TOWARDS DATA-FREE MODEL STEALING IN A HARD LABEL SETTING

Sunandini Sanyal **Sravanti Addepalli** **R. Venkatesh Babu**
Video Analytics Lab, Department of Computational and Data Science
Indian Institute of Science Bengaluru, India

ABSTRACT

Machine learning models deployed as a service are often susceptible to model stealing attacks. While existing attacks demonstrate near-perfect clone-model performance using softmax predictions of the classification network, most of the APIs allow access to only the top-1 labels. In this work, we show that it is indeed possible to steal Machine Learning models by accessing only top-1 predictions (Hard Label setting), without access to model gradients (Black-Box setting) or even the training dataset (Data-Free setting) within a low query budget. We propose a novel GAN-based framework[1] that trains a clone model and generator in tandem to steal the model effectively while utilizing gradients of the clone network as a proxy to the victim's gradients. We overcome the large query costs by utilizing publicly available (potentially unrelated) datasets as a weak image prior. We additionally show that even in the absence of such data, it is possible to achieve state-of-the-art results within a low query budget using synthetically crafted samples. We are the first to demonstrate the scalability of Model Stealing on a 100 class dataset.

1 INTRODUCTION

Deep learning based systems have progressed leaps and bounds over the past few years. Organizations often provide pretrained machine learning models as a service (MLaaS) where the end user is allowed to query the model and get access to its predictions via APIs for use in various applications. However, exposing the predictions of the models through queries makes the model susceptible to model stealing attacks, which attempt to clone the victim model without access to its gradients, in a black-box setting. Protecting the privacy of an ML model is of paramount importance as organizations invest significant resources on cutting edge research and also on gathering and labelling large amounts of training data Halevy et al. (2009) for achieving competent performance on various tasks. In addition, recent works Papernot et al. (2017); Tramèr et al. (2017); Zhou et al. (2020); Wang et al. (2021a) have shown that an adversary could train a substitute model via model stealing and use it further for crafting adversarial examples Goodfellow et al. (2014b) in a black-box setting, which poses a serious threat when the model is deployed in security critical applications. A stolen model could also compromise the privacy of users by leaking confidential data through a membership inference attack Shokri et al. (2017) or via model inversion Zhang et al. (2020); Zhao et al. (2021). Figure-1 showcases some of the possible malicious outcomes of Model Stealing. In order to prevent model stealing attacks, some defenses attempt to perturb the softmax predictions of the model, while preserving the top-1 prediction Lee et al. (2018). In this work, we consider the problem of model stealing in a more practical and challenging hard label setting, where only the top-1 prediction of the model is accessible, and is thus effective even in the presence of such defenses.

In a model stealing attack, an adversary first queries a black-box victim model $\mathcal{V}$ with input data and obtains a prediction for it as shown in Fig.1. This data along with its labels are used to train a clone model $\mathcal{C}$. In a practical scenario, the attacker would not have access to the training data, and hence we consider the problem of Data-Free Model Stealing (DFMS) in this work. In such a data-free scenario, the attacker could use publicly available related datasets Papernot et al. (2017); Orekondy et al. (2019a), or synthetically generated samples Truong et al. (2021) to query the model. While the use of publicly available datasets assumes access to related data, the

[1]Project Page: `https://sites.google.com/view/dfms-hl`

data-free generative approach suffers from a large query budget, as the synthetic data can be far from the true training data distribution. In this work we overcome both challenges by utilizing the available data that may be unrelated to the original training dataset, as a weak image prior. This enables the generation of representative samples under a low query budget, which is a crucial requirement in model stealing attacks, since MLaaS APIs work on a pay-per-query basis.

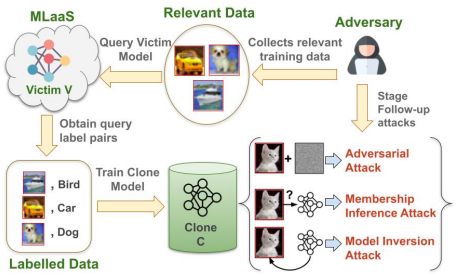

Figure 1: **Model Stealing Attack and its vulnerabilities**

While existing algorithms for Data-Free Knowledge Distillation Addepalli et al. (2020); Nayak et al. (2019); Lopes et al. (2017); Yin et al. (2020); Fang et al. (2019) and Model Extraction Kariyappa et al. (2021); Truong et al. (2021) achieve near perfect clone-model accuracy, there are additional challenges in a Model Stealing framework due to the lack of access to gradients and a hard-label setting. Therefore, we consider a practical setup of data-free hard-label model stealing and overcome the challenges by utilizing the clone model's gradients as a proxy to the gradients of the victim model. As the clone model starts training, it acts as a useful proxy for the victim model, and helps the generator learn to generate rich informative samples, which boosts the clone accuracy further. We explicitly enforce the generation of a class-balanced dataset from the generator that is also more aligned with the distribution of the training dataset. Additionally, we also utilize an adversarial loss in a GAN framework Goodfellow et al. (2014a), by using publicly available potentially unrelated data, which we refer to as proxy data Addepalli et al. (2020). While this could be completely unrelated to the original training dataset, it still helps in enforcing a weak image prior in the generated data. This in turn reduces the number of victim model queries needed to perform Model Stealing. In fact, we show that it is possible to even use synthetic samples, such as multiple overlapping shapes with a planar background, to steal a model in a completely data-free setting. Our method achieves a significant improvement over ZSDB3KD Wang (2021), a zero-shot data-free method in a similar hard label setting using only synthetic samples.

Key Contributions: Our key contributions are as follows.

- We propose DFMS-HL, a data-free model stealing (DFMS) attack in a hard-label (HL) setting to train a clone model with the help of unrelated proxy data. We show that DFMS-HL outperforms the existing baseline ZSDB3KD Wang (2021) and results in a significant reduction of around $500\times$ in the number of queries to the victim model.
- We demonstrate state-of-the-art results on the CIFAR-10 dataset using unrelated proxy samples, such as a given subset (containing 40 or 10 non-overlapping classes) from CIFAR-100, or synthetic data.
- We are the first to show noteworthy results of data-free model stealing on a dataset with a larger number of classes such as CIFAR-100.
- The soft-label variant (DFMS-SL) achieves a significant boost of 3% over the state-of-the-art model stealing attacks MAZE Kariyappa et al. (2020) and DFME Truong et al. (2021).

2 Proposed Approach

We propose a data-free model stealing approach **DFMS-HL** that requires only hard-label access. At first, we train a DCGAN by imposing an image prior using synthetic or unrelated proxy data. This gives a good initialization for the generator $\mathcal{G}$. We also train an initial clone model with the proxy images. Following this, we begin our procedure of alternately training the clone model and the generator. The data flow is shown in Fig. 2 wherein the generator $\mathcal{G}$ generates data $x = \mathcal{G}(z)$ from a random vector z. The victim model takes input x and generates input, label pairs $(x, \hat{y}(x))$ for each instance x. Since, the victim model is black-box, we do not backpropagate the gradients through it. The labelled input pairs are used to train the clone model with the cross-entropy loss as

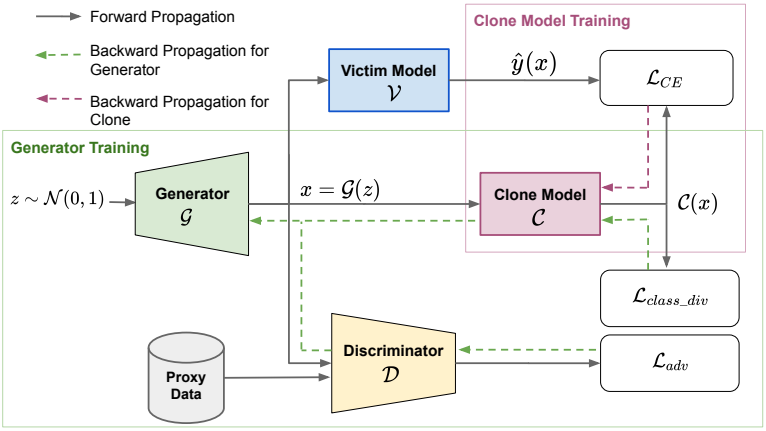

Figure 2: **Architecture of DFMS-HL**: Generator $\mathcal{G}$ generates data x with a proxy image prior. The clone model $\mathcal{C}$ is trained using the labels from the victim model $\mathcal{V}$ with a cross-entropy loss objective $\mathcal{L}_{CE}$. The discriminator $\mathcal{D}$ learns to discriminate between proxy and generated samples from $\mathcal{G}$. The generator $\mathcal{G}$ is trained with the adversarial loss $\mathcal{L}_{adv}$ along with the class-diversity loss $\mathcal{L}_{class\_div}$. The generator and clone model are trained alternately in every iteration of the algorithm.

follows:

$$\mathcal{L}_C = \mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,I)} \left[\mathcal{L}_{CE}(\hat{y}(x), \mathcal{C}(x)) \right], \; x = \mathcal{G}(z) \tag{1}$$

where $\hat{y}(x) = \underset{i}{\operatorname{argmax}} \, \mathcal{V}_i(x)$ is the class label for the maximum probability class and $\mathcal{C}(x)$ is the output logits from the clone model. The generator is trained with the adversarial loss Goodfellow et al. (2014a) and a unique diversity loss as shown below:

$$\mathcal{L}_{adv,real} = \mathop{\mathbb{E}}_{x \sim p_{data}(x)} \left[log \mathcal{D}(x) \right], \; \mathcal{L}_{adv,fake} = \mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,I)} \left[log(1 - \mathcal{D}(\mathcal{G}(z))) \right] \tag{2}$$

Across a batch of N samples, we take the expected confidence value over the batch as α_j for every class j and obtain the entropy over K classes. Hence, the generator model learns to generate samples from different classes by minimizing the diversity loss formulation as below,

$$\mathcal{L}_{class\_div} = \sum_{j=0}^{K} \alpha_j \log \alpha_j, \; \alpha_j = \frac{1}{N} \sum_{i=1}^{N} \text{softmax}(\mathcal{C}(x_i))_j \tag{3}$$

The equations below describe the generator and discriminator losses, that are minimized alternately for training.

$$\mathcal{L}_G = \mathcal{L}_{adv,fake} + \lambda_{div} \mathcal{L}_{class\_div}, \quad \mathcal{L}_D = \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake} \tag{4}$$

We refer the reader to Appendix B for details of the proposed method.

3 Experiments

Comparison with Knowledge distillation (KD) methods: We perform experiments on CIFAR-10 as the True dataset as shown in Table 1 for comparing with existing KD methods. DeGAN Addepalli et al. (2020) and ZSKD Nayak et al. (2019) are data-free knowledge distillation methods with white-box teacher access. KnockoffNets Orekondy et al. (2019a) and Black-Box Ripper Barbalau et al. (2020) are data-free KD methods in a black-box setting. Similar to the experimental setting of prior works Addepalli et al. (2020); Barbalau et al. (2020), we use 40 unrelated classes from CIFAR-100 dataset as the proxy dataset for CIFAR-10 model stealing. We also show results using 10 classes sampled randomly from these 40 classes. We achieve comparable results with the data-free KD methods despite having more restrictions on access to the victim model.

We also show results by using synthetically crafted data for imposing image priors using the discriminator. For this, we generate a synthetic dataset of 50k samples by including random shapes (triangle, rectangle, ellipse or circles) of randomly sampled sizes at random locations on a plain

Table 1: Comparison of DFMS-HL with state-of-the-art KD methods(Top) and ZSDB3KD(Bottom) on CIFAR-10 with AlexNet as victim and AlexNet-half as the clone model

Method	Hard Label	Black Box	Data Free	Victim Acc	Data Free	CIFAR-100 (40C)	CIFAR-100 (10C)
Victim Accuracy = 82.5%							
ZSKD	×	×	✓	82.50	**69.50**	69.50	69.50
DeGAN	×	×	✓	82.50	-	76.30	72.60
KnockoffNets	×	✓	✓	82.50	-	65.70	46.60
Black-Box Ripper	×	✓	✓	82.50	-	**76.50**	**77.90**
DFMS-HL (Ours)	✓	✓	✓	82.52	65.70	76.02	71.36
Victim Accuracy ∼ 80%							
ZSDB3KD	✓	✓	✓	79.30	59.46	59.46	59.46
DFMS-HL (Ours)	✓	✓	✓	80.18	**67.03**	**74.27**	**70.57**

Table 2: Comparison of DFMS-HL with data-free model stealing methods MAZE and DFME (Top, Victim: ResNet-34) and with ZSDB3KD (Bottom, Victim: ResNet-18) on CIFAR-10. Clone model architecture is ResNet-18.

Method	Hard Label	Black Box	Data Free	Victim Acc	Data Free	CIFAR-100 (40C)	CIFAR-100 (10C)
Victim Accuracy ∼ 95.5%							
MAZE	×	✓	✓	95.50	45.60	-	-
DFME	×	✓	✓	95.50	88.10	-	-
DFMS-HL (Ours)	✓	✓	✓	95.59	84.51	**92.06**	**85.53**
DFMS-SL (Ours)	×	✓	✓	95.59	**91.24**	**93.96**	**90.88**
Victim Accuracy ∼ 93.7%							
ZSDB3KD	✓	✓	✓	93.65	50.18	50.18	50.18
DFMS-HL (Ours)	✓	✓	✓	93.83	**85.92**	**90.51**	**83.37**

Table 3: Performance of DFMS-HL on CIFAR-100 with different proxy datasets

Method	Proxy Data	Victim Acc	Clone Acc
DeGAN	CIFAR-10	78.52	75.62
DFMS-HL (Ours)	CIFAR-10	78.52	72.83
DFMS-HL (Ours)	Data Free	78.52	43.56

Table 4: Clone model accuracy (%) using DFMS-HL with different proxy datasets.

Victim training Data:	CIFAR-10					Fashion MNIST
Proxy Data:	SVHN	Data Free	CelebA	Tiny imagenet	Imagenette	CIFAR-10
ZSDB3KD	-	50.18	-	-	-	-
DFMS-HL (Ours)	84.83	**84.51**	85.82	92.26	90.06	81.98

background of random color (details in the Supplementary). We also generate textured images by increasing the maximum number of shapes to 100 and reducing the maximum region occupied by the shapes in the image. These manually crafted images are converted to grey-scale and then used as proxy data.

From Table 1, it can be observed that our approach not only outperforms ZSDB3KD by a large margin, but also achieves a comparable accuracy with respect to the DeGAN and Black-Box Ripper for the CIFAR-100 40 classes proxy data. We also use a significantly lower query budget of 8M as compared to ZSDB3KD which requires 4000M queries. We also perform experiments on CIFAR-100 (Table 3) with CIFAR-10 Addepalli et al. (2020); Barbalau et al. (2020) and synthetic data as proxy datasets. DFMS-HL reaches a comparably close accuracy of 72.83% using CIFAR-10 as the proxy without any access to the victim model's gradients and only using hard labels. We report the clone model accuracy with other proxy datasets in Table 4. For the data-free approaches, we report the numbers under the "Synthetic" column across all tables since "Synthetic" means that any additional is not used in this case.

Comparison with Model Stealing methods. We compare our approach with the state-of-the-art data-free Model Stealing approaches Kariyappa et al. (2021); Truong et al. (2021) in Table 2. We obtain an accuracy of 84.51% by merely using synthetic samples in a completely data-free hard-label setting. We use a lower query budget of 8M, as compared to that of DFME and MAZE that require 20M queries for CIFAR-10. We further extend our attack to the soft-label black-box scenario (denoted as DFMS-SL in Table 2) where the softmax predictions of the victim model are available. We get a boost of almost 3% using synthetic data and CIFAR-100 10 classes with the same query budget of 20M.

4 Conclusions

In this paper, we propose an effective model stealing attack in a practical setting of having access to only hard-labels of a black-box victim model. Extensive experiments show that our method DFMS-HL performs better than the state-of-the art model stealing method at a 500x lower query budget. We further show that it is possible for an attacker to craft a synthetic dataset of images containing various shapes on a planar background and use it to attack a victim model in a completely data-free setting. We demonstrate the scalability of the proposed model stealing attack to CIFAR-100 as well with a low query budget, which has not been attempted in prior works.

Acknowledgements This work was supported by a project grant from MeitY (No.4(16) /2019-ITEA), Govt. of India and a grant from Uchhatar Avishkar Yojana (UAY, IISC_010), MHRD, Govt. of India. Sunandini Sanyal is supported by Prime Minister's Research Fellowship, and Sravanti Addepalli is supported by Google PhD Fellowship. We are thankful for the support.

REFERENCES

Sravanti Addepalli, Gaurav Kumar Nayak, Anirban Chakraborty, and Venkatesh Babu Radhakrishnan. DeGAN: Data-enriching gan for retrieving representative samples from a trained classifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Romero Adriana, Ballas Nicolas, K Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and B Yoshua. Fitnets: Hints for thin deep nets. *Proc. ICLR*, 2015.

Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing GANs using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019.

Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, and Marius Popescu. Black-Box Ripper: Copying black-box models using generative evolutionary algorithms. *arXiv preprint arXiv:2010.11158*, 2020.

Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

Varun Chandrasekaran, Hengrui Jia, Anvith Thudi, Adelin Travers, Mohammad Yaghini, and Nicolas Papernot. Sok: Machine learning governance. *arXiv preprint arXiv:2109.10870*, 2021.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Jacson Rodrigues Correia-Silva, Rodrigo F Berriel, Claudine Badue, Alberto F de Souza, and Thiago Oliveira-Santos. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *International Joint Conference on Neural Networks (IJCNN)*, 2018.

Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

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

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

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

Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020.

Sanjay Kariyappa and Moinuddin K Qureshi. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2020.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Protecting dnns from theft using an ensemble of diverse models. In *International Conference on Learning Representations*, 2020.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*, 2019.

Taesung Lee, Benjamin Edwards, Ian Molloy, and Dong Su. Defending against model stealing attacks using deceptive perturbations. *arXiv preprint arXiv:1806.00054*, 2018.

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

Smitha Milli, Ludwig Schmidt, Anca D Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 1–9, 2019.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pp. 4743–4751. PMLR, 2019.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019a.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against dnn model stealing attacks. *arXiv preprint arXiv:1906.10908*, 2019b.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. A framework for the extraction of deep neural networks by leveraging public data. *arXiv preprint arXiv:1905.09165*, 2019.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 601–618, 2016.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Jean-Baptiste Truong, Pratyush Maini, Robert J Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4771–4780, 2021.

Wenxuan Wang, Bangjie Yin, Taiping Yao, Li Zhang, Yanwei Fu, Shouhong Ding, Jilin Li, Feiyue Huang, and Xiangyang Xue. Delving into data: Effectively substitute training for black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021a.

Yixu Wang, Jie Li, Hong Liu, Yan Wang, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Black-box dissector: Towards erasing-based hard-label model stealing attack. *arXiv preprint arXiv:2105.00623*, 2021b.

Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. *arXiv preprint arXiv:2106.03310*, 2021.

Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. Model compression with two-stage multi-teacher knowledge distillation for web question answering system. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 690–698, 2020.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Y Lim. Exploiting explanations for model inversion attacks. *arXiv preprint arXiv:2104.12669*, 2021.

Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

A Related Works

In this section, we discuss existing Knowledge Distillation and Model Stealing works with varied levels of access to the victim model as shown in Table-5.

A.1 Knowledge distillation

Knowledge distillation Hinton et al. (2015) aims to transfer the knowledge of a large pretrained teacher model to a smaller student model without a significant impact on accuracy. This is primarily used to compress models for deployment, in order to reduce the memory requirements and inference time Gou et al. (2021); Adriana et al. (2015); Yang et al. (2020); Aguinaldo et al. (2019). In practical scenarios, training data is kept confidential due to privacy concerns. Hence, there has been a lot of focus on developing data-free approaches for knowledge-distillation. ZSKD Nayak et al. (2019), DAFL Chen et al. (2019), DFKD Lopes et al. (2017) are popular knowledge distillation methods in a data-free setting. A data-free KD method DeGAN Addepalli et al. (2020) demonstrated that it is possible to use publicly available unrelated data (proxy dataset) to distill the knowledge of a teacher model to a smaller student model. However, all these methods require access to the teacher model's gradients. Following this, Black-Box Ripper Barbalau et al. (2020) was proposed to implement model stealing by querying a black-box teacher model with unrelated proxy data. A recent work ZSDB3KD Wang (2021) proposed knowledge distillation for a black box model with only hard-label outputs. However, this approach is highly computationally intensive due to the requirement of a very large number of queries (4000 million) to the teacher model. Our work considers the same setup of having access to only the top-1 labels, with a significantly lower query budget of 8 million.

A.2 Model Stealing

Tramèr et al. (2016) demonstrated that an attacker could use queries to steal a machine learning model with near perfect fidelity. Following this, model stealing has been implemented in various domains Krishna et al. (2019); Jagielski et al. (2020); Pal et al. (2019); Correia-Silva et al. (2018); Milli et al. (2019). A partial data approach JBDA Papernot et al. (2017) assumed access to a small set of samples from the data distribution. On the other hand, surrogate data approaches such as KnockOffNets Orekondy et al. (2019a) and Black-Box dissectorWang et al. (2021b) consider that attackers could use images from a different data source to steal a model. These methods fail to perform well without a suitable surrogate dataset. This motivated the development of data-free approaches which work well without using surrogate data or seed samples from the training data. Recent data-free approaches such as MAZE Kariyappa et al. (2021) and DFME Truong et al. (2021) attempt to extract models using GAN generated synthetic data. In these approaches, the generator is trained to produce images that maximize the dissimilarity score between the clone and victim models. The victim model's gradients are required to measure this dissimilarity score, and are estimated using zeroth-order gradient approximation. These approaches are computationally expensive as they require a lot of queries ($\sim$20 million) to the victim model for synthesizing data samples in a black-box setting. Moreover, these methods assume that the softmax vector from the teacher model

Table 5: Taxonomy of prior works on Knowledge Distillation (KD) and model stealing attacks. Our approach DFMS-HL is a data-free model stealing attack on a black-box victim model with access to only hard labels.

Approach	**White-Box Soft Label**	**Black-Box Soft Label**	**Black-Box Hard Label**
Data free	ZSKD Nayak et al. (2019) DeGAN Addepalli et al. (2020)	MAZE Kariyappa et al. (2021) DFME Truong et al. (2021)	ZSDB3KD Wang (2021) DFMS-HL (Ours)
Data	KD with Data Hinton et al. (2015)	KnockoffNets Orekondy et al. (2019a) JBDA Papernot et al. (2017)	-

is accessible. Contrary to this, we consider a practical setting that allows access to only hard labels from the victim model.

A.3 Defenses against model stealing

Lee et al. (2018) propose to defend against model stealing attacks by perturbing the model predictions while preserving its top-1 label, to maintain similar classification accuracy. On similar lines, Prediction Poisoning Orekondy et al. (2019b) perturbs model predictions by poisoning the output distribution at the cost of model accuracy. However, such defenses fail in a scenario where an attacker has access to only hard labels from the model. A more sophisticated approach EDM Kariyappa et al. (2020) introduces randomness into the predictions by using an ensemble of diverse models to produce dissimilar outputs for Out-of-Distribution (OOD) samples, that are likely to be used for querying the victim model in a model stealing attack. Similarly, Adaptive Misinformation Kariyappa & Qureshi (2020) perturbs the predictions for OOD inputs only. However, these approaches have been shown to cause utility degradation Orekondy et al. (2019b), or can be made ineffective using an adaptive query synthesis strategy Chandrasekaran et al. (2020). Further, Chandrasekaran et al. (2020; 2021) provide theoretical insights to demonstrate that "model extraction is inevitable", even in a realistic setting with only hard labels, and even when models use randomised defenses. Hence, a model with a reasonably good accuracy would always leak information that could lead to model extraction. In this work we demonstrate that it is indeed possible to perform model stealing in a severely restricted setting as well, and further achieve competent clone accuracy. This paves way to the development of better defenses for preserving model privacy in future.

B Details of the proposed method

B.1 Clone model Training

The clone model $\mathcal{C}$ is trained using the data samples generated from the generator $\mathcal{G}$. In every iteration, we sample an m-dimensional random vector z, whose elements are sampled from m *i.i.d.* Standard Normal distributions. This vector is forward propagated through $\mathcal{G}$ to generate images x. These images are then passed to the victim model to obtain its hard-labels. The clone model is trained with the cross-entropy loss objective using the victim predictions as ground truth, as shown below:

$$\mathcal{L}_C = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I)} \left[\mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))\right],\ x = \mathcal{G}(z) \tag{5}$$

where $\hat{y}(x) = \underset{i}{\operatorname{argmax}}\ \mathcal{V}_i(x)$ is the class label corresponding to the maximum probability class, I is an m dimensional identity matrix, and $\mathcal{C}(x)$ is the pre-softmax output from the clone model.

B.2 Generator Training

For imposing an image prior, we initially train a DCGAN generator using proxy data or synthetic images. However, we find that this is not sufficient as the generator could potentially suffer from mode collapse and lack of diversity. Moreover, lack of class diversity can severely impact the learning of tail classes in a hard-label setting. Hence, it crucial for the generator to generate a class-balanced set of images for learning the information across all classes. Therefore, we use a class-diversity loss for-

Algorithm 1 DFMS-HL : Algorithm for Model Stealing

Require: $N_Q, \mathcal{G}, \mathcal{D}, n_G, n_C$
// Initialize a Generator $\mathcal{G}$ with DCGAN parameters
// Train the clone model $\mathcal{C}$ with DCGAN and proxy images using n_C queries for initialization.
while $n_G \neq 0$ **do**
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_G \leftarrow \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div}$
$\mathcal{L}_D \leftarrow \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$
$\theta_{\mathcal{G}} \leftarrow \theta_{\mathcal{G}} - \epsilon_{\mathcal{G}}\nabla_{\theta_G}\mathcal{L}_G$
$\theta_{\mathcal{D}} \leftarrow \theta_{\mathcal{D}} - \epsilon_{\mathcal{D}}\nabla_{\theta_D}\mathcal{L}_D$
$n_G \leftarrow n_G - 1$
end while
// Train clone model $\mathcal{C}$
while $n_C \neq 0$ **do**
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_C \leftarrow \mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))$
$\theta_{\mathcal{C}} \leftarrow \theta_{\mathcal{C}} - \epsilon_{\mathcal{C}}\nabla_{\theta_C}\mathcal{L}_C$
$n_C \leftarrow n_C - 1$
end while
// Start alternate training between $\mathcal{G}$ and $\mathcal{C}$
while $N_Q \neq 0$ **do**
// Train $\mathcal{G}$ and $\mathcal{D}$ with $\mathcal{C}$ as fixed
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_G \leftarrow \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div}$
$\mathcal{L}_D \leftarrow \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$
$\theta_{\mathcal{G}} \leftarrow \theta_{\mathcal{G}} - \epsilon_{\mathcal{G}}\nabla_{\theta_G}\mathcal{L}_G$
$\theta_{\mathcal{D}} \leftarrow \theta_{\mathcal{D}} - \epsilon_{\mathcal{D}}\nabla_{\theta_D}\mathcal{L}_D$
// Train $\mathcal{C}$ with $\mathcal{G}$ and $\mathcal{D}$ as fixed
$x = \mathcal{G}(z), z \sim \mathcal{N}(0, I)$
$\mathcal{L}_C \leftarrow \mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))$
$\theta_{\mathcal{C}} \leftarrow \theta_{\mathcal{C}} - \epsilon_{\mathcal{C}}\nabla_{\theta_C}\mathcal{L}_C$
end while

mulation Addepalli et al. (2020) to generate diverse samples from the generator $\mathcal{G}$ while remaining close to the manifold of the proxy/synthetic images.

The generator loss has two components. The first component is the adversarial loss Goodfellow et al. (2014a) which causes the generator to generate data close to the proxy data distribution. The second component is a class balancing loss Addepalli et al. (2020), to enforce a diversity constraint. The two loss formulations for the generator are described in more detail below.

Adversarial Loss Goodfellow et al. (2014a): The adversarial loss ensures that the distribution of images is close to the images in the proxy or synthetic dataset.

$$\mathcal{L}_{adv,real} = \mathop{\mathbb{E}}_{x \sim p_{data}(x)} [log\mathcal{D}(x)] \tag{6}$$

$$\mathcal{L}_{adv,fake} = \mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,I)} [log(1 - \mathcal{D}(\mathcal{G}(z))] \tag{7}$$

The discriminator $\mathcal{D}$ and generator $\mathcal{G}$ play a min-max game Goodfellow et al. (2014a) as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake} \tag{8}$$

Class Diversity Loss Addepalli et al. (2020): The class diversity loss encourages the generation of diverse images across all classes. In a batch of N samples, we consider the expected confidence value over the batch as α_j for every class j, and obtain the entropy over all K classes. The negative entropy, denoted as $\mathcal{L}_{class\_div}$ is computed as shown below:

$$\mathcal{L}_{class\_div} = \sum_{j=0}^{K} \alpha_j \log \alpha_j \tag{9}$$

$$\alpha_j = \frac{1}{N}\sum_{i=1}^{N} \text{softmax}(\mathcal{C}(x_i))_j \tag{10}$$

Using clone Model as a proxy for victim: Since, the victim model is black-box, backpropagation through $\mathcal{V}$ is not permitted. Hence, for imposing diversity we use the clone model parameters to compute the loss. Over the training process, the clone learns to imitate the gradients of the victim, making it a suitable proxy for enforcing diversity in the generated images.

The equations given below describe the overall generator and discriminator losses.

$$\mathcal{L}_G = \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div} \tag{11}$$

$$\mathcal{L}_D = \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake} \tag{12}$$

B.3 Algorithm

The overall training algorithm is outlined in Algorithm-1. We first train a DCGAN to initialize the generator model with an image prior. Following this, we train the clone model using a mix of images from the DCGAN and the proxy dataset to obtain a good initialization for the clone model. Using this clone model, we further fine-tune the generator for n_G epochs using the two proposed losses; adversarial loss and class-diversity loss. We then train a clone model from scratch for n_C epochs using the images from the diverse generator $\mathcal{G}$. Following this, we start the alternate training process for the generator and clone model. We train the generator for one iteration by freezing weights of the clone model and subsequently train the clone model for one iteration using labels from the victim model. This procedure is repeated until the query budget N_Q is exhausted.

B.4 Computing the Query Cost

In this section, we compute the total number of queries to the victim model. The number of samples in the proxy data is denoted as N_P. Initially, we require n_C queries to obtain a clone model to initialize the generator and an additional n_C queries to initialize the Classifier $\mathcal{C}$. For our experiments, we set n_C as 50,000. The alternate training of the clone and generator continues for E epochs and in each epoch, the victim model is queried N_P times. So the total query cost is computed as follows,

$$N_Q = E \cdot N_P \tag{13}$$

$$\text{Total Queries} = 2 \cdot n_C + N_Q \tag{14}$$

We set the query limit N_Q to 8 million for our proxy and synthetic data experiments on CIFAR-10.

B.5 Insights on Query Budget

Chandrasekaran et al. (2020) formulated the model extraction task as a query synthesis active learning problem where an adversary learns a hypothesis function with a query complexity $q_A(\epsilon, \delta)$. They show that, given a maximum query budget of $q_A(\epsilon, \delta)$ and a victim model $\mathcal{V}$ trained with a specific hypothesis $f^* \in \mathcal{F}$, there exists an adversary $\mathcal{A}$ which implements an ϵ-extraction attack with confidence $1 - \delta$. Adversary $\mathcal{A}$ trains a clone model $\mathcal{C}$ with hypothesis $\hat{f}$ such that the following holds true.

$$Pr[\mathcal{A} \text{ trains } \hat{f} \text{and Err}(\hat{f}) \leq \epsilon] \geq 1 - \delta \tag{15}$$

where $Err(\hat{f}) = ||w^* - w||_2$, w and $\hat{w}$ being the parameters of $\hat{f}$ and f^*, respectively. This shows that an adversary can implement a model stealing algorithm in a Query Synthesis scenario using active learning.

Chandrasekaran et al. (2020) further proved that model stealing is inevitable and there exists a query bound within which a model could be stolen. They show that even when a victim employs a randomized procedure for returning labels such that the upper bound on the probability of returning wrong labels $\rho_D(f^*) < \frac{1}{2}$, an adversary can implement an ϵ-extraction attack with confidence $1-2\delta$ within the following query bound:

$$q = \frac{8}{(1 - 2\rho_D(f^*))^2} q(\epsilon, \delta) ln \frac{q(\epsilon, \delta)}{\delta} \tag{16}$$

We empirically find the query budget needed for the proposed approach in the Query ablation (Appendix E).

C EXPERIMENTAL SETUP

We evaluate DFMS-HL on two datasets: CIFAR-10 and CIFAR-100. For evaluation, we first train a victim model with the same teacher accuracy as ZSDB3KD Wang (2021) for a fair comparison. The victim models are trained till the accuracy reaches the teacher accuracy. We evaluate two settings of ResNet18 and AlexNet as victim models, and ResNet18 and AlexNet-half as clone models, respectively. We use an SGD optimizer with a momentum of 0.9, learning rate of 0.1 and a weight decay of 5×10^{-4} to train the models. We also use a cosine annealed scheduler to decay the learning rate across epochs. At the start of the initial clone model training, we use the same SGD optimizer and train the model from scratch for 200 epochs. After this, the clone model is further trained with new images generated from the generator within the query budget or till the accuracy saturates.

For the generator, we use a DCGAN with upto five transpose convolution layers followed by batch-normalization and ReLU units, except the last layer. The last convolution layer is followed by Tanh activation units to convert the images in the normalised range of [-1,1]. The discriminator contains a stack of five convolution layers followed by batch normalization and Leaky ReLU units. The last layer of the discriminator uses a Sigmoid at the end. The GAN is trained with an Adam optimizer Kingma & Ba (2014) with a learning rate of 2×10^{-4} with (β_1, β_2) as $(0.5, 0.999)$.

D DATASETS

We perform experiments using different proxy datasets similar to prior works Addepalli et al. (2020); Barbalau et al. (2020) to evaluate the effectiveness of our method DFMS-HL. This section contains a description of the different datasets that we used to evaluate our attack with CIFAR-10 as the true dataset.

- **40-unrelated classes from CIFAR-100 Addepalli et al. (2020):** This consists of training data from CIFAR-100 belonging to non-overlapping classes with respect to CIFAR-10. The classes from the following categories are included: food containers, household electric devices, household furniture, large man-made outdoor things, large natural outdoor scenes, flowers, fruits and vegetables, trees.
- **10 random classes of CIFAR-100:** From the above 40 unrelated classes, we choose 10 classes randomly to demonstrate this setting. The classes used are : plate, rose, castle, keyboard, house, forest, road, television, bottle and wardrobe.
- **Synthetic Dataset:** We construct synthetic images which are far from the manifold of the training data distribution to simulate this setting. The images contain multiple overlapping shapes on top of a planar background. The creation of synthetic images is described in Sec. D.1.

D.1 CREATION OF SYNTHETIC DATASET

The algorithm to create a synthetic dataset is presented in Algorithm 2. At first, randomly sampled shapes (triangle, rectangle, circle or ellipse) are generated at random locations in the image with a randomly sampled colour. The shapes are generated using python skimage module[2]. A total of 50K images are generated. We generate two kinds of images. The first variant contains large overlapping shapes with number of shapes in the image (num_shapes) as 50 and the (min_size, max_size) of each shape as (20,50). The initial image generated is of size (100 x 100) which is scaled down to (32 x 32). The other variant contains textured images with (min_size, max_size) as (5,10) and num_shapes=50 to get small overlapping shapes on top of a planar background. A random colour is sampled and assigned to the background pixels. These images are then used to steal an ML model

[2] https://scikit-image.org/docs/stable/auto_examples/edges/plot_random_shapes.html

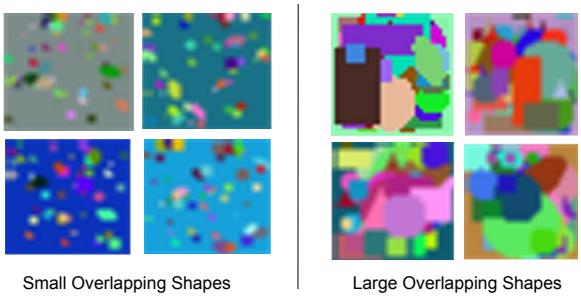

Figure 3: **Types of synthetic images used.** An equal share of large(right) and small(left) overlapping shapes on planar background used to train the clone model.

Algorithm 2 Algorithm for creating synthetic data

Require: Number of images to be generated N_P, num_shapes, max_size, min_size
  while $N_P \neq 0$ **do**
    Generates shapes on an image of size (100 x 100), with parameters: num_shapes, min_size, max_size
    Assign a random RGB colour to background pixels
    Perform blurring on the image using a 4 x 4 filter
    Resize image to (32 x 32)
    $N_P \leftarrow N_P - 1$
  end while

trained on CIFAR-10 and CIFAR-100. The generated images are shown in Fig. 3. We share our dataset here[3].

E Ablation Experiments

Effect of Query Budget: Query budget is one of the critical factors in model stealing as the number of queries to the victim model is usually restricted. We do an analysis on the accuracy of the clone model achieved with different number of queries. Our approach achieves a good accuracy with a query budget of 7.6 million on synthetic data for AlexNet as victim model and AlexNet-half as clone model. From Fig.4, we observe that even with a small query budget of 1.26M, our method performs well and it almost saturates within 8M. We report the saturating accuracies in Table 1 and 2. We use a query budget of 10M for the CIFAR-100 experiments (Table 3) and 8M for CIFAR-10 experiments (Tables1 and 2). The class-diversity loss has a huge impact on the clone accuracy as we observe a significant boost of 6% for the synthetic experiment for 7.6M queries. Hence, class balancing is an essential component of our approach.

Effect of Class Diversity Loss: The class diversity loss plays a major role in determining the diversity of samples generated by the Generator. We perform an ablation study to see the impact of the class diversity loss by gradually increasing the loss coefficient from 0 to 1000 for synthetic data as proxy with CIFAR-10 as the true dataset as shown in Fig. 5. We run the ablations for 150 epochs of training which limits the queries to 7.6M. We find that increasing the coefficient λ_{div} of class-diversity loss improves the clone model accuracy. We reported our final results with a λ_{div} value of 500 for CIFAR-10 experiments in Table 1 and 2 and set λ_{div} as 100 for CIFAR-100 experiments in Table 3.

Effect of alternate training: The generator and the clone model are trained once in every iteration. We make efforts to further reduce the query budget by training the clone model after every t iterations. We perform an ablation in Fig. 6 to study the effect of increasing the iteration gap from 0 to 4, where the clone model gets trained after every t number of iterations. A gap of 0 means that the

[3] https://drive.google.com/drive/folders/1CCMCYVRnvqZig9dYUYO_BupI8tImGZ2x

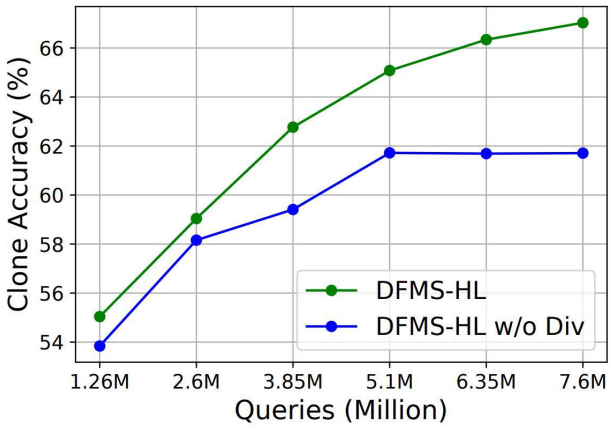

Figure 4: **Query Ablation:** Sensitivity Plot of clone model accuracy to number of queries. A significant boost of 6% in the clone model accuracy is evidenced after using class-diversity loss.

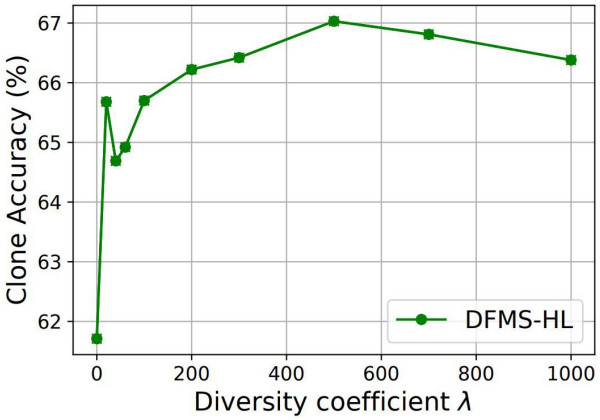

Figure 5: **Sensitivity Plot for Class-diversity:** Clone model accuracy increases with increase in diversity coefficient λ_{div}.

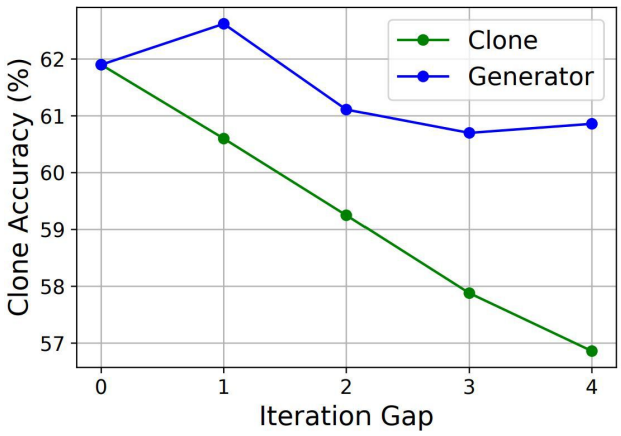

Figure 6: **Iteration Gap ablation:** Variation of clone model accuracy with varying gaps of training for clone and generator.

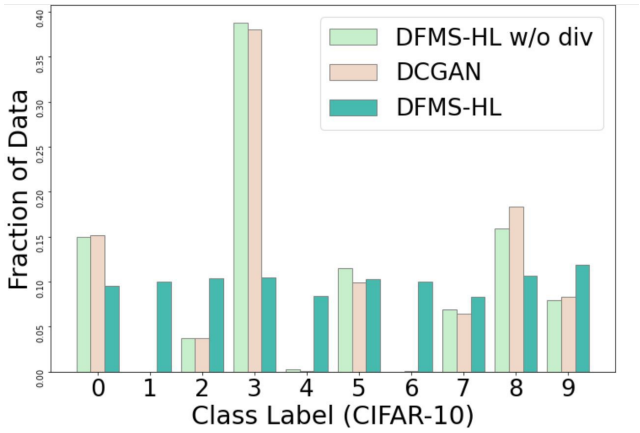

Figure 7: **Distribution of images over classes:** The images generated by DFMS-HL distribute evenly across all classes.

clone model is trained in every iteration. The results show that decreasing the iterations significantly impacts the clone accuracy. We perform the same ablation with the generator training gap g by increasing it from 0 to 4. We obtain a better accuracy for gap $g = 1$, where the generator is trained in alternate iterations. We use synthetic dataset as proxy data and CIFAR-10 as true dataset with 85 epochs of training for this ablation. We report our final results with g and t as 0.

Generation of Diverse Images: The DFMS-HL generator is initialised with a DCGAN generator at the start of the training process. As the training progresses, the generator learns a diverse distribution over the different classes of the victim model as shown in Fig. 7. The initial distribution of DCGAN looks skewed, with very few samples in classes 1, 4 and 6. We also plot the distribution of classes without the diversity loss, which also look skewed. From the plots, we observe that the class-diversity loss has a huge impact in making the class distribution uniform. We use synthetic data as proxy for this ablation with CIFAR-10 as the true dataset on AlexNet.

E.1 Impact of Synthetic Data

We tried two variants of the synthetic dataset. The first variant, "Large overlapping shapes" contains multiple overlapping shapes on a planar background. The second variant "Small overlapping shapes" contains multiple shapes of smaller size in an image. Each variant is shown in Fig 3. We report results obtained by using each of these datasets individually and both combined in Table-6. In

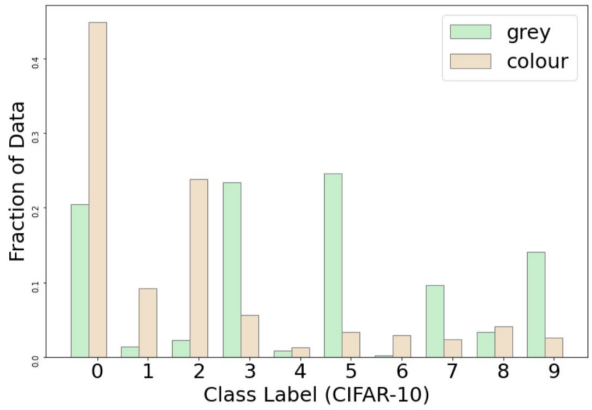

Figure 8: **Distribution of classes for grey vs colour images:** The grey synthetic images are more uniformly distributed across CIFAR-10 classes as compared to coloured images.

Table 6: **Impact of Synthetic Data**: Clone Model accuracy with different kinds of synthetic data images used, obtained on a ResNet-18 victim model of accuracy 93.65%, with ResNet-18 as the clone architecture.

Type of Synthetic Data	Clone Accuracy
Large overlapping shapes	80.34
Small overlaping shapes	56.30
Large + Small Combined	**85.92**

Table 7: **Impact of Synthetic Data**: Comparison for grey vs coloured images used as proxy data, with AlexNet as the victim model of accuracy 80.18% , trained on CIFAR-10, and AlexNet-half as the clone model.

Type of Synthetic Data	Clone Accuracy
Grey synthetic images	67.03
Coloured synthetic images	65.84

this experiment, we use grey scale images for training. After combining the two datasets, we obtain a competent accuracy of 85.92%.

We use grey-scale and coloured images individually from the synthetic dataset and observe its impact on the clone model accuracy with an AlexNet victim network. We find that the grey images are well-distributed across multiple classes as shown in Fig. 8. This makes grey images a better choice for initialization. In our method, we train a clone model with a mix of images from the proxy data and the generator to obtain a good initialisation. From our experiments, we observe that the initial clone model trained with grey-scale synthetic data achieves an accuracy of 44.57% and the one trained with coloured images has an accuracy of 37.31%. This shows that grey-scale images lead to a better initialization for the clone model. Hence, we reported the final results of our method using grey-scale synthetic images. We also report the results of using the grey-scale and colour images individually for training in Table 7 and observe that the final clone accuracy in both cases are comparable.

E.2 Hyperparameter tuning

The diversity loss plays a crucial role in ensuring that the distribution of images from the generator is class-balanced. The loss formulation of the generator with the class-diversity loss is shown below:

$$\mathcal{L}_G = \mathcal{L}_{adv,fake} + \lambda_{div} \cdot \mathcal{L}_{class\_div} \tag{17}$$

We show the impact of varying the class-diversity loss coefficient λ_{div} in Table 8. The true dataset is CIFAR-10 and the proxy dataset is 10 random classes from CIFAR-100. We use AlexNet as the victim architecture and train an AlexNet-half as the clone model for 500 epochs. We observe that as we increase the diversity loss coefficient, the clone model accuracy increases and reaches the maximum accuracy of 69.66% at λ_{div}=500. We note that the proposed method is not sensitive to minor variations in the hyperparameter λ_{div}.

Table 8: **Impact of class-diversity loss coefficient** λ_{div}: Performance (%) of the clone model on CIFAR-10 dataset trained using 10 random classes of CIFAR-100 as proxy, across variation in λ_{div}. The architecture of victim model is Alexnet and architecture of clone model is AlexNet-half. The proposed method is not sensitive to minor variations in λ_{div}.

Diversity Loss Coefficient	Clone Accuracy
100	69.29
200	69.59
300	69.42
500	**69.66**
700	69.54
1000	69.13

Table 9: **Impact of clone architecture on clone accuracy**: Clone Accuracy improves with a deeper CNN network

Clone Model Architecture	Clone Accuracy
ResNet-18	**83.37**
AlexNet	79.37
AlexNet_half	62.64
VGG-11	74.59
VGG-19	78.85
GoogleNet	84.50

E.3 Impact of clone architecture

In a practical scenario of Model Stealing, the architecture of the victim model is unknown to the attacker. Hence, we aim to stage a successful attack in a completely black-box condition. To evaluate the effectiveness of the attack in different scenarios, we perform an ablation experiment to see if the choice of the clone model architecture impacts the success of the attack. The clone model achieves a high accuracy of 83.37% using 10 random classes of CIFAR-100 when the same ResNet-18 architecture is used for both the victim and the clone. However, using a deeper CNN model such as GoogleNet gives a boost to the clone accuracy as shown in Table 9. We get lower clone accuracy for shallower networks such as AlexNet-half and VGG-11. Hence, we observe that it is beneficial for an adversary to use a deeper CNN architecture for capturing complex features from the victim model using proxy data.

E.4 Impact of Discriminator

The discriminator is an essential component of our approach. Across training epochs, the discriminator learns to differentiate between proxy data and fake images produced by the generator. We conduct an ablation experiment by disabling the discriminator updates. We use CIFAR-10 as the

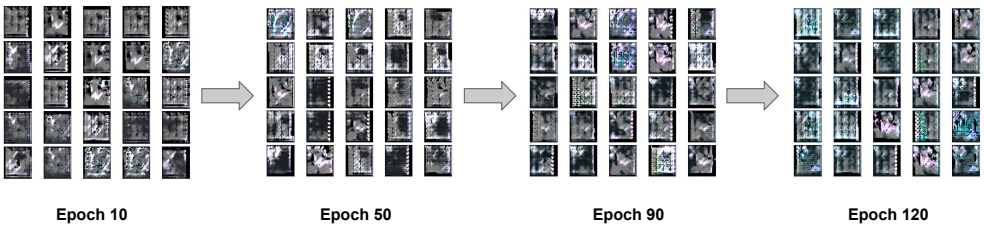

Figure 9: **Output of DFMS-HL after disabling the discriminator.** The images converge to degenerate cases after few epochs of training. Synthetic data is used as proxy data with an AlexNet victim model trained on CIFAR-10 and clone model as AlexNet-half.

Table 10: **Impact of L1 loss formulation on DFMS-SL (Soft-Label Setting)**: Clone Model accuracy increases by 3% after using L1-loss as compared to standard KL-divergence loss. Synthetic data is used as proxy for a ResNet-34 victim model trained on CIFAR-10 and ResNet-18 used as Clone model.

Method	**Teacher Acc**	Synthetic
DFME	95.5	88.10
DFMS-SL(L1 loss)	95.5	91.24
DFMS-SL(KL-div loss)	95.5	88.40

Table 11: **SVHN as Proxy Data ablation**: DFMS-HL achieves an accuracy of 84.83% using SVHN as Proxy data for a ResNet-34 victim model trained on CIFAR-10. ResNet-18 used as Clone architecture.

Method	Synthetic	CIFAR-100 (40C)	CIFAR-100 (10C)	SVHN
DFME	88.10	88.10	88.10	88.10
DFMS-HL (Ours)	84.51	92.06	85.53	84.83

true dataset and synthetic data as the proxy dataset for this experiment. For Alexnet as victim model and AlexNet-Half as clone model, DFMS-HL attains an accuracy of 67.03%. After disabling the discriminator, the clone accuracy drops to 57.06% and the images look degenerate as shown in Fig. 9. Hence, the discriminator also plays a crucial role in maintaining the distribution of images.

E.5 Impact of L1 loss in DFMS-SL

Prior works on Knowledge Distillation Hinton et al. (2015); Lopes et al. (2017); Nayak et al. (2019) train a student model using a KL-divergence loss between the student and teacher predictions. Let $\mathcal{V}_i(x)$ and $\mathcal{C}_i(x)$ be the output of class i (out of K classes) of the victim and clone models respectively. The KL divergence loss is written as follows,

$$\mathcal{L}_{KL} = \sum_{i=0}^{K} \mathcal{V}_i(x) log \left[\frac{\mathcal{V}_i(x)}{\mathcal{C}_i(x)} \right] \tag{18}$$

The DFME approach Truong et al. (2021) used an L1 loss formulation where they consider the L1 difference between the logits of the clone and the victim model. The logits are estimated by first taking log, then subtracting the mean of the predictions from it. The loss formulation is written as follows,

$$\mathcal{L}_{l1} = \sum_{i=0}^{K} \mid \mathcal{V}_i^{logits}(x) - \mathcal{C}_i^{logits}(x) \mid \tag{19}$$

where,

$$\mathcal{V}_i^{logits}(x) = log\mathcal{V}_i(x) - \frac{1}{K} \sum_{j=1}^{K} log\mathcal{V}_j(x) \tag{20}$$

We evaluate our approach in the soft-label setting with the two loss functions of L1 loss and KL-divergence loss as shown in Table 10. We observe an improvement in the clone accuracy using synthetic data by 3% by using L1 loss for distillation.

E.6 Using unrelated data as the Proxy Dataset

The amount of relatedness between the proxy data and true data is an important factor that influences the success of model stealing. We perform an ablation study using SVHN as the proxy dataset to steal a model originally trained on CIFAR-10. Since SVHN is a completely unrelated to CIFAR-10, it is indeed a difficult setting. Our method DFMS-HL attains a clone accuracy of 84.83% in this setting. This shows our attack is strong enough to work across a wide range of unrelated proxy datasets.

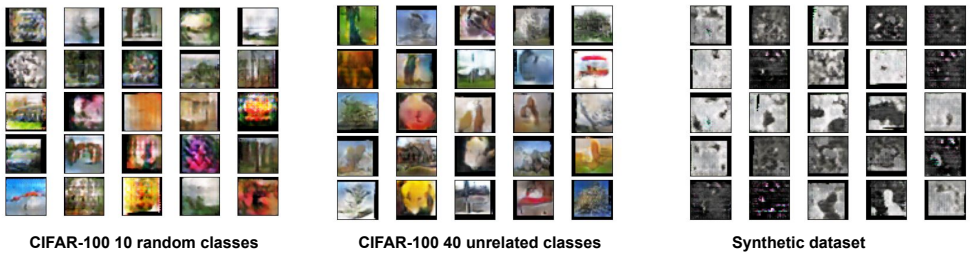

Figure 10: **DFMS-HL generator images.** The images generated by DFMS-HL generator for CIFAR-100 10 random classes, 40 unrelated classes and synthetic data as proxy for an AlexNet victim model of accuracy 80.18% trained on CIFAR-10 and clone model as AlexNet-half.

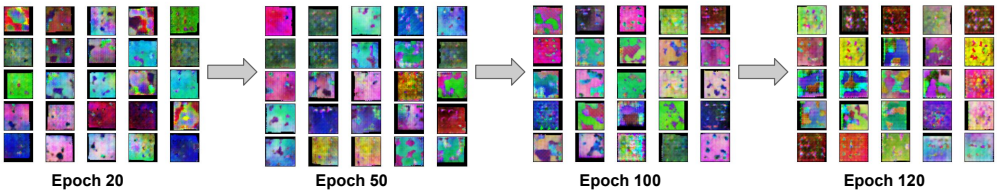

Figure 11: **DFMS-HL generator images.** The images generated by DFMS-HL generator using synthetic colour dataset as proxy for an AlexNet victim model of accuracy 80.18% trained on CIFAR-10 and clone model as AlexNet-half.

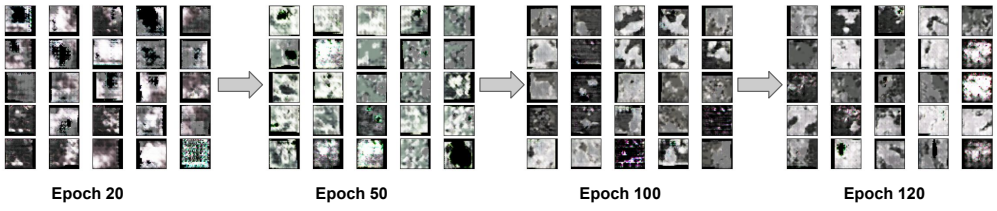

Figure 12: **DFMS-HL generator images.** The images generated by DFMS-HL generator using grey-scale synthetic images as proxy for an AlexNet victim model of accuracy 80.18% trained on CIFAR-10 and clone model as AlexNet-half.

F GAN GENERATED IMAGES

The images generated from the DFMS-HL GAN are shown in Fig. 10, 11 and 12. Initially, the generator starts generating images which closely resemble the proxy data. In the synthetic data experiments (Fig.11 and 12), as the training progresses, we observe that the shapes start merging with each other and start looking more continuous in nature. This makes the image look close to real images which have an object in front of a background. This shows that the generator starts capturing properties of the true training data distribution, as they look more intuitive than the original synthetic images. This helps the clone model learn intrinsic properties of the victim's training data.

G LIMITATIONS AND FUTURE DIRECTIONS

One of the crucial factors of a successful model stealing attack is its query budget. Our approach has reduced the number of queries required to 8 million, which is $\sim 500\times$ lesser than the query budget used by past methods of model stealing and knowledge distillation. We believe that reducing the query budget further would be an interesting area for future research. Another limiting factor for an adversary is the lack of relevant training data. Our approach addresses this limitation to quite an extent, as we showcase promising results in a limited data scenario by just using synthetic images. We believe that our approach would pave the way to address these limitations and develop stronger attacks and defenses in the area of hard-label model stealing.