $1e^{-5}$.

Regarding white-box attacks, we perform a gradient backpropagation throughout the entire pipeline of our CausalDiff approach, which includes purification, causal factors inference, and classification. This implies that the attacker possesses sufficient knowledge of the causal model.

Attack Evaluation. Attack evaluation for CIFAR-10 dataset includes a 100-step StAdv attack [60] with $\epsilon = 0.05$ under the ℓ_∞ norm bound, BPDA+EOT (EOT=20) against the ℓ_∞ threat model with $\epsilon = 8/255$, and AutoAttack [2] (AA), which comprises 100-step white-box attacks such as APGD-ce, APGD-t, FAB-t, and a 5000-step black-box Square Attack under both ℓ_2 ($\epsilon = 0.5$) and ℓ_∞ ($\epsilon = 8/255$) constraints.

For the GTSRB dataset, we utilize **four types of attacks as well as two types of corruptions** to evaluate robustness against adversarial attacks and the influence of the natural environment. Specifically, the attack methods include AutoAttack, which comprises 100-step white-box attacks such as APGD-ce, APGD-t, FAB-t, and a 5000-step black-box Square Attack under both ℓ_2 ($\epsilon = 0.5$) and ℓ_∞ ($\epsilon = 8/255$) constraints. The corruptions include Fog and Brightness. Following Nie et al. [33], we randomly sample 512 samples for evaluation.

B.3 Details for Pilot Study

Data Construction. For each data point of the 2000 samples, we sample a vector s with dimension $h_s = 8$ from a normal distribution with mean -1 and variance 1, i.e., $s \sim \mathcal{N}(-1, 1)$, and a vector $z \sim \mathcal{N}(1, 1)$ with dimension $h_z = 8$. Subsequently, we projected the concatenation of s and z, denoted as $[s; z]$, to a sample x, with a random initialized matrix $A_x (A_x \in \mathcal{R}^{(h_s+h_z)\times h_x})$, i.e., $x = [s; z] \cdot A_x$. Similarly, we produced the score y_s of x with $y_s = s \cdot A_y$, where $A_y \in \mathcal{R}^{h_s \times 1}$. To obtain samples with balanced binary labels, we consider the label y of the sample with y_s above the median as 1 and the others as 0.

Methods for Comparisons. In the pilot study detailed in Section 3, we conducted an investigation and analysis on four models: 1) Discriminative: a discriminative model that learns to classify the samples with a two-layer perceptron (MLP), 2) Generative [27]: a generative model that learns the generation of the sample x conditioning on its label y and predict the label of an adversarial example by calculating the $\max_y p(x|y)$, 3) Causal without Disent.: a causal model that models the generation of both x and y with the same causal factor v (Causal modeling without Disentanglement), 4) Causal with Disent.: our model that disentangles the label-causative factor s and another factor z during the generation of x. For the latter two causal models, given an adversarial example, the hidden vectors v or s, z are inferred for Causal without Disent. and with Disent. which are then used for the final label prediction. This section comprehensively presents the designed causal structures in Fig. 5 and the model architectures in Fig. 6.

Regarding implementation, we trained each of the four models for 20 epochs, optimizing the model parameters using the Adam optimizer with a learning rate of $1e^{-3}$. The latent dimension for each model was set to 64. For evaluation, we employed a 100-step PGD attack with $\epsilon = 0.3$ and $\alpha = 2/255$ within the ℓ_∞ norm boundary. The variation in latent variables and predicted logits between adversarial examples and clean images, as presented in Table 4, is measured on adversarial examples generated with $\epsilon = 10$ and $\alpha = 0.05$ within the ℓ_∞ norm boundary, using a 100-step PGD attack.

C More Experimental Results

C.1 Analyses on Core Components of Training

Effect of $\mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z)$ in Casual Information Bottleneck (CIB) To examine the impact of our introduced disentanglement term $\mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z)$ in CIB (See Equation (6)), we vary η in Equation (7) and evaluate the robustness against the most challenging attack in AutoAttack [2], i.e., with ℓ_∞ norm bounded by $\epsilon = 8/255$. Larger η will cause the diffusion model to collapse, so we do not include the results of larger η. As we mentioned in Section 4.2, our CIB regresses to the ELBO objective in [14] when $\eta = 0$. As shown in Fig. 7, CausalDiff has better clean accuracy as well as robustness when $\eta > 0$ and yields the best robustness when $\eta = 10^{-5}$ (69.14% compared to 65.04% when $\eta = 0$). It confirms that $\mathrm{I}^{\theta}_{\mathrm{CLUB}}(S;Z)$ in the loss function is beneficial to disentangle the Y-causative from Y-non-causative factors and can further enhance both clean accuracy and robustness.

C.2 Analyses on Core Components of Inference

Causal Factor Inference Method. We also evaluate the robustness using the encoder $f_{(s,z)}(x;\theta)$ to get latent variables instead of inferring s and z through the conditional diffusion model. The results reveal that classification by the encoder (without purification) achieves a clean accuracy of 91.99% but 0.00% against AutoAttack under both ℓ_∞ and ℓ_2 threat models. This decline might be attributed to imprecise modeling around x (i.e., $x + \delta$), which results in an inability to resist adversarial perturbations on x.

Timestep t Sampling Strategies for Purification As discussed in Section 5.2, our purification method (*CausalDiff*$_{\text{w/o Causal Factor Inference}}$ in Table 2) markedly surpasses the direct adaptation of likelihood maximization (*LM-DDPM* in Table 2), as proposed by Chen et al. [27], applied to DDPM. This improvement stems from a refined strategy in sampling the timestep t.

As demonstrated in Fig. 9, we found that smaller timesteps perform better in distinguishing between the distributions of clean and adversarial samples. Specifically, we presented the negative log-likelihood estimated by expectation $\mathbb{E}_\epsilon[w_t \|\epsilon_\theta(x_t, t) - \epsilon\|]$ for the given timestep over 512 examples. This may caused by a larger timestep implies a greater degree of noise addition for estimating likelihood, which might overshadow the adversarial perturbations, unexpected for purification.

C.3 Visualization of Cases

We visualize the generated images leveraging the conditional generation of our CausalDiff, providing an intuitive depiction of the label-causative factor s^* and the label-non-causative factor z^*. Fig. 8 illustrates that after inferring from the benign example x^*, perturbations are alleviated. The image x_{s^*} conditioned on s^* showcases that s^* captures the core predictive features, reflecting the general concept of the category (for instance, a common semantic representation of 'horse' from an image of a brown horse's head), or even enhances the predictive features such as the dog's face or the ship's body, whereas z^* retains specific and non-predictive image details.

We also visualize the purified example x^*, conditionally generated $x^*_{s^*}$ and $x^*_{z^*}$ when encountering an adversarial example $\tilde{x}$. These cases demonstrate that, despite the presence of perturbations in the clean images, our CausalDiff effectively captures the correct predictive information of s^*, maintaining alignment with the clean data.

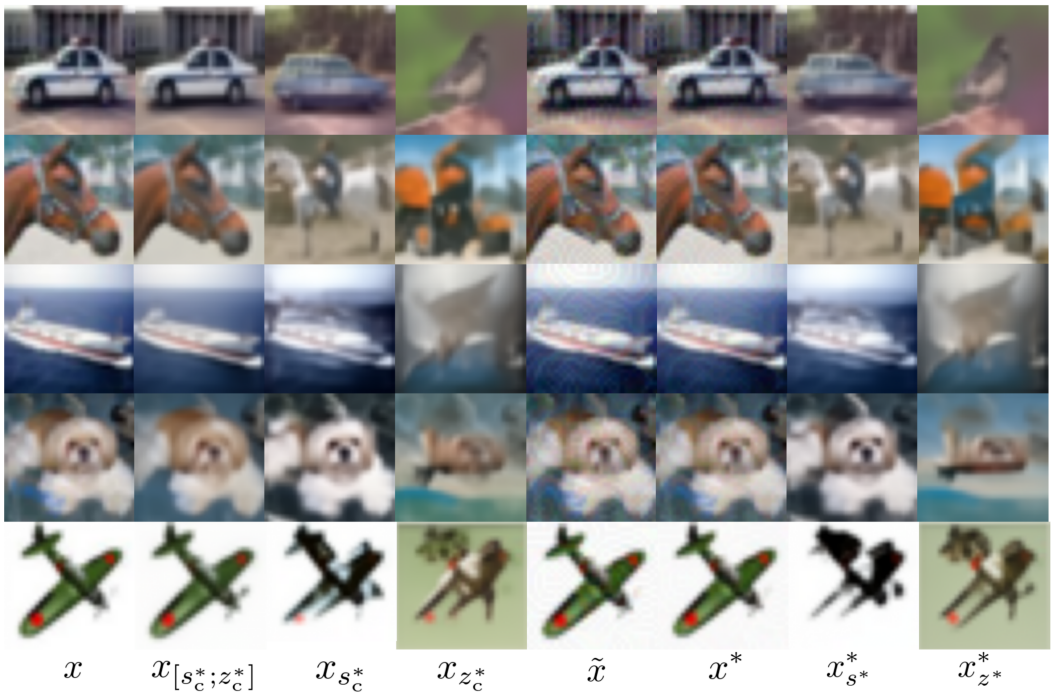


Figure 8: Reconstruction images $x_{[s_c^*;z_c^*]}$ when given clean example x, where s_c^* and z_c^* are inferred from the clean example x by our CausalDiff; generated images $x_{s_c^*}$ and $x_{z_c^*}$ conditioned on s_c^* and z_c^*, respectively; purified image x^* utilizing the unconditional diffusion (with s, z masked) when given an adversarial example $\tilde{x}$; generated images x_{s^*} and x_{z^*} conditioned on s^* and z^*, respectively, where s^* and z^* are inferred from the purified image x^*.

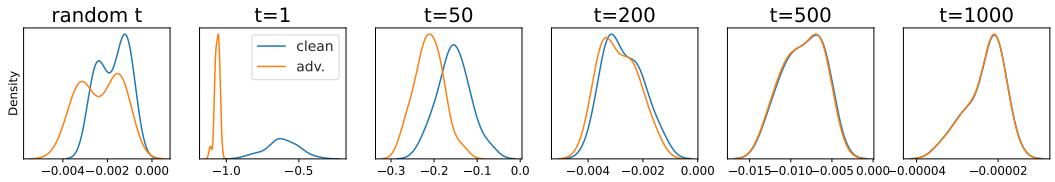


Figure 9: Distribution of likelihood for adversarial and benign examples across various timesteps t.

C.4 Speed Test of Inference Time

We evaluate the computational complexity of CausalDiff and DiffPure [33] as well as a discriminative model (WRN-70-16) by measuring the inference time in seconds for a single sample (average on 100 examples from CIFAR-10 dataset) on two types of GPUs, including NVIDIA A6000 GPU and 4090 GPU (Our experiments leverage 4 A6000 GPUs and 4 4090 GPUs). The results are shown in Table 5.

Table 5: Comparison of NFEs (Number of Function Evaluations) across different models on GPUs

	CausalDiff	CausalDiff w/o Purify	CausalDiff w/o Causal Factor Infer.	DiffPure	WRN-70-16
NFE	$N_1 + N_2 + 1$	$N_2 + 1$	$N_1 + 1$	N_1	1
Time (A6000)	4.97	4.62	0.29	2.22	0.011
Time (4090)	4.88	4.61	0.25	2.06	0.007

D Limitation

Although our CausalDiff significantly narrows the gap in classification accuracy between adversarial and clean examples, it requires an inference cost of $1 + N_1 + N_2$ NFEs (Number of Function Evaluations), where efficiency improvements are needed. Note that N_1 indicates the purification step (e.g. 5) and N_2 indicates the step of causal factor inference (e.g., 10) and 1 NFE for latent-S-based classification. Furthermore, our CausalDiff represents a new framework, meaning it requires training from scratch. Perhaps in the future, an efficient implementation of robust inference could be achieved by embedding causal mechanisms into the existing models in a plug-and-play manner.

E Broader Impact

Our CausalDiff model, built upon a powerful generative framework, aims to align with human decision-making mechanisms to enhance the stability and trustworthiness of neural networks. This approach holds potential for advancing the field of Machine Learning, particularly in safety-sensitive applications such as autonomous driving and facial recognition.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see abstract

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix D

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 4.2 and Appendix A

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5, Appendix C and Appendix B

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We described our experimental details in Appendix B and we will open source our code in camera-ready version.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Adversarial attack evaluation incurs high computational costs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C.4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Appendix E

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix E

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Appendix E

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix 4

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.