Fingerprinting Deep Neural Networks Globally via Universal Adversarial Perturbations (Supplementary Materials)

Zirui Peng[1*] Shaofeng Li[1*] Guoxing Chen[1,✉] Cheng Zhang[2] Haojin Zhu[1,✉] Minhui Xue[3,4]

[1] Shanghai Jiao Tong University [2] The Ohio State University [3] CSIRO's Data61 [4] The University of Adelaide

{pengzirui,shaofengli,guoxingchen,zhu-hj}@sjtu.edu.cn, zhang.7804@osu.edu, jason.xue@adelaide.edu.au

In this Supplementary Material, we provide details and results omitted in the main text.

- Appendix A: local variations of decision boundaries during model extraction (§ 4.2 of the main paper)
- Appendix B: effectiveness of multi-views fingerprints augmentations (§ 4.3 of the main paper)
- Appendix C: additional ablation studies (§ 5.3 of the main paper).
- Appendix D: additional results of model accuracy. (§ 5.2 of the main paper)
- Appendix E: additional analyses on the transferability of encoder. (§ 6 of the main paper)
- Appendix F: discussion on ethical issues. (§ 6 of the main paper)

A. Local Variations of Decision Boundaries During Model Extraction

In § 4.2 of the main paper, the visualization of fingerprints generated using UAP and Local Adversarial Perturbation (LAP) conducted on FashionMNIST dataset shows that LAP fingerprints are much less distinguishable than UAP fingerprints. This is because UAP based fingerprints capture the global information of decision boundaries, which is robust to model extraction process, as demonstrated in § 4.1. In this section, we will show that the local information representing by LAP based fingerprints is much less robust because decision boundaries vary significantly during model extraction.

To show the variation, we leverage a special kind of datapoints named borderpoints which are datapoints that lie on the decision boundary (*i.e.* $\{\mathbf{x} \,|\, argmax_1(f(\mathbf{x})) - argmax_2(f(\mathbf{x})) < 1e^{-6}\}$. As the last layer of DNN is a compact space, we can easily find such border points using dichotomy (*i.e.* for any $\mathbf{x_i}$ in class C_i and any $\mathbf{x_j}$ in class C_j, there exists $\lambda \in [0, 1]$ that $\lambda * \mathbf{x_i} + (1 - \lambda) * \mathbf{x_j}$ is a borderpoint). These borderpoints are used to query the piracy model of f and Figure A reports the differences between the largest confidence value and the second largest confidence value of piracy models on borderpoints. As shown in Figure A, the gap between the confidence value of the most probable class and the value of the second probable class is significant, indicating that borderpoints are far away from decision boundaries of piracy models. In this way, we show the variance of decision boundaries during model extraction quantitatively.

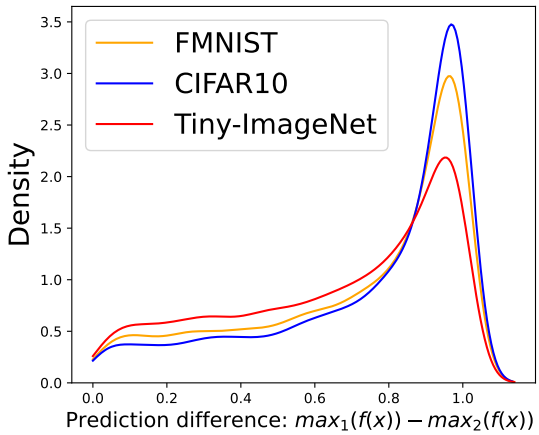
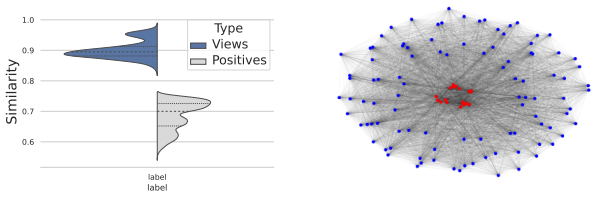


Figure A. Distributions of prediction gap of borderpoints on piracy models.

B. Effectiveness of Multi-views Fingerprints Augmentations

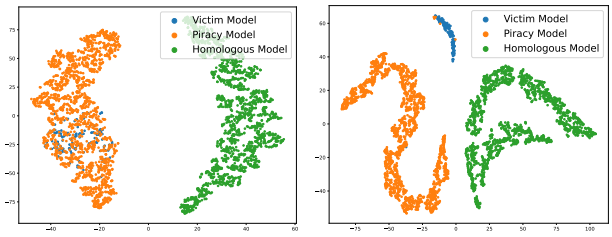
In § 4.3 of the main paper, we propose a data augmentation strategy which forms multiple views for one fingerprint and then use them to train the contrastive encoder. In this section, we verify the effectiveness of this strategy by proving that among all the positives (*i.e.*, samples belongs to the same class), those generated views of a specific fingerprint

*Equal contribution.



(a) Similarity distribution (b) Visualization of fingerprints

Figure B. (a) The left curve is the similarity between a fingerprint and its views, the right curve is the similarity between a fingerprints and its positives excluding views; (b)Visualization of fingerprints: red points are views generated from a single fingerprint and blue points are other positives with red points.



(a) Contrastive encoder (b) Non-contrastive encoder

Figure C. t-SNE visualization of representation vector outputted by the encoder of three types of fingerprints (FMNIST). Contrastive learning based encoder (left) can better differentiate homologous models from piracy models and can better mix piracy models with the victim model than non-contrastive encoder (right).

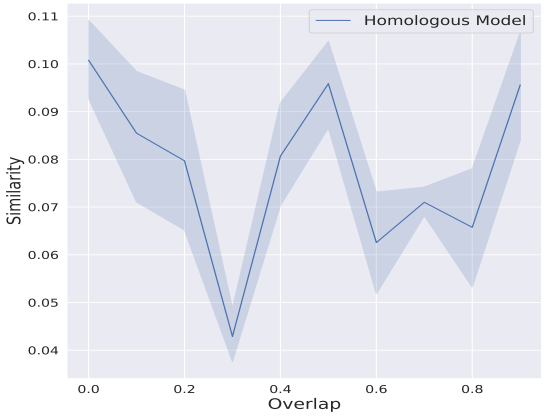


Figure D. Influence of the Overlapping rate between D_v and D_{homo} on similarities of homologous models and the victim model.

are the most similar to itself.

The similarities s between views and positives are measured by cosine similarities. Figure Ba reports the similarities between a pair of views and a pair of positives. We conclude that compared with positives, views are more similar with each other which is satisfied with our design choice. The graph in Figure Bb shows the distances between a pair of fingerprints measured by the reciprocal of their similarities $d = (1/s)$ which reveals the same results.

C. Additional Ablation Studies

C.1. Contrastive Loss

In this section, we aim at presenting the role of contrastive learning in encoder training. Recall that contrastive learning can distinguish the differences between homologous models and piracy models (*i.e.*, the representation vector of homologous models will be distant from the victim model whereas that of piracy models will be close to the victim model). This distance property in the latent space allows us to stably verify the similarity between two models and detect piracy models with high confidence.

To understand how well can contrastive learning based encoder differentiates models, we visualize the distribution of the representation vector of each fingerprint in the last layer of encoder and we compare the case of contrastive encoder with that of normal auto-encoder. Figure C is t-SNE visualization about the representation vector of fingerprints where each group of points with different colors represents one type of fingerprint. Figure Ca demonstrates the results of a contrastive learning based encoder. The representation vectors of the victim model (blue) are entangled with that of piracy models (orange) and lie on the left side of this hypersphere while the homologous models (green) lie on the opposite side. This is exactly what we expected because there exists an obvious decision boundary in Figure Ca that can easily split those three clusters into two parts. In contrast, Figure Cb shows the results of a non-contrastive encoder (*i.e.*, auto-encoder). As we can see, although each type of fingerprint is separated (recall that UAP based fingerprint itself is separable without any post-processing), the distances of any two points from different clusters variant. Besides, the representation vectors of victim's fingerprints and that of piracy models do not have any overlap. We thus conclude that the contribution of contrastive learning is vital in achieving nearly perfect detection rate.

C.2. Datasets Overlapping

In § 5.2 of the main paper, we give the implementation details about the training process of victim model, homologous models and piracy models. In this section, we will report the influence of the overlapping rate between the dataset of victim models and the dataset of homologous models with respect to the performance of our verification mechanism. Note that we are only interested in the overlapping rate between D_v and D_{homo} rather than D_v and the dataset of piracy models D_{pir}, as the data augmentation technology used in the model extraction process makes it

hard to measure their overlapping rate.

Intuitively, a homologous model trained on D_{homo} which overlaps more with D_v will assemble more the victim model and will be more difficult to be distinguished. A worth-trusty verification mechanism, however, need to be indifferent to such interference. To measure the effect of overlapping rate on our mechanism, we trained 20 homologous models with overlap rates ranging from 0 to 0.9 and calculated their similarities with $f_{\mathcal{V},u}$. Our experimental results in Figure D on FMNIST show that the overlapping rate does not undermine the performance of our verification mechanism. Which indicates our approach is effectinve on different datasets overlaps.

D. Model Accuracy

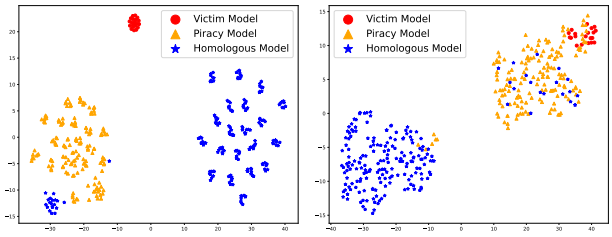
In this section, we present the accuracy on testset of all models used in our experiments in Table A as a supplement to § 5.2 of the main paper.

E. Transferability of Encoder

In § 4.3 of the main paper, the defender needs to train several homologous models and several piracy models in order to train an encoder which satisfies the detection demand. To ease the burden of defenders, in this section, we aim to verify one hypothesis: can an encoder trained for a specific victim model $f_{\mathcal{V},v_1}$ be used to protect another independent victim model $f_{\mathcal{V},v_2}$. Notice that $f_{\mathcal{V},v_1}$ is independent from $f_{\mathcal{V},v_2}$.

We evaluate this hypothesis on the FashionMNIST dataset. Specifically, given two independent victim models $f_{\mathcal{V},v_1}$ and $f_{\mathcal{V},v_2}$, *e.g.*, $f_{\mathcal{V},v_2}$ is a homologous model of $f_{\mathcal{V},v_1}$, we train an encoder E_{v_1} for $f_{\mathcal{V},v_1}$ to protect its ownership via our framework. To test the transferability of this encoder, we additionally prepare 10 homologous models and 10 piracy models for $f_{\mathcal{V},v_2}$ as well as its UAP. By far, we can generate three types of fingerprints for $f_{\mathcal{V},v_2}$, *i.e.*, the fingerprints of itself, its homologous models' fingerprints and its piracy models' fingerprints and we employ E_{v_1} to project these fingerprints to the representation space.

As demonstrated in Figure Ea, UAP based fingerprints are naturally separable, which unites with aforementioned experimental results. More importantly, as Figure Eb shown, after the projection of the encoder E_{v_1}, which is trained for another independent victim model, we observe that homologous fingerprints are indeed distant from the victim, whereas the piracy fingerprints are entangled with that of the victim. The distances of any two points belong to different clusters are maintained on most pairs of points. The average similarity $sim(f_{pir}, f_{\mathcal{V},v_2})$ for measuring the IP violation equals 0.85 and the average similarity $sim(f_{homo}, f_{\mathcal{V},v_2})$ for independent models equals 0.33. However, the similarity gap reduces slightly compared with



(a) Fingerprints without encoder (b) Representation vector outputed by encoder

Figure E. t-SNE visualization of fingerprints of three types of models associated with $f_{\mathcal{V},v_2}$ and their representation vector outputed by the encoder which is associated with $f_{\mathcal{V},v_1}$ (FMNIST).

the specific encoder that trained to protect itself (*i.e.*, $f_{\mathcal{V},v_2}$) and we claim that there is still space to improve the transferability of encoder in future work.

F. Ethical Considerations

This work is mainly a defense paper against model extraction attacks and it is hardly misused by ordinary people. In the worst case, an honest-but-curious adversary may adopt this technique to involve a normal MLaaS provider in a lawsuit that is destined to lose. This concern can be eliminated by deploying a trustworthy third party to audit the argument.

This work does not collect data from users or cause potential harm to vulnerable populations. It may arise concerns that the query data used by the defender reveal certain information about membership of their training dataset. Fortunately, in our work, the query data do not need to belong to the user's original dataset, which means that the defender can collect auxiliary data that fall in the problem domain as their query data.

The other concern is that, although our verification framework can achieve a nearly perfect accuracy with an AUC score of 1.0, we still need to pay attention to the negative impact caused by its false positive cases. Further evidence collected by social engineering can be applied as auxiliary evidence during the confirmation process to avoid the hardly appeared false positive cases mentioned above.

Table A. Accuracy of models of different architectures on different datasets. The values before brackets are the average and the values in brackets are STD. Arc.A and ResNet18 are chosen to be architecture of the victim model, so only one model of this architecture is generated.

	FashionMNIST			CIFAR10		TinyImageNet	
Architecture	Normal Training	Extraction	Architecture	Normal Training	Extraction	Normal Training	Extraction
Arc A	0.8978		ResNet18	0.8929		0.4768	
Arc B	0.8562(0.0314)	0.8786(0.0125)	ResNet34	0.8956(0.0034)	0.8918(0.0010)	0.4754(0.0582)	0.4354(0.0040)
Arc C	0.9102(0.0529)	0.8733(0.0071)	VGG16	0.9232(0.0021)	0.8860(0.0009)	0.4952(0.0520)	0.4392(0.0076)
Arc D	0.8648(0.0143)	0.8698(0.0057)	GoogLeNet	0.9185(0.0057)	0.9075(0.0007)	0.4561(0.0298)	0.3520(0.0232)
Arc E	0.8805(0.0223)	0.8845(0.0067)	DenseNet	0.9185(0.0057)	0.8962(0.0012)	0.5406(0.0305)	0.4114(0.0105)