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.
{yuvaraj.govindarajulu, pavan.kulkarni, manojkumar.parmar}@bosch.com

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.

1 Introduction

With the rise in MLaaS (Machine Learning as a Service), which performs tasks from minute levels [32], [2]&[12] to multitasking across domains [37], [9]; There has been a significant increase in model performance, correlating with their size and the ability to accommodate large input spaces. Previous model extraction attacks [48],[33], [41] & [49] 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 easily outperformed by current datasets and more 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 [48], [41], [16], [54], [51] to train a substitute models, providing a prior about the target dataset. However, studies finding a balance [16] 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[18], 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 [22], [5], [17], a generalized base approach can promote the development of more efficient and large-scale attacks. In this work, we limit ourselves to 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.

Preprint. Under review.

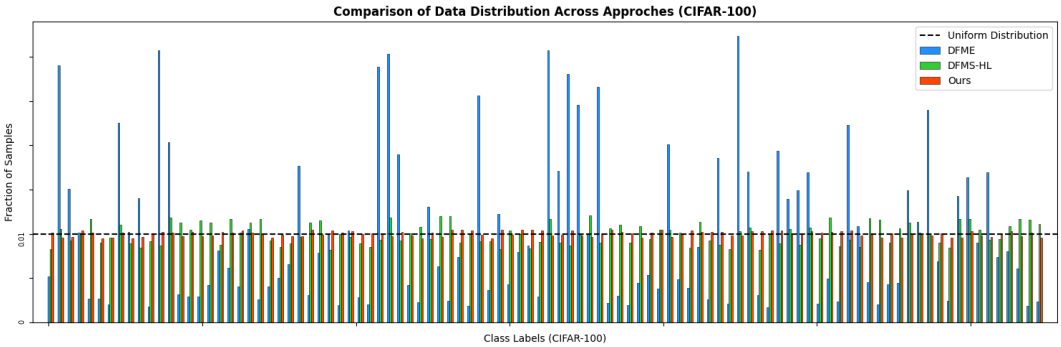


Figure 1: Distribution based on Victim model prediction on generated samples for CIFAR 100

We employ SHAP[29], 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 [20], 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 Eq. 1, This output can be from a regression, classification or any other open ended model. We use 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. 1.

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

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, BlackBox with soft labels, BlackBox with top-k & top-1 soft labels, BlackBox with top-k & top-1 hard labels

We also explore the limitations of this approach and provide considerations one should take into account when employing this strategy. To support further research and development in model extraction attacks, we have made the source code[1] available publicly.

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 have attempted to propose algorithms for model extraction in Softlabel settings and compute the approximate gradients for backpropagation of objectives [48], [33], [5], [20]. Works such as [41] extensively evaluate pipelines for Hardlabel settings, establishing a precedent for real-world model extraction. Although these approaches utilize similar frameworks with varied mechanisms for training the Generator, they share a common goal: optimizing the divergence between the Victim and Substitute as a discriminator for the Generator. However, these works are also limited in terms of

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

performance due to the query costs required to approximate gradients for a single sample using the Zeroth Order gradient approximation. Efforts have been made to train an efficient Generator using an Evolutionary Algorithm [38], [4], [39], [25] have demonstrated significantly lower extraction accuracy compared to the methods previously discussed [36]. Miscellaneous works like [50] focus on generating class-specific samples using minimum decision boundaries, which is superior to other approaches based on sample efficiency to train the Substitute model. However, computing these samples requires a high number of Victim Model queries, making it impractical in real-world scenarios due to the extensive querying required. Inferring from previous work, we understand the influence of samples significantly determines the extraction accuracy and efficiency of the approach. We attempt to address this trade-off by developing an auxiliary objective based on SHAP for the generator that is query-efficient and also improves the fidelity of the generated samples, enabling richer extraction of the Victim model.

2.2 Interpretable AI for GAN, Model Extraction

Research on utilizing Interpretable AI algorithms for training GANs remains sparse [34], largely because while explanations facilitate human interpretation, they are inferior in information density compared to gradients through the target network and discriminator. On the contrary, these methods have the potential for applications involving black box models, particularly in model extraction [52], [49], [35], [33]. While [52] and [35] simply train victim model to have the same explanation as that of the victim model, this does not direct the model to have better extraction accuracy and [33] assumes direct gradients from victim model defeating the purpose. In [49] it employs GradCam [42] to enhance sample augmentation by focusing on saliency maps from the substitute model to refine the loss function. The approach is constrained because GradCam depends on gradients from the model; Hence, using the substitute model for such operations leads to a noisy and unstable training process. Although the authors demonstrate stability within a confined study using predefined images from a surrogate dataset, the scalability of this approach to diverse real-world objectives remains dubious. While We Iterate on this by computing SHAP[43] values, which don't require gradient and can therefore be computed directly on the Victim model within a constrained `max_evals` budget for each sample. While acquiring SHAP values from the Victim model for `max_evals` for each sample is costly, we mitigate this by learning to estimate SHAP values within an Energy GAN framework [53], as deriving SHAP values [19] is more feasible than predicting gradients for the victim model.

2.3 Surrogate Dataset and Settings

In this subsection, we discuss the different settings and the utilization of surrogate datasets in prior research. Numerous studies have employed surrogate datasets [48], [41], [49], [25] each with different assessment on how samples should be selected from the surrogate or proxy dataset. Although these approaches significantly accelerate the extraction process, they require a prior understanding of the data distribution of the victim model, which complicates scalability as [47] summarizes the adverse effect of poor surrogate datasets. With the expansion of MLaaS platforms and the increasing number of classes for tasks, several aggregators [13], [1], [30], [31] now allow entities to deploy their models with the following settings: 1) Top-1 class labels 2) Top-k class labels 3) Top-1 prediction probability 4) Top-k prediction probability. We adopt these settings for our evaluations and also adopt to prediction probability of all classes to ensure comparative analysis with previous research. To offer a framework comparable to other algorithms for utilizing a surrogate dataset, we label the results under the grey box model extraction attack, with further details on the surrogate dataset used for each experiment in Sec.5.

3 Preliminary

In this section, we briefly introduce SHAP, an Additive explanation utilizing shapely values, particularly focusing on its application in defining objectives. We employ Partition Explainer[43], which recursively computes shapley value through a hierarchy of features; this hierarchy defines feature coalitions and results in the Owen values[28] from game theory. A detailed view on which is given in Appendix.B. Adhering to the fundamental principles of any SHAP explainer, we begin with the additive property presented in Eq. 1. In this equation f represents the target black-box model, M the size of input space, $\mathbb{E}[f(.)]$ the expected value of f over a uniform random distribution and ϕ

is the shapley value calculated over for the sample x, expressed as $\phi(f, x)$. Here x_i represents the i^{th} feature in x. The relationship between x' and x is given by the mapping function $x = h(x')$ as defined in [29, Section 2], with $x' \in [0, 1]^M$ standardised for the algorithms.

For generalizing to the different scenarios outlined in sec.2.3, we define our black-box victim model using Eq.2. This approach ensures consistent outputs across any top-k prediction setting. Here, $topk\_probs$ represents the probability values returned for topk predictions, and $topk\_indicies$ are the indices corresponding to these predictions. The output is a column vector of dimension $[0, 1]^{num\_classes}$, depicting a softmax output for a single class prediction scenario, which aligns closely with the intended application within the SHAP framework.

$$f_{sl} = \begin{cases} topk\_probs[i] & \text{if } i \in \text{topk\_indices}, \\ \dfrac{1 - sum(topk\_probs)}{num\_classes - k} & \text{otherwise.} \end{cases} \quad (2) \qquad f_{hl} = \begin{cases} 1/\text{k} & \text{if } i \in topk\_labels, \\ 0. & \text{otherwise} \end{cases} \quad (3)$$

For hard labels, we utilize the definition specified in Eq. 3. which assigns a straightforward binary output from the target black box model. We further explore how this method, although it conveys less information compared to the soft label approach, is adequately informative for calculating shapley values.

With the help of the definition Eq.1 which is a approximation under the local accuracy property given in [29, Section 3] and with choosing either function from Eq.2 or Eq.3, we derive Eq. 4a. We then represent the variables in the equation as vectors, reformulating $\phi = (\phi_1, \dots, \phi_i, \dots)^\top$ as a column vector and $x' = (x'_1, \dots, x'_i, \dots)^\top$ as a column vector, regardless of their original shape to obtain Eq. 4b. By applying this framework and forcing on specific class id c, we refine the formula to Eq. 4c.

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

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

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

Using 4 we define our objective to improve samples x to maximize class probability of the targeted model $f(.|c)$. Hence, we obtain eq. 5 as $\mathbb{E}[f(.|c)]$ is not dependent on x; we further simplify by replacing x' to reduce the final objective to be linearly proportional to the variable of interest. Hence we use j which is a column vector with 1's $j = (1, 1, \dots, 1)^\top$ of size M as $x' \in [0, 1]^M$ we use the upper bound and lower bound of the objective to be $0 \leq \phi^\top * x' \leq \phi^\top * j$ or $0 \geq \phi^\top * x' \geq \phi^\top * j$ based on the signum of $\phi^\top * x'$. While the use of $\phi^\top * j$ increases inaccuracy in objective, it enables us to shift the focus towards the contribution and coalition of each feature and not their magnitude. This was also another decision choice to solve the problem of exploding gradients observed while training. Finally, giving us the objective in Eq.6

$$\begin{aligned} \arg\max_x \quad f(x|c) &= \arg\max_x \quad \mathbb{E}[f(.|c)] + \phi(f(.|c), x)^\top * x' \\ &= \arg\max_x \quad \phi(f(.|c), x)^\top * x' \end{aligned} \tag{5}$$

$$ClassObj = \arg\max_x \quad \phi(f(.|c), x)^\top * j \quad \text{or} \quad ClassObj = \arg\max_x \quad \sum_{i=1}^{M} \phi_i(f(.|c), x) \tag{6}$$

Alongside the previously defined objective, a few crucial parameters are employed, which influence the accuracy of the approximations used in Eq. 1. The crucial parameters are: 1) `max_evals`: Partition explainer efficiently distributes Shapley value computations across a feature hierarchy rather than exhaustively calculating values for each feature. This approach significantly reduces the inference cost in high-dimensional settings, avoiding $M!$ inferences and instead requiring only `max_evals` calls to the targeted model. 2) `masker`: The Partition Explainer differs from other explainers by excluding either a single unit or multiple units of features at once, rather than offering granular control over each individual feature. The `masker` parameter defines this granularity we

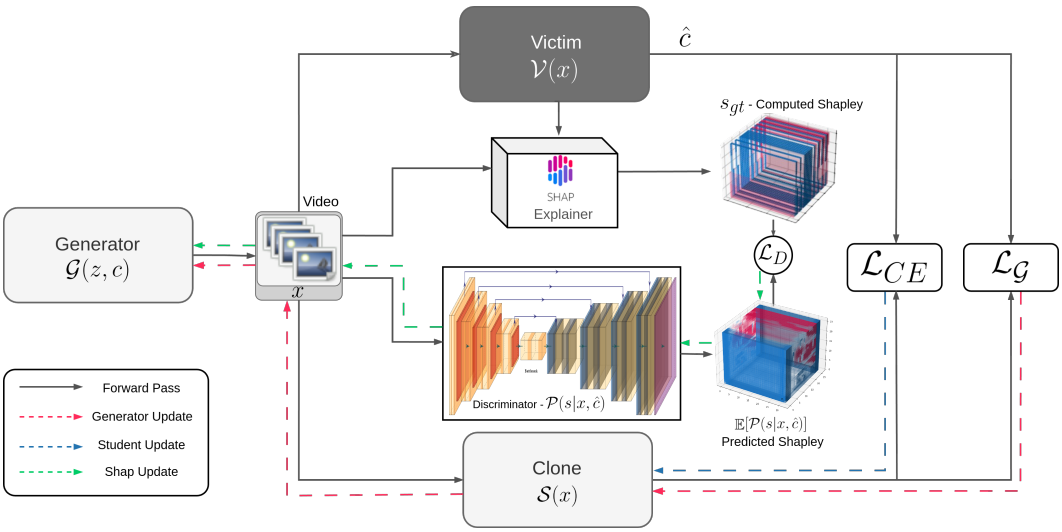


Figure 2: Model extraction diagram with additional objectives and SHAP explainers

employ a 3×3 & $3 \times 3 \times 3$ block of pixel for image & video models respectively. To obscure these sections, we use Gaussian Blur instead of zero-filling. This technique effectively reduces information without adversely affecting model predictions. Both `max_evals` & `masker` are highly influential and dependant on the target model's complexity and nuances.

4 Approach

The overall attack setup is well outlined by previous works [48], [41], 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[48] for soft label setting given in Eq.7a, and employ CrossEntropy Loss[41] for hard label setting given in Eq.7b to optimize $\mathcal{S}$. To optimize $\mathcal{G}$, we use an adversarial loss to increase the divergence between Student and victim model[48, 41] which is given by Eq. 8 As we use Conditional Generator instead we also specify c_T Target class index to generate samples for a particular class.

$$\mathcal{L}_{sl}(x) = \sum_{i \in topk\_indices} \mathcal{V}(x|i) \log \frac{\mathcal{V}(x|i)}{\mathcal{S}(x|i)} \quad (7a)$$

$$\mathcal{L}_{hl}(x) = - \sum_{i \in topk\_indices} \mathcal{V}(x|i) * \log(\mathcal{S}(x|i)) \quad (7b)$$

$$\begin{aligned} & z \sim \mathcal{N}(0,1) \\ & x = G(z, c_T) \\ \implies & \underset{\theta_{\mathcal{G}}}{argmax} \quad \underset{\theta_{\mathcal{S}}}{argmin} \quad \mathcal{L}(x) \end{aligned} \quad (8)$$

Alongside this setup, we additionally employ the ClassWise Objective defined in Eq.6, while ϕ value obtained from the explainer is not differentiable we introduce a estimator $\mathcal{P}(s|x, c_T)$ which estimates the SHAP value of the input given the input sample and the targetted class index. The architecture of $\mathcal{P}$ is a conditional UNet such that the shape of the predicted shap values is sample as the original input. The model $\mathcal{P}$ predicts a normal distribution of the SHAP value, this format is also helpful to compute the probability value of s_{gt} in the predicted distribution. We use the $\mathcal{P}(s_{gt}|x, c)$ as a mask to the objective to reduce error due to divergence between the prediction and the ground truth. We use this modified objective if s_{gt} is precomputed for the sample else use the initial objective as given in Eq.9. To improve the estimation of the model $\mathcal{P}$ we use Mean Absolute Error between the sampled shap value from $\mathcal{P}$ and shap value computed from the explainer given in Eq. 10.

$$\begin{aligned} ClassObj &= \underset{x}{argmax} \sum \mathbb{E}[\mathcal{P}(s|x,c)] \\ &= \underset{x}{argmax} \sum \mathbb{E}[\mathcal{P}(s|x,c)] \odot \mathcal{P}(s_{gt}|x,c) \end{aligned} \quad (9)$$

$$\mathcal{L}_{\mathcal{P}} = \sum |s_{gt} - \hat{s}| \,, \quad \text{where } \hat{s} \sim \mathcal{P}(x,c) \quad (10)$$

(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 3: Shap values and visualization at each stage of the Pipeline

The complete pipeline can be seen in Fig. 2, the shap estimator $\mathcal{P}$ is equivalent to a energy based discriminator in [53] only difference being they are not necessarily in an adversarial setting as $\mathcal{P}$ is attempting to improve accuracy in estimating the SHAP values for the generated samples, while $\mathcal{G}$ is being optimized generated samples to increase SHAP values of the samples they generate. Hence, in the paper, we call $\mathcal{P}$ and its objective as discriminator in the paper. This objective enables the generator to produce rich samples and improve samples to be balanced across classes. The probabilistic discriminator has an additional mask $\mathcal{P}(s_{gt}|x, c)$, which ensures no out-of-distribution or noise is introduced while training these components. We normalize the SHAP values to a range of $[-1, 1]$ as the magnitude of the shap values varies from $1 \times 10^{-8} - 1 \times 10^{-11}$ based on datasets and problem settings like image and video models.

Fig. 3 presents a series of visualizations that illustrate the data at each stage within the pipeline. Fig.3a displays the initial input to the victim model. This image is a substitute for a generated sample to simplify the interpretation of subsequent images. Fig.3b shows the SHAP value computed using the partition explainer. Fig. 3c is the expected value μ of the discriminator, denoted as $\mathbb{E}[\mathcal{P}(s|x, c)]$. Fig.3d is the probability mask used to stabilize the initial training phase. It computes the probability that the expected output s_{gt} matches the predicted distribution, essentially assessing the accuracy of the predictions relative to the ground truth. Finally, Fig.3e illustrates the final objective used to train the generator, as specified in Eq. 9 in energy gan-like architecture.

5 Experiments

This section evaluates our Vidmodex approach under diverse and challenging settings, detailed in sec. 2.3, using both image and video models across various datasets such as MNIST[11], CIFAR10, CIFAR100[23], Caltech101[24], Caltech256[15] and ImageNet1K[10] for images, and UCF11[26], UCF101[46], Kinetics 400[21], Kinetics 600[7] and Something-Something v2[14] for videos. These tests evaluate over increasing number of classes and complexities; we ensure the evaluation of image model extraction on high-resolution datasets to demonstrate efficiency in large search space. We compare the benchmark primarily across DFME[48], DFMS-HL[41] which we reproduce with our best efforts. We additional utilizes results reported from ZSDB3KD[50], MAZE[20], KnockoffNets[38] and BlackBox Dissector[49]. We chose not to replicate results from other studies since our selected methods have already outperformed them in prior works. Further, we assess the impact of `max_evals` on the extraction process and learning within the discriminator in Sec.5.2.1, identifying what we consider the best configuration. We conduct an ablation study to explore performance variations across different `top_k` settings. Our research primarily focuses on black box model extraction, but we also examine the implications of employing a surrogate dataset (grey box access), discussing its influence later in the paper. Additionally, we present hard label results with analysis over the availability of `top_k` labels to validate generalization. Our comprehensive qualitative analysis is detailed in the Appendix. D, further grounding our findings for the presented empirical evidence.

Experimental setup

DFME and DFMS-HL are integrated as configurable approaches within our pipeline, sharing similar outlines. We also provide scripts to facilitate reproducing these results in the code base. The experiments were conducted on the following hardware setup: 2 nodes of 8 x H100 GPUs(80GB), Intel(R) Xeon(R) Platinum 8480C CPUs (96 cores at 4 GHz Max boost), and 1.8 TB of RAM. We also test all the scripts on a machine with 4 x A100 GPUs (80GB), AMD EPYC 7V13 (64 cores at

4.8 GHz) and 867 GB RAM. While our primary setup is adept, we verify that our scripts run on a modest V100 GPUs(32 GB) system, ensuring reproducibility and facilitating development. Only the experiments involving Kinetics400, Kin600 and Something-Something v2 require the more capable system.

5.1 Results

We present our results for blackbox extraction results in Sec.5.1.1, While further we analyze the influence of top-k on softlabel and hard-label setting in Sec.5.1.2. We additionally share the results of Greybox extraction in Sec.5.1.3.

Table 1: Comparision of Blackbox Extraction Techniques on Image Models

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.%	Clone Acc.%	Query Budget
DFME[48]	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 [41]	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 [50]	MN‡ / LN-5†	-	99.33	**96.54**	100M
	C10‡ / RN-18†	-	82.5	59.46	400M
MAZE [20]	C10‡ / RN-18†	-	92.26	45.60	30M
	C100‡ / RN-34†	-	82.5	37.20	80M
KnockOff Nets[38]	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

5.1.1 BlackBox Extraction

For the blackbox extraction, our initial investigation centers on SoftLabel Setting with probabilities of all classes from the victim model, in line with previous studies like [50], [20], [38]. As illustrated in Table. 1, we present the accuracies for these methods as reported in their work and reproduced numbers from [41] and [48] alongside our work. To ensure reproducible comparative study, we provide the training epochs required to replicate the victim models, as prior studies often do not offer standardized or pre-trained weights. We train the Target victim architecture from a random initialized state on the target dataset with all configuration details including seeds are available in our code repository. We employ the same architecture for both the clone and the victim model, thereby eliminating any potential bias that might arise from architectural differences.

We also detail the Query Budget, for reported work we either present the reported value or calculate based on the algorithms described, particulary for [50]. Our approach demonstrates higher extraction across most tested configurations except in MNIST, where it performs comparably to [41] and slightly behind [50]. Notably, our method is 25$\times$ efficient than [50] based on Query Budget. We employ uniform Query Budget across the methods we reproduce, where we outperform [48] and [41] with equivalent budgets. We see a pattern of reducing extraction accuracy with increase in difficulty in the dataset which is correlated to the increase in resolution and increase in number of classes. The increase in difficulty can also be seen with reducing victim accuracy for a higher training epochs. Our approach outperforms [48] on a average of 16.45%, with a maximum improvement of 35.31%. Comparatively, Vidmodex and DFMS-SL Shows a mean improvement of 11.67% and a maximum improvement of 25.71%. These statistics underscore the efficacy of our approach across the evaluated datasets.

For video victim models, we employ similar Softlabel setting where we have probability predictions for all classes from the victim model. To facilitate reproduciblity, we have opted to use ViViT-B/16x2 [3] and Swin-T[27], primarily due to their popularity and the ease of use by the providor libraries. We maintain a uniform Query Budget across all three methods and document the training epoch and accuracy for the victim models. Importantly, we do not utilize any pre-trained weights fro these victim models, ensuring a level playing field, as the clone models also lacks access to pre-trained datasets or weights. This allows use to observe a correlation between the victim training epochs and the Query budget while offering a ealistic evaluation of the victim model accuracy, as opposed to the standard practice of using pre-trained weights for video models. As demonstrated in Table. 2, our method consistently outperforms DFME and DFMS-SL by a considerable margin, with the disparity growing as the complexity of both image and video models increases.

Our Vidmodex method consistently outperforms [48] and [41] in video model extraction. Specifically, Vidmodex achieves a mean improvement of 26.11% and a maximum improvement of 33.36% over [48]. In comparison to [41], Vidmodex shows a mean improvement of 21.52% and a maximum improvement of 31.47% in clone accuracy. Notably, these enhancements are achieved with a Query Budget that is equal to or lower than those utilized in the other two methods.

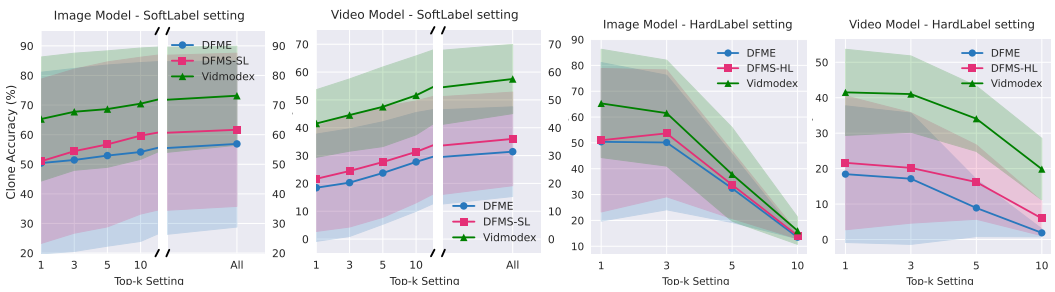
5.1.2 Impact of TopK Setting on Soft and Hardlabel extraction.

Table 2: Comparision of Blackbox Extraction on Video Models

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.%	Clone Acc.%	Query Budget
DFME[48]	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 [41]	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;

We explore the scenario where top-k labels are available for model extraction, illustrating the real-world applicability of our pipeline. This analysis adheres to the definition provided in Eq.2 for softlabel and Eq.3 for hardlabel, ensuring consistency across all scenarios. Our approach does not incorporate any specific methodology for handling top-k labels beyond these definitions. This analysis aims to demonstrate that the computation of the SHAP values and the introduction of the New Shap-based objective do not negatively impact performance when fewer labels are returned. As shown in Fig.4, we plot the mean clone accuracy for each value of K, with regions defined by the standard deviations of these points for each K. For Softlabel extraction, we report results for image and video

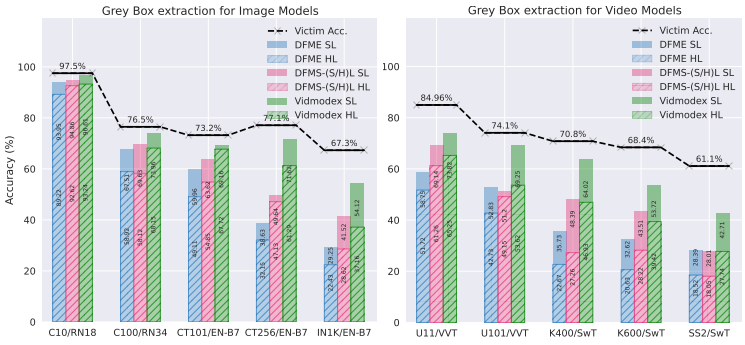


(a) (b) (c) (d)

Figure 4: Plots of the extraction accuracy across different K, for both Softlabel and Hardlabel setting

models across K $\in 1, 3, 5, 10$, ALL, While for Hardlabel extraction, we report K $\in 1, 3, 5, 10$; the 'All' category is omitted as it would equate to no information for hardlabel; hence the focus is limited to these K values. Notably, for datasets like MNIST, CIFAR10 and UCF11, we do not present numbers for K=10 in hardlabel as the total number of classes is 10 or 11, making hardlabels redundant in such scenarios. From Fig.4a for image models, there is an observed upward trend in extraction accuracy as richer information is introduced with each increasing label. A similar trend is observed for video models in Fig.4b. Conversely, Fig.4c and Fig.4d show a reverse trend for hardlabel extraction; accuracy decreases with increasing top-k as information diminishes with each additional label in hardlabel settings. These trends can be directly correlated with the entropy of the victim model in each scenario. Detailed configurations of these experiments are available in the Appendix: Table. 5 details the hard label extraction for image models, Table. 6 for video models, Table. 3 for softlabel extraction of image models, and Table. 4 for softlabel extraction of video models across various K values.

5.1.3 Grey Box extraction



(a) Image Model (b) Video Model

Figure 5: Comparision of GreyBox extraction methods

We also evaluate the efficacy of our approach using a surrogate dataset. Although enhancing grey box accuracy is not our primary focus, these experiments serve to confirm that our SHAP-based objective does not adversely affect the generator's learning process when paired with a proxy or surrogate dataset. Instead of delving into the methodology for selecting an appropriate surrogate dataset, we utilize portions of established datasets. Specifically, we incorporate ImageNet-22KK[40] for image models and Kinetics-700[8] and CHARADES[44] for video model extractions. These datasets are shuffled and employed without targeting any specific subclasses.

The experimental details and configurations are outlined in Table. 7 for image models and Table.8 for video models, with results visualized in Fig.5. Our analysis includes only three methods: [48], [41], and our own approach, examining both SoftLabel and HardLabel settings. As expected, extraction accuracy declines with increased difficulty, utilizing all labels for SoftLabel settings and only top-1 labels for HardLabel settings.

Despite this complexity, our method demonstrates notable robustness and effectiveness, particularly in the SoftLabel and image model contexts, where it shows a mean improvement of approximately 15.23% over DFME and 9.24% over DFMS-SL, with peaks of 32.99% and 21.98%, respectively. For the HardLabel setting in image models, our approach yields an average improvement of 15.15% over DFME and 9.24% over DFMS-HL, reaching up to 29.14% and 14.16%, respectively. In video model extractions under SoftLabel settings, we observe average enhancements of 19.04% over DFME and 12.65% over DFMS-SL, with maximum gains of 28.29% and 18.05%, respectively. The HardLabel setting shows our method outperforming DFME by 15.34% and DFMS-HL by 9.80% on average, with maximum improvements of 24.26% and 19.67%, respectively. These findings underscore the robustness and efficacy of our extraction pipeline.

5.2 Ablation study

5.2.1 Discriminator Learning

In this section, we examine the impact of the `max_eval` parameter on SHAP value computations, crucial for training the discriminator $\mathcal{P}$ with the objective defined in Eq. 10. A higher `max_eval` value results in more fine-grained SHAP values for each feature, thereby improving local accuracy as

outlined in Eq.1 and discussed in [29]. To enhance the approximation quality of $\mathcal{P}$, we initially set `max_eval` to a high value. However, a high `max_eval` also increases the number of victim model queries per sample. To manage this, we progressively reduce `max_eval` throughout the training process, similar to learning rate decay techniques [45]. This reduction strategy halves `max_eval` progressively until reaching a minimal threshold, beyond which the discriminator is no longer trained. This eliminates the need for further SHAP computations and reduces queries to the victim model. Additionally, we do not employ masking for SHAP optimization as specified in Eq.9.

We explore the stability of our approach using a dynamic `max_eval` adjustment mechanism, which can significantly alter the training objective. We conduct an extraction on the CIFAR100 dataset using a ResNet-18 model. The outcome of this experiment is visualized in Fig.6, where we evaluate the discriminator $\mathcal{P}$ against the SHAP ground truth, computed with `max_eval` set at 1024. For training purposes, we use the smaller values of `max_eval` specifically 32, 64, 128. To establish benchmarks for this configuration, we initially train the discriminator solely to minimize Eq.10 using fixed `max_eval` values of $\in 32, 64, 128$. Conversely, for the scheduled decay approach, we implement a descending `max_eval` sequence 128, 64, 32 over respective intervals: [(0, 500), (500, 1000), (1000, 1500)]. Our results indicate that while the hybrid decay strategy yields slightly inferior outcomes compared to constant high values of 128 and 64, it significantly outperforms the lowest fixed setting of 32. Moreover, the variance in validation loss under the hybrid approach is lower than 32 and 64 settings, making the approach viable to maintain efficiency while providing substantial feedback for optimizing the pipeline.

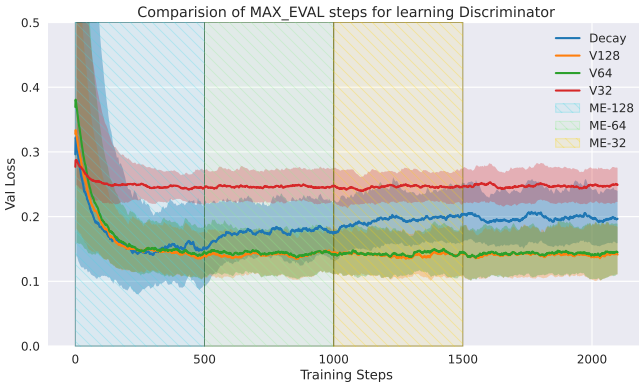


Figure 6: Variance in training Discriminator based on `max_evals`

6 Conclusion

In this study, we aimed to enhance the DataFree model extraction framework by integrating Explainable AI algorithms as an auxiliary objective alongside existing methodologies. We rigorously tested our approach in real-world scenarios, encompassing both hard and soft label settings across various top-k outputs, aligning with the constraints typical of contemporary MLaaS offerings. Our research extends the scope of model extraction attacks to video classification models, where we observed significant improvements over previous methods. Both quantitative and qualitative analyses were conducted to assess the impact of SHAP values, underscoring notable enhancements in model extraction. We also detailed the implementation of our pipeline and explored the influence of additional hyperparameters to facilitate reproducibility and further development. Although our approach is broadly applicable to any target model task—including audio, text, and tabular data—this paper focuses on a constrained study to substantiate our claims. Future work could explore the development of generalized extraction techniques for even larger models with billions of parameters, aiming to achieve this at a reasonable cost. While our work primarily details the attack on such models, our overarching goal is to enrich the community's awareness of the substantial MLaaS industry. We believe it is of utmost importance to understand the potential risks involved.

References

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

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

[3] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[4] A. Barbalau, A. Cosma, R. T. Ionescu, and M. Popescu. Black-box ripper: Copying black-box models using generative evolutionary algorithms. *Advances in Neural Information Processing Systems*, 33:20120–20129, 2020.

[5] J. Beetham, N. Kardan, A. Mian, and M. Shah. Dual student networks for data-free model stealing. *arXiv preprint arXiv:2309.10058*, 2023.

[6] N. Carlini, D. Paleka, K. D. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy, et al. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*, 2024.

[7] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[8] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

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

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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

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

[13] 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.

[14] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. 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.

[15] G. Griffin, A. Holub, and P. Perona. Caltech 256, Apr 2022.

[16] D. Han, R. Babaei, S. Zhao, and S. Cheng. Exploring the efficacy of learning techniques in model extraction attacks on image classifiers: A comparative study. *Applied Sciences*, 14(9): 3785, 2024.

[17] X. He, L. Lyu, Q. Xu, and L. Sun. Model extraction and adversarial transferability, your bert is vulnerable! *arXiv preprint arXiv:2103.10013*, 2021.

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

[19] N. Jethani, M. Sudarshan, I. C. Covert, S.-I. Lee, and R. Ranganath. Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2021.

[20] S. Kariyappa, A. Prakash, and M. 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.

[21] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

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

[23] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. *Department of Computer Science, University of Toronto*, 2009.

[24] F.-F. Li, M. Andreeto, M. Ranzato, and P. Perona. Caltech 101, Apr 2022.

[25] Z. Lin, K. Xu, C. Fang, H. Zheng, A. Ahmed Jaheezuddin, and J. 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.

[26] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *2009 IEEE conference on computer vision and pattern recognition*, pages 1996–2003. IEEE, 2009.

[27] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[28] S. López and M. Saboya. On the relationship between shapley and owen values. *Central European Journal of Operations Research*, 17:415–423, 2009.

[29] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

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

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

[32] 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.

[33] T. Miura, S. Hasegawa, and T. Shibahara. Megex: Data-free model extraction attack against gradient-based explainable ai. *arXiv preprint arXiv:2107.08909*, 2021.

[34] 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*, 2002.

[35] A. C. Oksuz, A. Halimi, and E. Ayday. Autolycus: Exploiting explainable ai (xai) for model extraction attacks against white-box models. *arXiv preprint arXiv:2302.02162*, 2023.

[36] D. Oliynyk, R. Mayer, and A. 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.

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

[38] T. Orekondy, B. Schiele, and M. 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.

[39] S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 865–872, 2020.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[41] S. Sanyal, S. Addepalli, and R. V. 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.

[42] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. 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.

[43] SHAP. SHAP PartitionExplainer Documentation. SHAP Documentation, 2024. URL `https://shap.readthedocs.io/en/latest/generated/shap.PartitionExplainer.html`. Accessed: 2024-05-21.

[44] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. 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.

[45] L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[46] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[47] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot. Data-free model extraction (supplementary material), June 2021. URL `https://openaccess.thecvf.com/content/CVPR2021/supplemental/Truong_Data-Free_Model_Extraction_CVPR_2021_supplemental.pdf`.

[48] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4771–4780, 2021.

[49] Y. Wang, J. Li, H. Liu, Y. Wang, Y. Wu, F. Huang, and R. Ji. Black-box dissector: Towards erasing-based hard-label model stealing attack. In *European conference on computer vision*, pages 192–208. Springer, 2022.

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

[51] A. Yan, H. Yan, L. Hu, X. Liu, and T. Huang. Holistic implicit factor evaluation of model extraction attacks. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[52] A. Yan, T. Huang, L. Ke, X. Liu, Q. Chen, and C. Dong. Explanation leaks: Explanation-guided model extraction attacks. *Information Sciences*, 632:269–284, 2023.

[53] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[54] S. Zhao, K. Chen, M. Hao, J. Zhang, G. Xu, H. Li, and T. Zhang. Extracting cloud-based model with prior knowledge. *arXiv preprint arXiv:2306.04192*, 2023.

A Appendix / supplemental material

B Derivation of Hierarchical Shapley Values from Owen Values

B.1 Proof of Shapley values and owen values

Given the complex interactions among features in high-dimensional models, it is crucial to attribute the model's prediction accurately back to the contributing features. The additive feature attribution methods provide a framework, with Shapley values offering a theoretically sound approach under certain axiomatic properties. Building upon the foundational Theorem 1, which establishes the uniqueness of Shapley values under properties of local accuracy, missingness, and consistency, we propose a new theorem for hierarchical Shapley values derived from Owen values.

Theorem 2: Hierarchical Shapley Values from Owen Values

Under the framework established by Theorem 1 and the hierarchical structure of the input space M, where M is divided into k groups such that $M = \bigcup_{i=1}^{k} G_i$ with $G_i \cap G_j = \emptyset$ for all $i \neq j$, the Shapley values derived from Owen values can uniquely be determined by the following approach:

Definition

Let f be the targeted black-box model mapping input vectors to outputs, and let $f(S)$ denote the output of the model when only the features in set $S \subseteq M$ are active. The Owen value for feature i considering the hierarchical decomposition is given by:

$$\phi_i^O(f) = \sum_{T:G_i \subseteq T \subseteq M \setminus \{i\}} \frac{|T \cap G_i|!(|G_i| - |T \cap G_i| - 1)!}{|G_i|!} \cdot \frac{|T|!(|M| - |T| - 1)!}{|M|!} \cdot [f(T \cup \{i\}) - f(T)] \tag{11}$$

Hierarchical Shapley Value Computation

The hierarchical Shapley value for each feature i in the input space M, which complies with the properties of local accuracy, missingness, and consistency as per Theorem 1, is computed as:

$$\phi_i(f, x) = \sum_{z_0 \subseteq x_0} \frac{|z_0|!(|M| - |z_0| - 1)!}{|M|!} [f_x(z_0 \cup \{i\}) - f_x(z_0 \setminus \{i\})] \tag{12}$$

where x_0 represents the simplified input mapping hx, and $z_0 \subseteq x_0$ represents all vectors z_0 where the non-zero entries are a subset of the non-zero entries in x_0.

Hierarchical Decomposition of Contributions in Cooperative Game Theory Applied to Machine Learning Models

In the analysis of machine learning models, particularly those considered as black-boxes, understanding the contribution of individual features is crucial. This can be analogously studied through concepts derived from cooperative game theory, notably the Shapley values and Owen values. In this context, we redefine the classical game-theoretic approach to accommodate a model's feature space.

Definitions and Notation

Let M be the set of all features in the input space of a black-box model f, which maps input vectors to an output space. The set M consists of k groups such that $M = \bigcup_{i=1}^{k} G_i$ and $G_i \cap G_j = \emptyset$ for all $i \neq j$. Each group G_i may represent a subset of features that interact or are related in a particular way, perhaps due to their similar nature or collective impact on the model's output.

The function f is a characteristic function in this cooperative game, assigning a real number to each subset of features, indicative of the output's dependency on those features.

Hierarchical Contribution Calculation

The contribution of each feature, and by extension each group of features, is assessed by the following hierarchical decomposition:

Group-Level Contribution

The contribution of a group G_i to the overall model output is computed by considering the group as a single entity within the larger set:

$$\phi_{G_i}(f) = \sum_{T \subseteq M \setminus G_i} \frac{|T|!(|M| - |T| - |G_i|)!}{|M|!} \cdot [f(T \cup G_i) - f(T)] \tag{13}$$

Here, T represents a coalition of groups excluding G_i.

Individual Feature Contribution Within Groups

Within each group G_i, the contribution of an individual feature i is calculated using the Shapley value formula adapted for the subgroup:

$$\phi_i^{G_i}(f) = \sum_{S \subseteq G_i \setminus \{i\}} \frac{|S|!(|G_i| - |S| - 1)!}{|G_i|!} \cdot [f(S \cup \{i\}) - f(S)] \tag{14}$$

Integration of Contributions

The Owen value for a feature i, reflecting both intra-group dynamics and inter-group interactions, is given by:

$$\phi_i^O(f) = \sum_{T: G_i \subseteq T \subseteq M \setminus \{i\}} \frac{|T \cap G_i|!(|G_i| - |T \cap G_i| - 1)!}{|G_i|!} \frac{|T|!(|M| - |T| - 1)!}{|M|!} \cdot [f(T \cup \{i\}) - f(T)] \tag{15}$$

This hierarchical approach to computing contributions of features provides a nuanced view that is particularly useful for interpreting the behavior of complex models in machine learning. By treating the feature space as a cooperative game, we can apply game-theoretical insights to understand how individual and groups of features influence model predictions, providing a bridge between machine learning interpretability and cooperative game theory.

C Additional Comparative Results

C.1 Soft Label

C.2 HardLabel

C.3 Grey Box extraction

D Qualitative Analysize

Table 3: Comparision of SoftLabel Blackbox Extraction on Image Models across TopK

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.	Clone Accuracy with given top-k classes in %					Query Budget
				1	3	5	10	All	
DFME[48]	MN‡ / RN18†	500	99.7	90.1	91.8	92.2	92.5	92.5	4M
	C10‡ / RN18†	1500	97.5	85.22	85.83	86.21	87.32	87.32	10M
	C100‡ / RN34†	3500	76.5	52.67	54.63	58.25	59.74	62.15	25M
	CT101‡ / EN-B7†	8000	73.2	44.96	45.62	48.21	49.61	53.56	70M
	CT256‡ / EN-B7†	10500	77.1	26.71	27.3	27.5	28.3	32.52	100M
	IN1K‡ / EN-B7†	15000	67.3	2.8	3.56	5.25	7.4	13.23	120M
DFMS-SL [41]	MN‡ / RN18†	500	99.7	86.6	89.34	91.5	95.1	95.1	4M
	C10‡ / RN18†	1500	97.5	81.29	86.46	89.92	91.24	91.24	10M
	C100‡ / RN34†	3500	76.5	53.62	57.93	59.61	60.64	65.04	25M
	CT101‡ / EN-B7†	8000	73.2	46.7	48.52	49.8	51.53	56.46	70M
	CT256‡ / EN-B7†	10500	77.1	32.35	32.84	35.85	37.0	38.54	100M
	IN1K‡ / EN-B7†	15000	67.3	5.56	11.45	13.64	22.53	23.56	120M
Vidmodex	MN‡ / RN18†	500	99.7	92.14	93.12	93.56	94.6	94.6	4M
	C10‡ / RN18†	1500	97.5	91.56	92.4	94.7	94.9	94.9	10M
	C100‡ / RN34†	3500	76.5	60.52	65.25	65.32	65.60	69.52	25M
	CT101‡ / EN-B7†	8000	73.2	60.03	62.73	62.61	67.46	68.14	70M
	CT256‡ / EN-B7†	10500	77.1	54.73	55.42	55.22	56.63	63.25	100M
	IN1K‡ / EN-B7†	15000	67.3	32.7	37.52	40.45	43.63	48.54	120M

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

Table 4: Comparision of SoftLabel Blackbox Extraction on Video Models across TopK

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.	Clone Accuracy with given top-k classes in %					Query Budget
				1	3	5	10	All	
DFME[48]	U11‡ / VVT†	800	84.96	49.75	50.93	52.79	55.28	55.27	70M
	U101‡ / VVT†	2000	74.1	31.95	34.75	36.8	40.83	43.56	200M
	K400‡ / SwT†	8000	70.8	7.52	10.42	17.24	21.56	28.49	350M
	K600‡ / SwT†	10000	68.4	2.3	3.95	7.31	12.69	18.26	420M
	SS2‡ / SwT†	17500	61.1	0.69	1.32	4.5	8.23	11.42	500M
DFMS-SL [41]	U11‡ / VVT†	800	84.96	52.57	58.29	60.23	61.36	61.34	70M
	U101‡ / VVT†	2000	74.1	30.6	33.81	39.37	41.4	47.53	200M
	K400‡ / SwT†	8000	70.8	20.29	20.61	21.94	27.42	34.56	350M
	K600‡ / SwT†	10000	68.4	3.82	6.32	9.2	13.85	20.15	420M
	SS2‡ / SwT†	17500	61.1	0.94	3.24	7.59	12.59	16.38	500M
Vidmodex	U11‡ / VVT†	800	84.96	60.14	64.53	70.24	72.71	72.64	50M
	U101‡ / VVT†	2000	74.1	48.56	53.50	57.12	61.48	68.23	200M
	K400‡ / SwT†	8000	70.8	38.02	39.63	41.52	48.53	57.45	350M
	K600‡ / SwT†	10000	68.4	36.91	37.59	38.31	43.51	51.62	420M
	SS2‡ / SwT†	17500	61.1	23.94	27.51	30.41	31.94	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 5: Comparision of HardLabel Blackbox Extraction on Image Models across TopK

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.	Clone Accuracy with given top-k classes in %				Query Budget
				1	3	5	10	
DFME[48]	MN‡ / RN18†	500	99.7	90.1	81.25	28.64	-	4M
	C10‡ / RN18†	1500	97.5	85.22	80.97	31.52	-	10M
	C100‡ / RN34†	3500	76.5	52.67	54.52	52.5	13.85	25M
	CT101‡ / EN-B7†	8000	73.2	44.96	46.14	43.63	14.16	70M
	CT256‡ / EN-B7†	10500	77.1	26.71	28.5	27.73	12.72	100M
	IN1K‡ / EN-B7†	15000	67.3	2.8	9.53	10.42	13.21	120M
DFMS-HL [41]	MN‡ / RN18†	500	99.7	86.6	86.4	23.62	-	4M
	C10‡ / RN18†	1500	97.5	81.29	82.52	25.74	-	10M
	C100‡ / RN34†	3500	76.5	53.62	53.61	54.93	12.62	25M
	CT101‡ / EN-B7†	8000	73.2	46.7	48.52	47.61	13.84	70M
	CT256‡ / EN-B7†	10500	77.1	32.35	32.9	31.52	13.96	100M
	IN1K‡ / EN-B7†	15000	67.3	5.56	18.22	19.19	15.96	120M
Vidmodex	MN‡ / RN18†	500	99.7	92.14	87.35	14.52	-	4M
	C10‡ / RN18†	1500	97.5	91.56	86.93	17.66	-	10M
	C100‡ / RN34†	3500	76.5	60.52	61.52	62.44	15.63	25M
	CT101‡ / EN-B7†	8000	73.2	60.03	58.16	57.25	9.33	70M
	CT256‡ / EN-B7†	10500	77.1	54.73	42.62	41.92	14.92	100M
	IN1K‡ / EN-B7†	15000	67.3	32.7	32.52	33.13	24.11	120M

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

Table 6: Comparision of HardLabel Blackbox Extraction on Video Models across TopK

Method	Target Dataset / Victim Model	Victim Train Epochs	Victim Acc.	Clone Accuracy with given top-k classes in %				Query Budget
				1	3	5	10	
DFME[48]	U11‡ / VVT†	800	84.96	49.75	47.15	21.94	-	70M
	U101‡ / VVT†	2000	74.1	31.95	30.56	14.62	3.64	200M
	K400‡ / SwT†	8000	70.8	7.52	4.62	4.72	1.92	350M
	K600‡ / SwT†	10000	68.4	2.3	2.7	2.1	0.48	420M
	SS2‡ / SwT†	17500	61.1	0.69	0.73	1.14	1.5	500M
DFMS-HL [41]	U11‡ / VVT†	800	84.96	52.57	41.71	24.05	-	70M
	U101‡ / VVT†	2000	74.1	30.6	31.42	25.63	11.76	200M
	K400‡ / SwT†	8000	70.8	20.29	22.61	24.44	9.78	350M
	K600‡ / SwT†	10000	68.4	3.82	4.15	5.15	1.36	420M
	SS2‡ / SwT†	17500	61.1	0.94	1.14	1.74	0.93	500M
Vidmodex	U11‡ / VVT†	800	84.96	60.14	52.92	36.14	-	70M
	U101‡ / VVT†	2000	74.1	48.56	53.62	45.39	31.62	200M
	K400‡ / SwT†	8000	70.8	38.02	35.72	32.63	21.62	350M
	K600‡ / SwT†	10000	68.4	36.91	37.26	38.62	18.96	420M
	SS2‡ / SwT†	17500	61.1	23.94	25.62	17.62	7.25	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 7: Comparision of GreyBox Extraction on Image Models

Method	Target Dataset / Victim Model	Surrogate / Percentage (%)	Victim Train Epochs	Victim Acc.	Clone Accuracy in %		Query Budget	Gen Iters
					SL	HL		
DFME[48]	C10‡ / RN18†	C100‡ / 10	1500	97.5	93.95	89.22	5M	1K
	C100‡ / RN34†	IN1K‡ / 10	3500	76.5	67.51	58.92	12.5M	4K
	CT101‡ / EN-B7†	CT256‡ / 10	8000	73.2	59.96	49.11	30M	7K
	CT256‡ / EN-B7†	IN1K‡ / 20	10500	77.1	38.63	32.15	45M	10K
	IN1K‡ / EN-B7†	IN22K‡ / 5	15000	67.3	29.25	22.43	58M	6K
DFMS-(S/H)L [41]	C10‡ / RN18†	C100‡ / 10	1500	97.5	94.86	92.62	5M	1K
	C100‡ / RN34†	IN1K‡ / 10	3500	76.5	69.63	58.12	12.5M	4K
	CT101‡ / EN-B7†	CT256‡ / 10	8000	73.2	63.62	54.85	30M	7K
	CT256‡ / EN-B7†	IN1K‡ / 20	10500	77.1	49.64	47.13	45M	10K
	IN1K‡ / EN-B7†	IN22K‡ / 5	15000	67.3	41.52	28.62	58M	6K
Vidmodex	C10‡ / RN18†	C100‡ / 10	1500	97.5	96.61	93.24	5M	1K
	C100‡ / RN34†	IN1K‡ / 10	3500	76.5	73.96	68.15	12.5M	4K
	CT101‡ / EN-B7†	CT256‡ / 10	8000	73.2	69.16	67.72	30M	7K
	CT256‡ / EN-B7†	IN1K‡ / 20	10500	77.1	71.62	61.29	45M	10K
	IN1K‡ / EN-B7†	IN22K‡ / 5	15000	67.3	54.12	37.16	58M	6K
BlackBox Dissector[49]	C10‡ / RN34†	IN1K‡ / 100	-	91.56	80.47	-	-	-
	CT256‡ / RN34†	IN1K‡ / 100	-	78.4	63.61	-	-	-

†Model Architecture RN-18: ResNet18; RN-34: ResNet34; EN-B7: EfficientNet-B7
‡Dataset MN: MNIST; C10: CIFAR10; C100: CIFAR100; CT101: Caltech101; CT256: Caltech256; IN1K: ImageNet1K; IN22K: ImageNet22K

Table 8: Comparision of GreyBox Extraction on Video Models

Method	Target Dataset / Victim Model	Surrogate Dataset / Percentage (%)	Victim Train Epochs	Victim Acc.	Clone Accuracy in %		Query Budget	Gen Iters
					SL	HL		
DFME[48]	U11‡ / VVT†	U101‡ / 10	800	84.96	58.75	51.72	45M	10K
	U101‡ / VVT†	K400‡ / 20	2000	74.1	52.83	42.73	60M	15K
	K400‡ / SwT†	K600‡ / 20	8000	70.8	35.73	22.67	120M	20K
	K600‡ / SwT†	K700‡ / 20	10000	68.4	32.62	20.63	200M	20K
	SS2‡ / SwT†	CHRD‡ / 10	17500	61.1	28.39	18.52	250M	10K
DFMS-(S/H)L [41]	U11‡ / VVT†	U101‡ / 10	800	84.96	69.14	61.26	45M	10K
	U101‡ / VVT†	K400‡ / 20	2000	74.1	51.20	49.15	60M	15K
	K400‡ / SwT†	K600‡ / 20	8000	70.8	48.39	27.26	120M	20K
	K600‡ / SwT†	K700‡ / 20	10000	68.4	43.51	28.22	200M	20K
	SS2‡ / SwT†	CHRD‡ / 10	17500	61.1	28.01	18.05	250M	10K
Vidmodex	U11‡ / VVT†	U101‡ / 10	800	84.96	73.82	65.25	45M	10K
	U101‡ / VVT†	K400‡ / 20	2000	74.1	69.25	53.62	60M	15K
	K400‡ / SwT†	K600‡ / 20	8000	70.8	64.02	46.93	120M	20K
	K600‡ / SwT†	K700‡ / 20	10000	68.4	53.72	39.42	200M	20K
	SS2‡ / SwT†	CHRD‡ / 10	17500	61.1	42.71	27.74	250M	10K

†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; K700: Kinetics-700; CHRD: CHARADES

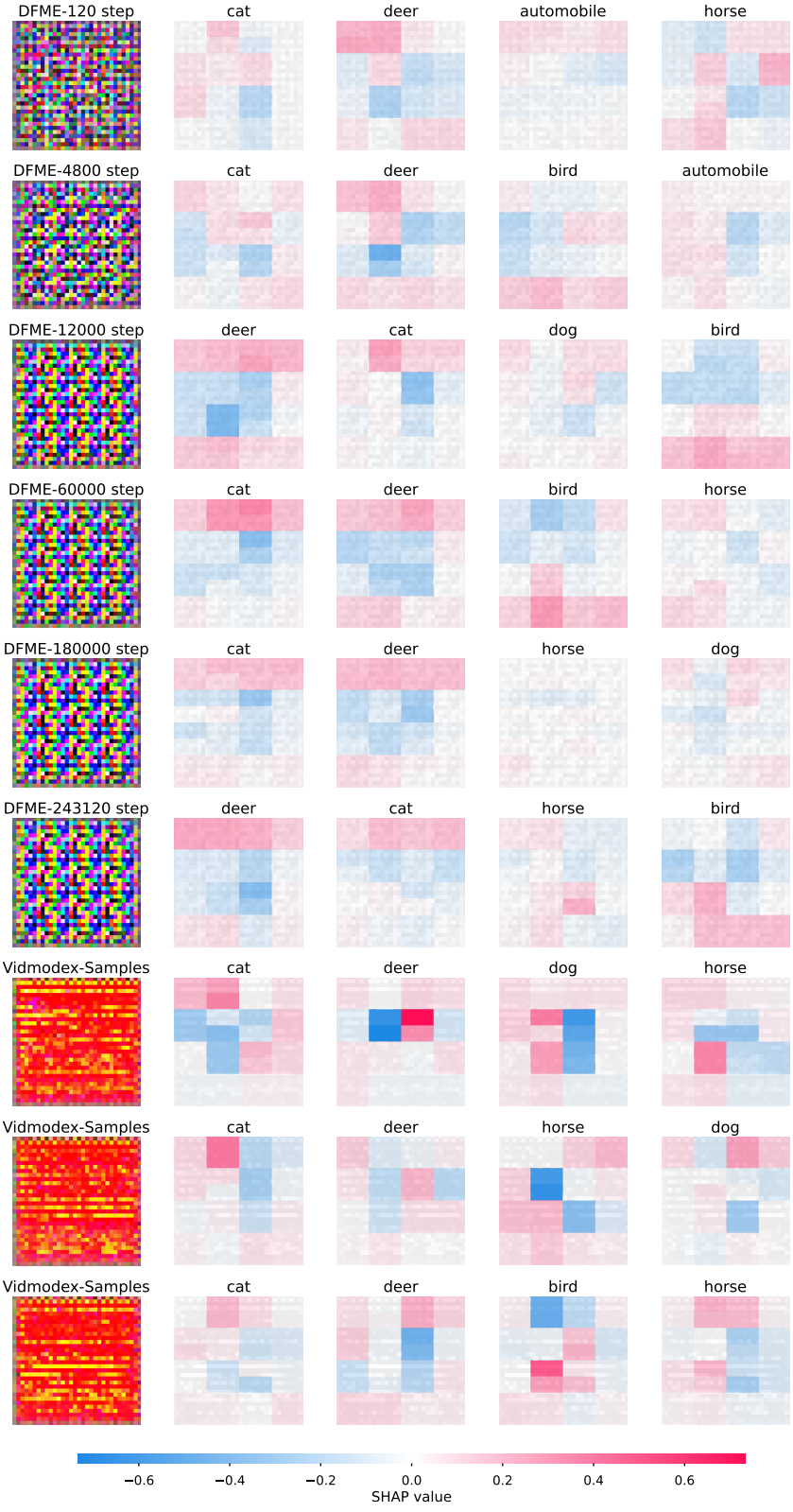


Figure 7: Qualitative analysis of learnt images