How to Enhance Downstream Adversarial Robustness (almost) without Touching the Pre-Trained Foundation Model?

Meiqi Liu [1] [*]   Zhuoqun Huang [2] [*]   Yue Xing [1]

Abstract

With the rise of powerful foundation models, a pre-training-fine-tuning paradigm becomes increasingly popular these days: A foundation model is pre-trained using a huge amount of data from various sources, and then the downstream users only need to fine-tune and adapt it to specific downstream tasks. However, due to the high computation complexity of adversarial training, it is not feasible to fine-tune the foundation model to improve its robustness on the downstream task. Observing the above challenge, we want to improve the downstream robustness without updating/accessing the weights in the foundation model. Inspired from existing literature in robustness inheritance (Kim et al., 2020), through theoretical investigation, we identify a close relationship between robust contrastive learning with the adversarial robustness of supervised learning. To further validate and utilize this theoretical insight, we design a simple-yet-effective robust auto-encoder as a data pre-processing method before feeding the data into the foundation model. The proposed approach has zero access to the foundation model when training the robust auto-encoder. Extensive experiments demonstrate the effectiveness of the proposed method in improving the robustness of downstream tasks, verifying the connection between the feature robustness (implied by small adversarial contrastive loss) and the robustness of the downstream task.

1. Introduction

In recent years, the development of foundation models has inspired people to consider a new training paradigm: Instead of training all layers of the neural network, the base large neural network is first trained by one party with a huge amount of data from various sources (pre-training). Then, the downstream users tune the last layers to adapt to the specific downstream tasks (fine-tuning) (Yang et al., 2023).

Meanwhile, although recent advances in deep learning and machine learning have led to breakthrough performance and have been widely applied in practice, both empirical evidence (*e.g.*, (Madry et al., 2017)) and theoretical investigations (*e.g.*, (Haldar et al., 2024)) reveal that deep learning models can be fragile and vulnerable against adversarial input which is intentionally perturbed to mislead the model. To improve the robustness of these models, adversarial training is one of the most popular ways (Madry et al., 2017).

With the new pre-training-fine-tuning paradigm, many studies consider improving adversarial robustness in the downstream task using adversarial pre-training and clean fine-tuning. For example, some works empirically observe the robustness of a robust pre-trained model using robust contrastive learning being inherited to downstream tasks (Shafahi et al., 2019; Salman et al., 2020; Deng et al., 2021b; Zhang et al., 2021; Kim et al., 2020; Fan et al., 2021).

However, although the rise of the pre-training-and-fine-tuning paradigm provides one possible way to reduce the computation cost for the downstream users, concerns have been raised regarding the computational cost of adversarial training: Compared to clean training, the cost of adversarial training is much higher since the attacks are recalculated in each training iteration. For example, while clean training of ResNet18 for CIFAR-10 takes 1 hour on a single NVIDIA V100 GPU, adversarial training can take more than 20 hours (Rice et al., 2020). Consequently, in the pre-training-fine-tuning paradigm, though pre-training parties are mostly resource-abundant entities like OpenAI, adversarial training can still be burdensome as the computation cost to clean pre-train a GPT-3 alone is already estimated to be up to $4.6m (Ohiri & Poole, 2024). The infeasible adversarial training cost leaves an open question to be answered:

How to leverage adversarial training in foundation models to maintain a low computation cost?

To address the above challenge, we provide a theoretical

[*]Equal contribution [1]Department of Statistics, Michigan State University, Michigan, United States [2]School of Computing and Information Systems, University of Melbourne, Melbourne, Australia. Correspondence to: Meiqi Liu <liumeiqi@msu.edu>, Zhuoqun Huang <calvin.huang@unimelb.edu.au>.

arXiv:2504.10850v1 [cs.LG] 15 Apr 2025

Figure 1: Use a robust auto-encoder to pre-process the downstream data. After obtaining the pre-trained foundation model, we use adversarial training to train a robust auto-encoder via leveraging adversarial contrastive loss. A robust auto-encoder is used to pre-process downstream data. These pre-processed inputs are then fed into the foundation model.

analysis to bound the downstream adversarial loss using adversarial contrastive loss. Further, based on the theoretical insights, we design a robust pre-processor to validate the importance of feature robustness (i.e., small adversarial contrastive loss), and pursue robustness in the downstream tasks with only clean foundation models.

Our main contributions are summarized as follows:

- We provide a theoretical analysis of the downstream adversarial loss. Based on our derivation, for classification tasks with cross-entropy loss, the downstream adversarial loss can be upper-bounded by a combination of the downstream clean loss and the adversarial contrastive loss (Theorem 1).

- In addition, in an (auto-encoder + foundation model + last layer adaptation) system, naively trained auto-encoders with only reconstruction loss cannot effectively output robust data (Proposition 1).

- Inspired from the above theoretical insights, we consider the following robust data pre-processor as in Figure 1: We train an auto-encoder leveraging adversarial contrastive loss to obtain a robust auto-encoder as a data pre-processor and feed the corrupted data to the pre-processor first and then feed the foundation model with the output of the pre-processor.

Specifically, the robust auto-encoder consists of an encoder-decoder structure, which is trained in an unsupervised manner without requiring labels. The pre-processed output is then passed to the foundation model for downstream tasks. The training of the auto-encoder is guided by adversarial contrastive loss, ensuring that the latent representations are robust against adversarial perturbations while maintaining the unsupervised nature of the learning process.

While existing literature, e.g., (Salehi et al., 2021; Zhou et al., 2023), attempts to add adversarial training, they need to access the foundation model in adversarial

training. In contrast, we do not access the foundation model when training the robust pre-processor. We name our simple-yet-effective approach as **C**ontrastive **Ro**bust **P**reprocessing **D**efense (*CRoPD*).

- We empirically evaluate the robustness of *CRoPD* using multiple datasets. Our results show a significant improvement in robustness in downstream tasks with only a clean foundation model. The empirical observations echo our theoretical observations, and highlight the importance of the feature robustness in robust supervised learning.

2. Preliminaries

2.1. Data Distribution

We consider classification in the downstream task. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i is the input data and $y_i \in \mathcal{Y}$ is the corresponding target values with label space $\mathcal{Y}$, we assume that these data points are drawn from an unknown joint probability distribution $q(x, y)$.

2.2. Adversarial Training

To formulate adversarial training, we first define adversarial attack. Given a loss function $\ell(\cdot, \cdot)$ and a model $f(\cdot)$, an adversarial attack aims to figure out the worst-case perturbation which maximizes the loss, i.e.,

$$x^{\mathrm{adv}} \triangleq \arg \max_{\bar{x} \in \mathcal{A}(x)} \ell(f(\bar{x}), \cdot),$$

where $\mathcal{A}(x)$, referred to as the threat model, defines the set of permissible adversarial perturbations for the input x. Specifically,

$$\mathcal{A}(x) = \{\bar{x} : \|\bar{x} - x\|_p \leq \epsilon\},$$

where p-norm ($p = 2$ or ∞) determines the metric used, and ϵ specifies the attack budget. For example, when $p = \infty$,

$\mathcal{A}(x)$ forms an $\mathcal{L}_\infty$-ball around x with radius ϵ, i.e., the pixel attack in image data. The second argument of $\ell(\cdot, \cdot)$ can be either the label for supervised learning (classification), or $f(x')$ for some other data x' in contrastive learning.

After defining the attack, adversarial training is a generic algorithm to iteratively train the neural network: In each iteration, given the current model f, we first calculate the attacked version of the training data using the above definition of adversarial attack, and then use the attacked data to calculate the loss and update the model correspondingly.

For clean training and adversarial training in this paper, we use "clean training" to minimize the clean loss and use "adversarial training" to minimize the adversarial loss.

2.3. Contrastive Learning and Robustness Inheritance

Different from supervised learning, contrastive learning is an unsupervised learning method and does not need labels/responses. The aim of contrastive learning is to figure out an encoder $f_{\text{en}}(\cdot)$ so that the similarity between similar pairs (positive pairs) of data is maximized, while the similarity between dissimilar pairs (negative pairs) is minimized. To theoretically connect contrastive learning with supervised learning, Arora et al. (2019) formalize the semantic similarity using latent classes and prove that minimizing the contrastive loss leads to a representation function that achieves a low average linear classification loss on downstream tasks. They establish that under certain conditions, the contrastive loss serves as an upper bound of the expected supervised loss, theoretically supporting the effectiveness of contrastive learning in downstream supervised tasks.

Extending from the clean contrastive learning, the aim of adversarial contrastive learning is to figure out an encoder $f_{\text{en}}(\cdot)$, and the adversarial contrastive loss L_{con} is

$$L_{\text{con}}(f_{\text{en}}) = \mathbb{E}_{q(x)}\left[\ell_{\text{con}}(f_{\text{en}}(x^{\text{adv}}), f_{\text{en}}(x))\right], \quad (1)$$

Empirically, this expectation is approximated by averaging the loss over n samples from the dataset:

$$\widehat{L}_{\text{con}}(f_{\text{en}}) = \frac{1}{n}\sum_{i=1}^{n}\ell_{\text{con}}(f_{\text{en}}(x_i^{\text{adv}}), f_{\text{en}}(x_i))).$$

The loss ℓ_{con} is defined as

$$\ell_{\text{con}}(f_{\text{en}}(x_i^{\text{adv}}), f_{\text{en}}(x_i))$$
$$= -\log\frac{\exp(\text{sim}(f_{\text{en}}(x_i^{\text{adv}}), f_{\text{en}}(x_i))/\tau)}{\sum_{x^{\text{neg}}\in X^{\text{neg}}}\exp(\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(x^{\text{neg}}))/\tau)},$$

where $\text{sim}(\cdot, \cdot)$ represents the cosine similarity between two latent vectors, defined as

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}.$$

The above formulation allows for adversarial examples to be integrated into the contrastive loss by maximizing the similarity between clean and adversarial representations while minimizing similarities with negative samples.

To construct the dissimilar set X^{neg} in the above, we first take X^{neg} as a set of clean examples $X = \{x_1, \ldots x_M\}$ to calculate the corresponding x_j^{adv}'s as the approximation to form a set X^{adv}, and then take $X^{\text{neg}} = X \bigcup X^{\text{adv}}\setminus\{x_i, x_i^{\text{adv}}\}$ when figuring out the x_i^{adv} on the numerator of ℓ_{con}.

Regarding the robustness inheritance phenomenon in adversarial contrastive learning, the common scenario is to use adversarial training to train a contrastive model and then use clean training to replace the projector of the contrastive model (the last linear layer) to adapt to the downstream task. In this training paradigm, the downstream robustness is significantly better than training the downstream task itself from scratch (Kim et al., 2020).

3. Robust Data Pre-Processing

We first present the theoretical analysis in Section 3.1 and the potential issue in auto-encoder in Section 3.2, and then introduce the considered practical algorithm in Section 3.3.

3.1. Main Theory

To analyze the robustness of the downstream task T with conditional distribution $q(y \mid x)$, the cross-entropy loss (i.e., ℓ_{sup}) given the encoder parameter θ is

$$\mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|x)}[-\log\hat{p}_T(y \mid z)]\right], \quad (2)$$

and the corresponding adversarial loss is

$$\mathbb{E}_{q(x,y)}\left[\max_{x^{\text{adv}}\in\mathcal{A}(x)} -\log\mathbb{E}_{p_\theta(z|x^{\text{adv}})}[\hat{p}_T(y \mid z)]\right], \quad (3)$$

where $\hat{p}_T$ is the model's output probability of label y. We aim to learn a classifier $\hat{p}_T(y \mid f_{\text{de}}(f_{\text{en}}(x)))$ that predicts y based on the robust decoded features $f_{\text{de}}(f_{\text{en}}(x))$.

The following theorem describes how the adversarial attack impacts the downstream classification performance, and how it is related to the adversarial contrastive loss:

Theorem 1. *Assume for all x, the encoder $f_{\text{en}}(x)$ generates a robust latent feature z such that $\|f_{\text{en}}(x^{\text{adv}}) - f_{\text{en}}(x)\|\leq \eta_1$, where η_1 is small, and for all pairs (x_1, y_1), (x_2, y_2) with $y_1 \neq y_2$, it holds that $\|f_{\text{en}}(x_1) - f_{\text{en}}(x_2)\|\geq \eta_2$ and $\|f_{\text{en}}(x_1) - f_{\text{en}}(x_2^{\text{adv}})\|\geq \eta_2$, where $\eta_2 > \eta_1$ is a larger constant. Additionally, assume that $-\log\hat{p}_T(y \mid f_{\text{de}}(z)) \leq M$ for all $z \in \mathcal{Z}$ and $y \in \mathcal{Y}$. Then, for some constant κ,*

$$\mathbb{E}_{q(x,y)}\left[\max_{x^{\text{adv}}\in\mathcal{A}(x)} -\log\hat{p}_T(y \mid f_{\text{de}}(f_{\text{en}}(x^{\text{adv}})))\right]$$

$$\leq \mathbb{E}_{q(x,y)} \left[-\log \hat{p}_T(y \mid f_{\text{de}}(f_{\text{en}}(x))) \right] + \kappa L_{\text{con}}(f_{\text{en}}) \quad (4)$$

where $L_{\text{con}}(f_{\text{en}})$ is the robust contrastive loss defined in (1).

The proof of Theorem 1 can be found in Appendix D. Theorem 1 illustrates how the downstream adversarial loss can be connected to the adversarial contrastive loss: it is upper bounded by the downstream clean loss plus the adversarial contrastive loss. This implies that, if a data pre-processor can achieve a small adversarial contrastive loss, it can also lead to the robustness of the downstream classification.

3.2. Potential Issue in Adversarial Reconstruction Loss

To highlight the importance of the feature robustness (i.e., a small adversarial contrastive loss), when there is no data pre-processing, or we only attack on the reconstruction loss to train the auto-encoder (the benchmark *ARAE*), we further show that these scenarios can result in a poor downstream robustness. The following proposition provides an example:

Proposition 1. *There exists a data generation model* (x, y), *an auto-encoder* $(f_{\text{en}}, f_{\text{de}})$, *and a classifier* $(f_{\text{pre}}, f_{\text{last}})$ *such that if the clean reconstruction loss* $\mathbb{E}_x \left[\|f_{\text{de}}(f_{\text{en}}(x)) - x\|^2 \right]$ *and the adversarial reconstruction loss* $\mathbb{E}_x \left[\|f_{\text{de}}(f_{\text{en}}(x^{\text{adv}})) - x\|^2 \right]$ *for adversarial examples* x^{adv} *within a perturbation budget* ϵ *are 0 and* $O(1/n)$ *respectively, then the adversarial classification loss satisfies*

$$\mathbb{E}_{q(x,y)} \left[\max_{x^{\text{adv}} \in \mathcal{A}(x)} -\log \hat{p}_T \left(y \mid f_{\text{last}}(f_{\text{pre}}(f_{\text{de}}(f_{\text{en}}(x^{\text{adv}})))) \right) \right]$$
$$\geq \Gamma > \mathbb{E}_{q(x,y)} \left[-\log \hat{p}_T \left(y \mid f_{\text{last}}(f_{\text{pre}}(x)) \right) \right] = O(\delta), \quad (5)$$

where $\Gamma = O(-\log(\delta))$ is a large value, and $\delta = o(1/n)$ associated with the data distribution.

Proposition 1 is a possible scenario where the auto-encoder is trained to achieve a small adversarial reconstruction loss while the downstream classification task has a poor robustness. It constructs a discrete data distribution with well-separated points and a classifier that assigns high probability to clean inputs and low probability to perturbed inputs. The detailed proof can be found in Appendix D.

Proposition 1 underscores the need for a better approach for robust auto-encoder. Further inspired by Theorem 1, by integrating adversarial contrastive loss when training the pre-processors, the robustness issue in auto-encoder can be mitigated. We present the algorithm as follows.

3.3. Practical Algorithm

To design the robust data pre-processor, we leverage adversarial contrastive learning when training the robust auto-encoder. Unlike the original contrastive learning framework, we assume that the foundation model already exists and try to avoid accessing it to reduce computational costs. In this case, since the foundation model receives images as inputs,

we need to develop a model to pre-process the images before feeding them into the foundation model. Consequently, we leverage contrastive learning to train an auto-encoder, the latter of which is supposed to output an image. The proposed algorithm is summarized in Algorithm 1, and the graphical illustration can be found in the above Figure 1.

There are several components in whole framework:

First, for the robust auto-encoder, assume we have a robust auto-encoder defined by an encoder $f_{en}(x)$ and a decoder $f_{de}(z)$, where $z = f_{en}(x)$ represents the latent feature of the input x. This auto-encoder is trained using a combination of reconstruction loss and adversarial contrastive loss. The reconstruction loss, $\|f_{de}(f_{en}(x)) - x\|^2$ ensures that the decoder f_{de} can accurately reconstruct the original input from the encoded features. The adversarial contrastive loss, $L_{\text{con}}(f_{en}(x^{adv}), f_{en}(x))$ promotes robustness of the features $f_{en}(x)$ against adversarial perturbations x^{adv}.

Second, in addition to the robust auto-encoder, since the output format of the foundation model might be different from the downstream task, we further train a new last layer on top of the foundation model. Recall that the output of the robust auto-encoder is $f_{de}(f_{en}(x))$, the output of the foundation model then becomes $f_{pre}(f_{de}(f_{en}(x)))$. After passing this to the last linear layer, we get the output as $f_{last}(f_{pre}(f_{de}(f_{en}(x))))$, and we minimize the downstream loss L_{sup}.

To connect with Theorem 1, In *CRoPD*, since we leverage adversarial contrastive loss in training the auto-encoder, its value is well controlled. In contrast, similar to Proposition 1, when there is no data pre-processing, or we only attack on the reconstruction loss to train the auto-encoder, there is no expectation on how the adversarial contrastive loss behaves in those models, and the robustness can be poor.

4. Experiments

In the experiments, we aim to demonstrate the effectiveness of the proposed robust pre-processor method. The expected result is that, *CRoPD* leads to robustness much stronger than using a non-robust-contrastive-learning-based data pre-processor, highlighting the importance of feature robustness (a small adversarial training loss). In addition, since the robust pre-processor is a small model, the final adversarial robustness of the downstream task may be a bit worse than using adversarial training to fine-tune the foundation model. However, the computational cost of *CRoPD* is much smaller than fine-tuning a foundation model using adversarial training.

Algorithm 1 **C**ontrastive **Ro**bust **P**reprocessing **D**efense

---

1: Use pre-training dataset $\mathcal{S}_{pretrain}$ to train a neural network $f_{last}(f_{pre}(\cdot))$.

2: Use the downstream dataset $\mathcal{S}_{down}$ to train a robust auto-encoder via minimizing the loss

$$\min_{f_{en}, f_{de}} \sum_{x \in \mathcal{S}_{down}} \|f_{de}(f_{en}(x)) - x\|^2 + \lambda \sup_{x^{adv} \in \mathcal{A}(x)} L_{con}(f_{en}(x^{adv}), f_{en}(x)),$$

3: Use the labeled downstream dataset $\mathcal{S}_{label}$ to adjust the last layers f_{last} for the downstream task:

$$\min_{f_{last}} \sum_{(x,y) \in \mathcal{S}_{label}} L_{sup}(f_{last}(f_{pre}(f_{de}(f_{en}(x)))), y).$$

4.1. Experimental Setups

Datasets We conduct experiments on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011) and ImagenetTe (Howard, 2020) (a subset of 10 classes from Imagenet (Deng et al., 2009)). We also consider a special variation of CIFAR-10, dubbed CIFAR-2, that subsets the first two classes of CIFAR-10 (*airplane* and *automobile*) for computationally intensive robust training experiments. We use the original dataset split to train and evaluate our models. We briefly describe these datasets below: **CIFAR-10**: This dataset consists of $60\,000 - 32 \times 32$ color images across 10 categories. **CIFAR-2**: A binary subset of CIFAR-10, containing only the first two classes (airplane and automobile). **CIFAR-100**: This dataset contains $60\,000 - 32 \times 32$ color images with 100 categories. **SVHN**: This dataset contains $630\,420 - 32 \times 32$ color images of digits (0-9) cropped from house numbers in Google Street View images. **ImagenetTe**: A subset of 13 000 images of 10 classes from the ImageNet dataset (Deng et al., 2009).

Pre-processors Following Zhou et al. (2023), we use a variant of ViT-MAE architecture that utilizes 50% deterministic masking for consistent reconstruction. The detailed configuration is postponed to Appendix A.

In addition to *Vanilla*, which utilizes an auto-encoder trained with reconstruction loss only, we also include two other pre-processor baselines, *ARAE* (Salehi et al., 2021) and pre-trained *VAE* (Kingma & Welling, 2014) from HuggingFace Diffusers (von Platen et al., 2022). For *ARAE*, although it utilizes adversarial training to train an auto-encoder, the main purpose is to improve the output quality of the auto-encoder rather than distinguishing similar and dissimilar data. We also include *VAE* because its denoising capabilities may also mitigate adversarial attacks. Finally, *Identity* represents the case without any pre-processor and is the most vulnerable baseline.

Following observations by Chen et al. (2020b), though ViT-MAE naturally outputs latent embeddings, it is ineffective for *CRoPD* or *ARAE* if we naively use this embedding to

align the latents. As a result, we use pooling and a two-layered projector to reduce the latent to a 128-dimension vector for training *CRoPD* and *ARAE*.

Before performing the downstream task, we first train the *CRoPD*, *Vanilla*, and *ARAE* using the downstream dataset to obtain pre-processors. Some sample image reconstructions by *ARAE* and *CRoPD* are demonstrated in Figure 2.

Figure 2: Sample image reconstructions of each dataset. Top row: original images, middle row: *ARAE* reconstructions, bottom row: *CRoPD* reconstructions. Columns correspond to different datasets. *ARAE* reconstructions are sharper as expected, while *CRoPD* reconstructions are purified and more robust for downstream tasks.

Foundation Model and Downstream Task For all experiments, we use the HuggingFace Transformers (Wolf et al., 2020) to load a pre-trained large ViT-MAE (He et al., 2021) as the foundation model. Due to the different characteristics of the datasets, we consider different ways of using the foundation model, and postpone the details to Appendix A. Finally, to perform classification for all the scenarios above, we instantiate and train a linear layer that maps features of the foundation model to labels.

Attacks We evaluate the robustness of the system using a true white-box attack scenario and PGD with two settings

Pre-processor	Fine-tuning foundation	Clean				Robust			
		Natural	PGD-10	PGD-20	AutoAttack*	Natural	PGD-10	PGD-20	AutoAttack*
Identity		**99.54**	32.78±2.15	19.14±1.78	2.11±0.63	**97.28**	73.46±1.96	65.76±2.11	45.28±2.14
+ LoRA		98.44	43.11±2.20	33.55±2.09	8.38±1.23	97.54	65.50±2.08	55.96±2.16	28.18±2.04
VAE		94.71	57.23±2.18	46.45±2.20	34.78±2.04	92.80	71.47±2.03	62.54±2.14	56.03±2.15
Vanilla		99.31	34.10±2.03	22.09±1.77	7.87±1.20	97.05	73.92±1.96	66.84±2.07	55.45±2.11
ARAE		99.20	46.84±2.17	36.22±2.08	70.62±1.98	97.29	78.83±1.79	73.16±1.93	81.78±1.67
CRoPD, $\lambda = 0.1$		99.00	45.57±2.12	34.65±1.94	19.48±1.76	97.30	77.07±1.85	71.10±1.99	63.90±2.13
CRoPD, $\lambda = 1$		96.81	82.04±1.69	77.42±1.83	87.06±1.48	96.31	88.00±1.45	85.64±1.55	91.80±1.24
CRoPD, $\lambda = 10$		96.66	**84.60±1.60**	**80.28±1.73**	**90.56±1.30**	96.41	**89.04±1.38**	**87.03±1.44**	**93.39±1.07**
Identity	✓	99.40	70.13±2.02	64.45±2.17	21.02±1.83	99.30	95.96±0.87	95.39±0.92	96.41±0.83

Table 1: Natural and robust performance comparison of different pre-processors on CIFAR-2. The results are presented for both clean and robust trained (Robust) models, evaluated under clean conditions and against adversarial attacks (PGD-10, PGD-20, and AutoAttack). The best values in each column, excluding full-rank fine-tuned models, are highlighted. *CRoPD* achieves the best robust accuracies for both PGD and AutoAttack, beating the runner-up by 11% to 27%.
* We use APGD-CE, FAB, Square Attack for CIFAR-2, because the other attacks are not compatible with the binary classification task (require at least 4 classes).

(PGD-10 and PGD-20). Adversarial examples are dynamically generated during evaluation by calculating gradients through the entire pipeline, which includes the robust auto-encoder (encoder and decoder), the foundation model, and the linear classifier. Specifically, we use an $\mathcal{L}_\infty$ attack with a maximum perturbation strength $\epsilon = 8/255$ for CIFAR-2, CIFAR-10, and CIFAR-100, and $\epsilon = 4/255$ for Imagenette, consistent with the training setup. Unlike training, where FGSM is applied, evaluation employs iterative PGD attacks to generate more challenging adversarial samples. For PGD-10, the attack is iterated for 10 steps with each step size limited to $\epsilon/5$. Similarly, PGD-20 performs 20 iterations with each step size limited to $\epsilon/10$.

4.2. Binary Classification

Since CIFAR-2 contains only 10 000 training samples and the computation is much less expensive than the other datasets, we benchmark the performance of all methods on CIFAR-2 with both clean and adversarial training. This includes the proposed method *CRoPD*, other pre-processors (*Vanilla*, *ARAE*, *VAE*), no pre-processor (*Identity*), and a benchmark of fine-tuning the whole foundation model using clean and adversarial training. Among all settings, robust fine-tuning is far more expensive than clean training, up to 10 times slower. The slow convergence rate of robust fine-tuning makes it the slowest approach.

The results are summarized in Table 1. In general, *CRoPD* outperforms the other methods when trained with both clean and adversarial training. The following presents some details about the experiment results.

First, as mentioned in the experiment setup above, we provide a benchmark of robust fine-tuning for the foundation model. From Table 1, it achieves a robust test accuracy of

around 96% for both PGD and AutoAttack, which is the highest compared to others.

Second, comparing (*Identity*, Clean) and the other settings with a pre-processor, without a pre-processor, all the small perturbations/noise contained in the input are fed into the foundation model, which leads to the worst robust test accuracy of only 32.78% for PGD-10, 19.14% for PGD-20 and 2.11% for AutoAttack. Consistent with our theorem, the naïve approach to adversarial auto-encoder, *Vanilla*, only provides a slight improvement to robust accuracy compared to *Identity*. Besides, comparing (*Identity*, Clean) with (*Identity*, Robust), when training the last linear layer using adversarial training, the adversarial test accuracy can be significantly improved to 73.46%, 65.76% and 45.28% for PGD-10, PGD-20 and AutoAttack. However, these accuracy values are still much lower than the expensive, robust fine-tuning setting of around 96%.

For the proposed method *CRoPD*, the weight for the adversarial contrastive loss (λ) controls the clean and robust accuracy trade-off as expected: The higher the λ is, the more we favor robust accuracy over clean accuracy. Recall that the no pre-processor (*Identity*) robust training with and without fine-tuning achieves around 95% and 70% robust accuracy, respectively. When taking the best $\lambda = 10$ and training the linear classification layer using clean training, robust accuracy of *CRoPD* outperforms robust training without fine-tuning by upto 48% with less than 1% degradation in clean accuracy. When the linear layer of *CRoPD* ($\lambda = 10$) is trained with robust training, the robust accuracy can be further enhanced to 89.04% for PGD-10, 87.03% for PGD-20 and 93.39% for AutoAttack, Lastly, our proposed method significantly improves the robustness of the whole system by merely compromising around 3% in the natural test accuracy.

On the other hand, for the other pre-processors, to compare *Vanilla* with *VAE* and *ARAE*, when the last linear layer is trained using clean training, the adversarial test accuracy can be significantly enhanced under both PGD and AutoAttack. Although these two methods are not specifically designed to remove adversarial attacks, since they still perform de-noising, they are expected to be robust to adversarial attacks to some extent. This aligns with the literature that random smoothing can defend against adversarial attack (Cao & Gong, 2017; Liu et al., 2018; Cohen et al., 2019). However, the robustness is not as strong as models trained from adversarial training, i.e., the adversarial test accuracies are much lower than 89.04% and 87.03% achieved by our proposed method *CRoPD*. We also try LoRA as a surrogate for faster fine-tuning, but the performance is as similar as *Identity* only.

4.3. Multi-class Experiments

In this section, we conduct experiments using full datasets for multi-class tasks, including CIFAR-10, ImagenetTe, SVHN, CIFAR-100, and Tiny-Imagenet. The results for all five datasets are summarized in Table 2. Similar to CIFAR-2, *CRoPD* generally has the largest improvement on the adversarial test accuracy compared to other methods, especially for CIFAR-10 (2.83% to 47.99%).

Besides, although in some scenarios, e.g., (*CRoPD* vs. *VAE* for ImagenetTe), (*CRoPD* vs. *Vanilla* for SVHN and CIFAR-100), some other methods achieve similar performance to *CRoPD* under PGD-10, *CRoPD* achieves better robustness under PGD-20 and AutoAttack in general. The similar performance of the other methods compared to *CRoPD* is also caused by special issues: For ImagenetTe, *VAE* is originally trained using a superset of ImagenetTe, so it is not surprising that it attains the highest performance under weak attacks; For CIFAR-100, the inflated number of classes requires more features to be retained. Combined with small input dimension, the impact of the contrastive loss is weakened, resulting in a similar performance between *Vanilla* and *CRoPD*. This is further validated by results on Tiny-Imagenet, a dataset with even larger number of classes but much larger input dimension, where *CRoPD* outperforms *Vanilla* and *ARAE* by a large margin under PGD-10 and PGD-20, Additional ablation study on this hypothesis is provided in Appendix B.2.

4.4. Transfer Ability

In this experiment, we use CIFAR-10/CIFAR-100 to train the pre-processor and evaluate the adversarial test accuracy on CIFAR-100/CIFAR-10 respectively[1]. When leveraging contrastive learning, the auto-encoder is expected to com-

---

[1]Same as previous settings, we fine-tune the foundation model for CIFAR-100 and leave CIFAR-10 foundation model as is.

Pre-processor	Natural	PGD-10	PGD-20	AutoAttack
Identity	**91.56**	$2.83_{\pm 0.33}$	$1.19_{\pm 0.21}$	$5.08_{\pm 0.44}$
+ LoRA	88.10	$3.90_{\pm 0.37}$	$1.78_{\pm 0.26}$	—
VAE	66.62	$14.10_{\pm 0.68}$	$7.70_{\pm 0.52}$	$15.27_{\pm 0.70}$
Vanilla	90.93	$10.04_{\pm 0.61}$	$4.95_{\pm 0.42}$	$6.93_{\pm 0.51}$
ARAE	87.96	$18.38_{\pm 0.76}$	$7.73_{\pm 0.51}$	$13.44_{\pm 0.65}$
CRoPD	79.31	$\mathbf{47.99_{\pm 0.98}}$	$\mathbf{40.31_{\pm 0.99}}$	$\mathbf{66.05_{\pm 0.95}}$

(a) CIFAR-10

Pre-processor	Natural	PGD-10	PGD-20	AutoAttack
Identity	96.89	$30.97_{\pm 1.44}$	$21.57_{\pm 1.26}$	$1.69_{\pm 0.42}$
+ LoRA	**96.97**	$33.28_{\pm 1.47}$	$21.43_{\pm 1.29}$	—
VAE	96.57	$\mathbf{59.91_{\pm 1.51}}$	$45.31_{\pm 1.57}$	$13.03_{\pm 1.10}$
Vanilla	92.30	$54.55_{\pm 1.52}$	$47.88_{\pm 1.58}$	$30.62_{\pm 1.45}$
ARAE	93.32	$57.32_{\pm 1.55}$	$53.69_{\pm 1.56}$	$34.41_{\pm 1.46}$
CRoPD	83.84	$58.15_{\pm 1.59}$	$\mathbf{54.80_{\pm 1.49}}$	$\mathbf{60.68_{\pm 1.54}}$

(b) ImagenetTe

Pre-processor	Natural	PGD-10	PGD-20
Identity	**99.85**	$33.60_{\pm 0.35}$	$30.33_{\pm 0.36}$
VAE	88.09	$14.17_{\pm 0.26}$	$8.27_{\pm 0.20}$
Vanilla	99.79	$93.43_{\pm 0.18}$	$67.79_{\pm 0.34}$
ARAE	99.21	$59.08_{\pm 0.35}$	$43.32_{\pm 0.37}$
CRoPD	99.78	$\mathbf{95.81_{\pm 0.14}}$	$\mathbf{95.22_{\pm 0.16}}$

(c) SVHN

Pre-processor	Natural	PGD-10	PGD-20	AutoAttack
Identity	**81.11**	$6.15_{\pm 0.47}$	$4.39_{\pm 0.41}$	$9.79_{\pm 0.58}$
VAE	44.21	$6.67_{\pm 0.50}$	$3.82_{\pm 0.38}$	$13.76_{\pm 0.67}$
Vanilla	80.09	$\mathbf{58.18_{\pm 0.99}}$	$\mathbf{55.94_{\pm 0.94}}$	$18.48_{\pm 0.74}$
ARAE	76.02	$50.28_{\pm 1.01}$	$47.86_{\pm 0.96}$	$\mathbf{23.54_{\pm 0.84}}$
CRoPD	78.89	$56.82_{\pm 0.99}$	$54.17_{\pm 0.95}$	$18.25_{\pm 0.75}$

(d) CIFAR-100

Pre-processor	Natural	PGD-10	PGD-20
Identity	67.20	$8.39_{\pm 0.54}$	$7.13_{\pm 0.52}$
VAE	54.43	$15.42_{\pm 0.72}$	$9.97_{\pm 0.59}$
Vanilla	67.23	$13.07_{\pm 0.68}$	$10.70_{\pm 0.60}$
ARAE	67.34	$19.82_{\pm 0.79}$	$16.64_{\pm 0.72}$
CRoPD	65.84	$\mathbf{22.16_{\pm 0.82}}$	$\mathbf{18.73_{\pm 0.78}}$

(e) Tiny-Imagenet

Table 2: Clean and robust accuracy comparison of various pre-processors across CIFAR-10, ImagenetTe, SVHN, CIFAR-100 and Tiny-Imagenet. The results are evaluated under clean conditions and against adversarial attacks (PGD-10, PGD-20, and AutoAttack). The best values in each column are highlighted.

Source	Target	Pre-processor	Natural	PGD-10	PGD-20
CIFAR-10	CIFAR-100	*Identity*	**81.11**	6.15±0.47	4.39±0.41
		ARAE	75.62	49.38±1.00	46.03±0.99
		CRoPD	78.43	**56.41±0.98**	**55.16±0.97**
CIFAR-100	CIFAR-10	*Identity*	**91.56**	2.83±0.33	1.19±0.21
		ARAE	88.04	13.34±0.65	4.73±0.40
		CRoPD	86.46	**21.29±0.80**	**10.84±0.62**

Table 3: Transfer learning results between CIFAR-10 and CIFAR-100 datasets. The table shows the clean and adversarial accuracy for each pre-processor. The best values in each column are highlighted. The pre-processor is trained on the source dataset and applied to reconstruct and classify images in the target dataset. $\lambda = 1$ is used when targeting CIFAR-100, as the foundation model is fine-tuned and more robust, and we set $\lambda = 6$ for the less robust CIFAR-10 foundation model. Compared to *Identity* and *ARAE*, *CRoPD* attained meaningful features and significantly improved robust accuracies.

prehensively learn all features from the data, thus we expect that *CRoPD* can also transfer across datasets, better than other methods. The results are summarized in Table 3.

From Table 3, both *ARAE* and *CRoPD* learn meaningful features for reconstruction, providing excellent natural performance similar to the upper bound by *Identity*. Meanwhile, our *CRoPD* manages to exceed the robust performance of *ARAE* by up to 10% across all settings.

5. Related Works

This section lists related works in the field of adversarial training and contrastive learning.

Adversarially Robust Pre-processing In literature, some existing works, e.g., (Sahay et al., 2019; Zhou et al., 2021; Cann et al., 2022; Zhou et al., 2023), consider implementing a robust data pre-processor to defend against adversarial attacks. Compared to the existing literature, our proposed method avoids utilizing the pre-trained model when training the robust data pre-processor. In contrast, these existing works use the supervised learning loss to train the pre-processor, thus heavily relying on the pre-trained model. Another work Salehi et al. (2021) proposes to design an attack to corrupt the latent space to train a robust auto-encoder to improve the output quality against adversarial attacks.

There is also other literature that designs attacks in auto-encoder, e.g., (Tabacof et al., 2016). Tabacof et al. (2016) designs an auto-encoder that receives an input image of one class but outputs another image similar to the input but belongs to another class.

Adversarial Training There are fruitful studies in the area of adversarial training. For methodology, there are many techniques, e.g., (Goodfellow et al., 2015; Zhang et al., 2019; Wang et al., 2019b; Cai et al., 2018; Zhang et al., 2020a; Carmon et al., 2019; Gowal et al., 2021; Mo et al., 2022; Wang et al., 2022). Theoretical investigations have

also been conducted for adversarial training from different perspectives. For instance, Chen et al. (2020a); Javanmard et al. (2020); Taheri et al. (2021); Yin et al. (2018); Raghunathan et al. (2019); Najafi et al. (2019); Min et al. (2020); Hendrycks et al. (2019); Dan et al. (2020); Wu et al. (2020b); Javanmard & Mehrabi (2021); Deng et al. (2021a); Javanmard & Soltanolkotabi (2022) study the statistical properties of adversarial training. And Sinha et al. (2018); Wang et al. (2019a); Xiao et al. (2022a;b) study the optimization aspect of adversarial training. Lastly, Zhang et al. (2020b); Wu et al. (2020a) work on theoretical issues related to adversarial training with deep learning.

Contrastive Learning Contrastive learning is a popular self-supervised learning algorithm. It uses unlabeled images to train representations that distinguish different images invariant to non-semantic transformations (Mikolov et al., 2013; Oord et al., 2018; Arora et al., 2019; Dai & Lin, 2017; Chen et al., 2020b; Tian et al., 2020; Chen et al., 2020b; Khosla et al., 2020; HaoChen et al., 2021; Chuang et al., 2020; Xiao et al., 2020; Li et al., 2020). Besides empirical studies, there are also many theoretical studies, e.g., (Saunshi et al., 2019; HaoChen et al., 2021; 2022; Shen et al., 2022; HaoChen & Ma, 2022; Saunshi et al., 2022). Based on both empirical and theoretical studies, a common understanding is that contrastive learning can capture the features from the input via comparing positive (similar) and negative (dissimilar) pairs.

6. Conclusion

In this paper, observing the computation challenge in fine-tuning a foundation model using adversarial training, we examine the role of adversarial contrastive learning to seek a strategy where downstream robustness can be obtained without using adversarial training on the foundation model. We theoretically upper bound the downstream adversarial loss by a combination of the downstream clean loss and the adversarial contrastive loss, which implies that if a data

pre-processor can result in a small adversarial contrastive loss, the robustness of the whole system can be improved. Leveraging this insight, an auto-encoder can be used to develop a data pre-processor that purifies the downstream data to remove potential adversarial attacks. Experiments demonstrate that the proposed method results in an improvement in the downstream robustness, highlighting the importance of the feature robustness.

There are two future directions. First, one can deepen the theoretical understanding via considering different data assumption, e.g., the sparse coding model in (Allen-Zhu & Li, 2020). Second, while this work considers image data, the ideas can be borrowed to natural language processing to enhance the robustness of large language models.

Impact Statement

This paper presents work whose goal is to advance the understanding of adversarial robustness. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Allen-Zhu, Z. and Li, Y. Feature purification: How adversarial training performs robust deep learning. *arXiv preprint arXiv:2005.10190*, 2020.

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Cai, Q.-Z., Du, M., Liu, C., and Song, D. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

Cann, A., Colbert, I., and Amer, I. Robust transferable feature extractors: Learning to defend pre-trained networks against white box adversaries. *arXiv preprint arXiv:2209.06931*, 2022.

Cao, X. and Gong, N. Z. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC '17, pp. 278–287, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450353458. doi: 10.1145/ 3134600.3134606. URL https://doi.org/10. 1145/3134600.3134606.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pp. 11192–11203, 2019.

Chen, L., Min, Y., Zhang, M., and Karbasi, A. More data can expand the generalization gap between adversarially robust and standard models. In *International Conference on Machine Learning*, pp. 1670–1680. PMLR, 2020a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020b.

Chuang, C.-Y., Robinson, J., Yen-Chen, L., Torralba, A., and Jegelka, S. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ cohen19c.html.

Dai, B. and Lin, D. Contrastive learning for image captioning. *arXiv preprint arXiv:1710.02534*, 2017.

Dan, C., Wei, Y., and Ravikumar, P. Sharp statistical guarantees for adversarially robust gaussian classification. In *International Conference on Machine Learning*, pp. 2345– 2355. PMLR, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Deng, Z., Zhang, L., Ghorbani, A., and Zou, J. Improving adversarial robustness via unlabeled out-of-domain data. In *International Conference on Artificial Intelligence and Statistics*, pp. 2845–2853. PMLR, 2021a.

Deng, Z., Zhang, L., Vodrahalli, K., Kawaguchi, K., and Zou, J. Adversarial training helps transfer learning via better representations. *arXiv preprint arXiv:2106.10189*, 2021b.

Fan, L., Liu, S., Chen, P.-Y., Zhang, G., and Gan, C. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34:21480–21492, 2021.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*, 2015.

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

Haldar, R., Xing, Y., and Song, Q. Effect of ambient-intrinsic dimension gap on adversarial vulnerability. In *International Conference on Artificial Intelligence and Statistics*, pp. 1090–1098. PMLR, 2024.

HaoChen, J. Z. and Ma, T. A theoretical study of inductive biases in contrastive learning. *arXiv preprint arXiv:2211.14699*, 2022.

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

HaoChen, J. Z., Wei, C., Kumar, A., and Ma, T. Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations. *arXiv preprint arXiv:2204.02683*, 2022.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021. URL https://arxiv.org/abs/2111.06377.

Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. *arXiv preprint arXiv:1901.09960*, 2019.

Howard, J. imagenette, 2020. URL https://github.com/fastai/imagenette/.

Javanmard, A. and Mehrabi, M. Adversarial robustness for latent models: Revisiting the robust-standard accuracies tradeoff. *arXiv preprint arXiv:2110.11950*, 2021.

Javanmard, A. and Soltanolkotabi, M. Precise statistical analysis of classification accuracies for adversarial training. *The Annals of Statistics*, 50(4):2127–2156, 2022.

Javanmard, A., Soltanolkotabi, M., and Hassani, H. Precise tradeoffs in adversarial training for linear regression. In *Conference on Learning Theory*, pp. 2034–2078. PMLR, 2020.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Kim, M., Tack, J., and Hwang, S. J. Adversarial self-supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, 2009.

Li, J., Zhou, P., Xiong, C., and Hoi, S. C. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, pp. 381–397, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01233-5. doi: 10.1007/978-3-030-01234-2_23. URL https://doi.org/10.1007/978-3-030-01234-2_23.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Min, Y., Chen, L., and Karbasi, A. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. *arXiv preprint arXiv:2002.11080*, 2020.

Mo, Y., Wu, D., Wang, Y., Guo, Y., and Wang, Y. When adversarial training meets vision transformers: Recipes from training to architecture. *arXiv preprint arXiv:2210.07540*, 2022.

Najafi, A., Maeda, S.-i., Koyama, M., and Miyato, T. Robustness to adversarial perturbations in learning from incomplete data. In *Advances in Neural Information Processing Systems*, pp. 5542–5552, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Ohiri, E. and Poole, R. What is the cost of training large language models?, Apr 2024. URL https://www.cudocompute.com/blog/what-is-the-cost-of-training-large-language-model

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. *arXiv preprint arXiv:2002.11569*, 2020.

Sahay, R., Mahfuz, R., and El Gamal, A. Combatting adversarial attacks through denoising and dimensionality reduction: A cascaded autoencoder approach. In *2019 53rd Annual conference on information sciences and systems (CISS)*, pp. 1–6. IEEE, 2019.

Salehi, M., Arya, A., Pajoum, B., Otoofi, M., Shaeiri, A., Rohban, M. H., and Rabiee, H. R. Arae: Adversarially robust training of autoencoders improves novelty detection. *Neural Networks*, 144:726–736, 2021.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pp. 5628–5637. PMLR, 2019.

Saunshi, N., Ash, J., Goel, S., Misra, D., Zhang, C., Arora, S., Kakade, S., and Krishnamurthy, A. Understanding contrastive learning requires incorporating inductive biases. *arXiv preprint arXiv:2202.14037*, 2022.

Shafahi, A., Saadatpanah, P., Zhu, C., Ghiasi, A., Studer, C., Jacobs, D., and Goldstein, T. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019.

Shen, K., Jones, R. M., Kumar, A., Xie, S. M., HaoChen, J. Z., Ma, T., and Liang, P. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 19847–19878. PMLR, 2022.

Sinha, A., Namkoong, H., and Duchi, J. C. Certifying some distributional robustness with principled adversarial training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hk6kPgZA-.

Tabacof, P., Tavares, J., and Valle, E. Adversarial images for variational autoencoders. *arXiv preprint arXiv:1612.00155*, 2016.

Taheri, M., Xie, F., and Lederer, J. Statistical guarantees for regularized neural networks. *Neural Networks*, 142:148–161, 2021.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

Wang, Q., Wang, Y., Zhu, H., and Wang, Y. Improving out-of-distribution generalization by adversarial training with structured priors. *arXiv preprint arXiv:2210.06807*, 2022.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, pp. 6586–6595, 2019a.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019b.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Wu, B., Chen, J., Cai, D., He, X., and Gu, Q. Does network width really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2020a.

Wu, D., Wang, Y., and Xia, S.-t. Adversarial weight perturbation helps robust generalization. *arXiv preprint arXiv:2004.05884*, 2020b.

Xiao, J., Fan, Y., Sun, R., Wang, J., and Luo, Z.-Q. Stability analysis and generalization bounds of adversarial training. *arXiv preprint arXiv:2210.00960*, 2022a.

Xiao, J., Qin, Z., Fan, Y., Wu, B., Wang, J., and Luo, Z.-Q. Adaptive smoothness-weighted adversarial training for multiple perturbations with its stability analysis. *arXiv preprint arXiv:2210.00557*, 2022b.

Xiao, T., Wang, X., Efros, A. A., and Darrell, T. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020.

Yang, S., Nachum, O., Du, Y., Wei, J., Abbeel, P., and Schuurmans, D. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*, 2023.

Yin, D., Ramchandran, K., and Bartlett, P. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 2019.

Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pp. 11278–11287. PMLR, 2020a.

Zhang, J., Sang, J., Yi, Q., Yang, Y., Dong, H., and Yu, J. Pre-training also transfers non-robustness. *arXiv preprint arXiv:2106.10989*, 2021.

Zhang, Y., Plevrakis, O., Du, S. S., Li, X., Song, Z., and Arora, S. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. *arXiv preprint arXiv:2002.06668*, 2020b.

Zhou, D., Wang, N., Gao, X., Han, B., Yu, J., Wang, X., and Liu, T. Improving white-box robustness of pre-processing defenses via joint adversarial training. *arXiv preprint arXiv:2106.05453*, 2021.

Zhou, D., Chen, Y., Wang, N., Liu, D., Gao, X., and Liu, T. Eliminating adversarial noise via information discard and robust representation restoration. In *International Conference on Machine Learning*, pp. 42517–42530. PMLR, 2023.

A. Extra Experimental Details

We describe our detailed experimental settings in this section. All experiments in this paper are conducted using a private cluster with Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz and NVIDIA A100 GPUs. We report the confidence intervals by bootstrapping the results equal to the full size of the test set for $1\,000$ repeats.

Foundation Model and Downstream Task   For all experiments, we use the HuggingFace Transformers (Wolf et al., 2020) to load a pre-trained large ViT-MAE (He et al., 2021) as the foundation model. We consider different scenarios of using the foundation model. First, for some datasets, *e.g.*, CIFAR-2, CIFAR-10, and ImagenetTe, their distribution lines up with the pre-trained foundation model. Thus, we freeze the foundation model as-is for these datasets to achieve a low-cost but efficient prediction. Second, we fine-tune the pre-trained foundation model using clean training for better alignments for CIFAR-100 and SVHN. Third, for CIFAR-2, we use adversarial training to fine-tune the foundation model with Projected Gradient Descent (PGD) (Madry et al., 2017), which serves as a benchmark and is expected to be the most robust baseline.

Finally, to perform classification for all the scenarios above, we instantiate and train a linear layer that maps features of the foundation model to labels.

Pre-processor   We use a modified version of ViT-MAE built on top of the codebase by Zhou et al. (2023) for *CRoPD*, *Vanilla*, and *ARAE*. Our ViT-MAE applies deterministic masking with a mask rate of 50% and consists of a 8-layered encoder and 2-layered decoder, where the attentions have 3-heads each. For datasets with lower resolution, we set the patch size to 2 with a 192-dimension embedding, and for ImagenetTe we use a patch size of 14 with a 768-dimension embedding size. Before feeding the latent embeddings to the projector, we first perform an average pooling with a pooling factor of 8. The resulting tensor is then flattened and fed into a two-layered projector with 2048 hidden size and 128 output size. Table 4 shows the parameters used in setting up *CRoPD*, *ARAE* and *VAE*. For reconstruction, we use standard mean squared error instead of binary cross entropy for better performance.

We apply data augmentation (Table 5) and/or adversarial perturbation to the inputs while training the auto-encoders. For the adversarial perturbation, we use the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) with ℓ_{inf}-norm with a max perturbation of $8/255$ and $4/255$ for low- and high-resolution datasets. Finally, the remaining optimization settings are shown in Table 6.

Pre-processor settings	
ARAE	$\gamma = 0.1$
CRoPD	$\lambda = 10$ for CIFAR-10, ImagenetTe
	$\lambda = 1$ for SVHN, CIFAR-100
VAE	diffusers.AutoencoderKL stabilityai/sd-vae-ft-mse-original

Table 4: Pre-processor settings for different datasets.

Augmentation settings
CIFAR-2, CIFAR-10, SVHN, CIFAR-100
RandomResizedCrop(32) RandomHorizontalFlip() RandomApply([ColorJitter(0.4, 0.4, 0.4, 0.1)], p=0.8) RandomGrayscale(p=0.2)
ImagenetTe
RandomResizedCrop(224) RandomHorizontalFlip() RandomApply([ColorJitter(0.4, 0.4, 0.4, 0.1)], p=0.8) RandomGrayscale(p=0.2)

Table 5: Augmentation settings for *CRoPD* and *Vanilla*. Note that *ARAE* does not use augmented images.

Optimization settings	
Optimizer	AdamW
Learning rate	1.5×10^{-4}
Weight decay	5×10^{-2}
Warmup epochs	20
Scheduler	CosineAnnealingLR
Batch size	CIFAR-2: 32 (until epoch 150) 256 (after epoch 150)
	CIFAR-10: 32
	SVHN: 32
	CIFAR-100: 96
	ImagenetTe: 96
Max epochs	CIFAR-2: 400 epochs
	CIFAR-10: 400 epochs
	SVHN: 400 epochs
	CIFAR-100: 150 epochs
	ImagenetTe: 150 epochs

Table 6: Optimization settings for training pre-processors on different datasets.

Foundation Model and Linear Layer   After training the pre-processors, we chain it with the foundation model (`facebook/vit-mae-large`) and train a linear classification layer. As described before, we will also optionally fine-tune the foundation model. Since the ViT-MAE expects 224×224 sized inputs, which mismatch with the image size of CIFAR-10, SVHN and CIFAR-100, we use the differentiable torch bi-linear interpolation to upscale image tensors from their original 32 shape. We also apply the preprocessing normalization steps with parameters described by their configurations. When training the linear classification layer (and optionally fine-tuning the foundation models), we perform optimization until convergence or when it reaches max epochs. For robust training, we use PGD-10 with a larger step size of 0.007 to generate adversarial examples and mix them with natural samples. The other detailed training parameters we used are shown in Table 7.

Optimization settings	
Optimizer	AdamW
Learning rate	1×10^{-2} (classification head only)
	1×10^{-4} (fine-tuning foundation)
Batch size	64
Max epochs	150
LR scheduler	multiply by 0.1 after epochs 30, 70, 100

Table 7: Optimization settings for training linear classification head with optional fine-tuning of the foundation model.

B. Additional Experimental Results

In this appendix, we show some additional experimental results.

Dataset	Method	Clean	PGD-10	PGD-20	Time
CIFAR-10	Encoder-Only	65.58±0.94	28.08±0.86	25.61±0.86	315 sec * 50 epoch
	CRoPD	79.31±0.40	47.99±0.98	40.31±0.99	250 sec * 100 epoch
ImagenetTe	Encoder-Only	66.09±1.47	41.95±1.57	41.57±1.55	310 sec * 90 epoch
	CRoPD	83.84±0.62	58.15±1.59	54.80±1.49	360 sec * 120 epoch

Table 8: Comparison of encoder-only adversarial training and *CRoPD* under PGD attacks.

B.1. Comparison with Encoder-Only Adversarial Training

Table 8 compares two training strategies across the CIFAR-10 and ImagenetTe datasets. The "Encoder-Only" approach employs supervised adversarial training using only the encoder and the downstream dataset. In contrast, *CRoPD* utilizes a robust auto-encoder (encoder+decoder) together with a frozen foundation model. On both datasets, *CRoPD* yields consistently higher accuracy and adversarial robustness. For example, on CIFAR-10, the clean accuracy improves from 65.58% to 79.31%, while PGD-10 and PGD-20 robustness increase from 28.08 and 25.611% to 47.991% and 40.311%, respectively. A similar trend is observed for ImagenetTe. Although *CRoPD* requires longer training times (250 sec * 100 epoch for CIFAR-10 and 360 sec * 120 epoch for ImagenetTe) compared to the Encoder-Only method, this additional computational cost is offset by the significant performance gains achieved by leveraging robust contrastive learning and the foundation model.

B.2. Why *CRoPD* perform worse on CIFAR-100?

Sample size	λ	Clean	PGD-10	PGD-20
10%	0.15	79.28±0.82	11.92±0.64	8.43±0.55
20%	1	79.81±0.79	16.01±0.73	9.98±0.58
50%	10	81.85±0.78	25.75±0.84	14.33±0.71
100%	10	79.31±0.79	47.99±0.98	40.31±0.99

Table 9: Ablation study of *CRoPD* on CIFAR-10 by reducing the training set to 10%, 20%, 50% and 100%. We use different λ values to balance the reconstruction and contrastive objectives given the size of the training set. *CRoPD* shows substantial gains with increased training data, underscoring its data efficiency and scalability.

In CIFAR-100, the proposed method exhibits limited performance compared to its results on CIFAR-10. This difference is primarily due to CIFAR-100 providing significantly fewer samples per class, which makes it more challenging to learn the well-separated robust features required by our adversarial contrastive loss. With limited per-class data, the preprocessor struggles to learn meaningful representations. In this context, a smaller adversarial contrastive loss weight (e.g., $\lambda = 0.1$) might better balance the reconstruction and contrastive objectives; note that the *Vanilla* baseline corresponds to $\lambda = 0$.

To validate that data scarcity, not merely the number of classes, is the core issue, we conducted an ablation study on CIFAR-10 by reducing its training set to 10%, 20%, and 50%, thereby matching the per-class data scale of CIFAR-100 (see Table 9). *CRoPD* demonstrates substantial gains with increased training data, underscoring its data efficiency and scalability.

These observations imply that the lower performance observed on CIFAR-100 does not indicate a fundamental limitation of *CRoPD* in multi-class settings. Instead, they highlight the critical importance of sufficient per-class data in achieving high robustness, which is consistent with our theoretical insights (Theorem 1). With adequate data, *CRoPD* consistently outperforms reconstruction-only baseline (*Vanilla*) in both clean accuracy and adversarial robustness.

B.3. ROCL Trained Foundation Model

Table 10 demonstrates another set of experiments on CIFAR-2 with a custom-trained model (using RoCL). Consistent with Table 1, *CRoPD* out-performs the competing benchmark *ARAE* by a large margin. Surprisingly, *CRoPD* even outperforms the (*Identity*, fine-tune) setting, though the gap is narrower.

Pre-processor	Fine-tuning foundation	Clean			Robust		
		Natural	PGD-10	PGD-20	Natural	PGD-10	PGD-20
Identity		**98.41**	1.40±0.53	0.74±0.38	**98.77**	0.21±0.19	0.00±0.00
Vanilla		98.31	1.00±0.42	0.35±0.25	96.08	9.15±1.26	5.04±0.96
ARAE		92.56	32.83±1.98	27.11±2.01	89.67	50.38±2.11	45.41±2.11
CRoPD, $\lambda = 0.1$		98.26	2.61±0.68	1.40±0.51	96.34	13.83±1.52	11.37±1.39
CRoPD, $\lambda = 1$		95.05	**75.98±1.86**	70.42±2.04	94.62	79.48±1.76	75.66±1.81
CRoPD, $\lambda = 10$		95.16	75.27±1.94	**71.60±1.96**	94.91	**79.59±1.72**	**76.64±1.80**
Identity	✓	98.77	0.21±0.19	0.00±0.00	97.66	76.51±1.87	71.83±2.00

Table 10: Natural and robust performance comparison of different pre-processors on CIFAR-2 with a custom foundation model trained with *RoCL*. The results are presented for both clean and robust trained (Robust) models, evaluated under clean conditions and against adversarial attacks (PGD-10 and PGD-20). The best values in each column, excluding fine-tuned models, are highlighted. All foundation model weights are frozen during training except for the fine-tuned models. *CRoPD* achieves competitive robust accuracies for both PGD-10 and PGD-20, demonstrating strong performance.

C. Computation Requirements

We conduct all experiments using a private cluster with Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz and NVIDIA A100 GPUs. We execute our experiments using six cores, 48G Memory, and 1 GPU. We document the estimated experiment runtime for CIFAR-2 experiments in Table 11. The total computation required by our method (*CRoPD*+Linear) can be much cheaper than that of robust fine-tuning. In addition, we also report the parameter counts for each model in Table 12. Our method only requires training of a fraction of parameters compared to the foundation model employed.

Dataset	*CRoPD* (h)	Linear (h)	Fine-tune (h)	Robust fine-tune (h)
CIFAR-100	5.0	3.0	7.3	19.6
ImagenetTe	9.6	1.7	3.6	22.3

Table 11: Estimated computation cost for training each model type on CIFAR-100 and ImagenetTe. The training time of our method includes the *CRoPD* and Linear columns, while the alternative adversarial fine-tuning can take drastically longer. Regular fine-tuning, though beneficial for robustness, can taker longer than the Linear to train due to slower convergence.

Dataset type	*CRoPD*	Foundation	Fraction
Small	1.8×10^7	3.0×10^8	6.1%
Large	8.3×10^7	3.0×10^8	27.6%

Table 12: Parameter counts of models for different datasets. Fraction shows the Small and large dataset types refer to non-ImagenetTe and ImagenetTe. By incorporating a pre-processor, we drastically reduced the number of parameters to tune during training by more than 90% and 70%, respectively.

D. Proofs

Outline of the proof of Theorem 1 We now outline the key steps in proving Theorem 1 below, with detailed derivations provided in the appendix.

We begin by applying Jensen's inequality to the expectation of the negative log probability of the model's prediction $\mathbb{E}_{q(x,y)}\left[-\log\mathbb{E}_{p_\theta(z|a(x))}\left[\hat{p}_T(y\mid z)\right]\right]$, where $a(x) = \arg\max_{x^{\mathrm{adv}}\in\mathcal{A}(x)}\log\mathbb{E}_{p_\theta(z|x^{\mathrm{adv}})}[\hat{p}_T(y\mid z)], \forall x$, transforming the inner expectation into a more tractable form as $\mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}\left[-\log\hat{p}_T(y\mid z)\right]\right]$. Next, we express the difference between the expected values of the loss function under the original distribution $p_\theta(z|x)$ and the adversarial distribution $p_\theta(z|a(x))$ using the Kantorovich-Rubinstein duality

$$\mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}\left[-\log\hat{p}_T(y\mid z)\right] - \mathbb{E}_{p_\theta(z|x)}\left[-\log\hat{p}_T(y\mid z)\right]\right]$$
$$\leq M_C \cdot \mathbb{E}_{p^T(x,y)}\left[W\left(p_\theta(z\mid a(x)),\, p_\theta(z\mid x)\right)\right].$$
$$(6)$$

This allows us to quantify the deviation of the adversarial distribution from the original one in latent space via the Wasserstein distance.

Assuming accurate feature reconstruction by the foundation model's decoder $f_{de}(z) \approx x$ where $z = f_{en}(x)$, and robust latent feature production by the encoder, such that $\|f_{en}(a(x)) - f_{en}(x)\|$ is small, then the encoder produces deterministic and precise latent representations for given inputs, and the decoder reconstructs features accurately. Consequently, the distributions $p_\theta(z \mid x)$ and $p_\theta(z \mid a(x))$ are expected to be highly concentrated around $f_{en}(x)$ and $f_{en}(a(x))$ with very small variance. Given this concentration, it is reasonable to approximate these distributions as Dirac delta functions centered at $f_{en}(x)$ and $f_{en}(a(x))$. Respectively, we treat $p_\theta(z \mid a(x))$ and $p_\theta(z \mid x)$ as Dirac delta distributions centered on $f_{en}(a(x))$ and $f_{en}(x)$. This allows us to bound the Wasserstein distance by the Euclidean distance $\|f_{en}(a(x)) - f_{en}(x)\|$ between the adversarial and original latent representations.

We then relate the Euclidean distance between $f_{en}(a(x))$ and $f_{en}(x)$ to the adversarial contrastive loss L_{con} through a scaling constant κ. This formalizes the relationship between minimizing contrastive loss and improving adversarial robustness $\|f_{en}(a(x)) - f_{en}(x)\| \leq \kappa \cdot L_{con}(f_{en}(a(x)), f_{en}(x))$. Finally, combining these elements yields the main inequality of Theorem 1, bounding the adversarial loss in the downstream task by the clean downstream loss plus a scaled version of the adversarial contrastive loss.

This proof leverages the Lipschitz continuity of the decoder to bound the impact of adversarial perturbations in latent space and demonstrates how contrastive learning serves as a mechanism for enhancing robustness by aligning clean and adversarial representations. Through this approach, we establish a theoretical connection between adversarial contrastive learning and the robustness of downstream tasks, providing a solid foundation for our proposed method.

Proof of Theorem 1. Let $a(x) = \arg\max_{x^{adv} \in \mathcal{A}(x)} \log \mathbb{E}_{p_\theta(z|x^{adv})}[\hat{p}_T(y \mid z)], \forall x$, where θ is the the parameters of f_{en}. z refers to the sample latent values generated by encoder distribution $f_{en}(x)$. $\hat{p}_T(y \mid z) = P(f_{last}(f_{pre}(f_{de}(z))) = y)$ refers to the model's prediction of a certain class. $\mathcal{A}(x)$ represents the threat model. We note that $-\log \hat{p}_T(y \mid f_{de}(z))$ was used in the Theorem 1 in the main body instead of the standard $\hat{p}_T(y \mid z)$ to help better highlight the procedure of our mechanism.

To prove theorem 1, it suffices to show the below general bound of our adversarial contrastive loss,

$$\mathbb{E}_{q(x,y)}\left[-\log \mathbb{E}_{p_\theta(z|a(x))}[\hat{p}_T(y \mid z)]\right] \leq \mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|x)}[-\log \hat{p}_T(y \mid z)]\right] + \kappa \sup_{x^{adv} \in \mathcal{A}(x)} L_{con}(f_{en}(x^{adv}), f_{en}(x)).$$

To show this, we first apply Jensen's Inequality and obtain the following relation,

$$\mathbb{E}_{q(x,y)}\left[-\log \mathbb{E}_{p_\theta(z|a(x))}[\hat{p}_T(y \mid z)]\right] \leq \mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}[-\log \hat{p}_T(y \mid z)]\right].$$

Consequentially, we further convert the inequality to the following form,

$$\mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}[-\log \hat{p}_T(y \mid z)]\right] - \mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|x)}[-\log \hat{p}_T(y \mid z)]\right]$$
$$= \mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}[-\log \hat{p}_T(y \mid z)] - \mathbb{E}_{p_\theta(z|x)}[-\log \hat{p}_T(y \mid z)]\right] \leq \kappa \sup_{x^{adv} \in \mathcal{A}(x)} L_{con}(f_{en}(x^{adv}), f_{en}(x)).$$

To show this, consider Proposition 2 and Proposition 3 below, there exists a constant M_C such that $g(z) = \frac{-\log \hat{p}_T(y|z)}{M_C}$ is 1-Lipschitz. This allows us to leverage the Kantorovich-Rubinstein duality to express the difference between the expectations under the distributions $p_\theta(z \mid a(x))$ and $p_\theta(z \mid x)$ in terms of a Wasserstein distance:

$$\mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}[-\log \hat{p}_T(y \mid z)] - \mathbb{E}_{p_\theta(z|x)}[-\log \hat{p}_T(y \mid z)]\right]$$
$$= M_C \cdot \mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z|a(x))}[g(z)] - \mathbb{E}_{p_\theta(z|x)}[g(z)]\right]$$
$$\leq M_C \cdot \mathbb{E}_{q(x,y)}\left[\sup_{\|h\|_L \leq 1}\left(\mathbb{E}_{p_\theta(z|a(x))}[h(z)] - \mathbb{E}_{p_\theta(z|x)}[h(z)]\right)\right] \quad \text{(arises for } g(z) \text{ belonging to 1-Lipschitz functions.)}$$
$$= M_C \cdot \mathbb{E}_{q(x,y)}\left[W\left(p_\theta(z \mid a(x)), p_\theta(z \mid x)\right)\right] \quad \text{(follows from the Kantorovich-Rubinstein theorem.),}$$

Consider the Wasserstein distance $W\left(p_\theta(z \mid a(x)), p_\theta(z \mid x)\right)$, where W refers to the Wasserstein distance between the two distributions $p_\theta(z \mid a(x))$ and $p_\theta(z \mid x)$.

Since the decoder f_{de} has been pre-trained, in an idealized scenario, we can consider both $p_\theta(z \mid a(x))$ and $p_\theta(z \mid x)$ as Dirac delta distributions, centered at the points $f_{\mathrm{en}}(a(x))$ and $f_{\mathrm{en}}(x)$, respectively.

Assuming that the function $h(z)$ is Lipschitz continuous with respect to the encoded representations $f_{\mathrm{en}}(a(x))$ and $f_{\mathrm{en}}(x)$, we can then express the Wasserstein distance between these two Dirac Delta distributions as the Euclidean distance between the encoded representations. Specifically, we have:

$$W\left(p_\theta(z \mid a(x)), p_\theta(z \mid x)\right) \leq \|f_{\mathrm{en}}(a(x)) - f_{\mathrm{en}}(x)\|_2.$$

Considering empirical distributions derived from the data, let $\{z_i = f_{\mathrm{en}}(x_i)\}_{i=1}^N$ be samples from $p_\theta(z \mid x)$ and $\{z_i' = f_{\mathrm{en}}(a(x_i))\}_{i=1}^N$ be samples from $p_\theta(z \mid a(x))$. We assume the following properties between z_i and z_i' holds for all i and $j \neq i$ with $y_j \neq y_i$,

$$\|z_i - z_i'\| = \|f_{\mathrm{en}}(x_i) - f_{\mathrm{en}}(a(x_i))\| \leq \eta_1, \tag{7}$$
$$\|z_i - z_j\| = \|f_{\mathrm{en}}(x_i) - f_{\mathrm{en}}(x_j)\| \geq \eta_2, \tag{8}$$
$$\|z_i - z_j'\| = \|f_{\mathrm{en}}(x_i) - f_{\mathrm{en}}(a(x_j))\| \geq \eta_2. \tag{9}$$

While Equation (7) guarantees similar pairs stay sufficiently close together in the embedding space, Equation (8) and Equation (9) constrains dissimilar examples to be far apart, which aligns with the goal of our contrastive loss.

Considering these distances, a feasible transport plan in the Wasserstein distance computation is to match each z_i with its corresponding z_i'. This matching incurs a cost of at most η_1 per pair. Matching z_i with any z_j or z_j' where $y_i \neq y_j$ would incur a higher cost of at least η_2. Therefore, this transport plan yields:

$$W\left(p_\theta(z \mid a(x)), p_\theta(z \mid x)\right) \leq \frac{1}{N} \sum_{i=1}^N \|z_i - z_i'\| = \mathbb{E}_{q(x,y)}\|f_{\mathrm{en}}(a(x)) - f_{\mathrm{en}}(x)\|_2.$$

Thus, the Wasserstein distance W between the two distributions $p_\theta(z \mid a(x))$ and $p_\theta(z \mid x)$ is bounded above by the Euclidean distance between the encoded representations $f_{\mathrm{en}}(a(x))$ and $f_{\mathrm{en}}(x)$, under the assumption that $h(z)$ is Lipschitz continuous. This bound can be formally expressed as:

$$\mathbb{E}_{q(x,y)}\left[-\log \mathbb{E}_{p_\theta(z \mid a(x))}[\hat{p}_T(y \mid z)]\right]$$
$$\leq \mathbb{E}_{q(x,y)}\left[\mathbb{E}_{p_\theta(z \mid x)}[-\log \hat{p}_T(y \mid z)]\right] + M_C \cdot \mathbb{E}_{q(x,y)}\|f_{\mathrm{en}}(a(x)) - f_{\mathrm{en}}(x)\|_2.$$

This inequality suggests that the adversarial loss in the downstream task can be bounded by the distance between the encoded representations of the original samples and their attacked (adversarial) counterparts.

First, instead of using a specific, predefined form of contrastive loss, we are leveraging the general idea behind contrastive learning. The central concept in contrastive learning is to maximize the similarity (minimize the distance) between pairs of similar samples (e.g., an original sample and its adversarial example). The following argument provided is an informal idea and a brief proof that by minimizing the distance between such pairs, we can effectively bound the adversarial loss. Now, considering the relationship between this distance and contrastive loss, informally, design a contrastive loss L_{con} to minimize the distance between the encoded representations of similar pairs (e.g., a sample and its adversarial version):

Given this, we can connect the Euclidean distance $\|f_{\mathrm{en}}(a(x)) - f_{\mathrm{en}}(x)\|_2$ to the contrastive loss by introducing a scaling constant C.

$$\mathbb{E}_{q(x,y)}\|f_{\mathrm{en}}(a(x)) - f_{\mathrm{en}}(x)\|_2 \leq C \cdot L_{\mathrm{con}}(f_{\mathrm{en}}(x), f_{\mathrm{en}}(a(x))).$$

It is important to note that value of C depends on the contrastive loss and learning schedule. Assuming that our choice of the loss function is a well-behaved model, then we can obtain a reasonable value of C for bounding adversarial Euclidean distance.

By substituting this into the earlier inequality, we obtain:

$$\mathbb{E}_{q(x,y)} \left[-\log \mathbb{E}_{p_\theta(z|a(x))}[\hat{p}_T(y \mid z)] \right]$$
$$\leq \mathbb{E}_{q(x,y)} \left[\mathbb{E}_{p_\theta(z|x)}[-\log \hat{p}_T(y \mid z)] \right] + M_C \cdot C \cdot L_{\mathrm{con}}(f_{\mathrm{en}}(x), f_{\mathrm{en}}(a(x)))$$
$$\leq \mathbb{E}_{q(x,y)} \left[\mathbb{E}_{p_\theta(z|x)}[-\log \hat{p}_T(y \mid z)] \right] + \kappa \sup_{x^{\mathrm{adv}} \in \mathcal{A}(x)} L_{\mathrm{con}}(f_{\mathrm{en}}(x^{\mathrm{adv}}), f_{\mathrm{en}}(x)).$$

where $\kappa = M_C \cdot C$.

Using the above bounds, in our study, we specifically consider the adversarial contrastive loss L_{con} defined as follows:

$$L_{\mathrm{con}}(f_{\mathrm{en}}(x^{\mathrm{adv}}), f_{\mathrm{en}}(x)) = \mathbb{E}_{q(x_i, y_i)} \left[\ell_{\mathrm{con}}(f_{\mathrm{en}}(x_i^{\mathrm{adv}}), f_{\mathrm{en}}(x_i)) \right],$$

where provided with a set of clean examples $X = \{x_1, \ldots x_N\}$, each with an adversarial counterpart, $X^{\mathrm{adv}} = \{x_1^{\mathrm{adv}}, \ldots x_N^{\mathrm{adv}}\}$. Without an explicit negative sampling process, for x_i, we now consider x_i^{adv} as the similar example and the remaining samples in these sets, $X^{\mathrm{neg}} = X \bigcup X^{\mathrm{adv}} \backslash \{x_i, x_i^{\mathrm{adv}}\}$ dissimilar and define the loss function as

$$\ell_{\mathrm{con}}(f_{\mathrm{en}}(x_i^{\mathrm{adv}}), f_{\mathrm{en}}(x_i)) = -\log \frac{\exp(\mathrm{sim}(f_{\mathrm{en}}(x_i^{\mathrm{adv}}), f_{\mathrm{en}}(x_i))/\tau)}{\sum_{x^{\mathrm{neg}} \in X^{\mathrm{neg}}} \exp(\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(x^{\mathrm{neg}})/\tau)}.$$

In the above, the similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$ is measured using cosine similarity and is defined as:

$$\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|},$$

which measures the cosine of the angle between $\mathbf{u}$ and $\mathbf{v}$, normalized by their magnitudes.

So we have:

$$M_c \mathbb{E}_{q(x_i, y_i)} \|f_{\mathrm{en}}(x_i) - f_{\mathrm{en}}(a(x_i))\|_2$$
$$= M_c \mathbb{E}_{q(x_i, y_i)} \sqrt{\|f_{\mathrm{en}}(x_i) - f_{\mathrm{en}}(a(x_i))\|_2^2}$$
$$= M_c \mathbb{E}_{q(x_i, y_i)} \sqrt{2} \sqrt{1 - \mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)))}$$
(considering the output for encoder is normalized, $\|f_{\mathrm{en}}(x_i)\|^2 = 1$)
$$\leq \sqrt{2} M_c C_{\mathrm{sqrt}} \mathbb{E}_{q(x_i, y_i)} (1 - \mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i))))$$
$$\leq \sqrt{2} M_c C_{\mathrm{sqrt}} \mathbb{E}_{q(x_i, y_i)} \exp(-\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)))$$
(according to Taylor expansion for $\exp(x)$ and $|\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)| < 1$)
$$\leq \sqrt{2} M_c C_{\mathrm{sqrt}} C_{\mathrm{log}} \mathbb{E}_{q(x_i, y_i)} \log(1 + \exp(-\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)))$$
$$\leq \sqrt{2} M_c C_{\mathrm{sqrt}} C_{\mathrm{log}} C_{\mathrm{M}} \mathbb{E}_{q(x_i, y_i)} \log((\sum_{x^{\mathrm{neg}} \in X^{\mathrm{neg}}} \exp(\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(x^{\mathrm{neg}}))) * \exp(-\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)))$$
$$= \sqrt{2} M_c C_{\mathrm{sqrt}} C_{\mathrm{log}} C_{\mathrm{M}} \mathbb{E}_{q(x_i, y_i)} \left(-\log \frac{\exp(\mathrm{sim}(f_{\mathrm{en}}(x_i^{\mathrm{adv}}), f_{\mathrm{en}}(x_i)))}{\sum_{x^{\mathrm{neg}} \in X^{\mathrm{neg}}} \exp(\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(x^{\mathrm{neg}})))} \right)$$
$$= \sqrt{2} M_c C_{\mathrm{sqrt}} C_{\mathrm{log}} C_{\mathrm{M}} \mathbb{E}_{q(x_i, y_i)} \left[\ell_{\mathrm{con}}(f_{\mathrm{en}}(x_i^{\mathrm{adv}}), f_{\mathrm{en}}(x_i)) \right]$$
$$\leq \kappa \sup_{x^{\mathrm{adv}} \in \mathcal{A}(x)} L_{\mathrm{con}}(f_{\mathrm{en}}(x^{\mathrm{adv}}), f_{\mathrm{en}}(x)).$$

where $\kappa = \sqrt{2} M_c C_{\mathrm{sqrt}} C_{\mathrm{log}} C_{\mathrm{M}}$.

- C_{sqrt} is a scaling factor designed to linearize the expression involving the square root function, ensuring that the inequality holds. The scaling factor C_{sqrt} is defined by the formula:

$$C_{\mathrm{sqrt}} = \frac{1}{\min_{t=\mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)))} \frac{\sqrt{1-t}}{1-t}},$$

where $t = \mathrm{sim}(f_{\mathrm{en}}(x_i), f_{\mathrm{en}}(a(x_i)))$ represents the similarity between the clean and adversarial examples.

- In the context of adversarial contrastive training, given that $\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i)))$ will be less than 1, there exists a bounded C_{sqrt}. This ensures that the linear approximation $1 - t$ is a valid upper bound for $\sqrt{1-t}$ across the relevant range of t.

- $C_{\log}$ is a scaling factor introduced to ensure that the logarithmic expression is correctly bounded by the exponential function, maintaining the inequality's validity. The scaling factor $C_{\log}$ is defined by the formula:

$$C_{\log} = \frac{1}{\min_{t=\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i)))} \frac{\log(1+\exp(-t))}{\exp(-t)}},$$

where $t = \text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i)))$ denotes the similarity between the clean and adversarial examples.

- In the context of adversarial contrastive training, given that $\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i)))$ lies within the range $-1 < t < 1$, the scaling factor $C_{\log}$ can be bounded above by $\frac{e}{\log(1+e)}$. This ensures that the logarithmic term is appropriately scaled relative to the exponential term, preserving the bound.

- C_{M} is a scaling factor that ensures the logarithm of the product is greater than or equal to the logarithm of the sum $\log(1 + \exp(-\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i)))))$.

- Given that for large M (especially when $M > 10$), the sum $\sum_{x^{\text{neg}} \in X^{\text{neg}}} \exp(\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(x^{\text{neg}})))$ will likely exceed $e(e + 1)$, making the product of this sum with $\exp(-\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i))))$ naturally larger than $1 + \exp(-\text{sim}(f_{\text{en}}(x_i), f_{\text{en}}(a(x_i))))$, the scaling factor C_{M} can be considered close to 1, or even negligible.

- In practical terms, C_{M} can be ignored when M is sufficiently large, since the dominant terms in the sum ensure that the inequality holds naturally without needing additional scaling.

$\square$

Proposition 2. *For the robust auto-encoder, assume we have a robust auto-encoder defined by an encoder $f_{\text{en}}(x)$ and a decoder $f_{\text{de}}(z)$, where $z = f_{\text{en}}(x)$ represents the latent feature of the input x. This auto-encoder is trained using a combination of reconstruction loss and adversarial contrastive loss. The reconstruction loss, $\|f_{\text{de}}(f_{\text{en}}(x)) - x\|^2$, ensures that the decoder f_{de} can accurately reconstruct the original input from the encoded features. The contrastive loss, $L_{\text{con}}(f_{\text{en}}(x + \delta), f_{\text{en}}(x'))$, promotes robustness of the encoded features $f_{\text{en}}(x)$ against adversarial perturbations δ. There exists a constant C such that $\frac{f_{\text{de}}(z)}{C}$ is 1-Lipschitz.*

Proof of Proposition 2. Given the training process of the robust auto-encoder, two key losses are employed: the reconstruction loss $\|f_{\text{de}}(f_{\text{en}}(x)) - x\|^2$ and the adversarial contrastive loss $L_{\text{con}}(f_{\text{en}}(x + \delta), f_{\text{en}}(x'))$. These losses are designed to ensure that the encoder $f_{\text{en}}(x)$ and decoder $f_{\text{de}}(z)$ not only preserve the integrity of the original input x in the reconstruction but also maintain robustness against adversarial perturbations.

We denote the Lipschitz constant of our encoder as L_{en}. Additionally, given that in a ViT-MAE, the representation space is typically larger than the input space, we assume there exists a lower bound on the gradient between any samples in our dataset and within the threat model $\mathcal{A}$ of samples of our dataset. Formally, we write this as

$$l_{en}\|x_1 - x_2\| \le \|z_1 - z_2\| \le L_{en}\|x_1 - x_2\|,$$

where $z_1 = f_{\text{en}}(x_1)$ and $z_2 = f_{\text{en}}(x_2)$ represent the latent features of the inputs x_1 and x_2, respectively.

Similarly, the reconstruction loss $\|f_{\text{de}}(f_{\text{en}}(x)) - x\|^2$ optimizes the decoder f_{de} to ensure accurate reconstruction of the input from the latent features. In most cases, it is safe to assume this optimization results in a finite-Lipschitz constant L_{rec} that describes how the output $f_{\text{de}}(z)$ changes with respect to variations in the input x:

$$\|f_{\text{de}}(z_1) - f_{\text{de}}(z_2)\| \le L_{rec}\|x_1 - x_2\|.$$

Given the relationships established by these losses, we can now derive the Lipschitz constant between the encoded feature z and the decoder output $f_{\text{de}}(z)$. By combining the encoder's and decoder's Lipschitz constants, we obtain:

$$\|f_{\mathrm{de}}(z_1) - f_{\mathrm{de}}(z_2)\| \le L_{rec}\|x_1 - x_2\| \le L_{rec} \cdot \frac{1}{l_{en}}\|z_1 - z_2\|.$$

Thus, the Lipschitz constant for the decoder with respect to the encoded feature z, denoted as $L_{de}^{(z)}$, is:

$$L_{de}^{(z)} = \frac{L_{rec}}{l_{en}}.$$

Finally, by selecting the constant $C = L_{de}^{(z)}$, we can ensure that the scaled decoder function $g(z) = \frac{f_{\mathrm{de}}(z)}{C}$ satisfies the 1-Lipschitz condition:

$$|g(z_1) - g(z_2)| = \frac{1}{C}|f_{\mathrm{de}}(z_1) - f_{\mathrm{de}}(z_2)| \le \|z_1 - z_2\|.$$

This concludes the proof, demonstrating that such a constant C exists and that $\frac{f_{\mathrm{de}}(z)}{C}$ is indeed 1-Lipschitz, ensuring the stability and robustness of the auto-encoder.

$\square$

Proposition 3. *Assume* $-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z)) \le M$ *for all* $z \in \mathcal{Z}, y \in \mathcal{Y}$. *If there exists a constant* C *such that* $\frac{f_{\mathrm{de}}(z)}{C}$ *is 1-Lipschitz, then there exists a constant* M_C *such that* $\frac{-\log \hat{p}_T(y|f_{\mathrm{de}}(z))}{M_C}$ *is 1-Lipschitz.*

Proof of Proposition 3. Given that $-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z)) \le M$ for all $z \in \mathcal{Z}$ and $y \in \mathcal{Y}$, we know that the function $-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z))$ is bounded above by M. Consequently, $\hat{p}_T(y \mid f_{\mathrm{de}}(z))$ is bounded below by e^{-M}.

Now, assume that there exists a constant C such that $\frac{f_{\mathrm{de}}(z)}{C}$ is 1-Lipschitz, which implies:

$$\left| \frac{f_{\mathrm{de}}(z_1)}{C} - \frac{f_{\mathrm{de}}(z_2)}{C} \right| \le \|z_1 - z_2\|, \quad \forall z_1, z_2 \in \mathcal{Z}.$$

Next, consider the function $-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z))$. To show that $\frac{-\log \hat{p}_T(y|f_{\mathrm{de}}(z))}{M_C}$ is 1-Lipschitz for some constant M_C, we recognize that the derivative of the log function is bounded by $\frac{1}{\hat{p}_T(y|f_{\mathrm{de}}(z))}$. Since $\hat{p}_T(y \mid f_{\mathrm{de}}(z))$ is bounded below by e^{-M}, the derivative is bounded by e^M.

Thus, we define M_C as:

$$M_C = C \times e^M.$$

We then scale the function $-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z))$ by M_C and consider the difference:

$$\left| \frac{-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z_1))}{M_C} - \frac{-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z_2))}{M_C} \right| = \frac{1}{M_C}\left| -\log \hat{p}_T(y \mid f_{\mathrm{de}}(z_1)) + \log \hat{p}_T(y \mid f_{\mathrm{de}}(z_2)) \right|.$$

Given the earlier bound on the derivative of the log function, we have:

$$|\log \hat{p}_T(y \mid f_{\mathrm{de}}(z_1)) - \log \hat{p}_T(y \mid f_{\mathrm{de}}(z_2))| \le e^M \cdot C \cdot \|z_1 - z_2\|.$$

Substituting into our earlier equation, we get:

$$\left| \frac{-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z_1))}{M_C} - \frac{-\log \hat{p}_T(y \mid f_{\mathrm{de}}(z_2))}{M_C} \right| \le \frac{e^M \cdot C}{M_C} \cdot \|z_1 - z_2\| = \frac{e^M \cdot C}{e^M \cdot C} \cdot \|z_1 - z_2\| = \|z_1 - z_2\|.$$

Thus, $\frac{-\log \hat{p}_T(y|f_{\mathrm{de}}(z))}{M_C}$ is indeed 1-Lipschitz, which completes the proof.

$\square$

Proof of Proposition 1. To illustrate an extreme scenario in proving Proposition1, we construct a data generation model (x, y) consisting of a discrete set $\{x_i\}$ with labels y_i, an identity auto-encoder $(f_{\text{en}}, f_{\text{de}})$ that achieves negligible reconstruction losses for both clean and adversarial inputs, and a classifier $(f_{\text{pre}}, f_{\text{last}})$ whose adversarial classification loss can be made arbitrarily large despite near-perfect reconstructions.

Step 1: Construct a data distribution $q(x, y)$. We select a finite (or discrete) set of points

$$\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^d,$$

each assigned a label $y_i \in \{1, \ldots, K\}$. We assume these points are well-separated by at least 2ϵ, i.e.,

$$\|x_i - x_j\| \geq 2\epsilon \quad \text{for all } i \neq j.$$

We define a discrete distribution

$$q(x, y) = \frac{1}{N} \sum_{i=1}^{N} \delta_{(x_i, y_i)}(x, y),$$

so that each pair (x_i, y_i) occurs with probability $\frac{1}{N}$.

Step 2: Define an auto-encoder $(f_{\text{en}}, f_{\text{de}})$ with small reconstruction loss. Consider the *identity mapping*:

$$f_{\text{en}}(x) = x, \quad f_{\text{de}}(z) = z.$$

Then for any input x (including adversarially perturbed inputs x^{adv}),

$$f_{\text{de}}(f_{\text{en}}(x)) = x \quad \Longrightarrow \quad \|f_{\text{de}}(f_{\text{en}}(x)) - x\|^2 = 0.$$

Hence for both clean data x and any x^{adv} with $\|x^{\text{adv}} - x\| \leq \epsilon$, the reconstruction loss is *exactly zero*, i.e.,

$$\mathbb{E}_x \left[\|f_{\text{de}}(f_{\text{en}}(x)) - x\|^2 \right] = 0,$$

and

$$\mathbb{E}_x \left[\|f_{\text{de}}(f_{\text{en}}(x^{\text{adv}})) - x\|^2 \right] = \epsilon^2 = O(1/n)$$

if taking $\epsilon = O(1/\sqrt{n})$. Thus both reconstruction errors (clean and adversarial) can be made as small as desired.

Step 3: Construct a brittle classifier $(f_{\text{pre}}, f_{\text{last}})$ with large adversarial loss. Define f_{pre} to be the identity as well (or any feature extractor that yields x effectively), so $f_{\text{pre}}(x) = x$. Then let $f_{\text{last}} : \mathbb{R}^d \to \Delta_K$ (the probability simplex) be chosen as follows:

1. For each clean point x_i, assign label y_i with probability very close to 1:

   $$\hat{p}_T(y_i \mid f_{\text{last}}(f_{\text{pre}}(x_i))) = 1 - \delta,$$

   for some arbitrary small constant $\delta > 0$, then $-\log \hat{p}_T(y_i \mid f_{\text{last}}(f_{\text{pre}}(x_i))) = O(\delta)$, ensuring near-zero loss on all clean (x_i, y_i).

2. For any x^{adv} satisfying $\|x^{\text{adv}} - x_i\| \leq \epsilon$, assign *zero* (or near-zero) probability to the label y_i, i.e.

   $$\hat{p}_T(y_i \mid f_{\text{last}}(f_{\text{pre}}(x^{\text{adv}}))) = O(\delta).$$

   Hence $-\log \hat{p}_T(y_i \mid f_{\text{last}}(f_{\text{pre}}(x^{\text{adv}}))) \gg 0$, so the classification loss is made arbitrarily large on these adversarial perturbations.

Because the ϵ-balls around different x_i do not intersect ($\|x_i - x_j\| \geq 2\epsilon$), we can define such a piecewise decision rule without conflict.

$\square$