ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10887780

Efficient and Effective Model Extraction

*Hongyu Zhu[1], Wentao Hu[1], Sichu Liang[2], Fangqi Li[1], Wenwen Wang[3], Shilin Wang[1]**

[1]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China
[2]School of Artificial Intelligence, Southeast University, China [3]Carnegie Mellon University, USA

***Abstract*—Model extraction aims to steal a functionally similar copy from a machine learning as a service (MLaaS) API with minimal overhead, typically for illicit profit or as a precursor to further attacks, posing a significant threat to the MLaaS ecosystem. However, recent studies have shown that model extraction is highly inefficient, particularly when the target task distribution is unavailable. In such cases, even substantially increasing the attack budget fails to produce a sufficiently similar replica, reducing the adversary's motivation to pursue extraction attacks. In this paper, we revisit the elementary design choices throughout the extraction lifecycle. We propose an embarrassingly simple yet dramatically effective algorithm, Efficient and Effective Model Extraction (E^3), focusing on both query preparation and training routine. E^3 achieves superior generalization compared to state-of-the-art methods while minimizing computational costs. For instance, with only $0.005\times$ the query budget and less than $0.2\times$ the runtime, E^3 outperforms classical generative model based data-free model extraction by an absolute accuracy improvement of over 50% on CIFAR-10. Our findings underscore the persistent threat posed by model extraction and suggest that it could serve as a valuable benchmarking algorithm for future security evaluations.**

***Index Terms*—Model Extraction, Functionality Stealing, Data-Free Knowledge Transfer**

I. Introduction

Machine Learning as a Service (MLaaS) APIs from major companies like Microsoft, Google, and OpenAI provide users with access to powerful deep learning models via a pay-per-query interface. These models span applications such as visual recognition, natural language processing, speech analysis, and code generation, benefiting both industry and everyday life [1] - [4] . However, the public availability of these APIs raises concerns about model theft [5]. Besides direct illegal distribution of model parameters [6] - [8], clients may attempt to reverse-engineer the models through **model extraction**, a tactic that is gaining increasing attention [9].

In model extraction, an adversary queries a victim API and uses the resulting probability vectors to train a surrogate model that approximates the victim's functionality [9] [10]. In data-dependent scenarios, the adversary samples in-distribution (IND) queries, while in data-free scenarios, they lack prior knowledge of the target distribution and synthesize substitute queries to reveal the private knowledge. The process operates in a black-box setting, where the adversary only has access to query inputs and responses, without any information about the victim's architecture, parameters, or gradients. It is assumed that model extraction can achieve comparable performance with minimal effort, bypassing the costs of data labeling and training from scratch. This poses a significant threat to the intellectual property and confidentiality of models [10] - [16] , making model extraction one of the most pressing concerns in industrial machine learning [17].

However, the implicit assumption that model extraction is cheaper in terms of data or computation [10] [18] has recently been questioned [19]. Building a model primarily involves data preparation, computational resources, and expertise. In data-dependent scenarios, adversaries must still gather IND samples, and using the victim as a labeling oracle may not significantly reduce costs compared to established data pipelines [19]. In data-free extraction, while the need for IND samples is avoided, adversaries often rely on computationally intensive generative models to synthesize samples [11] [12], which can demand significantly more queries (e.g., 20 million) and longer training cycles to converge [11] [19]. Moreover, implementing sophisticated extraction algorithms often requires more expertise than training a model from scratch [12] [15]. Thus, achieving a high-performing model with minimal resources is challenging, and the cost-effectiveness of model extraction has been exaggerated.

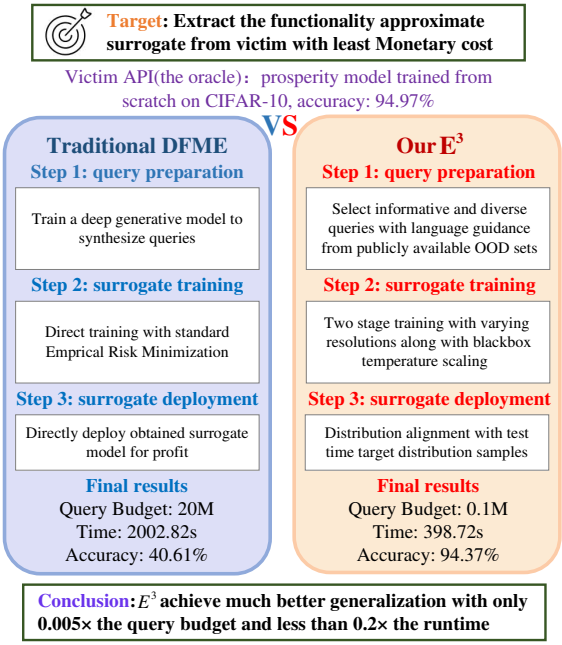

Fig. 1: Comparison of E^3 with traditional DFME.

Does this mean model extraction is entirely ineffective? In this paper, we revisit key design choices throughout the lifecycle of model extraction and reveal the inefficiency of mainstream methods. We further propose surprisingly simple techniques that achieve impressive generalization with minimal cost. For **query preparation**, we show that sophisticated deep generative models are unnecessary when IND samples are unavailable. Instead, publicly available OOD samples, selected via language-guidance, outperform synthetic samples in functionality approximation. For **surrogate training**, low-resolution pretraining significantly reduces costs while enhancing victim guidance. We also mathematically prove, for the first time, that temperature scaling—previously overlooked in the extraction community—is applicable in black-box settings and accelerates convergence while improving generalization. Finally, during **deployment stage**, we introduce a novel perspective where unlabeled target distribution samples naturally encountered by the surrogate can be leveraged for

* Corresponding author: Shilin Wang (wsl@sjtu.edu.cn).

This work was supported by the National Science Foundation of China (62271307).

rapid unsupervised adaptation, bridging the distribution gap between training and testing in data-free scenarios.

By incorporating these nearly cost-free components, we propose a novel E^3 extraction algorithm that achieves state-of-the-art performance with the lowest query budget and computational cost compared to previous methods. This serves as a proof of concept that model extraction continues to pose a significant threat to MLaaS. Figure 1 highlights the effectiveness of our approach.

Our contributions are summarized as follows:

- We highlight the inefficiency of current model extraction techniques, identifying resource-intensive yet ineffective components by revisiting the extraction pipeline.
- At each extraction stage, we introduce computationally efficient strategies that optimize both efficiency and effectiveness under a fixed query budget, resulting in the E^3 algorithm.
- With the lowest query budget and runtime, we achieve significant improvements in generalization, confirming that model extraction remains a substantial threat.

II. Related Work

In model extraction, the adversary $\mathcal{A}$ prepares a query set D^Q to access a *black-box* victim API f_{θ_v}, obtaining response probability vectors $y = f_{\theta_v}(x)$ for each sample $x \in D^Q$. Early *data-dependent* extraction strategies [10] select IND samples as D^Q, sometimes assuming direct access to the victim training set [20]. A surrogate model f_{θ_s} is then trained via knowledge distillation [21] to replicate f_{θ_v}. This process is formalized as:

$$\theta_s^* = \arg\min_{\theta_s} KLD(f_{\theta_v}(x)||f_{\theta_s}(x))^1. \tag{1}$$

Here, $KLD(\cdot||\cdot)$ represents the Kullback-Leibler divergence, measuring the difference between the prediction distributions of f_{θ_s} and f_{θ_v}. Acquiring IND sample, even without labels, is costly and challenging in privacy-sensitive contexts. Consequently, *data-free* extraction, which does not rely on target distribution samples, has garnered widespread attention. Inspired by data-free knowledge distillation [22], synthesizing queries with **deep generative models** has become mainstream [11] [12]. DFME [11] formulates a three-player game between the victim, surrogate and generator using a generative adversarial network framework [24], with zero-order optimization to estimate gradients from the black-box f_{θ_v}. IDEAL [12] optimizes the generator based on the prediction entropy of f_{θ_s}, bypassing gradient estimation on f_{θ_v} and significantly lowering the query budget. However, inverting training samples from the model is challenging, and the synthesized samples often deviate from the original distribution, lacking realistic visual patterns, which hinders the generalization of f_{θ_s}. Moreover, training the generator from scratch also introduces considerable computational overhead.

Another approach involves randomly sampling OOD samples and adding adversarial perturbations [25] to maximize the prediction entropy of f_{θ_v} [13] [14]. These high-entropy **adversarial samples** better characterize the decision boundary of f_{θ_v}, increasing information leakage from OOD queries. However, generating adversarial perturbations is time-consuming, and the responses are often erroneous or ambiguous, limiting f_{θ_s} from learning well-generalized mappings.

Lastly, **active sampling**-based extraction progressively select representative samples from candidate OOD set as D^Q [15] [26] [27]. However, selecting samples with a limited number of queries to f_{θ_v} is challenging, as the adversary must construct structural representation of the entire candidate set and recalculate selection criteria at each iteration. Notably, model extraction has also been adapted for legitimate purposes like model compression as a service [28] [29]. Whether for legitimate or illicit use, designing cost-effective extraction strategies remains crucial.

[1]In model extraction, to reduce labeling costs, cross-entropy between predictions and ground-truth labels is typically omitted from the loss function.

III. Methodology

In this section we elaborate on our proposed strategies in E^3. The first two apply to both data-dependent and data-free scenarios, while the last two are specifically tailored for data-free model extraction.

A. Two-Stage Extraction with Varying Resolution

The ultimate goal of model extraction is to replicate the victim model's functionality at a lower cost than training from scratch. However, without ground-truth supervision, the surrogate often requires the same sample volume as the victim's training set and longer training times to converge [19]. To address this, we propose a two-stage training routine with varying resolutions (VarRes), balancing training cost and generalization.

In image classification, the victim typically processes square images of size $R \times R$. When the query set is limited, using the same input for the surrogate can lead to overfitting. Therefore, in the first stage, we randomly crop a rectangle from the image, rescale it to $r \times r$ (where $\mathbf{r} < \mathbf{R}$), and optimize the following loss function:

$$\theta_s^* = \arg\min_{\theta_s} \mathrm{KL}(f_{\theta_v}(x) \,||\, f_{\theta_s}(\mathrm{transform}(x))) \tag{2}$$

where *transform* refers to the input transformation applied to f_{θ_s}, such as the *RandomResizedCrop* operation [30] in PyTorch [31]. In the second stage, we fine-tune the surrogate at full resolution during the final epochs as in Equation 1. Notably, the victim always receives the original image x as input, ensuring that the query budget remains unchanged. Low-resolution training in the first stage reduces computational costs while encouraging robustness to input scale variations. As shown in Fig. 2, training the surrogate with the same input size as the victim causes the training loss to rapidly converge to zero, limiting further guidance from the victim. In contrast, VarRes preserves a meaningful prediction difference between the victim and the surrogate, enabling the surrogate to continuously absorb task-specific knowledge from the victim. This teacher-student discrepancy is widely regarded as highly advantageous [10] [11] [15].

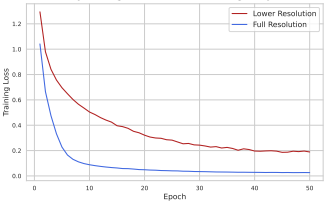

Fig. 2: Training Loss at Low vs Full Resolution Across Epochs.

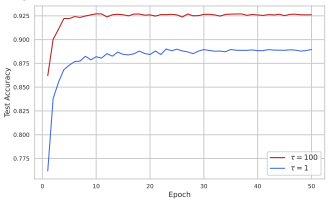

Fig. 3: Test Accuracy with τ = 1 or 100 Across Epochs.

B. Temperature Scaling in Black-box Extraction

When using KL divergence to measure the prediction difference between the victim and the surrogate, the optimization objective can be expressed as:

$$\min -\sum_{j=1}^{c} \sigma_j(z_v) \log \sigma_j(z_s) \tag{3}$$

where z_v and z_s are the output logits of f_{θ_v} and f_{θ_s}, respectively, and are transformed into probability distributions via the softmax function $\sigma(\cdot)$. However, the output of modern deep neural networks tends to be over-confident [32], where only the predicted class t of the victim receives significant weight in $\sigma_t(z_v)$, ignoring relative differences among other classes [33]. This results in near one-hot

labels, making it difficult to capture the "dark knowledge" needed for the surrogate to effectively learn the victim's unique functionality.

To address this, the classic solution in knowledge distillation [21] is to soften the model's predictions with temperature scaling, transforming the prediction probabilities into $\sigma(z_v/\tau)$ and $\sigma(z_s/\tau)$, where τ is the temperature parameter. This reduces class differences in the probabilities, allowing the loss function to focus on matching the entire logits vector [33] [34]. However, in model extraction, the adversary only has access to the black-box probability output $\sigma(z_v)$ from the victim. Since $\sigma(\cdot)$ is not injective and therefore non-invertible [35], it was considered impossible to derive the specific temperature-scaled $\sigma(z_v/\tau)$. As a result, temperature scaling has been overlooked in model extraction [10] - [15].

In this paper, we show that simple elementary operations can derive $\sigma(z_v/\tau)$ from $\sigma(z_v)$ in a mathematically equivalent way.
Proposition: $\forall z \in R^c, \tau \in R, \sigma(z_i/\tau) = \sigma(\log(\sigma(z_i))/\tau)$
Proof Sketch: $\log(\sigma(z_i))/\tau = (z_i - \log(\sum_j \exp(z_j)))/\tau = z_i/\tau + C$, where $C = \log(\sum_j \exp(z_j))/\tau$. Since $\sigma(x) = \sigma(x + C)$ [13], $\sigma(\log(\sigma(z_i))/\tau) = \sigma(z_i/\tau + C) = \sigma(z_i/\tau)$

This is the first demonstration that temperature scaling can be applied to black-box model extraction seamlessly. Fig. 3 shows a performance comparison of the surrogate on CIFAR-10, with and without temperature. Higher temperature significantly accelerates convergence and improves final generalization.

C. Language-Guided Query Selection

In data-free scenarios, the absence of IND samples for the target task makes it difficult to effectively transfer knowledge from the victim to the surrogate. Traditional approaches tackle this by synthesizing substitute samples using generative models or adversarial perturbations, both of which are computationally expensive. In this paper, we find that publicly available OOD samples from the Internet can offer valuable image priors, such as general shape and texture features. While the class semantics may not fully align with the target task, these samples are a cost-effective choice for queries. However, OOD samples that deviate significantly from the target distribution are less effective for knowledge transfer [36] [37]. The challenge, then, is efficiently constructing an optimal query set from these OOD samples. Active sampling [15] progressively select the most informative samples from the OOD set as queries, but significantly increases the computational cost and query budget, as it requires generating representations or confidence scores for all samples using either the victim or surrogate [15] [37].

In this paper, we propose a novel multimodal strategy that leverages a language model to select a query set from an OOD set with a predefined class structure, semantically similar to the target task. Specifically, given a labeled OOD set D_O with class names $\{C_{O_i}\}_{i=1}^{k_O}$ and target task class names $\{C_{T_j}\}_{j=1}^{k_T}$ (which may not overlap), we compute the semantic similarity between each OOD class and the target task using a pretrained language encoder $E_l(\cdot)$ as follows:

$$\text{similarity}_i = \sum_j \frac{\left(E_l(C_{O_i}) \cdot E_l(C_{T_j})\right)}{\left(|E_l(C_{O_i})| \cdot |E_l(C_{T_j})|\right)}$$

$$\text{similarity}'_i = \frac{\text{similarity}_i - \min(\text{similarity}_i)}{\max(\text{similarity}_i) - \min(\text{similarity}_i)}$$

Thus, $\text{similarity}'_i \times \left(\frac{|D^Q|}{|D_O|}\right)$ represents the sampling probability for each class in D_O. This language-guided sampling strategy is extremely efficient, performed in a single pass without requiring additional queries to the victim. Moreover, it is model-agnostic, avoiding the bias towards either the victim or the surrogate that often arises in active sampling strategies.

D. Test-Time Distribution Alignment

Model extraction aims for the surrogate to replicate the behavior of the victim during deployment, ensuring that for a test sample x_{test}, the surrogate's output approximates the victim's conditional distribution, $p_v(y|x_{test})$. Although the surrogate is trained on $p_v(y|x_{train})$, which aligns with the desired behavior $p_v(y|x_{test})$ during testing, the absence of IND samples from the target task causes a mismatch. The marginal distribution of the surrogate's training data $p(x_{train})$ inevitably differs from the deployment distribution $p(x_{test})$, leading to covariate shift [38] and impairing generalization. Recent data-free knowledge transfer methods alleviate this issue through distribution-invariant learning [39], but they increase computational costs and require access to the victim's internal features, which is unfeasible in black-box extraction.

Inspired by test-time classifier adjustment [40], we propose a low-cost solution for unsupervised online adaptation, leveraging the unlabelled target distribution samples that the surrogate naturally encounters post-deployment, which enables the surrogate to better align with the target distribution. We decompose f_{θ_s} into a feature extractor $f_e(\cdot)$ and a classification head $f_c(\cdot)$, where the classification head is parameterized by weights $w \in \mathbb{R}^{c \times d}$ and biases $b \in \mathbb{R}^c$, with c representing the number of classes and d the feature dimension output by $f_e(\cdot)$. In prototype learning [41], each $w_i \in \mathbb{R}^d$ represents the prototype for class i. The classification process of $f_c(\cdot)$ can thus be interpreted as assigning a sample to the class with the highest similarity to its prototypes. Thus, we propose fine-tuning the classification head by updating the prototypes. For each class i, we dynamically maintain a support set $S_i = \{x_{i_j}\}_{j=1}^k$ consisting of the k samples with the lowest prediction entropy that f_{θ_s} predicted as class i. The prototype w_i is updated as follows:

$$w_i = (1 - \alpha) \cdot w_i + \alpha \cdot \sum\nolimits_j f_e(x_{i_j})$$

where α is the interpolation coefficient balancing the old weights and the new prototypes. The fine-tuning of w can be viewed as slightly rotating the classification hyperplane to adapt to shifts in the distribution $p(x)$, thereby aligning the surrogate with the target distribution. Test-Time Distribution Alignment (TTDA) requires no additional labeled data, with updates performed online during testing. It only fine-tunes the classification head without extra forward or backward passes, resulting in minimal computational overhead.

IV. Experiments

A. Experimental Settings

We use ResNet-18 and ResNet-34 [43] trained on CIFAR-10 and CIFAR-100 [42] as victims. ResNet-18 is employed as the surrogate architecture to approximate the victims' functionality. We compare E^3 against classical and SOTA extraction methods from top-tier AI and security conferences. In the data-dependent (DD) scenario, we include Knockoff Nets [10] from CVPR 2019. In the data-free (DF) scenario, we evaluate techniques including DFME [11] from CVPR 2021 and IDEAL [12] from ICLR 2023, based on deep generative models, as well as CloudLeak [13] from NDSS 2020, SPSG [14] from CVPR 2024 based on adversarial perturbations, and Marich [15], the SOTA active sampling strategy from NeurIPS 2023.

For E^3, we use the minimal query budget from the comparison methods in both DD and DF scenarios. On CIFAR-10/100, with full resolution $R = 32$, we use a reduced resolution of $r = 24$ during the first stage of VarRes. For temperature scaling, we set high temperatures of $\tau = 10^2$ and $\tau = 10^3$ in the DD and DF scenarios, respectively. In the language-guided query selection, we leverage the lightweight text encoder from MobileCLIP [44] to generate class

TABLE I: Comparison of E^3 with SOTA model extraction algorithms. AC: Adversarial Capability; DD: Data Dependent; DF: Data-Free. Q_b: Query Budget. ↑ (higher is better), ↓ (lower is better). Best results in **bold**, second-best in •.

AC	Method	CIFAR-10					CIFAR-100				
		$Q_b(\downarrow)$	Victim: Resnet18 Acc: 94.97%		Victim: Resnet34 Acc: 95.94%		$Q_b(\downarrow)$	Victim: Resnet18 Acc: 77.33%		Victim: Resnet34 Acc: 81.88%	
			Time(↓)	Acc(↑)	Time(↓)	Acc(↑)		Time(↓)	Acc(↑)	Time(↓)	Acc(↑)
DD	KnockoffNets	0.05M	395.28•	91.56 ± 0.38	416.9•	91.28 ± 0.86	0.05M	395.2•	71.65 ± 0.26	417.13•	70.97 ± 1.29
	E^3 w/o VarRes	0.05M	394.93•	94.11 ± 0.20•	417.08•	94.48 ± 0.37•	0.05M	395.19•	75.95 ± 0.42•	417.19•	75.72 ± 0.21•
	E^3	**0.05M**	**343.58**	**95.48 ± 0.25**	**366.05**	**95.62 ± 0.14**	**0.05M**	**342.56**	**78.37 ± 0.43**	**364.43**	**79.16 ± 0.47**
DF	DFME	20M	2002.82	40.61 ± 2.39	2093.25	31.18 ± 2.68	20M	2060.73	7.27 ± 0.92	2117.98	5.29 ± 1.34
	IDEAL	0.25M	4049.14	65.22 ± 1.88	4129.13	62.26 ± 1.68	0.3M	4564.98	19.13 ± 0.58	4716.18	17.10 ± 0.40
	CloudLeak	0.5M	1860.66	87.65 ± 0.59	1894.72	85.48 ± 2.13	0.5M	1867.77	53.89 ± 1.69	1884.91	48.82 ± 1.16
	SPSG	5M	1904.18	88.07 ± 0.42	1917.48	85.47 ± 0.49	5M	1909.07	62.37 ± 0.76	1925.53	57.80 ± 1.21
	Marich	0.1M	958.75	90.66 ± 0.41	1029.60	88.85 ± 0.99	0.1M	1171.92	69.49 ± 1.49	1265.86	69.98 ± 1.51
	E^3 w/o VarRes	0.1M	459.85•	94.23 ± 0.17•	482.44•	93.22 ± 0.69•	0.1M	464.27•	**74.94 ± 0.56**	485.72•	**72.71 ± 1.32**
	E^3	**0.1M**	**398.72**	**94.37 ± 0.26**	**421.39**	**94.01 ± 0.07**	**0.1M**	**401.86**	72.31 ± 0.65•	**423.90**	72.34 ± 1.73•

name embeddings, using the LSVRC-2012 [45] as the candidate OOD set. For test-time distribution alignment, α is set to 0.1. The code and detailed settings are available at https://github.com/GradOpt/E3.

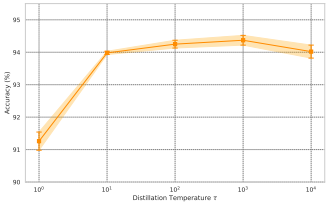

Fig. 4: Test Accuracy with Varying Temperatures.

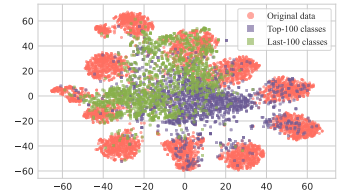

Fig. 5: Visualization of Selected Queries in the Feature Space.

TABLE II: Improvement from TTDA on CIFAR-10 Variants.

Scenario	Acc w/o TTDA	Acc w/ TTDA	Improvement
CIFAR-10	94.35 ± 0.22	94.37 ± 0.26	0.02
CIFAR-10_{34}	93.94 ± 0.06	94.01 ± 0.07	0.07
CIFAR-10_{GN}	79.19 ± 0.85	79.71 ± 0.87	0.52

B. Comparison with SOTA Extraction Algorithms

We compare the query budget, runtime on a single RTX 4090 GPU, and test accuracy of all methods on CIFAR-10/100 in Table I. E^3 consistently outperforms all competitors across different scenarios. VarRes not only reduces runtime but also improves generalization in some cases, allowing the surrogate model to achieve similar functionality to the victim at a lower computational cost. In the DD scenario, E^3 requires the same query budget as KnockoffNets but delivers approximately 4% and 7% higher accuracy, with a shorter runtime. When both the victim and surrogate use the ResNet-18 architecture, the surrogate trained by E^3 even surpasses the victim, benefiting from the regularization effect of knowledge distillation [33], while previous extraction methods failed to achieve similar effects due to their inefficient training routines.

In the DF scenario, VarRes exhibits a slight trade-off between performance and efficiency, though generalization remains strong and could be improved by extending the second stage. In contrast, methods based on generative models requires 5 to 10 × more runtime and a much larger query budget, yet completely failed on more complex tasks like CIFAR-100. Approaches based on adversarial perturbations and active sampling perform slightly better but are still far inferior to E^3. Compared to the top-performing competitor, Marich, E^3 uses less than half the runtime and achieves over 3% higher accuracy on both datasets. Notably, E^3 in the DF scenario surpasses Knockoff Nets with IND queries, with the surrogate closely approximating the labeling oracle, i.e., the victim. Interestingly, when the victim becomes more sophisticated (e.g., using ResNet-34 instead of ResNet-18), most surrogates experience a decline in performance, consistent with previous findings [46]. This can be attributed to the challenge smaller surrogates face in aligning with the victim when the query set is limited. However, E^3 exhibits minor degradation, highlighting its potential to effectively target larger and better victims.

C. Ablation Study

We present a brief analysis of the contributions of each component in E^3. Figure 4 shows the effect of temperature on CIFAR-10. Higher temperatures significantly improve generalization, and beyond $\tau = 10$, accuracy stabilizes, simplifying parameter selection. Figure 5 visualizes the feature space of ResNet-18, comparing the first and last 100 LSVRC class samples selected via language-guided query selection, along with original CIFAR-10 samples. The first 100 classes effectively capture CIFAR-10 features, demonstrating the efficacy of semantic selection. However, the last 100 classes also provide meaningful features, leading us to sample from all classes with normalized probabilities. This ensures both informativeness and diversity, resulting in a 1.22% improvement over CIFAR-100 and 0.37% over random sampling from LSVRC as queries.

Table II summarizes the improvements from test-time distribution alignment (TTDA) across various scenarios. Since the surrogate's performance is already close to that of the victim, TTDA provides only a marginal boost. However, when test samples are corrupted by Gaussian noise, TTDA proves particularly effective, making it valuable in cases of distribution shift between the surrogate and victim deployment. Furthermore, TTDA introduces only about 1% extra latency during deployment and allows model fixation after brief adaptation, resulting in negligible overhead.

V. Conclusion

In this paper, we address the inefficiency of current model extraction by thoroughly revisiting design choices throughout the pipeline. From query preparation to training routine and surrogate deployment, we propose E^3, a novel algorithm that achieves superior generalization with minimal computational overhead. Our strategies can be seamlessly integrated in a plug and play manner to optimize existing and future extraction algorithms. For future work, we plan to extend E^3 to other data modalities and model families [47] [48], and to experiment with larger datasets and real-world APIs.

REFERENCES

[1] Microsoft Azure, "AI Services", 2016, https://azure.microsoft.com/en-us/products/ai-services/

[2] Google Cloud, "Vision API", 2016, https://cloud.google.com/vision/

[3] OpenAI, "Whisper", 2022, https://openai.com/index/whisper/

[4] OpenAI, "Codex", 2021, https://openai.com/index/openai-codex/

[5] D. Oliynyk, R. Mayer, A. Rauber, "I know what you trained last summer: A survey on stealing machine learning models and defences", in ACM Computing Surveys, 2023, 55(14s): 1-41.

[6] T. Nayan, Q. Guo, M. Al Duniawi, et al, "SoK: All You Need to Know About On-Device ML Model Extraction-The Gap Between Research and Practice", in Proceedings of the 33rd USENIX Security Symposium, 2024.

[7] P. Ren, C. Zuo, X. Liu, et al, "DEMISTIFY: Identifying On-device Machine Learning Models Stealing and Reuse Vulnerabilities in Mobile Apps", in Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, 2024.

[8] D. Yang, P. J. Nair, M. Lis, "Huffduff: Stealing pruned DNNs from sparse accelerators", in Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 2023.

[9] F. Tramèr, F. Zhang, A. Juels, et al, "Stealing machine learning models via prediction APIs", in Proceedings of the 25th USENIX Security Symposium, 2016.

[10] 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, 2019.

[11] 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, 2021.

[12] J. Zhang, C. Chen, and L. Lyu, "IDEAL: Query-efficient data-free learning from black-box models", in ICLR 2023.

[13] H. Yu, K. Yang, T. Zhang et al, "CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples", in Proceedings of the NDSS Symposium, 2020.

[14] Y. Zhao, X. Deng, Y. Liu et al, "Fully Exploiting Every Real Sample: SuperPixel Sample Gradient Model Stealing", in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.

[15] P. Karmakar, D. Basu, "Marich: A Query-Efficient Distributionally Equivalent Model Extraction Attack using Public Data", in Proceedings of the Advances in Neural Information Processing Systems, 2023.

[16] H. Zhang, G. Hua, W. Yang, "Poisoning-Free Defense Against Black-Box Model Extraction", in Proceedings of the ICASSP 2024 IEEE International Conference on Acoustics, Speech and Signal Processing, 2024.

[17] K. Grosse, L. Bieringer, T. R. Besold, et al, "Machine learning security in industry: A quantitative survey", in IEEE Transactions on Information Forensics and Security, 2023, 18: 1749-1762.

[18] M. Jagielski, N. Carlini, D. Berthelot, et al, "High accuracy and high fidelity extraction of neural networks", in Proceedings of the 29th USENIX Security Symposium, 2020.

[19] A. Shafran, I. Shumailov, M. A. Erdogdu, and N. Papernot, "Beyond Labeling Oracles - What does it mean to steal ML models?" in Transactions on Machine Learning Research, 2024. Available: https://openreview.net/forum?id=950naKZIyh

[20] H. Zhu, S. Liang, W. Hu, et al. "Reliable Model Watermarking: Defending Against Theft without Compromising on Evasion", in Proceedings of the ACM International Conference on Multimedia, 2024.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network", in Proceedings of the NIPS Workshop, 2015.

[22] G. Fang, J. Song, X. Wang, et al, "Contrastive Model Inversion for Data-Free Knowledge Distillation", in Proceedings of the International Joint Conference on Artificial Intelligence, 2021.

[23] 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, 2021.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al, "Generative Adversarial Nets", in Proceedings of the NIPS Conference, 2014.

[25] C. Szegedy, W. Zaremba, I. Sutskever et al, "Intriguing properties of neural networks", in ICLR 2014.

[26] P. Ren, Y. Xiao, X. Chang et al, "A Survey of Deep Active Learning", in ACM Computing Surveys, 2021, 54(9):1-40.

[27] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli et al, "Exploring Connections Between Active Learning and Model Extraction", in Proceedings of the USENIX Security Symposium, 2020.

[28] Y. Xu, Y. Wang, H. Chen et al, "Positive-Unlabeled Compression on the Cloud", in Proceedings of the Advances in Neural Information Processing Systems, 2019.

[29] J. Hong, L. Lyu, J. Zhou and M. Spranger, "Outsourcing Training without Uploading Data via Efficient Collaborative Open-Source Sampling", in Annual Conference on Neural Information Processing Systems, 2022.

[30] C. Szegedy, W. Liu, Y. Jia, et al. "Going Deeper with Convolutions", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[31] A. Paszke, S. Gross, F. Massa, et al. "Pytorch: An imperative style, high-performance deep learning library", in Advances in neural information processing systems, 2019.

[32] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, "On calibration of modern neural networks", in International conference on machine learning, 2017.

[33] S.Stanton., P. Izmailov, P. Kirichenko, A. A. Alemi, and A. G. Wilson. "Does knowledge distillation really work?", in Advances in Neural Information Processing Systems, 2021.

[34] T. Kim, J. Oh, N. Y. Kim, S. Cho, S. Y. Yun, "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation", in IJCAI, 2021.

[35] Y. Bengio, I. Goodfellow, and A. Courville, "Deep learning", Cambridge, MA, USA: MIT press, 2017.

[36] D. Alvarez-Melis, N. Fusi, "Geometric Dataset Distances via Optimal Transport", in Advances in Neural Information Processing Systems, 2020.

[37] H. Chen, T. Guo, C. Xu, et al, "Learning student networks in the wild", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[38] S. Schneider, E. Rusak, L. Eck, et al, "Improving robustness against common corruptions by covariate shift adaptation", in Advances in neural information processing systems, 2020.

[39] J. Tang, S. Chen, G. Niu, M. Sugiyama, C. Gong, "Distribution shift matters for knowledge distillation with webly collected images", in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.

[40] Y. Iwasawa, Y. Matsuo, "Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization", in Advances in Neural Information Processing Systems, 2021.

[41] J. Snell, K. Swersky, R. Zemel, "Prototypical networks for few-shot learning", in Advances in neural information processing systems, 2017.

[42] A. Krizhevsky, G. Hinton, and others, "Learning multiple layers of features from tiny images," 2009.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[44] P. K. A. Vasu, H. Pouransari, F. Faghri, R. Vemulapalli, and O. Tuzel, "Mobileclip: Fast image-text models through multi-modal reinforced training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15963–15974.

[45] 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, 2009, pp. 248–255.

[46] J. Liang, R. Pang, C. Li, and T. Wang, "Model extraction attacks revisited," in Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, 2024, pp. 1231–1245.

[47] H. Zhu et al., "Improve Deep Forest with Learnable Layerwise Augmentation Policy Schedules," in ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024, pp. 6660–6664.

[48] J. Yuan, H. Chen, R. Luo, and F. Nie, "A Margin-Maximizing Fine-Grained Ensemble Method," *arXiv preprint arXiv:2409.12849*, 2024.