ProFeAT: Projected Feature Adversarial Training for Self-Supervised Learning of Robust Representations

Sravanti Addepalli * **Priyam Dey** * **R. Venkatesh Babu**

Vision and AI Lab, Department of Computational and Data Sciences, IISc Bangalore

Reviewed on OpenReview: https://openreview.net/forum?id=AUC0Kmn70N

Abstract

The need for abundant labelled data in supervised Adversarial Training (AT) has prompted the use of Self-Supervised Learning (SSL) techniques with AT. However, the direct application of existing SSL methods to adversarial training has been sub-optimal due to the increased training complexity of combining SSL with AT. A recent approach DeACL (Zhang et al., 2022) mitigates this by utilizing supervision from a standard SSL teacher in a distillation setting, to mimic supervised AT. However, we find that there is still a large performance gap when compared to supervised adversarial training, specifically on larger models. In this work, investigate the key reason for this gap and propose Projected Feature Adversarial Training (ProFeAT) to bridge the same. We show that the sub-optimal distillation performance is a result of mismatch in training objectives of the teacher and student, and propose to use a projection head at the student, that allows it to leverage weak supervision from the teacher while also being able to learn adversarially robust representations that are distinct from the teacher. We further propose appropriate attack and defense losses at the feature and projector, alongside a combination of weak and strong augmentations for the teacher and student respectively, to improve the training data diversity without increasing the training complexity. Through extensive experiments on several benchmark datasets and models, we demonstrate significant improvements in both clean and robust accuracy when compared to existing SSL-AT methods, setting a new state-of-the-art. We further report on-par/ improved performance when compared to TRADES, a popular supervised-AT method. Our code is available at: https://github.com/val-iisc/ProFeAT.

1 Introduction

Deep Neural Networks are known to be vulnerable to crafted imperceptible input-space perturbations known as *Adversarial attacks* (Szegedy et al., 2013), which can be used to fool classification networks into predicting any desired output, leading to disastrous consequences. Amongst the diverse attempts at improving the adversarial robustness of Deep Networks, Adversarial Training (AT) (Madry et al., 2018; Zhang et al., 2019) has been the most successful. This involves the generation of adversarial attacks by maximizing the training loss, and further minimizing the loss on the generated attacks for training. While adversarial training based methods have proved to be robust against various attacks developed over time (Carlini et al., 2019; Croce & Hein, 2020; Sriramanan et al., 2020), they require significantly more training data when compared to standard training (Schmidt et al., 2018), incurring a large annotation cost. This motivates the need for self-supervised learning (SSL) of robust representations, followed by lightweight standard training of the classification head. Motivated by the success of contrastive learning for standard self-supervised learning (Van den Oord et al., 2018; Chen et al., 2020b; He et al., 2020), several works have attempted to use contrastive learning for self-supervised adversarial training as well (Jiang et al., 2020; Kim et al., 2020; Fan et al., 2021). While this strategy works well in a full network fine-tuning setting, the performance is sub-optimal when the robustly

*Equal Contribution.

Correspondence to Sravanti Addepalli <sravantia@iisc.ac.in>, Priyam Dey <priyamdey@iisc.ac.in>

pretrained feature encoder is frozen while training the classification head (linear probing), demonstrating that the representations learned are indeed sub-optimal. A recent work, Decoupled Adversarial Contrastive Learning (DeACL) (Zhang et al., 2022), demonstrated significant improvements in performance and training efficiency by splitting this combined self-supervised adversarial training into two stages; first, where a standard self-supervised model is trained, and second, where this pretrained model is used as a teacher to provide supervision to the adversarially trained student network. Although the performance of this method is on par with supervised adversarial training on small model architectures (ResNet-18), we find that it does not scale to larger models such as WideResNet-34-10, which is widely reported in the adversarial ML literature.

In this work, we aim to bridge the performance gap between self-supervised and supervised adversarial training methods, and improve the scalability of the former to larger model capacities. We utilize the distillation setting discussed above, where a standard self-supervised trained teacher provides supervision to the student. In contrast to a typical distillation scenario, the student's objective or its *ideal goal* is not to replicate the teacher, but to leverage weak supervision from the teacher while also learning adversarially robust representations. This involves a trade-off between the sensitivity towards changes that flip the class of an image (for better clean accuracy) and invariance towards imperceptible perturbations that preserve the true class (for adversarial robustness) (Tramèr et al., 2020). Towards this, we propose to impose similarity with respect to the teacher in the appropriate dimensions by applying the distillation loss in a *projected space* (output of a projection MLP layer), while enforcing the smoothness-based robustness loss in the *feature space* (output of a backbone/ feature extractor). However, we find that enforcing these losses at different layers results in training instability, and thus introduce the complementary loss (clean distillation loss or robustness loss) as a regularizer to improve training stability. We further propose to reuse the pretrained projection layer from the teacher model for better convergence. In line with the training objective, the attack used for training aims to find images that maximize the smoothness loss in the feature space, and cause misalignment between the teacher and student in the projected space. Further, since data augmentations are known to increase the training complexity of adversarial training resulting in a drop in performance (Zhang et al., 2022; Addepalli et al., 2022), we propose to use augmentations such as AutoAugment (or *strong augmentations*) only at the student for better attack diversity, while using spatial transforms such as pad and crop (PC) (or *weak augmentations*) at the teacher. We summarize our contributions below:

- We propose Projected Feature Adversarial Training (ProFeAT) - a teacher-student distillation setting for self-supervised adversarial training, where the projection layer of the standard self-supervised pretrained teacher is utilized for student distillation. We further propose appropriate attack and defense losses for training, coupled with a combination of weak and strong augmentations for the teacher and student respectively.
- Towards understanding *why* the projector helps, we first show that the compatibility between the training methodology of the teacher and the ideal goals of the student plays a crucial role in the student model performance in distillation. We further show that the use of a projector can alleviate the negative impact of the inherent misalignment of the above.
- We demonstrate the effectiveness of the proposed approach on the standard benchmark datasets CIFAR-10 and CIFAR-100. We obtain significant gains of $3.5 - 8\%$ in clean accuracy and $\sim 3\%$ in robust accuracy on larger model capacities (WideResNet-34-10), and improved performance on small model architectures (ResNet-18), while also outperforming TRADES supervised training (Zhang et al., 2019) on larger models.

2 Preliminaries

We consider the problem of self-supervised learning of robust representations, where a self-supervised standard trained teacher model $\mathcal{T}$ is used to provide supervision to a student model $\mathcal{S}$. The feature, projection and linear probing layers of the teacher are denoted as $\mathcal{T}_f$, $\mathcal{T}_p$ and $\mathcal{T}_l$, respectively. We will use projection layer and projector interchangeably in the rest of the paper. A composition of the feature extractor followed by the projector of the teacher is denoted as $\mathcal{T}_{pf} = \mathcal{T}_p \circ \mathcal{T}_f$, and a composition of the feature extractor followed by the linear probing layer is denoted as $\mathcal{T}_{lf} = \mathcal{T}_l \circ \mathcal{T}_f$. An analogous notation is followed for the student as well. The dataset used for self-supervised pretraining $\mathcal{D}$ consists of images x_i where $i \leq N$. An adversarial

image corresponding to the image x_i is denoted as $\tilde{x}_i$. We consider the ℓ_∞ based threat model where $||\tilde{x}_i - x_i||_\infty \leq \varepsilon$. The value of ε is set to 8/255 for CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), as is standard in literature (Madry et al., 2018; Zhang et al., 2019). To evaluate the learned representations, we freeze the pretrained backbone, and perform linear layer training on a downstream labeled dataset consisting of image-label pairs using cross-entropy loss on clean samples, popularly referred to as linear probing (Kumar et al., 2022). We compare the robustness of the representations on both in-distribution data and in a transfer learning setting, where the distribution of images is different from pretraining.

3 Related Works

Self Supervised Learning (SSL): Early works on SSL had focused on designing well-posed tasks called "pretext tasks", such as rotation prediction (Gidaris et al., 2018) and solving jigsaw puzzles (Noroozi & Favaro, 2016), to provide a meaningful supervisory signal in the absence of true labels. However, the design of these hand-crafted tasks involves manual effort, and is generally specific to the dataset and training task. To alleviate this problem, contrastive learning based SSL approaches have emerged as a promising direction (Van den Oord et al., 2018; Chen et al., 2020b; He et al., 2020), where different augmentations of a given anchor image form the *positives*, and augmentations of other images in the batch form the *negatives*. The training objective involves pulling the representations of the positives together, and pushing away the representations of negatives. Strong augmentations like random cropping and color jitter are applied to make the learning task sufficiently hard, while also enabling the learning of invariant representations.

Self Supervised Adversarial Training: To alleviate the large sample complexity and training cost of adversarial training, there have been several works that have attempted self-supervised learning of adversarially robust representations. Chen et al. (2020a) propose AP-DPE, an ensemble adversarial pretraining framework where several pretext tasks like Jigsaw puzzles (Noroozi & Favaro, 2016), rotation prediction (Gidaris et al., 2018) and Selfie (Trinh et al., 2019) are combined to learn robust representations without task labels. Jiang et al. (2020) propose ACL, that combines the popular contrastive SSL method - SimCLR (Chen et al., 2020b) with adversarial training, using Dual Batch normalization layers for the student model - one for the standard branch and another for the adversarial branch. RoCL (Kim et al., 2020) follows a similar approach to ACL by combining the contrastive objective with adversarial training to learn robust representations. Fan et al. (2021) propose AdvCL, that uses high-frequency components in data as augmentations in contrastive learning, performs attacks on unaugmented images, and uses a pseudo label based loss for training to minimize the cross-task robustness transferability. Luo et al. (2023) study the role of augmentation strength in self-supervised contrastive adversarial training, and propose DynACL, that uses a "strong-to-weak" annealing schedule on augmentations. Additionally, motivated by Kumar et al. (2022), they propose DynACL++ that obtains pseudo-labels via k-means clustering on the clean branch of the DynACL pretrained network, and performs linear-probing (LP) using these pseudo-labels followed by adversarial full-finetuning (AFT) of the backbone. This is a generic strategy that can be integrated with several algorithms including ours.

Decoupled Adversarial Contrastive Learning (DeACL): While most self-supervised adversarial training methods aimed at integrating contrastive learning methods with adversarial training, Zhang et al. (2022) showed that combining the two is a complex optimization problem due to their conflicting requirements. The authors propose DeACL, where a *teacher* model is first trained using existing self-supervised training methods such as SimCLR (Chen et al., 2020b), followed by the adversarial training of a *student* model using the supervision from the earlier-trained teacher in a distillation framework, as shown in Figure 4 in Appendix A.1. The loss used during distillation is a combination of the cosine similarity between the representations of the clean image at the output of the teacher and student, along with a smoothness loss at the output of the student, as shown in Equation (5). The attack used during training is generated by a minimizing the cosine similarity between the teacher and student representations at the feature output (see Equation (6)). We present a more detailed description of DeACL in Appendix A.1. The robust student model is used for the downstream tasks. While existing methods used ~ 1000 epochs for contrastive adversarial training, the compute requirement for DeACL is much lesser since the first stage does not involve adversarial training, and the second stage is similar in complexity to supervised adversarial training (details in Appendix E). In this work, we investigate and address the shortcomings of the distillation framework presented in DeACL by introducing a projection layer during the distillation process, along with well-designed training and attack losses, and an appropriate augmentation scheme for better attack diversity, when compared to DeACL.

Table 1: **Diverse representations of Standard Trained (ST) and Adversarially Trained (AT) models (CIFAR-100, WRN-34-10):** ST models achieve 0% robust accuracy even with adversarial training of the linear layer, and AT models lose their robustness with standard full-finetuning. SA: Standard Accuracy, RA-G: Robust accuracy against GAMA Sriramanan et al. (2020), RA-PGD20: Robust accuracy against PGD-20 Madry et al. (2018) attack.

Training/ LP Method	SA	RA-PGD20	RA-G	Training/ LP Method	SA	RA-PGD20	RA-G
Standard trained model	80.86	0.00	0.00	TRADES AT model	60.22	28.67	26.36
+ Adversarial Linear Probing	80.10	0.00	0.00	+ Standard Full Finetuning	76.11	0.37	0.11

4 Proposed Method

4.1 Projection Layer in Self-supervised Distillation

In this work, we follow the setting proposed by Zhang et al. (2022), where a standard self-supervised pretrained teacher provides supervision for self-supervised adversarial training of the student model. This is different from a standard distillation setting (Hinton et al., 2015) because the representations of standard and adversarially trained models are known to be inherently different (Engstrom et al., 2019). Ilyas et al. (2019) attribute the adversarial vulnerability of models to the presence of non-robust features which can be disentangled from robust features that are learned by adversarially trained models. The differences in representations of standard and adversarially trained models can also be justified by the fact that linear probing of standard trained models using adversarial training cannot produce robust models as shown in Table 1. Similarly, standard full finetuning of adversarially trained models destroys the robust features learned (Chen et al., 2020a; Kim et al., 2020; Fan et al., 2021), yielding 0% robustness, as shown in the table. Due to the inherently diverse representations of these models, the ideal goal of the student in the considered distillation setting is not to merely follow the teacher, but to be able to take weak supervision from it while being able to differ considerably. In order to achieve this, we take inspiration from standard self-supervised learning literature (Van den Oord et al., 2018; Chen et al., 2020b; He et al., 2020; Navaneet et al., 2022; Gao et al., 2022; Gupta et al., 2022) and propose to utilize a projection layer following the student backbone, so as to isolate the impact of the enforced loss on the learned representations. Bordes et al. (2022) show that in standard supervised and self-supervised training, a projector is useful when there is a misalignment between the pretraining and downstream tasks, and aligning them can eliminate the need for the same. Xue et al. (2024) further shows theoretically that lower layers represent more features evenly resulting in better generalizability of the learned representations. Motivated by this, we hypothesize the following for the setting of self-supervised distillation:

Student model performance improves by matching the following during distillation:

1. *Training objectives of the teacher and the ideal goals of the student,*
2. *Pretraining and linear probe training objectives of the student.*

Here, the ideal goal of the student depends on the downstream task, which is clean (or standard) accuracy in standard training, and clean and robust accuracy in adversarial training. On the other hand, the training objective of the standard self-supervised trained teacher is to achieve invariance to augmentations of the same image when compared to augmentations of other images.

We now explain the intuition behind the above hypotheses and empirically justify the same by considering several distillation settings involving standard and adversarial, supervised and self-supervised trained teacher models in Tables 2 and 3. The results are presented on CIFAR-100 (Krizhevsky et al., 2009) with WideResNet-34-10 (Zagoruyko & Komodakis, 2016) architecture for both teacher and student. The standard self-supervised model is trained using SimCLR (Chen et al., 2020b). Contrary to a typical knowledge distillation setting where a cross-entropy loss is also used (Hinton et al., 2015), all the experiments presented involve the use of only self-supervised losses for distillation (cosine similarity between representations), and labels are used only during linear probing. Adversarial self-supervised distillation in Table 3 is performed using a combination of

Table 2: **Role of projector in self-supervised distillation (CIFAR-100, WRN-34-10):** The drop in accuracy of student ($\mathcal{S}$) w.r.t. the teacher ($\mathcal{T}$) indicates distillation performance, which improves by matching the training objective of the teacher with ideal goals of the student (S3/ S4 vs. S1), and by using similar losses for pretraining and linear probing (LP) (S2 vs. S1). Using a projector improves performance in case of mismatch in the above (S5 vs. S1). The similarity between teacher and student is significantly higher at the projector space when compared to the feature space in S5.

Exp #	Teacher training	Teacher acc (%)	Projector	LP Loss	Student accuracy after linear probe		$\cos(\mathcal{T},\mathcal{S})$	
					Feature space (%)	Projector space (%)	Feature space	Projector space
S1	Self-supervised	70.85	Absent	CE	64.90	-	0.94	-
S2	Self-supervised	70.85	Absent	$\cos(\mathcal{T},\mathcal{S})$	68.49	-	0.94	-
S3	Supervised	80.86	Absent	CE	80.40	-	0.94	-
S4	Supervised	69.96	Absent	CE	71.73	-	0.98	
S5	Self-supervised	70.85	Present	CE	73.14	64.67	0.19	0.92

Table 3: **Role of projector in self-supervised adversarial distillation (CIFAR-100, WRN-34-10):** Student performance after linear probe at feature space is reported. The drop in standard accuracy (SA) of the student ($\mathcal{S}$) w.r.t. the teacher ($\mathcal{T}$), and the robust accuracy (RA-G) of the student improve by matching the training objective of the teacher with ideal goals of the student (A3 vs. A1), and by using similar losses for pretraining and linear probing (LP) (A2 vs. A1). Using a projector improves performance in case of mismatch in the above (A4 vs. A1).

Exp #	Teacher training	Teacher accuracy		Projector	LP Loss	Student accuracy		$\cos(\mathcal{T},\mathcal{S})$
		SA (%)	RA-G (%)			SA (%)	RA-G (%)	
A1	Self-supervised (standard training)	70.85	0	Absent	CE	50.71	24.63	0.78
A2 (DeACL)	Self-supervised (standard training)	70.85	0	Absent	$\cos(\mathcal{T},\mathcal{S})$	54.48	23.20	0.78
A3	Supervised (TRADES adversarial training)	59.88	25.89	Absent	CE	54.86	27.17	0.94
A4	Self-supervised (standard training)	70.85	0	Present	CE	57.51	24.10	0.18

distillation loss on natural samples and smoothness loss on adversarial samples as shown in Equation (2) (Zhang et al., 2022). A randomly initialized trainable projector is used at the output of student backbone in S5 of Table 2 and A4 of Table 3. Here, the training loss is considered in the projection space of the student ($\mathcal{S}_p$) rather than the feature space ($\mathcal{S}_f$).

1. Matching the training objectives of teacher with the ideal goals of the student: Consider task-A to be the teacher's training task, and task-B to be the student's downstream task or its ideal goal. The representations in deeper layers (last few layers) of the teacher are more tuned to its training objective, and the early layers contain a lot more information than what is needed for this task (Bordes et al., 2022). Thus, features specific to task-A are dominant or replicated in the final feature layer, and other features that may be relevant to task-B are sparse. When a similarity based distillation loss is enforced on such features, higher importance is given to matching the task-A dominated features, and the sparse features which may be important for task-B are suppressed further in the student (Addepalli et al., 2023). On the other hand, when the student's task matches with the teacher's task, a similarity based distillation loss is very effective in transferring the necessary representations to the student, since they are predominant in the final feature layer. Thus, matching the training objective of the teacher with the ideal goals of the student should improve downstream performance.

To test this hypothesis, we first consider the standard training of a student model, using either a self-supervised or supervised teacher, in Table 2. One can note that in the absence of a projector, the drop in student accuracy w.r.t. the respective teacher accuracy is 6% with a self-supervised teacher (S1), and $< 0.5\%$ with a supervised teacher (S3). To ensure that our observations are not a result of the 10% difference in teacher accuracy between S1 and S3, we present results and similar observations with a supervised sub-optimally trained teacher in S4. Thus, a supervised teacher is significantly better than a self-supervised teacher for distilling representations specific to a given task. This justifies the hypothesis that, *student performance improves by matching the training objectives of the teacher and the ideal goals of the student.*

We next consider adversarial training of a student, using either a standard self-supervised teacher, or a supervised adversarially trained (TRADES) teacher, in Table 3. Since the TRADES model is more aligned

with the ideal goals of the student, despite its lower clean accuracy, the clean and robust accuracy of the student are better than those obtained using a standard self-supervised model as a teacher (A3 vs. A1). This further justifies the first hypothesis.

2. Matching the pretraining and linear probe training objectives of the student: For a given network, aligning the pretraining task with downstream task results in better performance since the matching of tasks ensures that the required features are predominant, and they are easily used by, for e.g., an SVM or a linear classifier trained over it (Addepalli et al., 2023). In context of distillation, since the features of the student are trained by enforcing similarity based loss w.r.t. the teacher, we hypothesize that enforcing similarity w.r.t. the teacher is the best way to learn the student classifier as well. To illustrate this, let's consider task-A to be the teacher pretraining task, and task-B to be the downstream task or ideal goal of the student. As discussed above, the teacher's features are aligned to task-A and these are transferred effectively to the student. The features related to task-B are suppressed in the teacher and are further suppressed in the student. As the features specific to a given task become more sparse, it is harder for an SVM (or a linear) classifier to rely on that feature, although it important for classification (Addepalli et al., 2023). Thus, training a linear classifier for task-B is more effective on the teacher when compared to the student. The linear classifier of the teacher in effect amplifies the sparse features, allowing the student to learn them more effectively. Thus, training a classifier on the teacher and distilling it to the student is better than training a classifier directly on the student.

We now provide empirical evidence in support of this hypothesis. To align pretraining with linear probing, we perform linear probing on the teacher model, and further train the student by maximizing the cosine similarity between the logits of the teacher and student. This boosts the student accuracy by 3.6%, in Table 2 (S2 vs. S1) and by 3.8% in Table 3 (A2 vs. A1). The projector isolates the representations of the student from the training loss, as indicated by the lower similarity between the student and teacher at feature space when compared to that at the projector (in S5 and A4), and prevents overfitting of the student to the teacher training objective. This makes the student robust to the misalignment between the teacher training objective and ideal goals of the student, and also to the mismatch in student pretraining and linear probing objectives, thereby improving student performance, as seen in Table 2 (S5 vs. S1) and Table 3 (A4 vs. A1).

4.2 ProFeAT: Projected Feature Adversarial Training

We now present our proposed approach ProFeAT which is illustrated in Figure 1. ProFeAT training happens in two stages. Firstly, a teacher model $\mathcal{T}$ is trained using a standard self-supervised learning algorithm such as SimCLR (Chen et al., 2020b), whose weights are used as an initialization for the student $\mathcal{S}$ for better convergence in the next stage. Next, the student model is trained robustly in a distillation framework where the teacher model supervises the student model (see Figure 1). Both the student and the teacher are trained with a *projection layer* on top of their feature backbone. The projection layer is kept frozen and is initialized using the learned weights from the earlier teacher training. We use well-designed *attack* and *defense* losses along with appropriate *augmentations* to ensure the student model is trained robustly while taking appropriate supervision from the teacher model. We now describe in detail the individual design components of the proposed approach ProFeAT:

Projection Layer: As discussed in Section 4.1, to overcome the impact of the inherent misalignment between the training objective of the teacher and the ideal goals of the student, and to mitigate the mismatch between the pretraining and linear probing objectives, we propose to use a projection head at the output of both the teacher and the student backbone. Most self-supervised pretraining methods (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; Chen & He, 2021; Zbontar et al., 2021) use similarity based losses at the output of a projection layer for standard training, resulting in a projected space where similarity has been enforced during pretraining, thus giving higher importance to the key dimensions.

Defense Loss: As is common in adversarial training literature (Zhang et al., 2019), we use a combination of (1) loss on clean samples and (2) a smoothness loss, to enforce adversarial robustness in the student model. Since the loss on clean samples utilizes supervision from the self-supervised pretrained teacher, it is enforced at the outputs of the respective projectors of the teacher and student. The goal of the smoothness loss is to enforce local smoothness in the loss surface of the student backbone (Zhang et al., 2019), and is

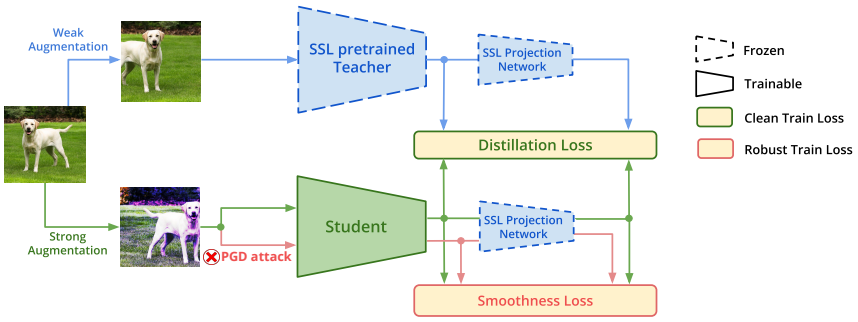

Figure 1: **Proposed approach (ProFeAT):** The student is trained using a distillation loss on clean samples using supervision from an SSL pretrained teacher, and a smoothness loss to enforce adversarial robustness (details of exact loss formulation is presented in Section 4). A frozen pretrained projection layer is used at the teacher and student to prevent overfitting to the clean distillation loss. The use of strong augmentations at the student increases attack diversity, while weak augmentations at the teacher reduce the training complexity.

ideally enforced at the feature space of the student network, since these representations are directly used for downstream applications. While the ideal locations for the clean and adversarial losses are the projected and feature spaces respectively, we find that such a loss formulation is hard to optimize, resulting in either a non-robust model, or collapsed representations (shown in Table 18). We therefore use a complimentary loss as a regularizer in the respective spaces. This results in a combination of losses at the feature and projection spaces as shown below:

$$\mathcal{L}_{pf} = -\sum_i \cos\left(\mathcal{T}_{pf}(x_i), \mathcal{S}_{pf}(x_i)\right) + \beta \cos\left(\mathcal{S}_{pf}(x_i), \mathcal{S}_{pf}(\tilde{x}_i)\right) \tag{1}$$

$$\mathcal{L}_{f} = -\sum_i \cos\left(\mathcal{T}_{f}(x_i), \mathcal{S}_{f}(x_i)\right) + \beta \cos\left(\mathcal{S}_{f}(x_i), \mathcal{S}_{f}(\tilde{x}_i)\right) \tag{2}$$

$$\mathcal{L}_{\text{ProFeAT}} = \frac{1}{2}\left(\mathcal{L}_{pf} + \mathcal{L}_{f}\right) \tag{3}$$

$$\tilde{x}_i = \underset{\tilde{x}_i : ||\tilde{x}_i - x_i||_\infty \leq \varepsilon}{\arg\min} \cos\left(\mathcal{T}_{pf}(x_i), \mathcal{S}_{pf}(\tilde{x}_i)\right) + \cos\left(\mathcal{S}_{f}(x_i), \mathcal{S}_{f}(\tilde{x}_i)\right) \tag{4}$$

Here, L_{pf} and L_f are the defense losses enforced at the projection and feature spaces, respectively. $\tilde{\mathcal{S}} = \mathcal{S}(\tilde{x})$ is the student representation for the adversarial input $\tilde{x}$ (see Section 2 for notations). The first terms in Equations (1) and (2) represents the **Distillation loss** (see Figure 1) involving only the clean input x_i, whereas the second terms correspond to the **Smoothness loss** at the respective layers of the student, involving both clean and adversarial input $\tilde{x}_i$, and is weighted by a hyperparameter β that controls the robustness-accuracy trade-off in the downstream model. The overall loss $\mathcal{L}_{\text{ProFeAT}}$ in Equation (3) is minimized during training.

Attack generation: The attack used during training is generated by a maximizing a combination of losses at both projection and feature spaces as shown in Equation (4). Since the projection space is primarily used for enforcing similarity with the teacher, we minimize the cosine similarity between the teacher and student representations for attack generation. Since the feature space is primarily used for enforcing local smoothness in the loss surface of the student, we utilize the unsupervised formulation that minimizes similarity between representations of clean and adversarial samples at the student.

Augmentations: Standard supervised and self-supervised training approaches are known to benefit from the use of strong data augmentations such as AutoAugment (Cubuk et al., 2018). However, such augmentations, which distort the low-level features of images, are known to deteriorate the performance of adversarial training (Rice et al., 2020; Gowal et al., 2020). Addepalli et al. (2022) attribute the poor performance to the larger domain shift between the augmented train and unaugmented test set images, in addition to the increased complexity of the adversarial training task, which overpower the superior generalization attained due to the use of diverse augmentations. Although these factors influence adversarial training in the self-supervised regime as well, we hypothesize that the need for better generalization is higher in self-supervised training,

Table 4: **SOTA comparison:** Standard Linear Probing performance (%) on CIFAR-10 and CIFAR-100 datasets on ResNet-18 and WideResNet-34-10 models. Mean and standard deviation across 3 reruns are reported for the best baseline DeACL (Zhang et al., 2022) and the proposed approach ProFeAT. Standard Accuracy (SA), Robust Accuracy against PGD-20 attack (RA-PGD20) and AutoAttack (RA-AA) reported.

Method	**CIFAR-10**			**CIFAR-100**		
	SA	**RA-PGD20**	**RA-AA**	**SA**	**RA-PGD20**	**RA-AA**
ResNet-18						
Supervised (TRADES)	83.74	49.35	47.60	59.07	26.22	23.14
AP-DPE	78.30	18.22	16.07	47.91	6.23	4.17
RoCL	79.90	39.54	23.38	49.53	18.79	8.66
ACL	77.88	42.87	39.13	47.51	20.97	16.33
AdvCL	80.85	50.45	42.57	48.34	27.67	19.78
DynACL	77.41	-	45.04	45.73	-	19.25
DynACL++	79.81	-	46.46	52.26	-	20.05
DeACL (Reported)	80.17	53.95	45.31	52.79	30.74	20.34
DeACL (Our Teacher)	80.05$\pm$ 0.29	52.97$\pm$ 0.08	**48.15**$\pm$ 0.05	51.53$\pm$ 0.30	30.92$\pm$ 0.21	21.91$\pm$ 0.13
ProFeAT (**Ours**)	**81.68**$\pm$ 0.23	49.55$\pm$ 0.16	47.02$\pm$ 0.01	**53.47**$\pm$ 0.10	27.95$\pm$ 0.13	**22.61**$\pm$ 0.14
WideResNet-34-10						
Supervised (TRADES)	85.50	54.29	51.59	59.87	28.86	25.72
DynACL++	80.97	48.28	45.50	52.60	23.42	20.58
DeACL	83.83$\pm$ 0.20	57.09$\pm$ 0.06	48.85$\pm$ 0.11	52.92$\pm$ 0.35	32.66$\pm$ 0.08	23.82$\pm$ 0.07
ProFeAT (**Ours**)	**87.62**$\pm$ 0.13	54.50$\pm$ 0.17	**51.95**$\pm$ 0.19	**61.08**$\pm$ 0.18	31.96$\pm$ 0.08	**26.81**$\pm$ 0.11

since the pretraining task is not aligned with the ideal goals of the student, making it important to use strong augmentations. However, it is also important to ensure that the training task is not too complex. We thus propose to use a combination of weak and strong augmentations as inputs to the teacher and student respectively, as shown in Figure 1. From Figure 5 we note that, the use of strong augmentations results in the generation of more diverse attacks, resulting in a larger drop when differently augmented images are used across different restarts of a PGD 5-step attack. The use of weak augmentations at the teacher imparts better supervision to the student, reducing the training complexity.

5 Experiments and Results

We first present an empirical evaluation of the proposed method, followed by several ablation experiments to understand the role of each component individually. Due to lack of space, details on the training, datasets and compute are presented in Appendices C and D.

For evaluation, we compare the proposed approach ProFeAT with several existing self-supervised adversarial training approaches (Chen et al., 2020a; Kim et al., 2020; Jiang et al., 2020; Fan et al., 2021; Zhang et al., 2022; 2019) by performing linear probing (LP) using cross-entropy loss on clean samples. To ensure a fair comparison, the same is done for the supervised AT method TRADES (Zhang et al., 2019) as well. The results are presented on CIFAR-10 and CIFAR-100 datasets Krizhevsky et al. (2009), and on ResNet-18 He et al. (2016) and WideResNet-34-10 (WRN-34-10) Zagoruyko & Komodakis (2016) architectures, as is common in the adversarial research community. The results of existing methods on ResNet-18 are as reported by Zhang et al. (2022). Since DeACL (Zhang et al., 2022) also uses a teacher-student architecture, we reproduce their results using the same teacher as used in our method, and report it as "DeACL (Our Teacher)". Since existing methods do not report results on larger architectures like WideResNet-34-10, we compare our results only with the best performing method (DeACL) and a recent approach DynACL (Luo et al., 2023). As the results with larger architectures are not reported in these papers, we run them using their official code.

Table 5: **Performance with additional evaluations:** Performance (%) of DeACL (best baseline) and ProFeAT (Ours) on CIFAR-10 and CIFAR-100 datasets with WideResNet-34-10 architecture. The model is first pretrained using the respective self-supervised adversarial training algorithm, and further we compute the standard accuracy (SA) and robust accuracy against GAMA (RA-G) using several methods such as standard linear probing (LP), training a 2 layer MLP head (MLP), and performing KNN in the feature space with $k = 10$. The proposed method achieves improvements over the baseline across all evaluation methods.

Method	LP Eval		MLP Eval		KNN Eval ($k = 10$)	
	SA	**RA-G**	**SA**	**RA-G**	**SA**	**RA-G**
CIFAR-10						
DeACL	83.60	49.62	85.66	48.74	87.00	54.58
ProFeAT (**Ours**)	**87.44**	**52.24**	**89.37**	**50.00**	**87.38**	**55.77**
CIFAR-100						
DeACL	52.90	24.66	55.05	22.04	56.82	31.26
ProFeAT (**Ours**)	**61.05**	**27.41**	**63.81**	**26.10**	**58.09**	**32.26**

The Robust Accuracy (RA) in the SOTA comparison table is presented against AutoAttack (RA-AA) (Croce & Hein, 2020) which is widely used as a reliable benchmark for robustness evaluation (Croce et al., 2021). In other tables, we also present robust accuracy against the GAMA attack (RA-G) (Sriramanan et al., 2020) which is known to be a competent and a reliable estimate of AutoAttack while being significantly faster to evaluate. We additionally present results against a 20-step PGD attack (RA-PGD20) (Madry et al., 2018) in the SOTA comparison table (Table 4), although it is a significantly weaker attack. A larger difference between RA-PGD20 and RA-AA indicates that the loss surface is more convoluted due to which weaker attacks are unsuccessful, yielding a false sense of robustness (Athalye et al., 2018). Thus this difference serves as a check for verifying the extent of gradient masking (Papernot et al., 2017; Tramèr et al., 2018). Therefore, in order to compare true robustness between any two defenses, accuracy against RA-AA or RA-G should be considered, while RA-PGD20 should not be considered. The accuracy on clean or natural samples is denoted as SA, which stands for Standard (Clean) Accuracy.

5.1 Comparison with the state-of-the-art

Table 4 presents the standard linear probing results of the proposed method ProFeAT in comparison to several SSL-AT baseline approaches. The proposed approach obtains superior robustness-accuracy trade-off when compared to the best performing baseline method DeACL, with $\sim 3{-}3.5\%$ gains in both robust and clean accuracy on CIFAR-10 dataset and similar gains in robustness on CIFAR-100 dataset with WideResNet-34-10 architecture. We obtain significant gains of $\sim 8\%$ on the clean accuracy on CIFAR-100. With ResNet-18 architecture, ProFeAT achieves competent robustness-accuracy trade-off when compared to DeACL on CIFAR-10 dataset, and obtains $\sim 2\%$ higher clean accuracy alongside improved robustness on CIFAR-100 dataset. We notice significantly less gradient masking, indicated by a lower value of (RA-PGD20 − RA-AA), for the proposed approach compared to all other baselines across all settings, indicating a reliable attack generation even in the absence of ground truth labels. Overall, the proposed approach significantly outperforms all the existing baselines, especially for larger model capacities (WRN-34-10), with improved results on smaller models (ResNet-18). Additionally, we obtain superior results when compared to the supervised AT method TRADES as well, at higher model capacities.

Performance comparison with other evaluation methods: We also present results of additional evaluation methods to evaluate the performance of the pretrained backbone in Table 5. We note that the proposed method achieves improvements over the baseline across all evaluation methods. Since the training of classifier head in LP and MLP is done using standard training and not adversarial training, the robust accuracy reduces as the number of layers increases (from linear to 2-layers), and the standard accuracy improves. The standard accuracy of KNN is better than the standard accuracy of LP for the baseline, indicating that the representations are not linearly separable. Whereas, as is standard, for the proposed approach, LP standard accuracy is higher than that obtained using KNN. The adversarial attack used for evaluating the robust accuracy using KNN is generated using GAMA attack on a linear classifier. The attack

Table 6: **Transfer Learning with Linear Probing:** Standard LP Performance (%) for transfer learning from CIFAR-10/100 to STL-10 dataset on ResNet-18 and WRN-34-10 model. ProFeAT (Ours) outperforms both DeACL and the supervised TRADES model.

Method	CIFAR-10 → STL-10		CIFAR-100 → STL-10	
	SA	**RA-AA**	**SA**	**RA-AA**
ResNet-18				
Supervised	54.70	22.26	51.11	19.54
DeACL	60.10	30.71	50.91	16.25
ProFeAT	**64.30**	**30.95**	**52.63**	**20.55**
WideResNet-34-10				
Supervised	67.15	30.49	**57.68**	11.26
DeACL	66.45	28.43	50.59	13.49
ProFeAT	**69.88**	**31.65**	56.68	**19.46**

Table 7: **Transfer Learning with Adv. Full-Finetuning:** AFF performance (%) using TRADES algorithm for 25 epochs, when transferred from CIFAR-10/100 to STL-10 and Caltech-101 dataset on WideResNet-34-10 model.

Method	SA	RA-G	SA	RA-G
	CIFAR-10 → STL-10		CIFAR-100 → STL-10	
Supervised	64.58	32.78	64.22	31.01
DeACL	61.65	28.34	60.89	30.00
ProFeAT	**74.12**	**36.04**	**68.77**	**31.23**
	CIFAR-10 → Caltech-101		CIFAR-100 → Caltech-101	
Supervised	62.46	39.40	**64.97**	41.02
DeACL	62.65	39.18	61.01	39.09
ProFeAT	**66.11**	**42.12**	64.16	**41.25**

Table 8: **Efficiency of Self-supervised adversarial training (CIFAR-100, WRN-34-10):** Performance (%) using lesser number of attack steps (2 steps) when compared to the standard case (5 steps) during adversarial training. Clean/ Standard Accuracy (SA) and robust accuracy against GAMA attack (RA-G) and AutoAttack(RA-AA) are reported. The proposed approach is stable at lower attack steps as well, while being better than both TRADES (Zhang et al., 2019) and DeACL (Zhang et al., 2022).

Method	2-step PGD attack			5-step PGD attack		
	SA	**RA-G**	**RA-AA**	**SA**	**RA-G**	**RA-AA**
Supervised (TRADES)	**60.80**	24.49	23.99	**61.05**	25.87	25.77
DeACL	51.00	24.89	23.45	52.90	24.66	23.92
ProFeAT (**Ours**)	60.43	26.90	**26.23**	**61.05**	27.41	**26.89**

is suboptimal since it is not generated by using the evaluation process (KNN), and thus the robust accuracy against such an attack is higher.

Transfer learning: To evaluate the transferrability of the learned robust representations, we compare the proposed approach with the best baseline DeACL in Table 6 under standard linear probing (LP). We consider transfer from CIFAR-10/100 to STL-10 (Coates et al., 2011). When compared to DeACL, the clean accuracy is $\sim 4-10\%$ higher on CIFAR-10 and $\sim 1.7-6\%$ higher on CIFAR-100. We also obtain $3-5\%$ higher robust accuracy when compared to DeACL on CIFAR-100, and higher improvements over TRADES. We also present transfer learning results using lightweight adversarial full finetuning (AFF) to STL-10 and Caltech-101 (Li et al., 2022) in Table 7. We defer the details on the process of adversarial full-finetuning and the selection criteria for the transfer datasets to Appendix D. The transfer is performed on WRN-34-10 model that is pretrained on CIFAR-10/100. As shown in Table 7, the proposed method outperforms DeACL by a significant margin. Note that by using merely 25 epochs of adversarial full-finetuning, the proposed method achieves improvements of around 4% on CIFAR-10 and 11% on CIFAR-100 when compared to the linear probing accuracy presented in Table 6, highlighting the practical utility of the proposed method. The AFF performance of the proposed approach is better than that of a supervised TRADES pretrained model as well.

Efficiency of self-supervised adversarial training Similar to prior works (Zhang et al., 2022), the proposed approach uses 5-step PGD based optimization for attack generation during adversarial training. In Table 8, we present results with lesser optimization steps (2 steps). The proposed approach is stable and obtains similar results even by using 2-step attack. Even in this case, the clean and robust accuracy of the proposed approach is significantly better than the baseline approach DeACL (Zhang et al., 2022), and also outperforms the supervised TRADES model (Zhang et al., 2019). We compare the computational aspects of the existing baselines and the proposed method in more detail in Appendix E.

Table 9: **Ablations on ProFeAT (CIFAR-100, WRN-34-10):** Performance (%) by enabling different components of the proposed approach. A tick mark in the Projector column means that a frozen pretrained projector is used for the teacher and student, with the defense loss being enforced at the feature and projector as shown in Equation (3). E1 represents the baseline or DeACL defense, and E9 represents the proposed defense or ProFeAT. E8*:Defense loss applied only at the projector. SA: Standard Accuracy, RA-G: Robust Accuracy against GAMA attack.

Ablation	Projector	Augmentations	Attack loss	SA	RA-G
E1				52.90	24.66
E2	✓			57.66	25.04
E3		✓		52.83	27.13
E4			✓	51.80	24.77
E5		✓	✓	55.35	**27.86**
E6	✓		✓	56.57	25.29
E7	✓	✓		**62.01**	26.89
E8	✓*	✓	✓	59.65	26.90
E9	✓	✓	✓	61.05	27.41

5.2 Ablations

We now present some of the ablation experiments to gain further insights into the proposed method, and defer more in-depth ablation results to Appendix F due to space constraints.

Effect of each component of the proposed approach: We study the impact of each component of the ProFeAT in Table-9, and make the following observations based on the results:

- *Projector*: A key component of the proposed method is the introduction of the projector. We observe significant gains in clean accuracy ($\sim 5\%$) by introducing the projector along with defense losses at the feature and projection spaces (E1 vs. E2). The importance of the projector is also evident by the fact that removing the projector from the proposed defense results in a large drop (5.7%) in clean accuracy (E9 vs. E5). We observe a substantial improvement of 9.2% in clean accuracy when the projector is introduced in the presence of the proposed augmentation strategy (E3 vs. E7), which is significantly higher than the gains obtained by introducing the same in the baseline DeACL (4.76%, E1 vs. E2). Further ablations on the projector are provided in Appendix F.1.
- *Augmentations*: The proposed augmentation strategy improves robustness across all settings. Introducing the proposed strategy in the baseline improves its robust accuracy by 2.47% (E1 vs. E3). Moreover, the importance of the proposed strategy is also evident from the fact that in the absence of the same, there is a 4.48% drop in SA and $\sim 2\%$ drop in RA-G (E9 vs. E6). Further, when combined with other components as well, the proposed augmentation strategy shows good improvements (E4 vs. E5, E2 vs. E7, E6 vs. E9). Detailed ablation on the respective augmentations used for the teacher and student model can be found in Appendix F.2.
- *Attack loss*: The proposed attack objective is designed to be consistent with the proposed defense strategy, where the goal is to enforce smoothness at the student in the feature space and similarity with the teacher in the projector space. The impact of the attack loss in feature space can be seen in combination with the proposed augmentations, where we observe an improvement of 2.5% in clean accuracy alongside notable improvements in robust accuracy (E3 vs. E5). However, in presence of projector, the attack results in only marginal robustness gains, possibly because the clean accuracy is already high (E9 vs. E7). More detailed ablations on the attack loss in provided in Appendix F.3.
- *Defense loss*: We do not introduce a separate column for defense loss as it is applicable only in the presence of the projector. We show the impact of the proposed defense losses in the last two rows (E8 vs. E9). The proposed defense loss improves the clean accuracy by 1.4% and robust accuracy marginally. Appendix F.4 provides further insights on various defense loss formulations and their impact on the proposed method.

Performance across different model architectures: We report performance of the proposed method ProFeAT and the best baseline DeACL on diverse architectures including Vision transformers (Dosovitskiy

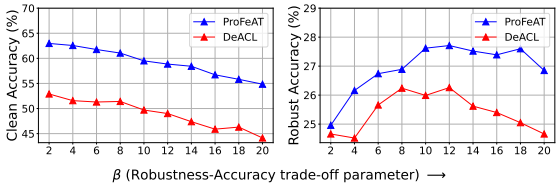

Figure 2: Performance of ProFeAT when compared to DeACL (Zhang et al., 2022) across variation in the robustness-accuracy trade-off parameter β on CIFAR-100 dataset with WRN-34-10 architecture.

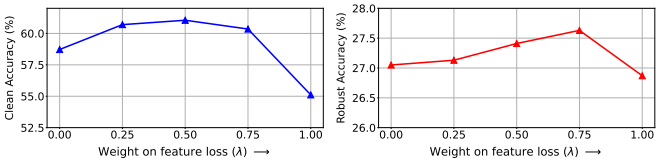

Figure 3: Performance (%) of ProFeAT by varying the weight λ between the defense losses at the feature and projector. The performance is stable across the range $\lambda \in [0.25, 0.75]$. We thus fix the value of λ to 0.5 in the proposed approach.

Table 10: **Performance across different architecture types and sizes:** Standard Linear Probing performance (%) of DeACL (Baseline) and ProFeAT (Ours) across different architectures on CIFAR-100. ViT-B/16 uses Imagenet-1K trained SSL teacher for training, while the SSL teacher in all other cases is trained on the CIFAR-100. SA: Standard Accuracy, RA-AA: Robust Accuracy against AutoAttack.

Method	**#params (M)**	**DeACL**		**ProFeAT (Ours)**	
		SA	**RA-AA**	**SA**	**RA-AA**
ResNet-18	11.27	51.53	21.91	**53.47**	**22.61**
ResNet-50	23.50	53.30	23.00	**59.34**	**25.86**
WideResNet-34-10	46.28	52.92	23.82	**61.08**	**26.81**
ViT-B/16	85.79	61.34	17.49	**65.08**	**21.52**

et al., 2021) on the CIFAR-100 dataset in Table 10. ProFeAT consistently outperforms DeACL in both clean and robust accuracy across various model architectures. An explanation behind the mechanism for successful scaling to larger datasets can be found in Appendix B.

Robustness-Accuracy trade-off: We present results across variation in the robustness-accuracy trade-off parameter β (Equations (1) and (2)) in Figure 2. Both robustness and accuracy of the proposed method are significantly better than DeACL across all values of β.

Weighting of defense losses at the feature and projector: In the proposed approach, the defense losses are equally weighted between the feature and projector layers as shown in Equation (3). In Figure 3, we present results by varying the weighting λ between the defense losses at the feature ($\mathcal{L}_f$) and projector ($\mathcal{L}_{pf}$) layers: $\mathcal{L}_{\text{ProFeAT}} = \lambda \cdot \mathcal{L}_f + (1-\lambda) \cdot \mathcal{L}_{pf}$, where L_{pf} and L_f are given by Equations (1) and (2), respectively. It can be noted that the two extreme cases of $\lambda = 0$ and $\lambda = 1$ result in a drop in clean accuracy, with a larger drop in the case where the loss is enforced only at the feature layer. The robust accuracy shows lesser variation across different values of λ. Thus, the overall performance is stable over the range $\lambda \in [0.25, 0.75]$, making the default setting of $\lambda = 0.5$ a suitable option.

6 Conclusion

In this work, we bridge the performance gap between supervised and self-supervised adversarial training approaches, specifically for large capacity models. We utilize a teacher-student setting (Zhang et al., 2022) where a standard self-supervised trained teacher is used to provide supervision to the student. Due to the inherent misalignment between the teacher training objective and the ideal goals of the student, we propose to use a projection layer to prevent the network from overfitting to the standard SSL trained teacher. We present a detailed analysis on the use of projection layer in distillation to justify our method. We additionally propose appropriate attack and defense losses in the feature and projector spaces alongside the use of weak and strong augmentations for the teacher and student respectively, to improve the attack diversity while maintaining low training complexity. The proposed approach obtains significant gains over existing self-supervised adversarial training methods, specifically for large models, demonstrating its scalability. While we improve the scalability of self-supervised AT methods to larger capacity models, we limit the datasets to CIFAR scale similar to prior works in this field due to the large computational costs. We hope future works explore the application of such SSL-AT methods to large scale datasets such as ImageNet.

References

Sravanti Addepalli, Samyak Jain, and R.Venkatesh Babu. Efficient and effective augmentation strategy for adversarial training. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:1488–1501, 2022.

Sravanti Addepalli, Anshul Nasery, Venkatesh Babu Radhakrishnan, Praneeth Netrapalli, and Prateek Jain. Feature reconstruction from outputs can mitigate simplicity bias in neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *The European Conference on Computer Vision (ECCV)*, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Improving deep networks generalization by removing their head. *arXiv preprint arXiv:2206.13378*, 2022.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1739–1747, 2020.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 699–708, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark, 2021.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations (ICLR)*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34, 2021.

Yuting Gao, Jia-Xin Zhuang, Shaohui Lin, Hao Cheng, Xing Sun, Ke Li, and Chunhua Shen. Disco: Remedy self-supervised learning on lightweight models with distilled contrastive learning. In *The European Conference on Computer Vision (ECCV)*, 2022.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Kartik Gupta, Thalaiyasingam Ajanthan, Anton van den Hengel, and Stephen Gould. Understanding and improving the role of projection head in self-supervised learning, 2022.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 33:16199–16210, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:2983–2994, 2020.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022.

Rundong Luo, Yifei Wang, and Yisen Wang. Rethinking the effect of data augmentation in adversarial contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Tsipras Dimitris, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

KL Navaneet, Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Simreg: Regression as a simple yet effective tool for self-supervised knowledge distillation. *arXiv preprint arXiv:2201.05131*, 2022.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *International Conference on Learning Representations (ICLR)*, 2021.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (ACM ASIACCS)*, 2017.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning (ICML)*, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and R Venkatesh Babu. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018.

Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *International Conference on Machine Learning (ICML)*, 2020.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pp. arXiv–1807, 2018.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations (ICLR)*, 2018.

Yihao Xue, Eric Gan, Jiayi Ni, Siddharth Joshi, and Baharan Mirzasoleiman. Investigating the benefits of projection head for representation learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Axi Niu, Jiu Feng, Chang D Yoo, and In So Kweon. Decoupled adversarial contrastive learning for self-supervised adversarial robustness. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pp. 725–742. Springer, 2022.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

Appendix

A Background

A.1 Decoupled Adversarial Contrastive Learning (DeACL)

DeACL introduces a teacher-student distillation framework to train robust models in a self-supervised setting without the need of labeled training data. While most self-supervised adversarial training methods aimed at integrating contrastive learning methods with adversarial training, the authors (Zhang et al., 2022) showed that combining the two is a complex optimization problem due to their conflicting training requirements. The training is thus decoupled to two stages as discussed below. The first stage trains a standard self-supervised model using any of the existing Self-supervised Learning (SSL) techniques such as SimCLR, BYOL, or BarlowTwins. This is followed by the second stage that involves adversarial training of a *student* model using the supervision from the earlier-trained teacher in a distillation framework as shown in Figure 4. The loss used during distillation is a combination of the cosine similarity between the representations of the clean image at the output of the teacher and student, along with a smoothness loss at the output of the student as shown in Equation (5).

$$\mathcal{L}_{\text{DeACL}} = -\sum_i \cos\left(\mathcal{T}_f(x_i), \mathcal{S}_f(x_i)\right) + \beta \cos\left(\mathcal{S}_f(x_i), \mathcal{S}_f(\tilde{x}_i)\right) \tag{5}$$

$$\tilde{x}_i = \underset{\tilde{x}_i : ||\tilde{x}_i - x_i||_\infty \leq \varepsilon}{\arg\min} \cos\left(\mathcal{T}_f(x_i), \mathcal{S}_f(\tilde{x}_i)\right) \tag{6}$$

In Equation (5), the first term is the **Distillation loss** on the feature backbone $\mathcal{T}_f$ of the teacher $\mathcal{T}$, and $\mathcal{S}_f$ of the student $\mathcal{S}$ for a clean input x_i. $\tilde{x}$ is the adversarial input. The second term corresponds to the **Smoothness loss** at the student to enforce similar representations for the clean and adversarial inputs, and is weighted by a hyperparameter β that controls the robustness-accuracy trade-off in the downstream model. The loss $\mathcal{L}_{\text{DeACL}}$ (Equation (5)) is minimized during training. The attack used during training is generated by a minimizing the cosine similarity between the teacher and student representations at the feature output as shown in Equation (6).

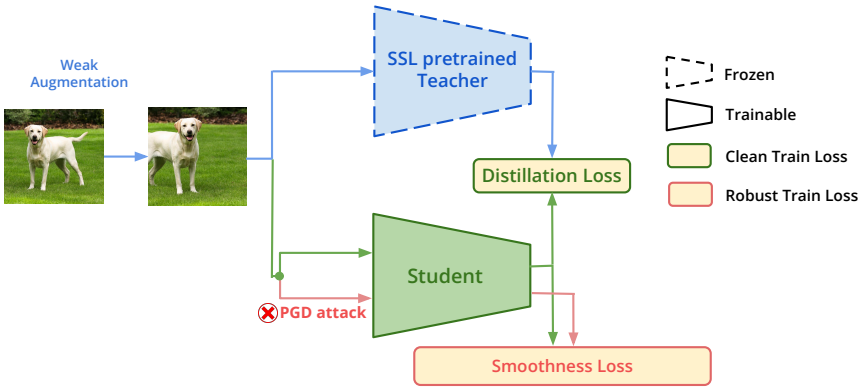

Figure 4: **Decoupled Adversarial Contrastive Learning (DeACL):** The student is trained using a distillation loss on clean samples using supervision from an SSL pretrained teacher, and a smoothness loss to enforce adversarial robustness (Equation (5)).

A.2 Supervised Adversarial Defenses

Following the demonstration of adversarial attacks by Szegedy et al. (2013), there have been several attempts of defending Deep Networks against them. Early defenses proposed intuitive methods that introduced non-differentiable or randomized components in the network to thwart gradient-based attacks (Buckman et al., 2018; Ma et al., 2018; Dhillon et al., 2018; Xie et al., 2018; Song et al., 2018). While these methods were efficient

and easy to implement, Athalye et al. (2018) proposed adaptive attacks which successfully broke several such defenses by replacing the non-differentiable components with smooth differentiable approximations, and by taking an expectation over the randomized components. Adversarial Training (Madry et al., 2018; Zhang et al., 2019) was the most successful defense strategy that withstood strong white-box (Croce & Hein, 2020; Sriramanan et al., 2020), black-box (Andriushchenko et al., 2020; Chen & Gu, 2020) and adaptive attacks (Athalye et al., 2018; Tramer et al., 2020) proposed over the years. PGD (Projected Gradient Descent) based adversarial training (Madry et al., 2018) involves maximizing the cross-entropy loss to generate adversarial attacks, and further minimizing the loss on the adversarial attacks for training. Another successful supervised Adversarial Training based defense was TRADES (Zhang et al., 2019), where the Kullback-Leibler (KL) divergence between clean and adversarial samples was minimized along with the cross-entropy loss on clean samples for training. Adjusting the weight between the losses gave a flexible trade-off between the clean and robust accuracy of the trained model. Although these methods have been robust against several attacks, it has been shown that the sample complexity of adversarial training is large (Schmidt et al., 2018), and this increases the training and annotation costs needed for adversarial training.

B Mechanism behind Scaling to Larger Datasets

For a sufficiently complex task, a scalable approach results in better performance on larger models given enough data. Although the task complexity of adversarial self-supervised learning is high, the gains in prior approaches are marginal with an increase in model size, while the proposed method results in significantly improved performance on larger capacity models (Table 4). We discuss the key factors that result in better scalability below:

- As discussed in Section 4.1, a mismatch between training objectives of the teacher and ideal goals of the student causes a drop in student performance. This primarily happens because of the overfitting to the teacher training task. As model size increases, the extent of overfitting increases. The use of a projection layer during distillation alleviates the impact of this overfitting and allows the student to retain more generic features that are useful for the downstream robust classification objective. Thus, a projection layer is more important for larger model capacities where the extent of overfitting is higher.
- Secondly, as the model size increases, there is a need for higher amount of training data for achieving better generalization. The proposed method has better data diversity as it enables the use of more complex data augmentations in adversarial training by leveraging supervision from weak augmentations at the teacher.

C Details on Datasets

We compare the performance of the proposed approach ProFeAT with existing methods on the benchmark datasets CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), that are commonly used for evaluating the adversarial robustness of models (Croce et al., 2021). Both datasets consist of RGB images of dimension 32×32. CIFAR-10 consists of 50,000 images in the training set and 10,000 images in the test set, with the images being divided equally into 10 classes - airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. CIFAR-100 dataset is of the same size as CIFAR-10, with images being divided equally into 100 classes. Due to the larger number of classes, there are only 500 images per class in CIFAR-100, making it a more challenging dataset when compared to CIFAR-10.

D Details on Training and Compute

Model Architecture: We report the key comparisons with existing methods on two of the commonly considered model architectures in the literature of adversarial robustness (Pang et al., 2021; Zhang et al., 2019; Rice et al., 2020; Croce et al., 2021) - ResNet-18 (He et al., 2016) and WideResNet-34-10 (Zagoruyko & Komodakis, 2016). Although most existing methods for self-supervised adversarial training report results

Table 11: **Total number of Forward (FP) or Backward (BP) Propagations during training** of the proposed approach when compared to prior works. Distillation based approaches - ProFeAT and DeACL require significantly lesser compute when compared to prior methods, and are only more expensive than supervised adversarial training.

Method	#Epochs	#Attack steps	#FP or BP		
			AT	Teacher model	Total
Supervised (TRADES)	110	10	1210	-	1210
AP-DPE	450	10	4950	-	4950
RoCL	1000	5	6000	-	6000
ACL	1000	5	12000	-	12000
AdvCL	1000	5	12000	1000	13000
DynACL	1000	5	12000	-	12000
DynACL++	1025	5	12300	-	12300
DeACL	100	5	700	1000	1700
ProFeAT (**Ours**)	100	5	800	1000	1800

only on ResNet-18 (Zhang et al., 2022; Fan et al., 2021), we additionally consider the WideResNet-34-10 architecture to demonstrate the scalability of the proposed approach to larger model architectures. We perform the ablation experiments on the CIFAR-100 dataset with WideResNet-34-10 architecture, which is a very challenging setting in self-supervised adversarial training, to be able to better distinguish between different variations adopted during training.

Training Details: The self-supervised training of the teacher model is performed for 1000 epochs with the SimCLR algorithm (Chen et al., 2020b) similar to prior work (Zhang et al., 2022). We utilize the `solo-learn` [1] GitHub repository for this purpose. For the SimCLR SSL training, we tune and use a learning rate of 1.5 with SGD optimizer, a cosine schedule with warmup, weight decay of $1e{-}5$ and train the backbone for 1000 epochs with other hyperparameters kept as default as in the repository.

The self-supervised adversarial training of the feature extractor using the proposed approach is performed for 100 epochs using SGD optimizer with a weight decay of $3e{-}4$, cosine learning rate with 10 epochs of warm-up, and a maximum learning rate of 0.5. We consider the standard ℓ_∞ based threat model Zhang et al. (2022); Fan et al. (2021) for attack generation during both training and evaluation, i.e., $||\tilde{x}_i - x_i||_\infty \leq \varepsilon$, where $\tilde{x}$ is the adversarial version of the clean sample x_i. The value of ε is set to 8/255 for both CIFAR-10 and CIFAR-100 datasets, as is standard in literature (Madry et al., 2018; Zhang et al., 2019). A 5-step PGD attack is used for attack generation during training with a step size of 2/255. We fix the value of β, the robustness-accuracy trade-off parameter (Ref: Equations (1) and (2) in the main paper) to 8 in all our experiments, unless specified otherwise.

Details on configurations in Tables 2 and 3: In these tables, we present the experiments showing the importance of the projector in self-supervised distillation, first in a standard setting (Table 2) and then in an adversarial setting (Table 3). In Table 2, the linear probing performance of the student model is shown at both the feature and projection space when linear probed using two different probing methods (column "LP Loss"): standard CE loss (S1, S3-S5), and cosine similarity loss between the student and the teacher (denoted as $cos(\mathcal{T}, \mathcal{S})$, S2). Experiments S1-S4 refer to settings where there is no projector involved during distillation, while S5 includes the projector network. S1 and S2 considers a self-supervised trained teacher for distillation, whereas S3 and S4 uses a standard teacher trained using label supervision. In the absence of a projector, when the training objectives of teacher matches with the student (S3/S4 vs. S1), or when similar losses are used for pretraining and linear probing (S2 vs. S1), we see an improvement in the downstream performance of the student. On the other hand, in the event of misalignment between the student and teacher objectives, one can employ a projector network (see S5 vs. S1) to alleviate the impact of the mismatch and gain back the performance of the student. In that case, the similarity between the teacher and the student is significantly higher at the projector space when compared to the feature space (S5).

[1] `https://github.com/vturrisi/solo-learn`

Table 12: **Floating Point Operations per Second (GFLOPS) and latency per epoch during training** of the proposed approach ProFeAT when compared to the baseline DeACL for ResNet-18 and WideResNet-34-10 models. The computational overhead during training is marginal with the addition of the projection layer, and reduces further for larger capacity models.

	ResNet-18			**WideResNet-34-10**		
	GFLOPS	**Time/epoch**	**#Params(M)**	**GFLOPS**	**Time/epoch**	**#Params(M)**
DeACL	671827	51s	11.27	6339652	4m 50s	46.28
ProFeAT (**Ours**)	672200	51s	11.76	6340197	4m 50s	46.86
% increase	0.056	0.00	4.35	0.009	0.00	1.25

Likewise, we show a similar behavior in the setting of adversarial self-supervised distillation in Table 3. Let's again consider the case when there's no projector. When using CE loss for linear probing (A3 vs. A1), using a supervised AT teacher for adversarial distillation (A3) which is also trained using CE loss, results in much better student performance compared to a self-supervised teacher (A1) which has been trained using a similarity-based objective, like a contrastive loss. On the other hand, when the linear probing objective is changed instead to a similarity-based loss (A2: $cos(\mathcal{T}, \mathcal{S})$), the mismatch of the student with a self-supervised teacher vanishes, and we recover the student performance (note similar clean performance of A2 and A3, w.r.t. A1). In the case of using a projector network, we see an improvement in the performance of the student model even when distilled SSL teacher for distillation and linear probed using CE loss (A4 vs. A1).

Details on Linear Probing: To evaluate the performance of the learned representations, we perform standard linear probing by freezing the adversarially pretrained backbone as discussed in Section 2 of the main paper. We use a class-balanced validation split consisting of 1000 images from the train set and perform early-stopping during training based on the performance on the validation set. The training is performed for 25 epochs with a step learning rate schedule where the maximum learning rate is decayed by a factor of 10 at epoch 15 and 20. The learning rate is chosen amongst the following settings — $\{0.1, 0.05, 0.1, 0.5, 1, 5\}$, with SGD optimizer, and the weight decay is fixed to $2e{-}4$. The same evaluation protocol is used for the best baseline DeACL (Zhang et al., 2022) as well as the proposed approach ProFeAT, for both in-domain and transfer learning settings.

Details on Transfer Learning: To perform transfer learning using Adversarial Full-Finetuning, a robustly pretrained base model is used as an initialization, which is then adversarially finetuned using TRADES Zhang et al. (2019) for 25 epochs. After finetuning, the model is evaluated using linear probing on the same transfer dataset. We consider STL-10 Coates et al. (2011) and Caltech-101 Li et al. (2022) as the transfer datasets. Caltech-101 contains 101 object classes and 1 background class, with 2416 samples in the train set and 6728 samples in the test set. The number of samples per class range from 17 to 30, and thus is a suitable dataset to highlight the practical importance of lightweight finetuning of an adversarial self-supervised pretrained representations for low-data regime.

Compute: The following Nvidia GPUs have been used for performing the experiments reported in this work - V100, A100, and A6000. Each of the experiments are run either on a single GPU, or across 2 GPUs based on the complexity of the run and GPU availability. Uncertainty estimates in the main SOTA table (Table 4 of the main paper) were obtained on the same machine and with same GPU configuration to ensure reproducibility. For 100 epochs of single-precision (FP32) training with a batch size of 256, the proposed approach takes ∼8 hours and ∼16GB of GPU memory on a single A100 GPU for WideResNet-34-10 model on CIFAR-100.

E Computational Complexity

In terms of compute, both the proposed method ProFeAT and DeACL (Zhang et al., 2022) lower the overall computational cost when compared to prior approaches. This is because self-supervised training in general requires larger number of epochs (1000) to converge (Chen et al., 2020b; He et al., 2020) when compared to supervised learning (<100). Prior approaches like RoCL (Kim et al., 2020), ACL (Jiang et al., 2020) and

Table 13: **Ablation on Projector training configuration (CIFAR-100, WRN-34-10):** Performance (%) using variations in projector (proj.) initialization (init.) and trainability. SA: Standard Accuracy, RA-G: Robust accuracy against GAMA attack.

Ablation	Student proj.	Proj. init. (Student)	Teacher proj	Proj. init. (Teacher)	SA	RA-G
AP1	Absent	-	Absent	-	55.35	27.86
AP2	Trainable	Random	Absent	-	63.07	26.57
AP3	Frozen	Pretrained	Absent	-	40.43	22.23
AP4	Trainable	Pretrained	Absent	-	62.89	26.57
AP5	Trainable	Random (common)	Trainable	Random (common)	53.43	27.23
AP6	Trainable	Pretrained (common)	Trainable	Pretrained (common)	54.60	27.41
AP7	Trainable	Pretrained	Frozen	Pretrained	58.18	27.73
Ours	Frozen	Pretrained	Frozen	Pretrained	61.05	27.41

Table 14: **Ablation on Projector architecture (CIFAR-100, WRN-34-10):** Performance (%) obtained by varying the projector configuration (config.) and architecture (arch.). A non-linear projector effectively reduces the gap in clean accuracy between the teacher and student. A bottleneck architecture for the projector is worse than other variants. SA: Standard Accuracy, RA-G: Robust Accuracy against GAMA attack.

Ablation	Projector config.	Projector arch.	Teacher SA	Student SA	Drop in SA	% Drop in SA	RA-G
APA1	No projector	-	70.85	55.35	15.50	21.88	**27.86**
APA2	Linear layer	640-256	68.08	53.35	14.73	21.64	27.47
Ours	2 Layer MLP	640-640-256	70.85	61.05	9.80	13.83	27.41
APA3	3 Layer MLP	640-640-640-256	70.71	60.37	10.34	14.62	27.37
APA4	2 Layer MLP	640-640-640	69.88	61.24	8.64	12.36	27.36
APA5	2 Layer MLP	640-2048-640	**70.96**	**61.76**	9.20	12.97	26.66
APA6	2 Layer MLP	640-256-640	69.37	57.87	11.50	16.58	27.56

AdvCL (Fan et al., 2021) combine the contrastive training objective of SSL approaches and the adversarial training objective. Thus, these methods require larger number of training epochs (1000) for the adversarial training task, which is already computationally expensive due to the requirement of generating multi-step attacks during training. ProFeAT and DeACL use a SSL teacher for training and thus, the adversarial training is more similar to supervised training, requiring only 100 epochs. In Table 11, we present the approximate number of forward and backward propagations for each algorithm, considering both pretraining of the auxiliary network used and the training of the main network. It can be noted that the distillation based approaches - ProFeAT and DeACL require significantly lesser compute when compared to prior methods, and are only more expensive than supervised adversarial training. In Table 12, we present the FLOPS required during training for the proposed approach and DeACL. One can observe that there is a negligible increase in FLOPS compared to the baseline approach.

F Additional Results

F.1 More ablations on the Projector

Training configuration of the Projector: We present ablations using different configurations of the projection layer in Table 13. As also discussed in Section 4.1 of the main paper, we observe a large boost in clean accuracy when a random (or pretrained) trainable projection layer is introduced to the student (AP2/ AP4 vs. AP1 in Table 13). While the use of pretrained frozen projection head only for the student degrades performance considerably (AP3), the use of the same for both teacher and student (Ours) yields a optimal robustness-accuracy trade-off across all variations. The use of a common trainable projection head for both teacher and student results in collapsed representations at the projector output (AP5, AP6), yielding results similar to the case where projector is not used for both teacher and student (AP1). This issue is overcome when the pretrained projector is trainable only for the student (AP7).

Ablation	Teacher	Student	SA	RA-G
AG1	PC	PC	56.57	25.29
AG2	AuAu	AuAu	60.76	27.21
AG3	PC1	PC2	56.95	25.39
AG4	AuAu1	AuAu2	59.51	**28.15**
AG5	AuAu	PC	57.28	26.14
Ours	PC	AuAu	**61.05**	27.41

Table 15: **Ablation on Augmentations used (CIFAR-100, WRN-34-10):** Performance (%) using different augmentations for the teacher and student. (PC: Pad+Crop, AuAu: AutoAugment). Standard Accuracy (SA) and Robust accuracy (RA-G) reported.

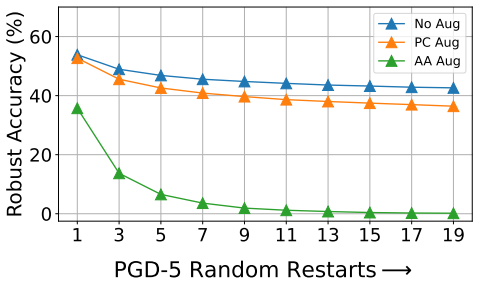

Figure 5: Robust accuracy of a supervised TRADES model across random restarts of PGD 5-step attack (CIFAR-100, WRN-34-10).

Architecture of the Projector: In the proposed approach, we use the following 2-layer MLP projector architecture for both SSL pretraining of the teacher and adversarial training of the student: (1) **ResNet-18: 512-512-256**, and (2) **WideResNet-34-10: 640-640-256**. In Table 14, we present results using different configurations and architectures of the projector. Firstly, the use of a linear projector (APA2) is similar to the case where projector is not used for student training (APA1), with $\sim 21\%$ drop in clean accuracy of the student with respect to the teacher. This improves to $12-17\%$ when a non-linear projector is introduced (APA3-APA6 and Ours). The use of a 2-layer MLP (Ours) is marginally better than the use of a 3-layer MLP (APA3) in terms of clean accuracy of the student. The accuracy of the student is stable across different architectures of the projector (Ours, APA4, APA5). However, the use of a bottleneck architecture (APA6) results in a higher drop in clean accuracy of the student.

F.2 Augmentation for the Student and Teacher

We present ablation experiments to understand the impact of different augmentations used for the teacher and student separately in Table 15. The base method (AG1) uses common Pad and Crop (PC) augmentation for both teacher and student. By using more complex augmentations —AutoAugment followed by Pad and Crop (denoted as AuAu in the table), there is a significant improvement in both clean and robust accuracy. By using separate augmentations for the teacher and student, there is an improvement in the case of PC (AG3), but a drop in clean accuracy accompanied by better robustness in case of AuAu. Finally by using a mix of both AuAu and PC at the student and teacher respectively (Ours), we obtain improvements in both clean and robust accuracy, since the former improves attack diversity (shown in Figure 5), while the latter makes the training task easier.

F.3 Attack Loss

For performing adversarial training using the proposed approach, attacks are generated by minimizing a combination of cosine similarity based losses as shown in Equation (4) of the main paper. This includes an unsupervised loss at the feature representations of the student and another loss between the representations of the teacher and student at the projector. As shown in Table 16, we obtain a better robustness-accuracy trade-off by using a combination of both losses rather than by using only one of the two losses, due to better diversity and strength of attack. These results also demonstrate that the proposed method is not very sensitive to different choices of attack losses.

F.4 Defense Loss

We present ablation experiments across variations in training loss at the feature space and the projection head in Table 17. In the proposed approach (Ours), we introduce a combination of clean and robust losses at both feature and projector layers, as shown in Equation (3) of the main paper. By introducing the loss only at the features (AD1), there is a considerable drop in clean accuracy as seen earlier, which can be recovered by introducing the clean loss at the projection layer (AD3). Instead, when only the robust loss is

Table 16: **Ablation on Attack Loss (CIFAR-100, WRN-34-10):** Performance (%) with variations in attack loss at feature (feat.) and projector (proj.). While the proposed defense is stable across several variations in the attack loss, minimizing a combination of both losses $\cos(\mathcal{T},\mathcal{S})$ and $\cos(\mathcal{S},\mathcal{S})$ gives the best robustness-accuracy trade-off. SA: Standard Accuracy, RA-G: Robust Accuracy against GAMA.

Ablation	Attack @ feat.	Attack @ proj.	SA	RA-G
AT1	$\cos(\mathcal{T},\mathcal{S})$	$\cos(\mathcal{T},\mathcal{S})$	60.84	26.78
AT2	$\cos(\mathcal{S},\mathcal{S})$	$\cos(\mathcal{S},\mathcal{S})$	61.30	26.75
AT3	$\cos(\mathcal{T},\mathcal{S})$	$\cos(\mathcal{S},\mathcal{S})$	60.69	**27.44**
AT4	$\cos(\mathcal{S},\mathcal{S})$	-	61.62	26.62
AT5	-	$\cos(\mathcal{S},\mathcal{S})$	61.09	27.00
AT6	$\cos(\mathcal{T},\mathcal{S})$	-	**62.01**	26.89
AT7	-	$\cos(\mathcal{T},\mathcal{S})$	61.18	27.24
Ours	$\cos(\mathcal{S},\mathcal{S})$	$\cos(\mathcal{T},\mathcal{S})$	61.05	27.41

Table 17: **Ablation on Defense Loss (CIFAR-100, WRN-34-10):** Performance (%) with variations in training loss at feature (feat.) and projector (proj.). "clean" denotes the cosine similarity between representations of teacher and student on clean samples. "adv" denotes the cosine similarity between representations of the corresponding clean and adversarial samples either at the output of student $(\mathcal{S},\mathcal{S})$ or between the teacher and student $(\mathcal{T},\mathcal{S})$. SA: Standard Accuracy, RA-G: Robust accuracy against GAMA.

Ablation	Loss @ feat.	Loss @ proj.	SA	RA-G
AD1	clean + adv$(\mathcal{S},\mathcal{S})$	-	55.35	**27.86**
AD2	-	clean + adv$(\mathcal{S},\mathcal{S})$	59.65	26.90
AD3	clean + adv$(\mathcal{S},\mathcal{S})$	clean	61.69	26.40
AD4	clean + adv$(\mathcal{S},\mathcal{S})$	adv$(\mathcal{S},\mathcal{S})$	49.59	25.35
AD5	adv$(\mathcal{S},\mathcal{S})$	clean	59.72	1.38
AD6	adv$(\mathcal{S},\mathcal{S})$	clean + adv$(\mathcal{S},\mathcal{S})$	59.22	26.50
AD7	clean	clean + adv$(\mathcal{S},\mathcal{S})$	62.24	25.97
AD8	clean + adv$(\mathcal{S},\mathcal{S})$	clean + adv$(\mathcal{T},\mathcal{S})$	63.85	23.91
AD9	clean + adv$(\mathcal{T},\mathcal{S})$	clean + adv$(\mathcal{T},\mathcal{S})$	**65.34**	22.40
Ours	clean + adv$(\mathcal{S},\mathcal{S})$	clean + adv$(\mathcal{S},\mathcal{S})$	61.05	27.41

Table 18: **Failure of AD5 (refer Table 17) defense loss for SSL-AT training of WRN-34-10 model on CIFAR-100:** Using clean and adversarial loss exclusively at projector and feature space respectively, results in an unstable optimization problem, giving either a non-robust model or collapsed representations at the end of training. As shown below, a lower value of β (the robustness-accuracy trade-off parameter) results in a non-robust model, while higher β results in the learning of collapsed representations.

β	Standard Accuracy (SA)	Robust Accuracy against GAMA (RA-G)
1	**67.34**	0.46
5	51.99	0.71
10	31.34	7.81
50	11.59	2.55
100	8.23	**2.61**

Table 19: **Ablations on the accuracy of the teacher SSL model (CIFAR-100, WRN-34-10):** Performance (%) obtained by varying the number of epochs for which the standard self-supervised teacher model is pretrained. Improvements in accuracy of the teacher result in corresponding gains in both standard and robust accuracy of the student. SA: Standard Accuracy, RA-G: Robust Accuracy against GAMA.

#Epochs of PT	Teacher SA	Student SA	Drop in SA	% Drop in SA	RA-G
100	55.73	49.37	6.36	11.41	20.86
200	65.43	56.16	9.27	14.17	24.15
500	69.27	59.62	9.65	13.93	26.75
1000	**70.85**	**61.05**	9.80	13.83	**27.41**

Table 20: **Ablations on the algorithm used for training the self-supervised teacher model (CIFAR-100, WRN-34-10):** Performance (%) of the proposed approach by varying the pretraining algorithm of the teacher model. A random trainable projector is used for training the student model, to maintain uniformity in projector architecture across all methods. SA: Standard Accuracy, RA-G: Robust Accuracy against GAMA attack.

Method (Teacher training)	Teacher SA	Student SA	RA-G
SimCLR	67.98	62.20	26.13
SimCLR (tuned)	70.85	63.07	26.57
BYOL	**72.97**	63.19	**26.82**
Barlow Twins	67.74	60.69	24.48
SimSiam	68.60	63.46	26.69
MoCoV3	72.48	**65.57**	26.65
DINO	68.75	60.61	24.80

introduced at the projection layer (AD4), there is a large drop in clean accuracy confirming that the need for projection layer is mainly enforcing the clean loss. When the combined loss is enforced only at the projection head (AD2), the accuracy is close to that of the proposed approach, with marginally lower clean and robust accuracy. Enforcing only adversarial loss in the feature space, and only clean loss in the projector space is a hard optimization problem, and this results in a non-robust model (AD5). As shown in Table 18, even by increasing β in AD5, we do not obtain a robust model, rather, there is a representation collapse. Thus, as discussed in Section 4.1 of the main paper, it is important to introduce the adversarial loss as a regularizer in the projector space as well (AD6). Enforcing only one of the two losses at the feature space (AD6 and AD7) also results in either inferior clean accuracy or robustness. Finally from AD8 and AD9 we note that the robustness loss is better when implemented as a smoothness constraint on the representations of the student, rather than by matching representations between the teacher and student. Overall, the proposed approach (Ours) results in the best robustness-accuracy trade-off.

F.5 Accuracy of the self-supervised teacher model

The self-supervised teacher model is obtained using 1000 epochs of SimCLR (Chen et al., 2020b) training in all our experiments. We now study the impact of training the teacher model for lesser number of epochs. As shown in Table 19, as the number of teacher training epochs reduces, there is a drop in the accuracy of the teacher, resulting in a corresponding drop in the clean and robust accuracy of the student model. Thus, the performance of the teacher is crucial for training a better student model.

F.6 Self-supervised training algorithm of the teacher

In the proposed approach, the teacher is trained using the popular self-supervised training algorithm SimCLR (Chen et al., 2020b), similar to prior works (Zhang et al., 2022). In this section, we study the impact of using different algorithms for the self-supervised training of the teacher and present results in Table 20. In

order to ensure consistency across different SSL methods, we use a *random trainable* projector (2-layer MLP with both hidden and output dimensions of 640) for training the student and do not employ any projection head for the pretrained frozen teacher. While the default teacher trained using SimCLR was finetuned across hyperparameters, we utilize the default hyperparameters from the `solo-learn` Github repository for this table, and thus present SimCLR also without tuning for a fair comparison. For uniformity, we report all results with $\beta = 8$ (the robustness-accuracy trade-off parameter). From Table 20, we note that in most cases, the clean accuracy of the student increases as the accuracy of the teacher improves, while the robust accuracy does not change much. We note that this table merely shows that the proposed approach can be effectively integrated with several base self-supervised learning algorithms for the teacher model. However, it does not present a fair comparison across different SSL pretraining algorithms, since the ranking on the final performance of the student would change if the pretraining SSL algorithms were used with appropriate hyperparameter tuning.