Improving Adversarial Transferability via Model Alignment

Avery Ma[1,2], Amir-massoud Farahmand[1,2], Yangchen Pan[3], Philip Torr[3], and Jindong Gu[3]

[1] University of Toronto, Toronto, Canada
ama@cs.toronto.edu
[2] Vector Institute, Toronto, Canada
[3] University of Oxford, Oxford, UK

A Summary of the Supplementary Material

The supplementary material is organized as follows. In Appendix B, we first describe the implementation details, including the exact optimization schedule, model architectures, data augmentations, attack algorithms, and methods for alignment in the embedding space. In Appendix C, we discuss additional empirical studies on understanding the model alignment process. Finally, additional experiment results and discussions are included in Appendix D.

B Implementation Details

For all models considered in our paper, we adhere to the optimization configurations as detailed in the official PyTorch repository[4].

CNN-based models: All CNN-based models are trained using the same configurations. Those models include ResNet18 (Res18), ResNet50 (Res50), ResNet101 (Res101), VGG19, DenseNet121 (DN121), and Inception-v3 (IncV3). They are trained for 90 epochs using SGD with an initial learning rate of 0.1 and a momentum coefficient of 0.9. We start training with a 5-epoch learning rate warmup, followed by a cosine decay schedule. The batch size is set at 256.

ViT-based models: The ViT-based models include ViT-T/16, ViT-S/16, ViT-B/16, and Swin Transformers (SWIN). They are all trained for 300 epochs using AdamW. The initial learning rates are set at 0.003 for ViT models and 0.001 for SWIN. For the ViT's, training begins with a 30-epoch learning rate warmup, followed by a cosine decay schedule, whereas for SWIN, the warmup is 20 epochs. The batch size is set at 1024. We use label smoothing during training. To ensure training stability, the global gradient norm is clipped at 1.

Data augmentations: For the training of all CNN-based models, augmentation techniques are random resizing, cropping, and flipping. For the training of all ViT-based models, we further incorporate Mixup [20] and Cutmix [19]. Only random resizing, cropping and flipping are used during the alignment process.

[4] https://github.com/pytorch/vision/tree/main/references/classification

Table 1: Comparing the similarity between the source and witness model. We focus on aligning a Res50 with a Res18 and a ViT-B/16 with a ViT-T/16. We first evaluate the similarity between each original source model and its corresponding witness model. This is then followed by an evaluation between the aligned model and the witness model.

Source/Witness	KL		Prediction Agreement		Input Gradient Cosine Similarity	
	Before	After	Before	After	Before	After
Res50 / Res18	0.63	0.31	80.7%	82.4%	0.043	0.096
ViT-B / ViT-T	0.75	0.24	78.9%	85.0%	0.026	0.063

Specifically, RandAugment [4] is applied to ViT's, while TrivialAugment [11] and random erasing [22] are applied to SWIN.

Model definitions: All model definitions are obtained from the torchvision library [10], with the exceptions of IncV3, ViT's, and SWIN, which are obtained from the timm library [16].

Attack algorithms: All the attack algorithms used in our paper are provided by the Torchattacks library [6].

Alignment in the embedding space: When using a witness model with a different architecture than the source model, the dimensions of their hidden representations are likely to be different, i.e., $\dim(z_s^{[q_s]}(x)) \neq \dim(z_w^{[q_w]}(x))$, where q_s and q_w represent the layers just before the fully-connected layer in their respective models. To address this dimensional mismatch, we follow previous work [2, 9, 13, 14] and apply a linear projection to z_s so its dimension matches that of z_w. The weights of the linear projection are treated as trainable parameters during the alignment process.

C Understanding Model Alignment

C.1 Evaluating Similarity Between the Source and Witness Model

To study the similarity between the source and witness models before and after alignment, we examine KL divergence, prediction agreement, and cosine similarity of input gradients. The latter is especially important for its role in the attack processes. Our investigation involves aligning a Res50 with a Res18 and a ViT-B/16 with a ViT-T/16. We first evaluate the similarity between each original source model and its corresponding witness model. This is then followed by an evaluation between the aligned model and the witness model. Tab. 1 presents the results, indicating reductions in KL divergence, increased cosine similarity, and improved prediction agreement post-alignment.

Table 2: Comparing the largest eigenvalue of the input Hessian on the original source model and the aligned source model. Results are averaged over 1000 randomly selected data points from the ImageNet test set. The largest eigenvalue decreases significantly when evaluated on all three types of inputs.

Data	Res50	
	Original	Aligned
Clean	4.16	1.25
Gaussian-perturbed	4.34	1.27
PGD-perturbed	0.10	0.05

Table 3: Aligning the Res50 source model with multiple Res18 as witness models. Using an increasing number of Res18 witness models during the alignment process results in a modest improvement in transferability for adversarial examples generated from the aligned Res50. This result suggests that simply quadrupling the number of witness models, without considering their diversity, does not lead to a proportional improvement in transferability.

# Witness	Target									
	Res18	Res50	Res101	VGG19	DN121	IncV3	ViT-T/16	ViT-S/16	ViT-B/16	SWIN
n/a	44.52	61.51	53.01	43.47	50.59	36.05	25.58	21.54	18.86	22.65
1	+35.30	+26.56	+25.20	+24.38	+27.86	+19.18	+5.49	+2.85	+1.61	+9.60
2	+35.01	+27.83	+24.82	+24.50	+27.88	+19.31	+8.28	+3.82	+1.19	+10.67
3	+34.69	+27.69	+27.39	+24.37	+29.44	+21.51	+8.19	+4.49	+3.27	+11.74
4	+35.57	+27.63	+27.94	+25.12	+28.61	+21.81	+8.29	+5.00	+3.78	+12.15

C.2 Smoothness Analysis Using the Largest Eigenvalue of the Input Hessian

In Sec. 3.3, we study the effect of model alignment from a geometric perspective. Our analysis demonstrates a smoothing effect on the loss surface due to model alignment, as shown by the significant decrease in gradient norm.

Another metric for evaluating the smoothness of the loss landscape is the largest eigenvalue of the input Hessian [21]. This analysis, based on a second-order Taylor expansion of the loss function, assumes that the local curvature of loss with respect to the input can be well represented by the eigenspectrum of the Hessian matrix. Concretely, we measure $\lambda_{\max}(H)$ where $H = \nabla_x^2 \ell(x + \mathit{\Delta} x, y, \theta)$, for $\mathit{\Delta} x \in \{0, \mathrm{Gaussian}, \mathrm{PGD}\}$ and for $\theta \in \{\theta_s, \theta_a\}$ over 1000 randomly sampled images from the ImageNet test set.

The results are summarized in Tab. 2. We only incude results for Res50, as PyTorch currently does not support computing second-order derivatives for ViT models. In line with the result presented in Sec. 3.3, the decrease in the largest eigenvalue for all inputs demonstrates a smoothing effect on the loss surface due to the alignment process.

Table 4: Aligning Res50 using witness models with different architectures. When combining multiple models of different architectures as witness models, we observe improvements for some target models.

Witness	Target						
	Res18	Res50	VGG19	DN121	IncV3	ViT-S	SWIN
Res50	54.13	71.41	50.06	59.56	40.84	22.19	25.30
Res50+IncV3+ViT-S	51.21	66.25	48.65	59.82	44.17	23.86	25.55

D Additional Experiment

D.1 More Witness Models, Higher Transferability

Evaluations in Sec. 4 focus on alignment using a single witness model. However, aligning the source model with multiple witness models can encourage learning features extracted by a group of witness models, potentially increasing transferability even further. This approach also serves as an effective strategy to prevent overfitting to a single witness model. Tab. 3 demonstrates the results of using an increasing number of Res18 models to align with Res50. While we do observe that a greater number of witness models tends to result in more transferable perturbations, the improvement is modest, indicating that simply increasing the number of witnesses without considering their diversity does not proportionally enhance transferability.

D.2 Witness Model Diversity Matters

Though having more witnesses can prevent overfitting, model diversity is crucial. For example, in Tab. 4, we compare aligning a Res50 with a single Res50 to aligning it with a Res50, ViT-S, and IncV3. The results show that aligning with models of different architectures improves transferability to some targets while slightly decreasing it for others. Results in Tab. 3 and 4 suggest that a strategy for selecting the optimal number and type of witness models is an interesting direction for future research.

D.3 Comparison with Model Modification-based Methods

A key advantage of our approach over model modification-based methods is its model-agnostic nature: alignment can be applied to any model without changing its forward or backward pass. In contrast, other methods require changes, such as those seen in LinBP and BPA, or even complete retraining from scratch. Additionally, our technique can be integrated with these model modification-based methods. In Tab. 5, we observe the transferability improvement of MI-FGSM-LinBP and PGD-BPA generated using the aligned Res50 across all considered target models.

Table 5: Model alignment can be integrated with model modification-based methods. Using a Res50 as the target model, we compare the transferabilty of MI-FGSM-LinBP and PGD-BPA perturbations from the aligned Res50 to those from the original, unaligned Res50. We observe increased transferability across all target models.

Attack	Source	Target						
		Res18	Res50	VGG19	DN121	IncV3	ViT-T	SWIN
MI-FGSM-LinBP	Res50	87.89	93.55	84.08	88.57	71.19	43.16	46.19
	Res50*	97.27	96.27	93.07	95.61	86.04	56.25	57.71
PGD-BPA	Res50	51.06	63.98	51.54	58.24	34.02	12.0	11.04
	Res50*	76.76	81.98	70.62	77.98	53.18	19.3	20.06

Table 6: Improved transferability on Res50 with defense methods. Although the defense methods can generally reduce transferability, using the aligned model still results in a significant improvement in transferability compared to the original model.

Source	Defense						
	n/a	AT [17]	Bit-Red [18]	JPEG [5]	FD [8]	RS [3]	NRP [12]
Res50	61.51	48.06	49.22	38.81	43.77	35.41	33.01
Res50*	88.07	52.15	54.70	53.03	50.59	43.99	41.19

Note that the evaluation data selection strategy in Tab. 5 differs slightly from that in Sec. 4, where perturbations are generated from both the source and target models, and transferability evaluations are based on those misclassified by their originating models. Our evaluation considers a wide range of target model architectures. However, since the attack algorithms are only avaiable for limited architectures, making it difficult to generate perturbations from some target models. As such, results for MI-FGSM-LinBP and PGD-BPA are evalated on 1000 randomly selected inputs and are not included in Tab. 5 with other attacks.

D.4 Improved Transferability on Defended Models

To demonstrate the improved transferability on defended models, we evaluate using a normally trained Res50 with five defense mechanisms, including Bit-Red [18], JPEG [5], FD [8], RS [3], NRP [12]. We adhere to the exact configuration of the defense methods as described in the previous work [15]. We also consider an adversarially trained Res50 (AT) [17]. Results in Tab. 6 show that although the defense methods can generally reduce transferability, using the aligned model still results in a significant improvement in transferability compared to the original source model.

D.5 Experiments on Additional Datasets

Evaluations in Sec. 4 focuses on the ImageNet dataset. Here, we supplement our results with experiments on Stanford Cars [7] and Food101 [1]. On Stanford Cars,

Table 7: Generalizability of model alignment on other datasets. We consider two additional datasets: Stanford Cars and Food101. On Stanford Cars, SWIN is aligned using Res50. On Food101, Res50 is aligned using ConvNeXt-B. The aligned source model is denoted with *. We demonstrate that improved transferability can also be observed on other datasets.

Dataset	Source	Target	
		ConvNeXt-B	ViT-B
Stanford Cars	SWIN	68.77	17.6
	SWIN*	88.89	50.67

Dataset	Source	Target	
		SWIN	ViT-B
Food101	Res50	26.67	12.36
	Res50*	31.88	16.16

Table 8: Improving transferability via embedding space alignment. We focus on ViT-B/16 as the source model and ViT-T/16 as the witness model.

Method	Target									
	Res18	Res50	Res101	VGG19	DN121	IncV3	ViT-T/16	ViT-S/16	ViT-B/16	SWIN
n/a	23.05	19.67	18.30	21.83	21.96	21.52	36.44	44.58	51.27	21.22
KL	+10.13	+9.71	+8.42	+10.71	+11.30	+9.89	+51.72	+44.99	+34.69	+20.01
KD [13]	+13.40	+8.47	+8.79	+11.54	+10.25	+11.37	+46.07	+43.41	+34.02	+17.51
EGA [9]	+12.55	+10.21	+9.21	+11.39	+9.30	+12.05	+53.08	+44.65	+33.50	+21.40
HINT [14]	+12.78	+10.02	+9.84	+12.21	+12.07	+12.31	+53.41	+47.05	+38.50	+25.51
NCE [2]	+8.90	+8.48	+7.16	+9.00	+7.26	+7.92	+45.09	+41.84	+32.57	+15.47

SWIN is aligned using Res50. On Food101, Res50 is aligned using ConvNeXt-B. Results in Tab. 7 show that the improved transferability achieved through model alignment is also evident on other datasets.

D.6 Embedding-space Alignment on Vision Transformers

The ablation study in Sec. 4 shows that Res50 generates more transferable perturbations when aligned with Res18 in the embedding space. Results in Tab. 8 show that ViT can also benefit from embedding-space alignment.

References

1. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
2. Chen, Y., Xian, Y., Koepke, A., Shan, Y., Akata, Z.: Distilling audio-visual knowledge by compositional contrastive learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
3. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning (ICML) (2019)
4. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: RandAugment: Practical automated data augmentation with a reduced search space. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)

5. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. In: International Conference on Learning Representations (ICLR) (2018)
6. Kim, H.: Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950 (2020)
7. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the International Conference on Computer Vision Workshops (ICCVW) (2013)
8. Liu, Z., Liu, Q., Liu, T., Xu, N., Lin, X., Wang, Y., Wen, W.: Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
9. Ma, Y., Chen, Y., Akata, Z.: Distilling knowledge from self-supervised teacher by embedding graph alignment. In: British Machine Vision Conference (BMVC) (2022)
10. maintainers, T., contributors: Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision` (2016)
11. Müller, S.G., Hutter, F.: Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
12. Naseer, M., Khan, S., Hayat, M., Khan, F.S., Porikli, F.: A self-supervised approach for adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
13. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
14. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
15. Wang, X., He, K.: Enhancing the transferability of adversarial attacks through variance tuning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
16. Wightman, R.: Pytorch image models. `https://github.com/rwightman/pytorch-image-models` (2019). `https://doi.org/10.5281/zenodo.4414861`
17. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. In: International Conference on Learning Representations (ICLR) (2020)
18. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: Network and Distributed System Security Symposium (2018)
19. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
20. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (ICLR) (2018)
21. Zhao, P., Chen, P.Y., Das, P., Ramamurthy, K.N., Lin, X.: Bridging mode connectivity in loss landscapes and adversarial robustness. In: International Conference on Learning Representations (ICLR) (2020)
22. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI Conference on Artificial Intelligence (AAAI) (2020)