Robust Fine-Tuning from Non-Robust Pretrained Models: Mitigating Suboptimal Transfer With Epsilon-Scheduling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Fine-tuning pretrained models is the standard approach in current machine learning practice, but simultaneously achieving adversarial robustness to adversarial examples remains a challenge. Despite the abundance of non-robust pretrained models in open-source repositories, their use for Robust Fine-Tuning (RFT) remains understudied. This work aims to bridge this knowledge gap by systematically examining RFT from such models. Our experiments reveal that fine-tuning non-robust models with a robust objective, even under small perturbations, can lead to poor performance, a phenomenon that we dub *suboptimal transfer*. In fact, we find that fine-tuning using a robust objective impedes task alignment at the beginning of training and eventually prevents optimal transfer. To promote optimal transfer, we propose *Epsilon-Scheduling*, a simple heuristic scheduling over perturbation strength. Additionally, we introduce *expected robustness*, a metric that measures performance across a range of perturbations. Experiments on six pretrained models and five datasets show that *Epsilon-Scheduling* prevents *suboptimal transfer* and consistently improves the expected robustness.

1 Introduction

Fine-tuning pretrained models is the standard workflow in machine learning, spanning NLP (Koroteev, 2021) and vision (Goldblum et al., 2023). This workflow offers clear benefits: (i) reusing a single foundation model across tasks (Devlin et al., 2019), (ii) faster convergence and better generalization than training from scratch (Yosinski et al., 2014), and (iii) reduced computation (Weiss et al., 2016), especially when labelled data is scarce (Pan & Yang, 2010).

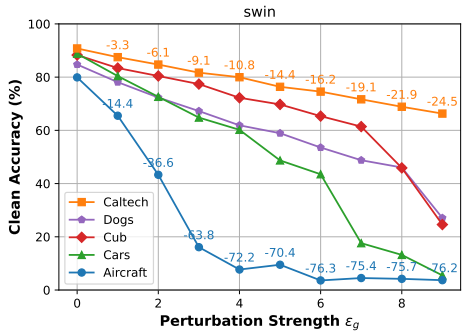

Figure 1: Robust Fine-Tuning can lead to *suboptimal transfer* even when optimizing for small perturbations.

However, in high-stakes applications, adversarial vulnerability remains a major concern (Biggio et al., 2013; Goodfellow et al., 2015). Adversarial Training (AT) (Madry et al., 2018) and its variants (Zhang et al., 2019; Wang et al., 2020; Ding et al., 2020; Shafahi et al., 2019a; Wong et al., 2020) are the most successful empirical defenses (Croce et al., 2020). Robust Fine-Tuning (RFT) is the integration of these methods in fine-tuning on downstream tasks (Shafahi et al., 2019b; Liu et al., 2023; Xu et al., 2024; Hua et al., 2024). RFT is challenging because it must balance alignment with the downstream task and robustness (Xu et al., 2024). Prior work mainly studies RFT from robust pretrained models (Hua et al., 2024; Liu et al.,

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

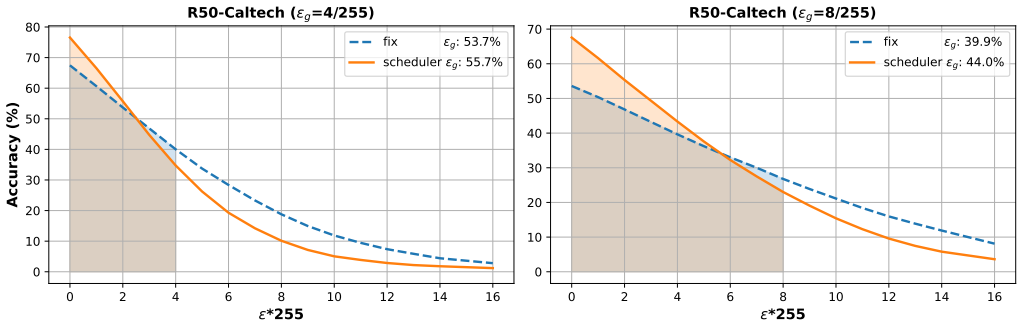

Figure 2: **Expected robustness**. The first value in the legend represents the evaluation over $[0, \epsilon_g]$ (shaded region) and the second value is over the whole interval $[0, 16]$.

2023; Xu et al., 2024), overlooking the more common non-robust ones (Wolf et al., 2020). Since robust models are costly and since pretraining typically targets general-purpose features, robustness can be considered as a property to be acquired on downstream tasks (Heuillet et al., 2025). Thus, improving RFT from non-robust backbones is essential and naturally aligns with current workflows.

In this work, we study Robust Fine-Tuning (RFT) of non-robustly pretrained backbones. We fine-tune various pretrained backbones on different datasets using adversarial training (Madry et al., 2018) with a fixed perturbation radius. We find that, even for small nonzero radii, this approach yields ***suboptimal transfer***, where performance falls short of that achieved by standard fine-tuning (without perturbation) and is often too low to be considered a successful transfer. Its severity depends on both the backbone and the downstream task. Unlike standard fine-tuning, where model adaptation to the downstream task occurs immediately, our study shows that in RFT, **task alignment is delayed until later epochs**, eventually leading to *suboptimal transfer*.

To mitigate this, we propose ***Epsilon-Scheduling*** (Figure 3): a simple scheduling that starts with standard fine-tuning (zero perturbation) for early epochs and linearly increases to the target perturbation at final epochs. This strategy prevents *suboptimal transfer* and improves both generalization and robustness.

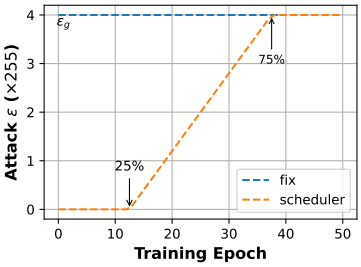

Figure 3: *Epsilon-Scheduling*

Finally, to better evaluate the fine-tuned models, instead of the standard evaluation that compares only clean and robust accuracy at target perturbation strength ϵ_g, we introduce ***expected robustness***, which evaluates the expectation of the accuracy of the model across the full perturbation range $[0, \epsilon_g]$. In particular, cases where an increase in generalization comes at the cost of reduced robustness can make model comparison subjective. The *expected robustness* provides a comprehensive evaluation of the accuracy-robustness trade-off. Under this metric, *Epsilon-Scheduling* consistently improves performance, even when worst-case robustness at ϵ_g is similar or lower.

2 Methodology

Robust Fine-Tuning Fine-tuning consists in training a classifier $f = c_{\theta_2} \circ h_{\theta_1}$, composed of a pretrained backbone h_{θ_1} and a randomly initialized classifier head c_{θ_2}, to maximize accuracy on a given data distribution $\mathcal{D}$. This work focuses on full fine-tuning where both θ_1 and θ_2 are trainable parameters. In Robust Fine-Tuning (RFT), the goal is to maximize robust accuracy $\mathrm{Acc}_{\epsilon_g}(f)$ at a target perturbation strength $\epsilon_g > 0$. We consider RFT with adversarial training (Madry et al., 2018) that minimizes the adversarial risk at ϵ as a surrogate objective:

$$R_\epsilon(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big(\max_{\|\delta\|_\infty<\epsilon} \ell_{\mathrm{CE}}(f(x+\delta), y)\Big) \tag{1}$$

where ℓ_{CE} the cross-entropy loss. The common practice in RFT for target perturbation strength ϵ_g consists of minimizing an empirical counterpart of $R_{\epsilon_g}(f)$ for a certain number of epochs, a strategy

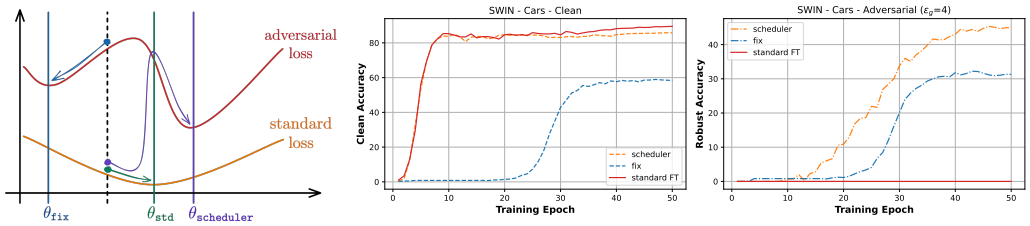

Figure 4: ***Epsilon-Scheduling* preserves task alignment while improving robustness.** Left: Illustrative example of the difference between RFT-`fix` and RFT-`scheduler`. Center and right: Evolution of clean and robust accuracy during the fine-tuning of the SWIN backbone on Cars dataset with $\epsilon_g = 4/255$.

that we refer to as RFT-`fix` (or `fix`), since the training objective remains the same during the whole fine-tuning process.

Epsilon-Scheduling In contrast with RFT-`fix`, we propose to achieve RFT for target perturbation strength ϵ_g by minimizing an empirical counterpart of $R_\epsilon(f)$ where the radius ϵ follows a simple schedule during the fine-tuning, as illustrated in (Figure 3). In effect, this strategy starts with standard fine-tuning at $\epsilon = 0$ for 25% of the number of epochs, then linearly increases from $\epsilon = 0$ to $\epsilon = \epsilon_g$ during half of the fine-tuning, until it finally minimizes $R_{\epsilon_g}(f)$ for the remaining 25%. From a transfer learning perspective, we can view this strategy as follows: begin with task adaptation, then gradually shift to the robust objective and conclude by minimizing the robust objective. In the sequel, we will refer to this strategy as RFT-`scheduler` (or `scheduler`).

Expected Robustness While RFT targets low adversarial risk $R_{\epsilon_g}(f)$, models are usually evaluated both for clean accuracy $\mathrm{Acc}_0(f)$ and robust accuracy $\mathrm{Acc}_{\epsilon_g}(f)$. We propose to extend this classical evaluation to take into account intermediary perturbation strengths within the range $[0, \epsilon_g]$. Evaluating models' accuracy at intermediate perturbation strengths reveals distinct patterns (See Figure 2). Such evaluation is helpful for comparing models with similar accuracies or when the clean–robust trade-off is ambiguous. We summarize these evaluations using the *expected robustness* metric, defined as the expectation under uniform distribution U of the accuracy over $[0, \epsilon_g]$:

$$\mathbb{E}_{\epsilon \sim U[0,\epsilon_g]}\big[\mathrm{Acc}_\epsilon(f)\big] = \frac{1}{\epsilon_g}\int_0^{\epsilon_g} \mathrm{Acc}_\epsilon(f)\, d\epsilon = \frac{1}{\epsilon_g}\mathrm{AUC}_{\epsilon_g}(f)$$

where $\mathrm{AUC}_{\epsilon_g}(f)$ represents the area under the accuracy curve from 0 to ϵ_g (See Figure 2). More details can be found in Appendix B.

3 Characterizing Suboptimal Transfer in Robust Fine-Tuning

We explore how high perturbation strength ϵ_g in RFT-`fix` affects the transfer accuracy of non-robust pre-trained models. For our experiment, we use two ImageNet-pretrained backbones, `SWIN` and `ViT`, and fine-tune them on five datasets: Caltech256, Cub200, Stanford Dogs, Stanford Cars, and FGVC-Aircraft. We consider perturbation strengths ϵ_g from 0 (standard fine-tuning) up to $9/255$. The results in Figure 1 show that as ϵ_g increases, the transfer accuracy drops significantly. For example, at $\epsilon_g = 4/255$, the `SWIN` models have performance drops of 10% to 72%, respectively, compared to standard fine-tuning. We refer to this phenomenon as ***suboptimal transfer***, where RFT-`fix` yields a transfer accuracy significantly lower than standard fine-tuning, at times to the point of no longer being considered an effective transfer. Results for `ViT` are in appendix (Figure 7)

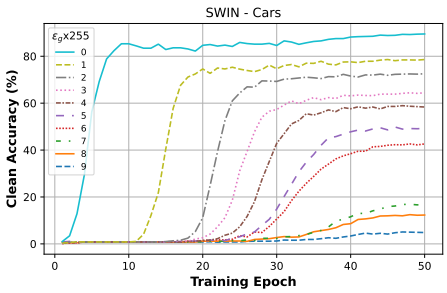

Figure 5: **RFT-`fix` delays task alignment**. The stronger the perturbation, the later the validation accuracy starts to improve.

Robust Fine-Tuning with Fixed Perturbation Strength Delays Task Alignment As shown in Figure 5 in standard fine-tuning, the task adaptation to the downstream task begins almost immediately–validation accuracy rises from the first epoch–since there are no robustness constraints that may conflict with task alignment. With nonzero values of ϵ_g, RFT-`fix` distorts task-relevant features, which prevents early alignment and delays the onset of task adaptation. For example, task alignment

	Dataset	Aircraft			Caltech			Cars			Cub			Dogs		
Model	Metric / Setting	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**
vit	fix	6.40	2.80	4.48	68.14	41.64	55.07	12.70	4.90	8.20	42.82	15.12	27.79	56.40	**19.97**	36.93
	scheduler	**58.60**	**13.20**	**34.95**	**78.73**	**41.69**	**60.71**	**73.40**	**19.10**	**46.71**	**73.40**	**23.63**	**48.09**	**70.69**	15.69	**41.62**
swin	fix	7.70	4.80	6.11	79.97	**57.16**	69.19	60.20	29.70	44.74	72.25	**41.87**	57.55	61.89	**26.89**	44.17
	scheduler	**73.80**	**32.00**	**53.75**	**85.43**	56.39	**72.04**	**84.70**	**43.20**	**66.41**	**82.29**	41.61	**63.82**	**72.70**	24.32	**48.50**
convnext	fix	7.60	4.50	5.86	83.27	**61.54**	73.08	69.60	43.20	57.52	76.34	**47.08**	62.59	68.90	**31.61**	50.61
	scheduler	**78.40**	**38.00**	**59.40**	**89.41**	61.45	**77.23**	**88.90**	**57.70**	**75.85**	**85.17**	44.99	**67.30**	**78.39**	26.31	**53.19**
r50	fix	8.40	2.90	4.56	67.47	**40.02**	53.74	4.20	2.90	3.49	49.19	19.35	33.58	57.05	**19.80**	37.73
	scheduler	**53.10**	**11.10**	**29.40**	**76.55**	34.74	**55.67**	**70.00**	**19.30**	**43.44**	**70.06**	**19.59**	**43.62**	**69.11**	15.94	**41.11**
clip_vit	fix	5.00	3.30	4.16	31.91	15.49	23.00	4.90	3.00	3.74	13.95	3.64	7.97	7.89	3.29	5.39
	scheduler	**69.80**	**33.90**	**52.79**	**74.83**	**46.64**	**60.99**	**86.70**	**58.60**	**75.01**	**74.35**	**35.67**	**55.54**	**63.17**	**20.87**	**41.05**
clip_convnext	fix	3.10	2.50	2.82	61.76	42.13	51.54	2.80	1.60	2.23	28.89	14.33	20.92	23.90	11.33	17.14
	scheduler	**81.70**	**50.70**	**67.88**	**81.19**	**52.68**	**67.71**	**90.90**	**74.10**	**84.33**	**79.06**	**42.11**	**61.45**	**70.85**	**25.85**	**48.19**

Table 1: ***Epsilon-Scheduling* mitigates *suboptimal transfers* and consistently improves expected robustness.** Results at moderate perturbation regime ($4/255$). See Table 2 for $\epsilon_g = 8/255$

begins around epoch 25 for $\epsilon_g = 4/255$. To the best of our knowledge, the delayed onset of task alignment in robust fine-tuning has not been previously reported.

4 Experimental Results

Pretrained Models and Datasets: We experiment with six pretrained models—Transformers (*Swin-Base*, *ViT-Base*), Convolutional networks (*ConvNext-Base*, *ResNet50*), and CLIP models (*CLIP-ViT*, *CLIP-ConvNext*)—spanning attention, convolution, supervised, and multi-modal paradigms. Fine-tuning is evaluated on five low-data benchmarks: **CUB-200-2011** (birds), **Stanford Dogs**, **Caltech256**, **Stanford Cars**, and **FGVC-Aircraft**.

Epsilon-Scheduling* mitigates *suboptimal transfer The results in Table 1 show that while RFT-`fix` often fails to transfer with low clean accuracy, RFT-`scheduler` achieves high clean accuracy for most models. At the same time, it maintains decent adversarial accuracy. For the perturbation target $\epsilon_g = 4/255$, while RFT-`fix` sometimes achieves better adversarial accuracy (9 out of 30 configurations), our scheduling strategy always obtains a higher clean and expected accuracy (see also Figure 6 for results aggregated across models). These results show that even at moderate perturbations ($4/255$), *epsilon-scheduling* prevents the steep degradation incurred by RFT-`fix`, allowing models to retain strong clean performance while achieving improved or similar adversarial accuracy at non-trivial levels. In high perturbation regime ($\epsilon_g = 8/255$), transfer fails more often with RFT-`fix` and RFT-`scheduler` becomes the only viable option for robust fine-tuning.

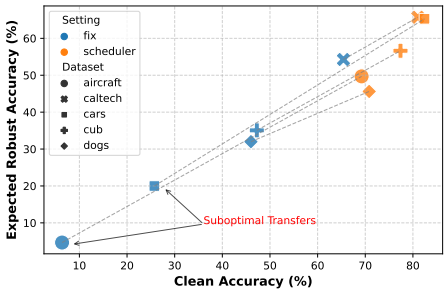

Figure 6: ***Epsilon-Scheduling* mitigates *suboptimal transfers* and improves *expected robustness***. Aggregated results across models from Table 1 ($\epsilon_g = 4/255$).

***Epsilon-Scheduling* preserves task alignment while improving robustness** Figure 4 shows the evolution of the validation accuracy during training for $\epsilon_g = 4/255$. As expected, the standard fine-tuning converges very fast, successfully learning the task with a high clean accuracy. RFT-`fix` negatively affects the clean accuracy and ultimately fails to learn the task effectively. In RFT-`scheduler`, delaying fine-tuning with perturbations helps achieve a high clean accuracy at the level of standard fine-tuning at the early stage. Once RFT starts, around epoch 12, with perturbation strengths above zero, robust accuracy begins to increase. Interestingly, the clean accuracy remains high and relatively stable.

Limitations & Future Work. While our study sheds light on the phenomenon of *suboptimal transfer* in RFT and proposes a mitigation via *epsilon-scheduling*, it also opens up several interesting research directions. We leave the study of different schedulers, the mechanistic understanding of *suboptimal transfer*, applications beyond image classification, parameter-efficient fine-tuning, and extensions to other modalities (e.g., language) for future work.

References

Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

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 *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Qi-Zhi Cai, Chang Liu, and Dawn Song. Curriculum adversarial training. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3740–3747. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/520. URL https://doi.org/10.24963/ijcai.2018/520.

Luiz Chamon and Alejandro Ribeiro. Probably approximately correct constrained learning. *Advances in Neural Information Processing Systems*, 33:16722–16735, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 225–253. IEEE, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkeryxBtPB.

Junhao Dong, Piotr Koniusz, Junxi Chen, Z Jane Wang, and Yew-Soon Ong. Robust distillation via untargeted and targeted intermediate adversarial samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28432–28442, 2024.

Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3996–4003, 2020.

Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, 36:29343–29371, 2023.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Maxime Heuillet, Rishika Bhagwatkar, Jonas Ngnawé, Yann Pequignot, Alexandre Larouche, Christian Gagné, Irina Rish, Ola Ahmad, and Audrey Durand. A guide to robust generalization: The impact of architecture, pre-training, and optimization strategy, 2025. URL https://arxiv.org/abs/2508.14079.

Andong Hua, Jindong Gu, Zhiyu Xue, Nicholas Carlini, Eric Wong, and Yao Qin. Initialization matters for adversarial transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24831–24840, 2024.

Mikhail V Koroteev. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.

Ziquan Liu, Yi Xu, Xiangyang Ji, and Antoni B Chan. Twins: A fine-tuning framework for improved transferability of adversarial robustness and generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16436–16446, 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. URL https://openreview.net/forum?id=rJzIBfZAb.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Xb8xvrtB8Ce.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019a.

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019b.

Naman Deep Singh, Francesco Croce, and Matthias Hein. Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models. *Advances in Neural Information Processing Systems*, 36:13931–13955, 2023.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rklOg6EFwS.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):1–40, 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. Autolora: an automated robust fine-tuning framework. In *The Twelfth International Conference on Learning Representations*, 2024.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

A Related Work

Adversarial Robustness in Transfer Learning with Robust-FineTuning There are two main ways to achieve adversarial robustness in Transfer Learning: Robust Distillation (Goldblum et al., 2020; Dong et al., 2024) and Robust Fine-Tuning. Previous work on RFT has focused on strategies to preserve the robustness of pretrained models (Liu et al., 2023; Xu et al., 2024; Hua et al., 2024). (Liu et al., 2023) proposed TWINS (TwoWIng NormliSation), a statistics-based fine-tuning framework that employs two neural networks with shared parameters: one maintains the population means and variances of the pretraining data in the batch normalization layers, while the other tracks the statistics of the downstream dataset. AutoLoRA (Xu et al., 2024) shows that there is often a divergence between natural and adversarial gradient directions in RFT and addresses it by disentangling the optimization objectives—using a low-rank LoRA branch for natural objectives and a robust, pretrained feature extractor for adversarial objectives. Hua et al. (2024) showed that linear probing best preserves the robustness of the adversarially pretrained model and proposed RoLi. This strategy initializes the linear classifier head via adversarially trained linear probing before performing RFT. These strategies only consider robust pretrained feature extractors. To the best of our knowledge, this work is the first to propose a method for RFT that directly targets non-robust, pretrained models without assuming robust pretrained features.

Tuning Perturbation Strength in Adversarial Training The idea of tuning or adapting the adversarial perturbation strength ϵ during training has appeared in various forms across the robustness literature. Early work like Gowal et al. (2018) used a linear ramp-up of ϵ in the Interval Bound Propagation (IBP) method. Ding et al. (2020) drew a theoretical connection between margin maximization and the loss at the smallest adversarial perturbation, motivating the use of adaptive, sample-specific ϵ values. Similarly, Balaji et al. (2019) explored instance-wise epsilon selection, though these approaches can be computationally intensive due to per-sample perturbation searches. Ding et al. (2020) additionally introduced PGDLS (PGD with Linear Scaling), which linearly ramps up the perturbation radius during adversarial training and shows little to no improvement at $\epsilon \leq {}^{16}/_{255}$ but only at high $\epsilon = {}^{24}/_{255}$. To better trade off clean and robust accuracy, Chamon & Ribeiro (2020) proposed sampling ϵ from a Beta distribution. Cai et al. (2018) proposed a curriculum adversarial training scheme that gradually increases the attack steps, which improves performance in combination with batch mixing and quantization. Unlike Pang et al. (2021), which showed that linear ϵ warmup had a limited effect in ResNets, Debenedetti et al. (2023) showed that it improved both clean and robust accuracy in vision transformers. In contrast to prior works, which have primarily applied perturbation tuning in classical adversarial training from scratch, our study frames *Epsilon-Scheduling* through the lens of transfer learning. In this context, *Epsilon-Scheduling* is not just an optional improvement over standard RFT with a fixed epsilon; rather, it constitutes a dependable alternative when standard RFT fails to transfer, which we show happens when training directly at large ϵ. In addition to previous work, we evaluate performance using a new metric, the *expected robustness*, and show that it is consistently beneficial, regardless of task and architecture, including ResNets.

B Additonal Details

Training Details We follow a similar setup described in Hua et al. (2024), using the AdamW optimizer with a cosine learning rate scheduler that includes a warmup period. We select the learning rate and weight decay via hyperparameter optimization (HPO) based on clean accuracy. HPO is performed only for the `fix` setting, and the resulting hyperparameters are reused for the `scheduler` setting to ensure a fair comparison. Adversarial training is performed by minimizing an empirical counterpart of the adversarial risk (Equation 1). More specifically, on a mini-batch B we minimize

$$L_\epsilon(f) = \frac{1}{|B|} \sum_{(x,y)\sim B} \ell_{\mathrm{CE}}(f(\tilde{x}), y)$$

where $\tilde{x}$ is an adversarial example crafted for x using APGD (instead of PGD) with cross-entropy loss as in (Singh et al., 2023; Heuillet et al., 2025), benefiting from APGD's adaptive step size, which removes the need for manual tuning across different perturbation thresholds. The number of APGD steps is 7 for training. As in Heuillet et al. (2025), we train for 50 epochs, and results are reported at the end of training because overfitting of the adversarial accuracy is negligible here (see Figure 4).

Model	Dataset	Aircraft			Caltech			Cars			Cub			Dogs		
	Metric	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**	Clean	Adv.	**E. Adv.**
	Setting															
vit	fix	3.00	2.00	2.50	44.95	19.52	31.43	3.60	2.00	2.74	17.40	2.80	8.56	8.64	2.88	5.35
	scheduler	**57.00**	**6.70**	**27.72**	**72.86**	**26.89**	**49.28**	**68.10**	**9.00**	**35.18**	**64.74**	**9.79**	**33.93**	**56.86**	**5.79**	**25.81**
swin	fix	4.20	2.70	3.47	68.87	38.10	53.40	13.20	5.60	8.66	45.89	13.60	28.56	46.05	**11.08**	26.69
	scheduler	**69.20**	**22.40**	**45.12**	**80.27**	**38.67**	**60.26**	**78.00**	**23.50**	**53.57**	**74.80**	**21.07**	**47.34**	**60.49**	8.73	**31.14**
convnext	fix	1.60	1.50	1.48	59.85	33.95	46.34	5.30	2.60	3.98	5.02	2.28	3.56	27.33	7.73	16.28
	scheduler	**75.00**	**28.80**	**50.90**	**84.99**	**41.82**	**64.92**	**85.60**	**35.90**	**65.04**	**80.69**	**24.28**	**53.07**	**68.94**	**9.78**	**36.51**
r50	fix	1.30	0.90	0.74	53.59	**26.78**	39.93	1.50	1.20	1.34	30.89	8.27	17.84	27.14	**6.95**	15.61
	scheduler	**42.80**	**5.30**	**20.38**	**67.56**	23.01	**44.03**	**57.10**	**8.50**	**29.56**	**59.49**	**8.68**	**29.95**	**50.89**	6.92	**25.26**
clip_vit	fix	3.60	2.20	3.05	23.02	7.29	14.52	3.00	2.50	2.73	11.11	2.30	5.73	2.20	1.38	1.77
	scheduler	**65.80**	**25.40**	**44.84**	**70.68**	**33.70**	**51.67**	**84.70**	**38.60**	**64.47**	**67.64**	**18.05**	**41.79**	**54.28**	**8.94**	**27.78**
clip_convnext	fix	1.80	1.30	1.62	51.94	28.37	39.44	1.30	1.10	1.25	6.37	2.30	4.05	8.36	3.97	5.98
	scheduler	**79.20**	**34.50**	**59.09**	**76.53**	**37.20**	**56.83**	**90.00**	**55.20**	**77.14**	**73.58**	**22.75**	**47.77**	**62.67**	**11.36**	**33.85**

Table 2: ***Epsilon-Scheduling* mitigates *suboptimal transfers* and consistently improves expected robustness in high perturbation regime** ($^8/_{255}$). The table shows clean accuracy (**Clean**), adversarial accuracy (**Adv.**), and the *expected adversarial accuracy* (**E. Adv.**). The models are evaluated under a fixed perturbation strength (`fix`) and an *Epsilon-Scheduling* (`scheduler`). See Table 1 for $\epsilon_g = {}^4/_{255}$

We consider two target evaluation thresholds $\epsilon = {}^4/_{255}$ (moderate perturbation) and $\epsilon = {}^8/_{255}$ (high perturbation) as two commonly used evaluation targets on these datasets.

Evaluation details For a given perturbation strength $\epsilon > 0$, the (L_∞-)robust accuracy $\mathrm{Acc}_\epsilon(f)$ of a classifier f is defined as

$$\mathrm{Acc}_\epsilon(f) = \mathbb{E}_{(x,y)\sim D}\mathbf{1}[\forall x'(\|x - x'\|_\infty \leq \epsilon \Rightarrow \arg\max f(x') = y)],$$

where $\mathbf{1}[\phi]$ equals 1 if ϕ holds and 0 otherwise. In particular, for $\epsilon = 0$, $\mathrm{Acc}_0(f) = \mathbb{E}_{(x,y)\sim D}\mathbf{1}[\arg\max f(x) = y]$ coincides with the usual clean accuracy of the classifier f. This robust accuracy is estimated using the AutoAttack library with the APGD method and 10 steps on a given test dataset. The expected robustness is estimated by using the trapezoidal rule with evaluations made with steps $^1/_{255}$, so for example with $\epsilon_g = {}^4/_{255}$:

$$\mathrm{AUC}_{4/255}(f) = \frac{1}{4}\sum_{i=0}^{3}\frac{\mathrm{Acc}_{\frac{i}{255}}(f) + \mathrm{Acc}_{\frac{i+1}{255}}(f)}{2}.$$

C Additional Results

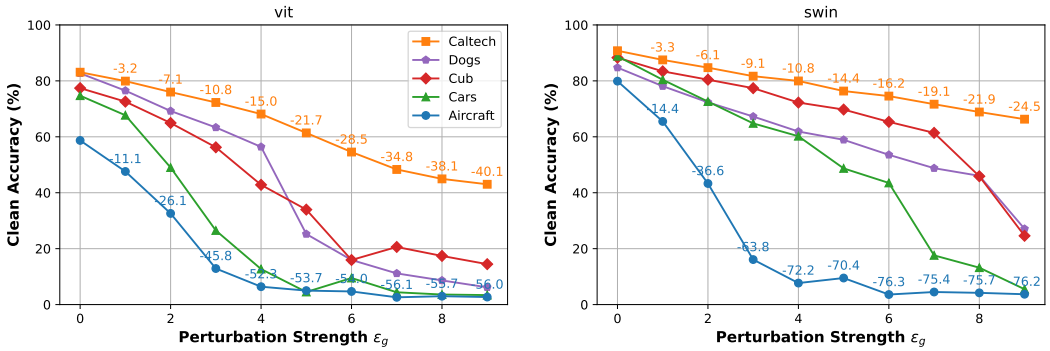

Figure 7: **RFT can lead to *suboptimal transfer* even for small ϵ.** The variation of transfer accuracy with the training perturbation strength ϵ_g is not always smooth and is highly model- and dataset-dependent.

1. **Claims**

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

   Answer: [Yes]

   Justification: The topic and claims in the abstract are accurately reflected in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See the Limitations & Future work paragraph in the Conclusion section.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper explains the experimental setup sufficient to reproduce the results.

   Guidelines:

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

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

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

   Answer: [No]

   Justification: Only data is open-source for now. Code will be released later.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Training details are provided in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The results do have error bars which is not unusual in adversarial robustness community due to compute requirements for adversarial training. Results presented are reproducible and cover various experimental conditions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper mitigates vulnerability in deep neural networks and respects all points in the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The societal impacts are discussed in the introduction.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper is based on already open-sourced and widely used tools.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have bibliographical references for all the datasets, and backbones, and open source software used for this study.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: We provide all the code base to reproduce the study and the dataset of the collected results. These support the study but are not new assets.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: The paper is based on open sourced datasets and backbones.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: No research with human subjects or crowdsourcing.

    Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM use does not impact core methodology.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.