Benchmarking Robust Self-Supervised Learning
Across Diverse Downstream Tasks

Antoni Kowalczuk [* 1]   **Jan Dubiński** [* 2 3]   **Atiyeh Ashari Ghomi** [* 4]   **Yi Sui** [4]   **George Stein** [4]   **Jiapeng Wu** [4]
Jesse C. Cresswell [4]   **Franziska Boenisch** [1]   **Adam Dziedzic** [1]

Abstract

Large-scale vision models have become integral in many applications due to their unprecedented performance and versatility across downstream tasks. However, the robustness of these foundation models has primarily been explored for a single task, namely image classification. The vulnerability of other common vision tasks, such as semantic segmentation and depth estimation, remains largely unknown. We present a comprehensive empirical evaluation of the adversarial robustness of self-supervised vision encoders across multiple downstream tasks. Our attacks operate in the encoder embedding space and at the downstream task output level. In both cases, current state-of-the-art adversarial fine-tuning techniques tested only for classification significantly degrade clean and robust performance on other tasks. Since the purpose of a foundation model is to cater to multiple applications at once, our findings reveal the need to enhance encoder robustness more broadly. Our code is available at github.com/layer6ai-labs/ssl-robustness.

1. Introduction

Foundation models trained through self-supervised learning (SSL) have become the backbone of many applications due to their versatility; one foundation model can be adapted to many downstream tasks with a small amount of data and training (or fine-tuning). Foundation models in the vision domain have even outperformed dedicated models on several tasks (Caron et al., 2021; He et al., 2022; Oquab et al., 2024). Despite their broad utility, the adversarial robustness of these models has only been explored for classification
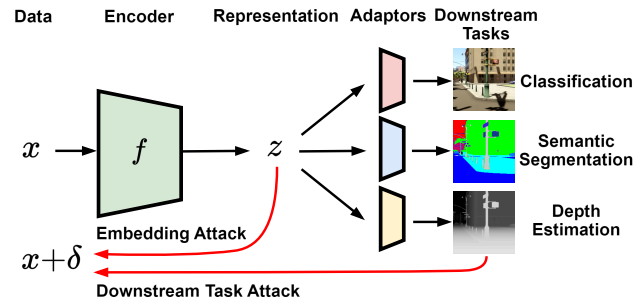

Figure 1: **Adversarial Attacks for SSL.** An SSL encoder is applied to downstream tasks through adaptors. Adversarial attacks can attack the downstream labels, or the embedding space directly.

tasks with linear probing (Naseer et al., 2020; Jiang et al., 2020; Fan et al., 2021; Zhang et al., 2022; Luo et al., 2023) while other common downstream tasks, such as semantic segmentation (Long et al., 2015) and depth estimation (Godard et al., 2019), remain unexplored. Recently, Li et al. (2023) showed that non-robust features extracted from adversarial examples for supervised models (and useful for classification) become largely useless when transferred to self-supervised learning paradigms. They advocated for a cross-paradigm examination of robustness, yet focused their analysis solely on classification. A major outstanding question is whether adversarial robustness transfers across downstream tasks.

We present an in-depth empirical evaluation of the adversarial robustness of self-supervised vision encoders (Chen and He, 2021; Caron et al., 2021) for downstream tasks beyond classification. We use attacks that operate in the encoder's embedding space (*EmbedAttack*) and those that leverage direct access to the downstream task outputs (*PGDAttacks*), *e.g., PGD* for classification (Madry et al., 2018) or *SegPGD* for semantic segmentation (Gu et al., 2022). Our main observation is that the state-of-the-art adversarial full fine-tuning of encoders (Zhang et al., 2022): (1) substantially lowers clean performance, (2) increases robustness only against the *EmbedAttack*, and (3) remains ineffective in improving robustness against the task-specific *PGDAttacks*. We observe only a slight improvement against the *PGDAttacks* for classification when the adversarial fine-tuning dataset and downstream dataset come from the same distribution. This

---

[*]Equal contribution   [1]CISPA Helmholtz Center for Information Security [2]Warsaw University of Technology [3]IDEAS NCBR [4]Layer 6 AI, Toronto, Canada. Correspondence to: Adam Dziedzic <adam.dziedzic@sprintml.com>.

indicates a need to rethink what it means for a foundation model to be robust. Finally, we offer potential approaches to bolster the cross-task robustness of SSL encoders.

2. Background and Related Work

Self-Supervised Learning. SSL aims to extract a representation of data which is useful for downstream tasks specified at test-time (Balestriero et al., 2023). In many frameworks, an input x is first modified by two semantic-preserving augmentations yielding x_1 and x_2, which are subsequently passed to an encoder f. The training objective aligns the output representations by minimizing a distance metric d (*e.g.,* Euclidean distance) as $L(f, x) = d(f(x_1), f(x_2))$ (Chen et al., 2020). Once trained, the encoder's representations are then used for various downstream tasks, such as *classification*, *semantic segmentation*, or *depth estimation* by fine-tuning adaptor networks. In this work, we focus on a state-of-the-art SSL framework, **DINO** (Caron et al., 2021). DINO utilizes two encoder networks, the *teacher* f_t, and *student* f_s. The student network is optimized to minimize the cross-entropy between $f_s(x_1)$ and the soft labels $f_t(x_2)$, as a form of knowledge distillation (Hinton et al., 2015). To prevent collapse, the gradients are only passed through f_s. Parameters of f_t are updated using the moving average of the student's parameters. **DINOv2** (Oquab et al., 2024) improves over DINO in terms of scale and efficiency of training, rather than proposing a new SSL method. Oquab et al. (2024) showed substantial improvements on dense (pixel-wise) downstream tasks like semantic segmentation and depth estimation compared to DINO encoders.

Adversarial Robustness in SSL. In this work, we focus on the state-of-the-art **Decoupled Adversarial Contrastive Learning (DeACL)** framework by (Zhang et al., 2022) to obtain robust SSL encoders. For an overview on other methods for robust SSL and a thorough discussion on the advantages of DeACL, see Appendix B.1. DeACL fine-tunes existing encoders for increased robustness using knowledge transfer from a pre-trained encoder to a robust one. The objectives for the distillation are to: (1) match the distilled encoder representations to those of the pre-trained encoder (high cosine similarity), and (2) bring the distilled encoder's representations of adversarial examples (*i.e.,* examples generated with the pre-trained encoder that maximize the distance to their original samples) close to their clean counterparts. By decoupling the encoder pre-training from increasing its robustness, DeACL provides high computational efficiency in comparison to prior methods and obtains state-of-the-art robust performance.

Downstream Tasks. To evaluate the quality of representations learned by SSL methods, we consider three common downstream tasks. **(1) Linear Classification** assesses the quality of the learned representations by training a down-

stream classifier and measuring classification performance. **(2) Semantic Segmentation** is a common computer vision task that categorizes every pixel in an image into a class or object. While downstream-agnostic adversarial examples against SSL encoders can be used to fool segmentation models, Gu et al. (2022) show with *SegPGD* that tailoring the attack to the segmentation task is even more effective. *SegPGD* aims at manipulating all pixel classifications of an image by introducing a weighted loss term between correctly classified and misclassified pixels. **(3) Depth Estimation** is another prevalent computer vision task aimed at estimating distances of objects in an image relative to the camera location, where each pixel is assigned a depth value. Targeted adversarial attacks against depth estimation can lead to strong deviations between actual and predicted depth (Wong et al., 2020). At the same time, they can also be leveraged for depth estimation-specific adversarial training to improve robustness (Cheng et al., 2022).

3. Attack and Defense Methods

We propose a framework to assess the robustness of foundation models at both the embedding level and for downstream tasks, as described in Section 2. The goal of benchmarking the robustness of foundation models across diverse downstream tasks restricts our possible selection of encoder models. Specifically, the encoder must generate representations that are applicable to a variety of tasks beyond classification. In our preliminary experiments, we evaluated the performance of SimCLR (Chen et al., 2020), SimSiam (Chen and He, 2021), and DINO encoders. We observed that the representations produced by SimCLR and SimSiam were insufficient to achieve high-quality downstream segmentation or depth estimation. For that reason, we use the foundation models DINO and DINOv2 as examples, and train a linear adaptor for each downstream task. For the embedding attack, we target the model at the representation layer. For downstream attacks, we evaluate three different tasks: classification, semantic segmentation, and depth estimation. Each attack is detailed in the following sections.

3.1. Embedding-level Attack

The *EmbedAttack* operates directly on the underlying encoder's embeddings (Kim et al., 2020; Jiang et al., 2020; Fan et al., 2021; Luo et al., 2023). The objective behind the approach is to make imperceptibly small modifications to an input image such that the resulting representation from the SSL encoder is changed substantially. More concretely, for a clean input image x, we find its adversarial perturbation $x_{\text{adv}} = x + \delta$ such that $\|\delta\|_\infty < \varepsilon$, where ε is the maximum allowed input distortion measured in the ℓ_∞-norm. Given an encoder f, the objective is to find x_{adv} such that the ℓ_2 distance between the representations from

the original $f(x)$ and adversarial images $f(x_{\text{adv}})$ is maximized: $\arg\max_{x_{\text{adv}}} \|f(x) - f(x_{\text{adv}})\|_2$. For sparse downstream tasks (classification) we target the CLS token embedding, while for dense tasks (semantic segmentation and depth estimation) we target patch embeddings. We leverage the projected gradient descent (*PGD*) attack (Madry et al., 2018) with the objective defined in representation space to find adversarial examples x_{adv}. We set the maximum perturbation to $\varepsilon = 8/255$, start with x_{adv} initialized from x with uniform noise added (defined as $\mathcal{U}(-\varepsilon, \varepsilon)$), and perform 20 steps of *PGD* with step size $2/255$. To ensure that the distance from the original image x is within the ε-ball, we clip the perturbation to $[-\epsilon, \epsilon]$ at every step of *PGD*.

3.2. Downstream Attacks

Classification. For the standard classification tasks, we use the *PGD* attack (Madry et al., 2018) with settings similar to those above: $\varepsilon = 8/255$, 20 steps with step size $2/255$, and initialization from randomly perturbed images. The target is to maximize cross-entropy loss for the perturbed images.

Semantic Segmentation. To attack semantic segmentation we leverage the *SegPGD* attack (Gu et al., 2022) which calculates a weighted average of the loss over correctly and incorrectly classified pixels,

$$L(f_{\text{seg}}(x_{\text{adv}}^t), y) = \frac{1 - \lambda_t}{HW} \sum_{j \in P_T} L_j + \frac{\lambda_t}{HW} \sum_{k \in P_F} L_k. \quad (1)$$

Here L_j represents the cross-entropy loss, λ_t is a hyperparameter, H and W denote the height and width of the image, while P^T and P^F are the sets of correctly and incorrectly classified pixels respectively. *PGD* is used to find adversarial examples with this loss, and we use similar settings as mentioned previously. The weight λ_t starts from zero and increases linearly each iteration. The main insight behind the *SegPGD* attack is to fool correctly classified pixels in the first attack iterations and then treat the correct and incorrect pixel classifications roughly equally in the later iterations. As a result, the *SegPGD* attack can achieve similar attack effectiveness as *PGD* but with substantially fewer iterations.

Depth Estimation. Similarly to semantic segmentation, we compute the average loss per pixel and then apply a *PGD* attack targeting this loss, referred to as *DepthPGD*. The loss terms used for depth estimation and its attack are akin to those in Oquab et al. (2024), incorporating the multi-scale gradient matching loss (Li and Snavely, 2018) and pixelwise depth loss (Farooq Bhat et al., 2021). For more details, refer to Appendix C.

3.3. DeACL Defense

We combat the above attacks using the state-of-the-art method of obtaining robust encoders, DeACL (Zhang et al.,

2022). We select DeACL for several compelling reasons. Firstly, unlike many other methods aimed at enhancing robustness, it does not rely on any specific downstream task and instead improves robustness at the representation layer in a self supervised manner. Besides, it is a state-of-the-art method with superior robustness compared to other techniques. Lastly, the proposed adversarial fine-tuning approach is significantly more computationally efficient compared to training models from scratch using traditional adversarial training methods. These advantages make DeACL feasible and practical, particularly given the substantial computational resources required to train state-of-the-art encoder models. We start from a pretrained encoder f and create its robust version f_R using fine-tuning with the following objective:

$$L(f_R, f) = d(f_R(x), f(x)) + \gamma d(f_R(x_{\text{adv}}), f_R(x)). \quad (2)$$

Here we set d as the standard cosine similarity, and x as the input image. Equation (2) aims to preserve representation quality, and improve robustness against adversarial examples. $\gamma = 2$ is a parameter used to balance the impact of each goal on the final objective function.

4. Empirical Evaluation

4.1. Setup

We present the results for encoders trained using the DINO and DINOv2 SSL frameworks, utilizing ViT (Dosovitskiy et al., 2020) backbones. The underlying encoders are either *Standard*, *i.e.*, provided by the SSL frameworks, or *DeACL*, further fine-tuned to enhance robustness. We present the hyperparameters that we use to train the linear layers for the various of downstream tasks. These hyperparameters are uniform across encoders and datasets, and vary only between different types of tasks, *i.e.*, classification, semantic segmentation, and depth estimation. Full insights are presented in Appendix C.

Classification. We use a learning rate of 0.5, batch size 16, and train the linear classifiers for 5 epochs using the Adam (Kingma and Ba, 2015) optimizer. As a train-time augmentation we use random horizontal flips.

Semantic segmentation. We follow the setup from the DINOv2 framework, and use a learning rate of 0.0001, batch size 16, weight decay 0.001, and train for 50 epochs using the AdamW (Loshchilov and Hutter, 2018) optimizer. For training as well as evaluation on non-uniformly sized images (*e.g.,* PASCAL VOC 2012) we utilize sliding window inference, *i.e.*, we divide the image into parts of uniform size, compute logits for all of the parts, and then combine them into one final logit map. Overlap between the parts is handled by averaging the logits in the overlap regions. We use random cropping, and random horizontal and vertical

flips as training-time augmentations.

Depth estimation. Since DINOv2 has achieved state-of-the-art performance in depth estimation, we adopt their settings. For training, we use their combination of gradient matching loss and pixel-wise depth loss. For the remaining hyperparameters, we use a learning rate of 0.0001, batch size 128, weight decay 0.01, and train for 20 epochs using AdamW. All hyperparameters are listed in Appendix C.1.

4.2. Results

Classification. We follow the widely used linear evaluation protocol (Chen et al., 2020; Chen and He, 2021), where a linear classifier is trained on top of the frozen base SSL encoder, and test accuracy is used as a proxy for representation quality. We compare the classification accuracy after linear probing for the standard vision benchmarks: CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), and STL10 (Coates et al., 2011). The evaluation is presented in Table 1. Contrary to the results shown by Zhang et al. (2022), we observe no improvement in robustness against tailored *PGD* attacks (right column) for the encoder fine-tuned using DeACL, with the only exception being on the STL10 dataset. We argue that the discrepancy in our results and ones reported by Zhang et al. (2022) stems from the underlying training sets of the fine-tuned encoder. Zhang et al. (2022) utilized encoders trained on CIFAR10, then fine-tuned and evaluated them on CIFAR10 as well. In contrast, we focus on ImageNet-trained encoders, use ImageNet for fine-tuning, and evaluate them on various datasets including CIFAR10. We assume that the discrepancy between training, fine-tuning, and evaluation sets leads directly to the inefficacy of DeACL in obtaining robust encoders against stronger adversarial attacks than *EmbedAttack*, like *PGD*. This idea is supported by the improved adversarial accuracy against *PGD* attacks on STL10 with the fine-tuned encoder, as it is a subset of ImageNet. We observe an increase (above random guessing) in accuracy compared to the standard encoder (see second last and the last row of Table 1, rightmost column), from 0 to 0.23.

Semantic segmentation. Similarly to classification, a single linear layer is trained on patch embeddings, to obtain a low-resolution logit map. Next, we interpolate the logits to obtain a logit map of a resolution matching the size of x. The minimized objective is a pixel-wise cross-entropy loss. We evaluate encoders on ADE20k (Zhou et al., 2017; 2019), CityScapes (Cordts et al., 2016), and PASCAL VOC 2012 (Everingham et al., 2010), and report mean Intersection over Union (mIoU↑) scores in Table 2. *EmbedAttack* proves to be a potent downstream task-agnostic method of obtaining adversarial examples for the segmentation task, achieving mIoU of 0 for all clean encoders across all

Dataset	SSL Framework	Encoder Type	*Clean* Accuracy↑	*EmbedAttack* Accuracy↑	*PGD* Accuracy↑
CIFAR10	DINO v2 ViT-S/14	Standard	0.94	0.01	0.00
CIFAR10	DINO v2 ViT-B/14	Standard	0.98	0.04	0.00
CIFAR10	DINO v1 ViT-B/16	Standard	0.94	0.01	0.00
CIFAR10	DINO v1 ViT-B/16	DeACL	0.91	0.73	0.02
CIFAR100	DINO v2 ViT-S/14	Standard	0.82	0.00	0.00
CIFAR100	DINO v2 ViT-B/14	Standard	0.86	0.00	0.00
CIFAR100	DINO v1 ViT-B/16	Standard	0.76	0.00	0.00
CIFAR100	DINO v1 ViT-B/16	DeACL	0.72	0.55	0.03
STL10	DINO v2 ViT-S/14	Standard	0.98	0.06	0.00
STL10	DINO v2 ViT-B/14	Standard	0.99	0.20	0.00
STL10	DINO v1 ViT-B/16	Standard	0.98	0.00	0.00
STL10	DINO v1 ViT-B/16	DeACL	0.97	0.83	0.23

Table 1: **Evaluating the robustness of SSL encoders for linear classification.** We observe that *EmbedAttack* is effective against all *Standard* encoders. Robustness against *EmbedAttack* is improved with DeACL fine-tuning, however, the downstream *PGD* attack maintains its effectiveness, with the possible exception of linear classification on STL10 dataset, a subset of ImageNet.

Dataset	SSL Framework	Encoder Type	*Clean* mIoU↑	*EmbedAttack* mIoU↑	*SegPGD* mIoU↑
ADE20k	DINOv2 ViT-S/14	Standard	0.42	0.01	0.01
ADE20k	DINOv2 ViT-B/14	Standard	0.45	0.00	0.01
ADE20k	DINOv1 ViT-B/16	Standard	0.27	0.01	0.01
ADE20k	DINOv1 ViT-B/16	DeACL	0.24	0.14	0.01
CityScapes	DINOv2 ViT-S/14	Standard	0.65	0.02	0.01
CityScapes	DINOv2 ViT-B/14	Standard	0.68	0.03	0.00
CityScapes	DINOv1 ViT-B/16	Standard	0.45	0.06	0.06
CityScapes	DINOv1 ViT-B/16	DeACL	0.36	0.31	0.03
PASCAL VOC 2012	DINOv2 ViT-S/14	Standard	0.83	0.00	0.01
PASCAL VOC 2012	DINOv2 ViT-B/14	Standard	0.83	0.00	0.01
PASCAL VOC 2012	DINOv1 ViT-B/16	Standard	0.56	0.06	0.00
PASCAL VOC 2012	DINOv1 ViT-B/16	DeACL	0.51	0.30	0.02

Table 2: **Evaluating the robustness of SSL encoders for semantic segmentation.** We note that *EmbedAttack* succeeds across all datasets and Standard encoders, achieving mIoU close to 0, performing on par with *SegPGD*. Fine-tuning with DeACL limits the effectiveness of *EmbedAttack*, while *SegPGD* remains successful.

datasets. Similarly to the linear classification task, we note that fine-tuning with DeACL improves robustness against *EmbedAttack*, however, it fails to achieve significant improvements for the downstream attack *SegPGD*.

SSL Framework	Encoder Type	*Clean* RMSE↓	*EmbedAttack* RMSE↓	*DepthPGD* RMSE↓
DINO v2 ViT-S/14	Standard	0.49	1.54	2.60
DINO v2 ViT-B/14	Standard	0.46	1.29	2.74
DINO v1 ViT-B/16	Standard	0.61	1.28	1.79
DINO v1 ViT-B/16	DeACL	0.68	0.92	1.71

Table 3: **Evaluating the robustness of SSL encoders for depth estimation.** On the NYU-Depth v2 dataset, we observe that both types of attacks remain effective against the DeACL defense.

Depth estimation. For depth estimation, following Oquab et al. (2024), we extract the final layer of the frozen transformer and concatenate the CLS token with each patch token. Then we apply bilinear upsampling to the tokens to

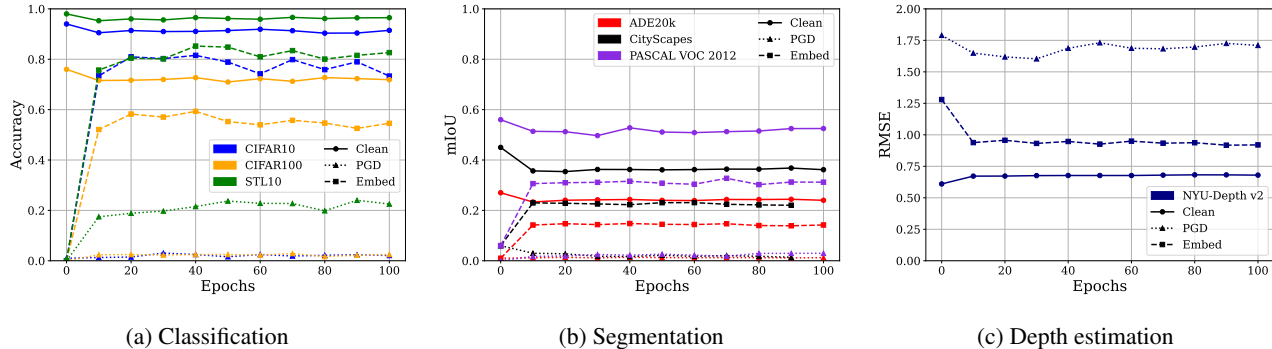| (a) Classification | (b) Segmentation | (c) Depth estimation |

Figure 2: **Evolution of robustness on different downstream tasks during DeACL fine-tuning.** Figure 2a presents classification accuracy on clean data for the *PGD* and *EmbedAttack* attacks. Figure 2b shows segmentation mIoU on clean data for the *SegPGD* and *EmbedAttack* attacks. In Figure 2c we present depth estiation RMSE on clean data for the *DepthPGD* and *EmbedAttack* attacks. Line colors indicate different datasets and line styles indicate different reported metrics.

enhance the resolution. Finally, we train a linear layer on top to estimate the depth of each pixel. We evaluate quality of the depth estimation using the standard Root Mean Square Error (RMSE) metric on the NYU-Depth-v2 dataset (Silberman et al., 2012). Our results in the Table 3 show that the *EmbedAttack* and *DepthPGD* attacks significantly increase the RMSE. The only instance where the RMSE remains below 1 after an attack is with DeACL fine-tuning against the *EmbedAttack*; however, this fine-tuning fails to provide a notable improvement in robustness against the *DepthPGD* attack, similarly to the classification and semantic segmentation tasks.

Evolution of Robustness During DeACL Fine-Tuning. Figure 2 presents the dynamics of model robustness for different downstream tasks during DeACL fine-tuning. Notably, robustness against *PGD*-based attacks exhibits minimal improvement, remaining unchanged during this process. The only exception is the improvement in robustness of linear classification on the STL10 dataset (which is a subset of ImageNet) observed during the first 20 epochs of training. We also observe that a relatively short period of fine-tuning—around 10 epochs—leads to noticeable improvements in robustness against *EmbedAttack*. However, further fine-tuning iterations show diminishing returns, with robustness metrics plateauing. Performance on clean data remains relatively stable throughout fine-tuning after a drop during the first 10 epochs. The simultaneous increase in robustness against *EmbedAttack* and decrease in performance on clean data observed at the start of the fine-tuning process confirms the trade-off between clean and adversarial model performance. The observed dynamics hold true across all downstream tasks. Our findings indicate that the adversarial fine-tuning method proposed by Zhang et al. (2022) exerts its greatest impact during the initial epochs, with little to no benefit from prolonging training to a larger (*e.g.* 100) number of epochs.

5. Discussion and Conclusions

SSL encoders are foundation models leveraged for a myriad of downstream vision tasks in critical domains, like autonomous driving (Liu et al., 2021) or medical imaging (Jiang et al., 2018). This motivates the necessity of ensuring the encoders' robustness. In this work, we argue that prior work on SSL encoder robustness mainly evaluates downstream classification tasks while leaving other popular tasks, such as semantic segmentation or depth estimation under-explored. Through our experimentation, we show that encoders are highly vulnerable to adversarial attacks on multiple downstream tasks, which pose a significant risk. Our results also highlight that the defenses that were developed with downstream classification in mind also harm the downstream performance on classification and other tasks. This suggests that more fundamental work is required to make foundational SSL encoders robust and effective for a wide variety of tasks.

Future directions for improving robustness. We observe that defenses against adversarial examples in SSL are effective only for a single attack type, namely *EmbedAttack*. However, they remain ineffective for other perturbations, especially task-specific attacks like *PGD*, *SegPGD*, and *DepthPGD*. To train SSL models that are simultaneously robust to multiple perturbation types, a potential solution is to apply multi-perturbation adversarial training, similar to the approach used for enhancing robustness in supervised models against various perturbations (Tramèr and Boneh, 2019), which involved concurrent adversarial training with first-order ℓ_1, ℓ_2, and ℓ_∞ attacks. Therefore, to enhance the robustness of SSL encoders, we should not only fine-tune them on adversarial examples in the embedding space but also potentially perform robust tuning for each intended downstream task.

References

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv:2304.12210*, 2023.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference*, pages 387–402, 2013.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, October 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607. PMLR, 13–18 Jul 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, June 2021.

Zhiyuan Cheng, James Chenhao Liang, Guanhong Tao, Dongfang Liu, and Xiangyu Zhang. Adversarial training of self-supervised monocular depth estimation against physical-world attacks. In *The Eleventh International Conference on Learning Representations*, 2022.

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*, pages 215–223, 2011.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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. In *International Conference on Learning Representations*, 2020.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010.

Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? In *Advances in Neural Information Processing Systems*, volume 34, pages 21480–21492, 2021.

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014.

Jindong Gu, Hengshuang Zhao, Volker Tresp, and Philip HS Torr. Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In *European Conference on Computer Vision*, pages 308–325. Springer, 2022.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, June 2022.

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

Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. In *Advances in Neural Information Processing Systems*, volume 33, pages 17081–17093. Curran Associates, Inc., 2020.

Feng Jiang, Aleksei Grigorev, Seungmin Rho, Zhihong Tian, YunSheng Fu, Worku Jifara, Khan Adil, and Shaohui Liu. Medical image semantic segmentation based on deep learning. *Neural Computing and Applications*, 29:1257–1265, 2018.

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

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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

Ang Li, Yifei Wang, Yiwen Guo, and Yisen Wang. Adversarial examples are not real features. In *Advances in Neural Information Processing Systems*, volume 36, pages 17222–17237. Curran Associates, Inc., 2023.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9816–9825, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

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

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

Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision – ECCV 2012*, pages 746–760, 2012. ISBN 978-3-642-33715-4.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in Neural Information Processing Systems*, 32, 2019.

Alex Wong, Safa Cicek, and Stefano Soatto. Targeted adversarial perturbations for monocular depth prediction. *Advances in Neural Information Processing Systems*, 33: 8486–8497, 2020.

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*, pages 725–742, 2022. ISBN 978-3-031-20056-4.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing Through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Understanding of Scenes Through the ADE20K Dataset. *Int. J. Comput. Vision*, 127(3):302–321, 2019. doi: 10. 1007/s11263-018-1140-0.

A. Societal Impact

Prior work on SSL encoder robustness has primarily focused on classification tasks, leading to a false sense of security among users. Our findings reveal that encoders are also susceptible to attack on other downstream tasks, underscoring the need for more comprehensive defenses. This paves the way for the development of robust solutions, thereby enhancing the trustworthiness and reliability of foundational SSL encoders for broader societal applications.

B. Extended Related Work

B.1. Adversarial Robustness in SSL

For supervised tasks, adversarial attacks produce imperceptible changes δ to an input x that result in the model predicting an incorrect label y (Biggio et al., 2013; Szegedy et al., 2013). To increase robustness, adversarial training (Goodfellow et al., 2014) incorporates the perturbed data with the correct label into the training data. Since SSL operates without labels, this approach is not directly applicable. The initial method towards robust SSL proposed by Naseer et al. (2020) introduces a *purifier network* to defend against adversarial examples, which attempts to recover the original input from an adversarially perturbed version before inputting it to the encoder. **Robust contrastive learning (RoCL)** (Kim et al., 2020) instead aims to make the encoder itself robust by maximizing the similarity between a random augmentation of a data point and its instance-wise adversarial perturbation. RoCL translates instance-level robustness to class-level robustness, at the cost of substantial degradation in clean performance.

Ho and Nvasconcelos (2020) propose adversarial examples specifically designed to challenge contrastive learning methods. Using these adversarial examples, they develop a novel adversarial training algorithm for self-supervised learning, which they call Contrastive Learning with Adversarial Examples (CLAE). Compared to standard contrastive learning, CLAE creates more difficult positive pairs by using adversarial examples. Additionally, by optimizing over all images in a batch, CLAE produces more challenging negative pairs through adversarial training. In essence, CLAE strengthens contrastive learning models by exposing them to tailored adversarial attacks during training.

Jiang et al. (2020) introduce **adversarial contrastive learning (ACL)** to improve robustness-aware self-supervised pre-training by learning representations that are consistent under *both data augmentations and adversarial perturbations*. They extend SimCLR (Chen et al., 2020) to learn robust representations by maximizing feature consistency between differently augmented views. Fan et al. (2021) build on top of ACL and propose **AdvCL**, which leverages labels in addition to instance-level robustness to further boost robust performance. Luo et al. (2023) propose **Dynamic Adversarial Contrastive Learning (DYNACL)** as an extension that uses pseudo-labels directly generated by the pre-trained encoder. All these methods require retraining the large SSL encoders from scratch to improve robustness which is highly impractical and computationally expensive. To solve the problem, (Zhang et al., 2022) propose a two-stage framework called **Decoupled Adversarial Contrastive Learning (DeACL)** which fine-tunes existing encoders for increased robustness. Therefore, the knowledge of a pre-trained encoder is distilled to a robust one. The objective for the distillation are to: (1) match the distilled encoder representations to those of the pre-trained encoder, and (2) bring the distilled encoder's representations of adversarial examples close to their clean counterparts. *Closeness* is defined by cosine similarity, and adversarial examples are just those examples generated on the original trained encoder to maximize the distance to the original samples. A compelling aspect is that the decoupling approach of DeACL is not limited to contrastive learning - the original encoder could potentially leverage other self-supervised learning (SSL) methods. Only the distillation loss may need adaptation for SSL frameworks like MAE (He et al., 2022), where cosine similarity may not be optimal. Through this approach, DeACL sets a new state-of-the-art by effectively and efficiently improving encoder robustness. This is achieved by decoupling the SSL pre-training stage from the adversarial fine-tuning stage. The flexibility of DeACL leaves room for exploring different SSL methods in the first pre-training stage. Given the many advantages of DeACL demonstrated thus far, we focus our evaluation on this approach.

C. Hyperparameters

C.1. Further Insights on Depth Estimation

The multi-scale gradient matching loss (Li and Snavely, 2018) encourages smoother transitions in depth predictions and penalizes differences in log-depth gradients across multiple scales:

$$L_{\text{grad}} = \frac{1}{n} \sum_k \sum_i |\nabla_x R_i^k| + |\nabla_y R_i^k|. \tag{3}$$

The loss is computed at multiple scales where R_i^k represents the value of the log-depth difference at position i and scale k. ∇_x and ∇_y denote the gradients in the x and y directions, respectively.

The pixel-wise depth loss (Farooq Bhat et al., 2021) measures the difference between the predicted and ground truth depth values in a scale-invariant manner:

$$L_{\text{pixel}} = \alpha \sqrt{\frac{1}{T} \sum_i g_i^2 - \frac{\rho}{T^2} \left(\sum_i g_i \right)}. \tag{4}$$

Where $g_i = \log \tilde{d}_i - \log d_i$, with $\tilde{d}_i$ representing the predicted depth and d_i the ground truth depth. The parameters α and ρ are set to 1 and 0.85 in our experiments.

The final loss we use is $\frac{1}{2} L_{\text{grad}} + L_{\text{pixel}}$.

C.2. DeACL fine-tuning

In this section, we describe the hyperparameters we adopt to perform the adversarial fine-tuning proposed by (Zhang et al., 2022) on DINOv1 with ViT B/16 backbone. We use a learning rate of 0.05 with a *cosine* scheduler and 10 epochs of warmup. We fine-tuned the model for 100 epochs with a SGD optimizer (momentum 0.9) and batch size of 128. The adversarial perturbation budget ε was set to 4/255. We did not use weight decay. We employed random crops, and random horizontal and vertical flips as training-time augmentations.