VidModEx: Interpretable and Efficient Black Box Model Extraction for High-Dimensional Spaces

Somnath Sendhil Kumar*†, Yuvaraj Govindarajulu‡, Pavan Kulkarni‡, Manojkumar Parmar‡

† Microsoft Research, India.

‡ AIShield, Bosch Global Software Technologies, Bangalore, India.

Abstract

In the domain of black-box model extraction, conventional methods reliant on soft labels or surrogate datasets struggle with scaling to high-dimensional input spaces and managing the complexity of an extensive array of interrelated classes. In this work, we present a novel approach that utilizes SHAP (SHapley Additive exPlanations) to enhance synthetic data generation. SHAP quantifies the individual contributions of each input feature towards the victim model's output, facilitating the optimization of an energy-based GAN towards a desirable output. This method significantly boosts performance, achieving a 16.45% increase in the accuracy of image classification models and extending to video classification models with an average improvement of 26.11% and a maximum of 33.36% on challenging datasets such as UCF11, UCF101, Kinetics 400, Kinetics 600, and Something-Something V2. We further demonstrate the effectiveness and practical utility of our method under various scenarios, including the availability of top-k prediction probabilities, top-k prediction labels, and top-1 labels.

Figure 1. Activation Atlas for SHAP (Eq. (6)) objective

1. Introduction

With the rise in MLaaS (Machine Learning as a Service), which performs tasks from minute levels [2, 14, 37] to multitasking across domains [11, 42]; There has been a significant increase in model performance, correlating with their size and the ability to accommodate large input spaces. However, these advancements also incentivise malicious parties to exploit vulnerabilities [45], particularly through adversarial attacks [61], privacy leaks [20], and model stealing [41]. In this work, we focus on model extraction attacks, which aim to replicate the target model with black-box access to the model and potentially the target data. Previous model extraction attacks [38, 47, 54, 55] have predominantly targeted small datasets such as MNIST and CIFAR, and at the best case scenario have achieved acceptable extraction accuracy on CIFAR-100, which are minuscule compared to current datasets and robust models. Although there are studies scaling to large real-world models like [6], these are specifically crafted for a target architecture or task, making a generalized approach challenging.

On the contrary, some methods employ surrogate datasets [18, 47, 54, 57, 60] to train a substitute model, providing a prior about the target dataset. However, studies finding a balance [18] between surrogate and target datasets are limited in terms of scalability. With the affordable cost of hardware and increased services offering model fine-tuning for user data [21], relying on surrogate datasets presents challenges in selecting the appropriate dataset. While every task in model extraction comes with its nuances, ranging from classification problems that might use soft labels or hard labels to top-k predictions/labels or top-1 prediction/label [5, 19, 25], a generalized base ap-

*Work partially done while an intern at AIShield.

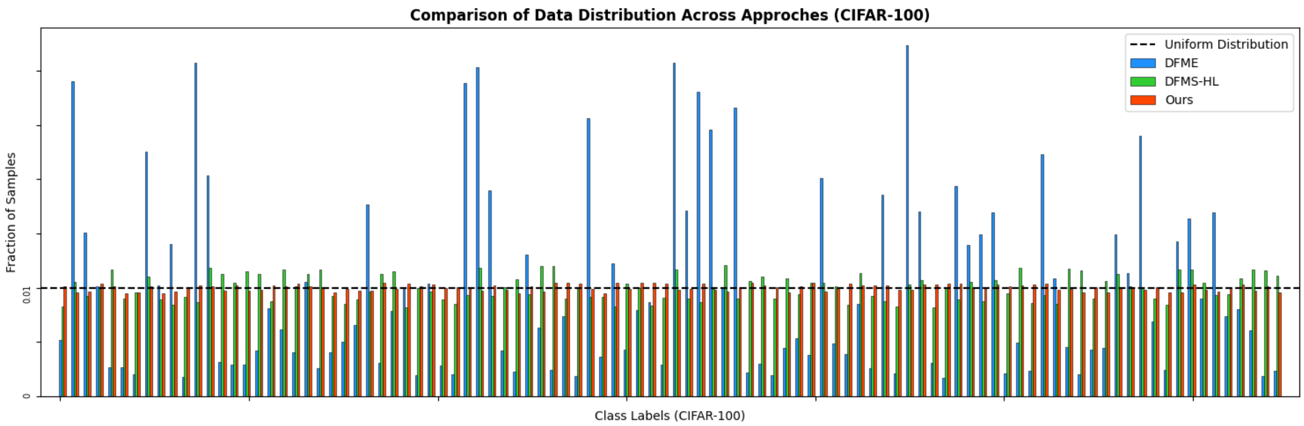


Figure 2. Distribution based on Victim model prediction on generated samples for CIFAR 100

proach can promote the development of more efficient and large-scale attacks. In this work, we limit ourselves to single-label Vision Classifiers, but we do not exploit any specific architectural constraints or any discrepancy present in these tasks, thus maintaining an approach that is easily adaptable to other domains.

We employ SHAP [32], an InterpretableAI Algorithm to act as a guide to the Generator improving performance and also supplementing as a weak prior to Zeroth Order Gradient approximation [23], which is employed in most of the Model extraction approaches. **SHAP** Stands for **SH**apley **A**dditive ex**P**lanations, It calculates feature importance indicating the contributing of sample towards a black box models output in Eq. (1), This output can be from a regression, classification or any other open-ended model. We introduce a differentiable pipeline that utilizes SHAP values to optimize the generator for custom objectives. Within this pipeline, we optimize the generation for each class by our conditional generator, which enhances the class distribution as evidenced in Fig. 2. Furthermore, the custom objective enhances sample quality, as demonstrated by Activation Atlases [9] in Fig. 1 is superior to common objectives employed in [23, 54] in Fig. D.III

$$f(x) = \mathbb{E}[f(.)] + \sum_{i=1}^{M} \phi_i * x'_i \tag{1}$$

In this work, our key contributions can be enumerated as below:

- We introduce an efficient class-targeting approach for model extraction, significantly enhancing the efficacy of the substitute model across all classes.
- We devise a query-efficient feedback mechanism to train a generator which facilitates the pipeline scale to higher dimensional spaces. We demonstrate this through a comparative analysis against prior works while being the first to extract Video Classification models with an acceptable accuracy and query budget.
- Our algorithm's versatility is demonstrated across various settings, including Greybox, BlackBox with all soft labels, BlackBox with top-k & top-1 soft labels, BlackBox with top-k & top-1 hard labels

We explore the limitations of this approach and offer considerations for using this strategy effectively. To advance further research in model extraction attacks, we've made our source code publicly available[1].

2. Related Work

We have outlined the motivation for this work in the introduction; this section will review the seminal literature related to each component or domain critical to our study.

2.1. Model Extraction Attacks

Previous efforts in model extraction in Softlabel settings have focused on computing approximate gradients for backpropagation objectives [5, 23, 38, 54]. Works such as [47] extensively evaluate pipelines for Hardlabel settings, establishing a precedent for real-world model extraction. These approaches, utilizing varied mechanisms for training the Generator, share a common goal: to optimize the divergence between the Victim and Substitute acting as a discriminator. However, the high query costs required to approximate gradients for a single sample using Zeroth Order gradient approximation often limit their performance. Efforts to train an efficient Generator using an Evolutionary Algorithm [4, 28, 43, 44] have demonstrated significantly lower extraction accuracy compared to earlier methods [41]. Meanwhile, initiatives like [56] focus on generating class-specific samples using minimum decision boundaries, which is superior to other approaches based on sam-

[1] https://github.com/vidmodex/vidmodex

ple efficiency to train the Substitute model. Yet, computing these samples requires a high number of Victim Model queries, rendering it impractical in real-world scenarios due to extensive querying. Drawing from these findings, we aim to address this trade-off by developing an auxiliary objective based on SHAP for the generator that is query-efficient and improves the fidelity of the generated samples, enabling richer extraction of the Victim model.

2.2. Interpretable AI for GAN, Model Extraction

Research on using Interpretable AI algorithms to train GANs is limited [39] because, although explanations help in human interpretation, they have less information than gradients from the target network and discriminator. However, these methods show promise for black box model applications, particularly in model extraction [38, 40, 55, 58]. Both [40, 58] focus on mimicking explanations from the victim model, which does not perforce improve extraction accuracy, and [38] relies on direct gradients, which defeats the purpose. In [55], GradCam [48] is used for sample augmentation by leveraging saliency maps from the substitute model to refine the loss function. This method is limited because GradCam depends on the substitute model's gradients, leading to a noisy and unstable training process. Despite demonstrating stability in a limited study with predefined surrogate dataset images, the scalability to diverse real-world objectives remains dubious. We iterate on this by computing SHAP [49], which are gradient-free and can be directly computed on the victim model within a specified `max_evals` budget for each sample. Acquiring SHAP values from the victim model are costly, but we address this by learning to estimate SHAP values within an Energy GAN framework [59] since learning SHAP values [22] is more feasible than predicting gradients for the victim model.

2.3. Surrogate Dataset and Settings

In this subsection, we examine the use of surrogate datasets in prior research and the settings employed for model deployment. Many studies, including [28, 47, 54, 55], have utilized surrogate datasets with varied methodologies for sample selection. These strategies accelerate the model extraction process but necessitate an understanding of the victim model's data distribution, complicating scalability due to the adverse effects of selecting a poor surrogate dataset as noted by Truong et al. [53]. With the growth of MLaaS platforms, such as [1, 15, 35, 36], entities can now deploy models with specific settings including Top-1 and Top-k class labels, and prediction probabilities for the Top-1 or Top-k classes. Our analysis adopts these settings and extends to prediction probabilities for all classes to align with previous research. We offer a framework for using a surrogate dataset under a grey box model extraction attack, detailing the surrogate dataset specifics in Sec. 5.

3. Preliminary

In this section, we introduce SHAP, an Additive explanation method using Shapley values, focusing on how it helps define objectives. We use the Partition Explainer [49], which calculates Shapley values recursively across a feature hierarchy, forming feature coalitions that result in Owen values from game theory [31], detailed in Appendix A.1. Following SHAP's basic principles, we start with the additive property shown in Eq. (1). In this formula, f is the target black-box model, M is the input space size, $\mathbb{E}[f(.)]$ the expected value of f over a uniform distribution, and ϕ represents the Shapley value for the sample x, noted as $\phi(f, x)$. x_i denotes the i^{th} feature of x, and the mapping between x' and x is defined by $x = h(x')$ as outlined in [32, Section 2], with $x' \in [0, 1]^M$ standardized for the algorithms. $h(x')$ is a explainer specific mapping function to reconstruct x from a standardized input space x'.

To generalize across the scenarios outlined in Section 2.3, we define our black-box victim model using Equation (2), which ensures consistent outputs across any top-k prediction setting. The terms $topk\_probs$ and $topk\_indices$ represent the probability values and corresponding indices for top-k predictions, respectively. The model produces a column vector of dimension $[0, 1]^{num\_classes}$, displaying a softmax output for a single class prediction, aligning well with the intended application within the SHAP framework.

$$f_{sl} = \begin{cases} topk\_probs[i] & \text{if } i \in \text{topk\_indices}, \\ \dfrac{1 - sum(topk\_probs)}{num\_classes - k} & \text{otherwise.} \end{cases} \tag{2}$$

$$f_{hl} = \begin{cases} 1/\text{k} & \text{if } i \in topk\_labels, \\ 0. & \text{otherwise} \end{cases} \tag{3}$$

For hard labels, we adhere to the definition in Eq. (3), which yields a binary output from the target black-box model. We also explore how, although this approach provides less information than soft labels, it remains sufficiently informative for computing Shapley values.

With the definition Eq. (1), an approximation under the local accuracy property given in [32, Section 3] and choosing either function from Eq. (2) or Eq. (3), we derive Eq. (4a). Representing the variables in the equation as vectors, reformulating $\phi = (\phi_1, \ldots, \phi_i, \ldots)^\top$ as a column vector and $x' = (x'_1, \ldots, x'_i, \ldots)^\top$ as a column vector and conditioning on specific class id c, we refine it to Eq. (4b).

$$f(x) = \mathbb{E}[f(.)] + \sum_{i=0}^{M} \phi_i * x'_i \tag{4a}$$

$$f(x|c) = \mathbb{E}[f(.|c)] + \phi(f(.|c), x)^\top * x' \tag{4b}$$

Using Eq. (4), we set our objective to enhance sample x effectiveness by maximizing the class probability of the target model $f(.|c)$, leading to Eq. (5). Since $\mathbb{E}[f(.|c)]$ does not depend on x, we simplify further by substituting x', making the objective linearly proportional to the variable of interest. We use vector $j = (1, 1, \ldots, 1)^\top$ of size M for $x' \in [0, 1]^M$ to define the bounds $0 \leq \phi^\top * x' \leq \phi^\top * j$ or $0 \geq \phi^\top * x' \geq \phi^\top * j$, depending on the sign of $\phi^\top * x'$. Using $\phi^\top * j$ simplifies the objective but introduces some inaccuracy, focusing on feature contribution over magnitude. This approach also addresses the issue of exploding gradients during training, culminating in the objective defined in Eq. (6).

$$\underset{x}{arg\,max}\ f(x|c) = \underset{x}{arg\,max}\ \ \phi(f(.|c), x)^\top * x' \tag{5}$$

$$ClassObj = \underset{x}{arg\,max}\ \ \phi(f(.|c), x)^\top * j \tag{6}$$

Building on previous work on Shap value computation [49], we use the Partition Explainer $\mathbf{E}$ to approximate the Shap values ϕ. The function $\mathbf{E}$ takes as inputs the target model $\mathcal{V}$, the sample x, and the maximum number of model evaluations $\max\_eval$. The approximate Shap value s for the input is given by Eq. (7).

$$s = \mathbf{E}(\mathcal{V}, x, max\_eval) \tag{7}$$

Alongside the previously defined objective, we employ several crucial parameters that influence the accuracy of the approximations used in Eq. (1). One key parameter is `max_evals`: The Partition Explainer efficiently distributes Shapley value computations across a feature hierarchy, significantly reducing inference costs in high-dimensional settings by avoiding $M!$ inferences and requiring only `max_evals`. Another less influential hyperparameter is `masker`, defaulting to Gaussian Blur with a kernel size of 3. Both parameters are tailored to the complexity and nuances of the target model.

4. Approach

The overall attack setup is well outlined by previous works [54], [47], with $\mathcal{V}$ the Victim black box model, $\mathcal{S}$ a substitute model and A generator $\mathcal{G}$ which is responsible for crafting input samples. While our objective is to learn $\mathcal{S}$ that closely mimics the $\mathcal{V}$. We employ KL divergence [54] for soft label setting given in Eq. (8a), and employ CrossEntropy Loss [47] for hard label setting given in Eq. (8b) to optimize $\mathcal{S}$. To optimize $\mathcal{G}$, we use an adversarial loss to increase the divergence between Student and victim model [47, 54] which is given by Eq. (9). As we use Conditional Generator instead, we also specify c_T Target class index to generate samples for a particular class.

$$\mathcal{L}_{sl}(x) = \sum_{i\, \in\, topk\_indices} \mathcal{V}(x|i) \log \frac{\mathcal{V}(x|i)}{\mathcal{S}(x|i)} \tag{8a}$$

$$\mathcal{L}_{hl}(x) = -\sum_{i\, \in\, topk\_indices} \mathcal{V}(x|i) * \log(\mathcal{S}(x|i)) \tag{8b}$$

$$\left.\begin{array}{r} z \sim \mathcal{N}(0, 1); \\ x = G(z, c_T); \end{array}\right| \implies \underset{\theta_G}{argmax}\ \underset{\theta_S}{argmin}\ \mathcal{L}(x) \tag{9}$$

We complement our setup with the ClassWise Objective from Eq. (6). Since the ϕ value from the explainer isn't differentiable, we use an estimator $\mathcal{P}(s|x, c_T)$ that predicts SHAP values(ϕ) based on the input(x) and targeted class index(c_T). $\mathcal{P}$, a conditional UNet, ensures predicted SHAP values(ϕ), is in a normal distribution and consistent with input's shape, aiding calculation of the probability over ϕ. As we only obtain an approximate value of ϕ from $\mathbf{E}$, we use it as the ground truth s_{gt}. Hence s_{gt} is used to train a differentiable and computationally efficient method to estimate Shap value ($\mathcal{P}$) similar to Jethani et al. [22]. To train $\mathcal{P}$, we optimize the Mean Absolute Error between $\mathcal{P}$'s SHAP output and the explainer's values as per Eq. (11). We apply $\mathcal{P}(s_{gt}|x, c)$ as a mask to minimize prediction errors when s_{gt} is known; otherwise, we revert to the initial objective in Eq. (10).

$$\begin{aligned} ClassObj &= \underset{x}{argmax}\ \ \sum \mathbb{E}[\mathcal{P}(s|x, c)] \\ &= \underset{x}{argmax} \sum \mathbb{E}[\mathcal{P}(s|x, c)] \odot \mathcal{P}(s_{gt}|x, c) \end{aligned} \tag{10}$$

$$\mathcal{L}_{\mathcal{P}} = \sum |s_{gt} - \hat{s}|\ \text{, where } \hat{s} \sim \mathcal{P}(x, c) \tag{11}$$

Algorithm 1: VidModEx: Data-Free Model Extraction with SHAP and Class-Wise Objectives

Input: Victim model $\mathcal{V}$, Clone model $\mathcal{S}$, Generator $\mathcal{G}$, explainer $\mathbf{E}$, Shap estimator $\mathcal{P}$, Query budget N_Q, Generator iterations n_G, Clone model iterations n_S, Learning rates η_G, η_S, η_P, Top-k labels k, Target classes C, Initial max evaluations `max_eval`, Decay threshold $threshold$, Decay schedule $D_S = \{d_1, d_2, \ldots, d_k\}$

Output: Trained Clone model $\mathcal{S}$ and Generator $\mathcal{G}$

while $N_Q > 0$ **do**
  foreach $c_T \in C$ **do**
    for $i = 1$ **to** n_G **do**
      Sample $z \sim \mathcal{N}(0, 1)$;
      $x = \mathcal{G}(z, c_T)$;
      if $max\_eval \geq threshold$ **then**
        $s_{gt} = \mathbf{E}(\mathcal{V}, x, \texttt{max\_eval})$;
        $\hat{s} \sim \mathcal{P}(x, c_T)$;
        $\theta_P \leftarrow \theta_P - \eta_P \nabla_{\theta_P} \mathcal{L}_D(s_{gt}, \hat{s})$;
      $\hat{s} \sim \mathcal{P}(x, c_T)$;
      $\theta_G \leftarrow \theta_G - \eta_G \nabla_{\theta_G} \mathcal{L}_G(\hat{s}, x, c_T)$;
    for $j = 1$ **to** n_S **do**
      Sample $z \sim \mathcal{N}(0, 1)$;
      $x = \mathcal{G}(z, c_T)$;
      $\theta_S \leftarrow \theta_S - \eta_S \nabla_{\theta_S} \mathcal{L}_{hl \text{ or } sl}(\mathcal{V}, \mathcal{S}, x)$;
    $N_Q \leftarrow N_Q - (n_S + n_G * (1 + \texttt{max\_eval}))$;
    if $N_Q \in D_S$ **then**
      $\texttt{max\_eval} \leftarrow \frac{\texttt{max\_eval}}{2}$;

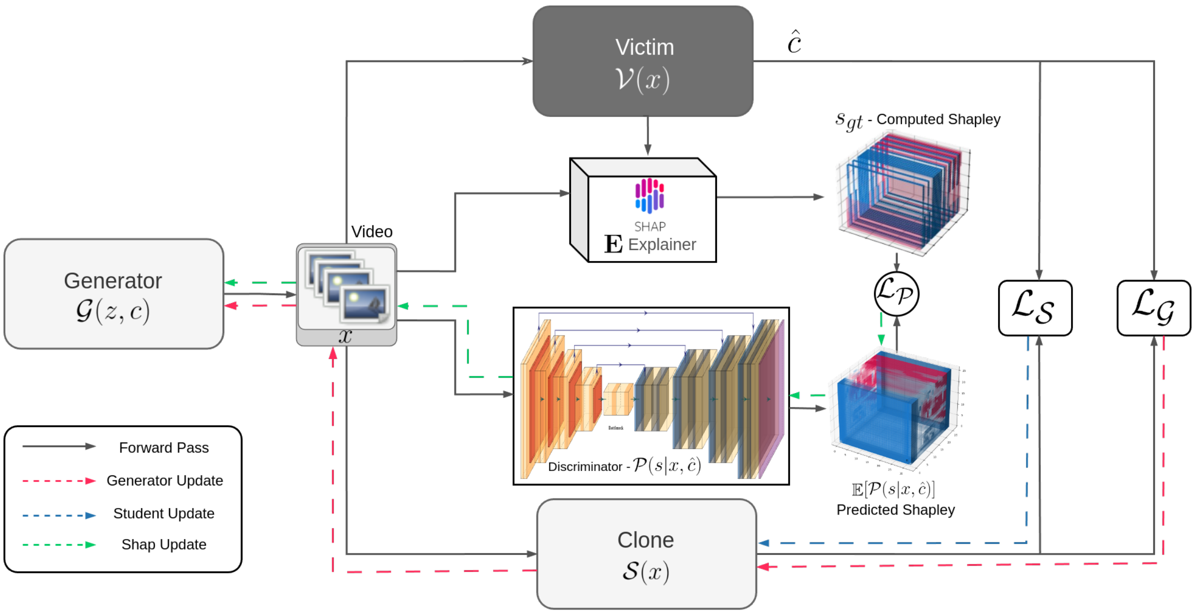


Figure 3. Model extraction diagram with additional objectives and SHAP explainers

(a) x - Original input (b) s_{gt} - SHAP GT (c) $\mathbb{E}[\hat{s}]$ - SHAP Pred (d) $\mathcal{P}(s_{gt})$ - P Mask (e) SHAP *Energy*

Figure 4. Shap values and visualization at each stage of the Pipeline

The complete pipeline is illustrated in Fig. 3 and detailed in Algorithm 1, where the shap estimator $\mathcal{P}$ operates akin to an energy-based discriminator, as detailed in [59]. Unlike typical adversarial settings, $\mathcal{P}$ focuses on accurately estimating SHAP values for generated samples, while $\mathcal{G}$ optimizes these samples to enhance their SHAP values. Consequently, $\mathcal{P}$ is termed a discriminator in this paper, enabling the generator to create rich and class-balanced samples. Additionally, the probabilistic discriminator incorporates a mask $\mathcal{P}(s_{gt}|x, c)$ to exclude any out-of-distribution signals or noise during training. SHAP values are normalized between $[-1, 1]$ to account for their variability from 1×10^{-8} to 1×10^{-11} across different datasets and model scenarios like images and videos.

Fig. 4 presents visualizations that illustrate data at each pipeline stage. Fig. 4a shows the initial input to the victim model, using a substitute image to simplify subsequent image interpretation. Fig. 4b displays the SHAP value computed with the partition explainer. Fig. 4c depicts the expected value μ of the discriminator, denoted as $\mathbb{E}[\mathcal{P}(s|x, c)]$. Fig. 4d shows the probability mask stabilizing the initial training phase, computing the probability that the expected output s_{gt} aligns with the predicted distribution, thus assessing prediction accuracy relative to the ground truth. Fig. 4e illustrates the final objective used to train the generator in an energy gan-like architecture, as specified in Eq. (10). To further concretize the stability of the joint training of estimator $\mathcal{P}$ and Clone model $\mathcal{S}$, we conduct experiments in Sec. 5.2.

5. Experiments

This section assesses our Vidmodex approach in diverse settings, outlined in Sec. 2.3, using image and video models across datasets like MNIST [13], CIFAR10, CIFAR100 [26], Caltech101 [27], Caltech256 [17], ImageNet1K [12] for images, and UCF11 [29], UCF101 [51], Kinetics 400 [24], Kinetics 600 [7], Something-Something v2 [16] for videos. These tests evaluate increasing class complexities, emphasizing high-resolution datasets to show efficiency in large search spaces. We benchmark primarily against DFME [54], DFMS-HL [47], and include results from ZSDB3KD [56], MAZE [23], KnockoffNets [43], and BlackBox Dissector [55], opting not to replicate other studies since our methods have surpassed them previously. We also assess `max_evals`' impact on the extraction process and learning within the discriminator in

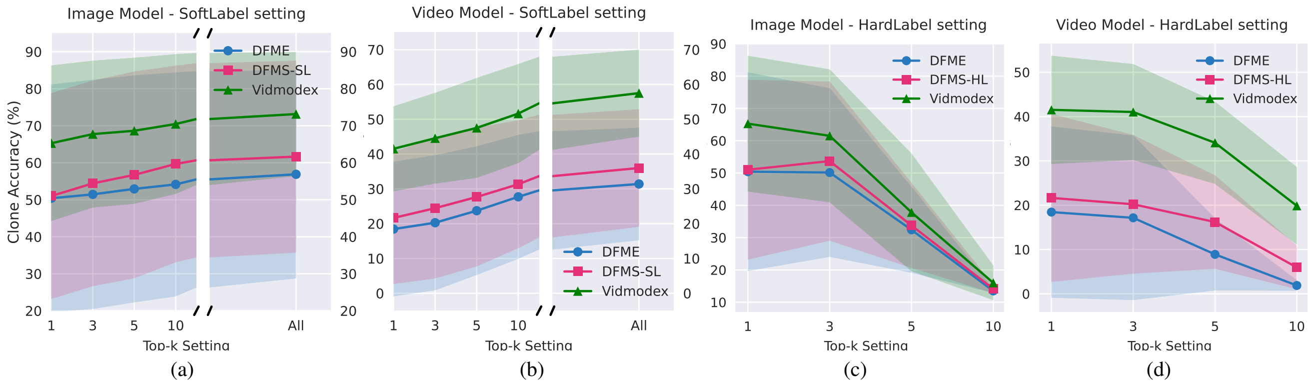


Figure 5. Plots of the extraction accuracy across different K, for both Softlabel and Hardlabel setting

Sec. 5.2, determining the optimal configuration. An ablation study on `top_k` settings explores performance variations. Our focus is on black box model extraction, but we also address the use of a surrogate dataset (grey box access) and its effects. Results with `top_k` label availability are analyzed to confirm generalization. Our detailed qualitative analysis in Appendix Appendix D further supports our empirical findings.

Experimental setup

DFME and DFMS-HL are integrated into our pipeline as configurable approaches, with outlines and scripts provided for result reproduction in our code base. Our experiments ran on 2 nodes with 8 x H100 GPUs (80GB), Intel(R) Xeon(R) Platinum 8480C CPUs (96-cores 4 GHz), and 1.8 TB of RAM, along with a setup featuring 4 x A100 GPUs (80GB), AMD EPYC 7V13 (64-core 4.8 GHz), and 867 GB RAM. Additionally, tested our scripts on a modest setup with V100 GPUs (32 GB), ensuring broad reproducibility and ease of development. The high-demand experiments, such as those involving Kin400, Kin600, and Something-Something-v2, necessitate more robust hardware configurations.

5.1. Results

We present our results for black-box extraction results in Sec. 5.1.1, and we further analyze the influence of top-k on both settings; Also present Greybox extraction results. A standard practice in previous model extraction literature is to use the same architecture for both the Victim and Clone models to reduce variance in results that may arise from architectural disparity.

5.1.1. BlackBox Extraction

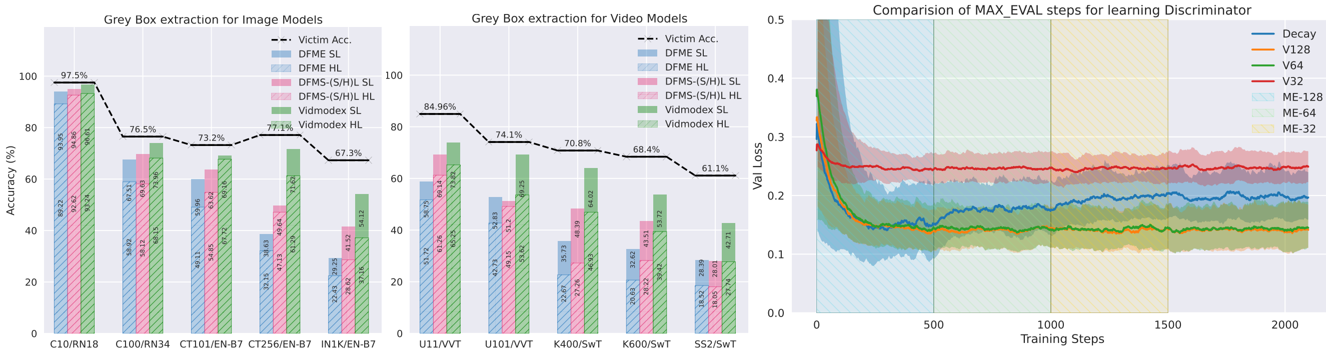
Our blackbox extraction initially focuses on the SoftLabel Setting, using class probabilities from the victim model, similar to previous studies like [56], [23], [43]. As shown in Table 1, we report accuracies from these methods and our reproduced results from [47] and [54]. To enable a reproducible comparison, we detail the training epochs needed to replicate the victim models, addressing the lack of standardized or pre-trained weights in prior research. We train the Target victim architecture from scratch on the dataset, with all configurations, including seeds, documented in our repository. Both the clone and the victim model use the same architecture to prevent bias from architectural differences.

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.%	Clone Acc.%	Query Budget
DFME [54]	MN‡ / RN-18†	500	99.7	92.5	4M
	C10‡ / RN-18†	1500	97.5	87.32	10M
	C100‡ / RN-34†	3500	76.5	62.15	25M
	CT101‡ / EN-B7†	8000	73.2	53.56	70M
	CT256‡ / EN-B7†	10500	77.1	32.52	100M
	IN1K‡ / EN-B7†	15000	67.3	13.23	120M
DFMS-SL [47]	MN‡ / RN-18†	500	99.7	**95.1**	4M
	C10‡ / RN-18†	1500	97.5	91.22	10M
	C100‡ / RN-34†	3500	76.5	65.04	25M
	CT101‡ / EN-B7†	8000	73.2	56.46	70M
	CT256‡ / EN-B7†	10500	77.1	38.54	100M
	IN1K‡ / EN-B7†	15000	67.3	23.56	120M
Vidmodex	MN‡ / RN-18†	500	99.7	94.6	4M
	C10‡ / RN-18†	1500	97.5	**94.9**	10M
	C100‡ / RN-34†	3500	76.5	**69.52**	25M
	CT101‡ / EN-B7†	8000	73.2	**68.14**	70M
	CT256‡ / EN-B7†	10500	77.1	**64.25**	100M
	IN1K‡ / EN-B7†	15000	67.3	**48.54**	120M
ZSDB3KD [56]	MN‡ / LN-5†	-	99.33	**96.54**	100M
	C10‡ / RN-18†	-	82.5	59.46	400M
MAZE [23]	C10‡ / RN-18†	-	92.26	45.60	30M
	C100‡ / RN-34†	-	82.5	37.20	80M
KnockOff Nets [43]	C10‡ / RN-18†	-	91.56	74.44	8M
	CT256‡ / RN-34†	-	78.4	55.28	8M

†Model Architecture RN-18: ResNet18; RN-34: ResNet34; EN-B7: EfficientNet-B7; LN-5: LeNet-5
‡Dataset MN: MNIST; C10: CIFAR10; C100: CIFAR100; CT101: Caltech101; CT256: Caltech256; IN1K: ImageNet1K

Table 1. Comparision of Blackbox Extraction Techniques on Image Models

We detail the Query Budget, presenting reported values or estimates from algorithms like [56]. Our method generally outperforms others, except on MNIST where it matches [47] and trails [56]. Notably, it is 25× more efficient than [56] in Query Budget



(a) Image Model (b) Video Model (c) Variance based on `max_evals`

Figure 6. (a) and (b) show GreyBox extraction accuracies; (c) illustrates variation in Discriminator training.

use. Using a uniform Query Budget, we exceed the performance of [54] and [47]. We note that extraction accuracy declines with increased dataset difficulty, linked to higher resolution and more classes. Our method achieves a 16.45% average improvement over [54], peaking at 35.31%. Comparatively, Vidmodex over DFMS-SL show improvements of 11.67% and 25.71%, respectively. These metrics underline our approach's robustness across various datasets.

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.%	Clone Acc.%	Query Budget
DFME [54]	U11‡/VVT†	800	84.96	55.27	70M
	U101‡/VVT†	2000	74.1	43.56	200M
	K400‡/SwT†	8000	70.8	28.49	350M
	K600‡/SwT†	10000	68.4	18.26	420M
	SS2‡/SwT†	17500	61.1	11.42	500M
DFMS-SL [47]	U11‡/VVT†	800	84.96	61.34	70M
	U101‡/VVT†	2000	74.1	47.53	200M
	K400‡/SwT†	8000	70.8	34.56	350M
	K600‡/SwT†	10000	68.4	20.15	420M
	SS2‡/SwT†	17500	61.1	16.38	500M
Vidmodex	U11‡/VVT†	800	84.96	**72.64**	50M
	U101‡/VVT†	2000	74.1	**68.23**	200M
	K400‡/SwT†	8000	70.8	**57.45**	350M
	K600‡/SwT†	10000	68.4	**51.62**	420M
	SS2‡/SwT†	17500	61.1	**37.63**	500M

† Model Architecture VVT: ViViT-B/16x2; SwT: Swin-T;
‡ Dataset U11: UCF-11; U101: UCF-101; K400: Kinetics-400; K600: Kinetics-600; SS2: Something-Something-v2;

Table 2. Comparision of Blackbox Extraction Techniques on Video Models

For video victim models, we use a similar Softlabel setting with probability predictions for all classes from the victim model. We've chosen ViViT-B/16x2 [3] and Swin-T [30] for reproducibility due to their popularity and ease of use of the provider library. We maintain a uniform Query Budget across all methods and present the training epochs and accuracy of the victim models. Importantly, we avoid using pre-trained weights for these models to ensure fair comparisons, as the clone models also lack access to pre-trained datasets or weights. As shown in Table 2, our method significantly outperforms DFME and DFMS-SL, with the disparity increasing as the complexity of the models rises.

Our approach consistently outperforms [54] and [47] in video model extraction. Specifically, Vidmodex achieves a mean improvement of 26.11% and a maximum improvement of 33.36% over [54]. And 21.52% and 31.47% over [47] respectively. Notably, these are achieved with a Query Budget that is lower or equal to the other two methods.

5.1.2. Impact of TopK Setting on Soft and Hardlabel extraction.

We explore scenarios where top-k labels facilitate model extraction, affirming our pipeline's real-world relevance. Adhering to definitions in Eq. (2) for softlabel and Eq. (3) for hardlabel ensures consistent analysis across scenarios. We do not employ specialized methods for handling top-k labels beyond these definitions. This study aims to demonstrate that computing SHAP values and introducing the SHAP-based objective does not negatively impact performance, even with fewer labels returned. As illustrated in Fig. 5, we plot mean clone accuracy for each value of K, with variability indicated by standard deviations. For softlabel extraction, we report on image and video models across $K \in \{1, 3, 5, 10, \text{ALL}\}$, while for hardlabel, $K \in \{1, 3, 5, 10\}$; the 'All' category is excluded in hardlabel as it offers no added information. We omit $K = 10$ for datasets like MNIST, CIFAR10, and UCF11 in hardlabel scenarios, where the total class count makes hardlabels redundant. Fig. 5a and Fig. 5b show an upward trend in extraction accuracy as more label information becomes available. Conversely, Fig. 5c and Fig. 5d display a downward trend in hardlabel settings, where additional labels decrease useful information. These trends align with the victim model's entropy in each scenario. Detailed experiment configurations are catalogued in the Appendix: Tab. B.I, Tab. B.II, Tab. B.I, and Tab. B.II detail each model type and label setting for various K values.

5.1.3. Grey Box extraction

We also evaluate our approach's efficacy using a surrogate dataset. While enhancing grey box accuracy is not our main focus, these tests ensure our SHAP-based objective doesn't negatively impact the generator's learning when using a proxy dataset. Instead of detailing the selection methodology for an appropriate surrogate dataset, we use parts of established datasets. Specifically, we incorporate ImageNet-22K [46] for image models, and Kinetics-700 [8] and CHARADES [50] for video models. A shuffled subset is used instead of targeted subclasses.

Experimental details and configurations are detailed in Table B.III, with results shown in Fig. 6. Our analysis covers three methods: [54], [47], and ours in both SoftLabel and HardLabel settings. We select the best `top_k` setting given All for SoftLabel settings and only top-1 labels in HardLabel settings.

Our method remains robust and effective, especially in SoftLabel image models, showing a mean improvement of 15.23% over DFME and 9.24% over DFMS-SL, peaking at 32.99% and 21.98%. In HardLabel image settings, we see average improvements of 15.15% over DFME and 9.24% over DFMS-HL, with highs of 29.14% and 14.16%. Video model extractions under SoftLabel conditions show enhancements of 19.04% over DFME and 12.65% over DFMS-SL, with top gains of 28.29% and 18.05%. HardLabel settings reveal our method surpassing DFME by 15.34% and DFMS-HL by 9.80%, with maximum improvements of 24.26% and 19.67%.

5.2. Ablation study on Discriminator Learning

In this section, we examine the impact of the `max_eval` parameter on SHAP value computations, crucial for training the discriminator $\mathcal{P}$, as detailed in Eq. (11). By increasing `max_eval`, we enhance the granularity of SHAP values, thereby improving accuracy as discussed in Eq. (1) and [32]. Initially set high, `max_eval` is progressively reduced during the training, akin to learning rate decay strategies. Our CIFAR100 experiments using a ResNet-18 model (see Appendix. C.1 for Fig. 6c details) demonstrate that the hybrid decay strategy, while slightly underperforming compared to constant high values, significantly outperforms the lowest setting and maintains lower variance in validation loss. Verifying the viability of an efficient and effective training process.

6. Limitations

While we aim to provide a query-efficient and interpretable approach for model extraction, we have achieved success with the limited experiments we have performed and presented. In adversarial contexts, such methods can also serve as a strategic probing tool to infer model behavior without full replication. The approach can be viewed as a dissector-style attack due to the SHAP value computation: even if the approach fails to fully replicate the target model, it still reveals local attribution signals from the black box, offering insights into decision boundaries as studied in Appendix D.3. This positions our work within a broader family of adversarial probing techniques studied in black-box security research.

A major limitation of the study is that we do not evaluate VidModEx on commercial MLaaS providers, as these platforms often incorporate proprietary defense mechanisms such as rate limiting, randomized responses, model fingerprinting, or obfuscation of prediction confidences[41], adding complexity that is difficult to simulate precisely. Our experiments are conducted under a standardized and controlled black-box interface to match the vanilla setup. While there are known workarounds to bypass[10] such defenses in real-world scenarios, integrating them is beyond the scope of this study. We anticipate that once access is normalized, VidModEx would remain competitive with—or outperform—existing model extraction techniques under comparable query budgets.

Another limiting factor in evaluating the pipeline on such MLaaS platforms is the lack of knowledge about the target dataset used to train the model or the absence of a surrogate dataset that approximates its distribution. Without this information, benchmarking the cloned model remains a non-trivial task. This limits the reliability of quantitative evaluation metrics unless challenges such as unsupervised task inference or dataset characterization are explicitly addressed [33, 34, 52].

7. Future Work

While VidModEx leverages SHAP-based objectives to guide the generation process, the outputs of the generator are not constrained to be visually interpretable by humans. Particularly in high-dimensional or fine-grained datasets, the generated samples may lack semantic coherence or exhibit abstract patterns that resist human interpretation. Although sample visualizations are included, along with activation atlases, they remain insufficient for drawing systematic insights into the generator's learning process.

The optimization structure of our approach builds upon established generator–teacher–student training paradigms used in prior model extraction works. While the integration of SHAP-based objectives is theoretically presented in Appendix A.1, the joint optimization of the generator, SHAP estimator, and substitute model introduces complex interactions that are not explicitly modeled, and we do not provide formal convergence guarantees for the overall system. Our empirical findings suggest stable training dynamics, with the improvements in extraction performance largely attributable to enhanced data representation driven by SHAP optimization. We are keen to see future works that extend or address these limitations and are open to exploring them in future iterations of this research direction.

8. Conclusion

In this study, we enhanced the DataFree model extraction framework by integrating Explainable AI algorithm. We tested our approach in real-world scenarios with both hard and soft label settings across various top-k outputs, aligning with typical MLaaS constraints. Our research extends model extraction to video classification models, observing significant improvements. We conducted quantitative and qualitative analyses to assess SHAP values' impact, noting enhanced extraction capabilities. We detailed our pipeline's implementation and explored additional hyperparameters to aid reproducibility. While applicable to audio, text, and tabular data, this paper focuses on video models to substantiate our claims. Future work could develop generalized techniques for larger models with billions of parameters, aiming for cost-effectiveness. Our primary goal is to enrich awareness of the potential impacts on the MLaaS industry and emphasize the importance of understanding associated risks.

References

[1] Amazon Web Services. Amazon rekognition video features. https://aws.amazon.com/rekognition/video-features/, 2024. Accessed: 2024-05-30. 3

[2] Amazon Web Services. Computer vision — amazon web services. https://aws.amazon.com/computer-vision/, 2024. Accessed: 2024-05-22. 1

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 7

[4] Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, and Marius Popescu. Black-box ripper: Copying black-box models using generative evolutionary algorithms. *Advances in Neural Information Processing Systems*, 33:20120–20129, 2020. 2

[5] James Beetham, Navid Kardan, Ajmal Mian, and Mubarak Shah. Dual student networks for data-free model stealing. *arXiv preprint arXiv:2309.10058*, 2023. 1, 2

[6] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*, 2024. 1

[7] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 5

[8] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 8

[9] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. https://distill.pub/2019/activation-atlas. 2

[10] Yanjiao Chen, Rui Guan, Xueluan Gong, Jianshuo Dong, and Meng Xue. D-dae: Defense-penetrating model extraction attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 382–399, 2023. 8

[11] Covariant. Rfm-1: Robotics foundation model. https://covariant.ai/rfm/, 2024. Accessed: 2024-05-22. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[13] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5

[14] edenai. Eden ai. https://www.edenai.co/, 2024. Accessed: 2024-05-22. 1

[15] Google Cloud. Get predictions for video classification with vertex ai. https://cloud.google.com/vertex-ai/docs/video-data/classification/get-predictions#output_format, 2021. Accessed: 2024-05-30. 3

[16] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 5

[17] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, 2022. 5

[18] Dong Han, Reza Babaei, Shangqing Zhao, and Samuel Cheng. Exploring the efficacy of learning techniques in model extraction attacks on image classifiers: A comparative study. *Applied Sciences*, 14(9):3785, 2024. 1

[19] Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. Model extraction and adversarial transferability, your bert is vulnerable! *arXiv preprint arXiv:2103.10013*, 2021. 1

[20] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022. 1

[21] Hugging Face. Autotrain – hugging face. https://huggingface.co/autotrain, 2024. Accessed: 2024-05-22. 1

[22] Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2021. 3, 4

[23] 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*, pages 13814–13823, 2021. 2, 5, 6

[24] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5

[25] Dmitry Kazhdan, Zohreh Shams, and Pietro Lio. Marleme: A multi-agent reinforcement learning model extraction library. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Department of Computer Science, University of Toronto*, 2009. 5

[27] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. 5

[28] Zijun Lin, Ke Xu, Chengfang Fang, Huadi Zheng, Aneez Ahmed Jaheezuddin, and Jie Shi. Quda: query-limited data-free model extraction. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pages 913–924, 2023. 2, 3

[29] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *2009 IEEE conference on computer vision and pattern recognition*, pages 1996–2003. IEEE, 2009. 5

[30] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 7

[31] Susana López and Martha Saboya. On the relationship between shapley and owen values. *Central European Journal of Operations Research*, 17:415–423, 2009. 3

[32] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 2, 3, 8

[33] Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. *arXiv preprint arXiv:2104.10706*, 2021. 8

[34] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092, 2024. 8

[35] Microsoft Azure. Azure ai vision services. https://azure.microsoft.com/en-us/products/ai-services/ai-vision/, 2024. Accessed: 2024-05-30. 3

[36] Microsoft Azure. Azure ai custom vision. https://azure.microsoft.com/en-us/products/ai-services/ai-custom-vision/, 2024. Accessed: 2024-05-30. 3

[37] Microsoft Azure. Azure ai vision with ocr and ai. https://azure.microsoft.com/en-in/products/ai-services/ai-vision/, 2024. Accessed: 2024-05-22. 1

[38] Takayuki Miura, Satoshi Hasegawa, and Toshiki Shibahara. Megex: Data-free model extraction attack against gradient-based explainable ai. *arXiv preprint arXiv:2107.08909*, 2021. 1, 2, 3

[39] V Nagisetty, L Graves, J Scott, and V Ganesh. xai-gan: enhancing generative adversarial networks via explainable ai systems (2020). *DOI: https://doi. org/10.48550/arxiv*, 2022. 3

[40] Abdullah Caglar Oksuz, Anisa Halimi, and Erman Ayday. Autolycus: Exploiting explainable ai (xai) for model extraction attacks against white-box models. *arXiv preprint arXiv:2302.02162*, 2023. 3

[41] Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55 (14s):1–41, 2023. 1, 2, 8

[42] OpenAI. Gptv system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. Accessed: 2024-05-22. 1

[43] 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*, pages 4954–4963, 2019. 2, 5, 6

[44] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 865–872, 2020. 2

[45] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Comput. Surv.*, 56(4), 2023. 1

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 8

[47] Sunandini Sanyal, Sravanti Addepalli, and R Venkatesh Babu. Towards data-free model stealing in a hard label setting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15284–15293, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[48] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[49] SHAP. SHAP PartitionExplainer Documentation. SHAP Documentation, 2024. Accessed: 2024-05-21. 3, 4

[50] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016. 8

[51] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[52] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*, 2020. 8

[53] Jean-Baptiste Truong, Pratyush Maini, Robert J. Walls, and Nicolas Papernot. Data-free model extraction (supplementary material), 2021. 3

[54] 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*, pages 4771–4780, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[55] 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. In *European conference on computer vision*, pages 192–208. Springer, 2022. 1, 3, 5

[56] Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. In *International conference on machine learning*, pages 10675–10685. PMLR, 2021. 2, 5, 6

[57] Anli Yan, Hongyang Yan, Li Hu, Xiaozhang Liu, and Teng Huang. Holistic implicit factor evaluation of model extraction attacks. *IEEE Transactions on Dependable and Secure Computing*, 2022. 1

[58] Anli Yan, Teng Huang, Lishan Ke, Xiaozhang Liu, Qi Chen, and Changyu Dong. Explanation leaks: Explanation-guided model extraction attacks. *Information Sciences*, 632:269–284, 2023. 3

[59] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 3, 5

[60] Shiqian Zhao, Kangjie Chen, Meng Hao, Jian Zhang, Guowen Xu, Hongwei Li, and Tianwei Zhang. Extracting cloud-based model with prior knowledge. *arXiv preprint arXiv:2306.04192*, 2023. 1

[61] Meixi Zheng, Xuanchen Yan, Zihao Zhu, Hongrui Chen, and Baoyuan Wu. Blackboxbench: A comprehensive benchmark of black-box adversarial attacks. *arXiv preprint arXiv:2312.16979*, 2023. 1