SCIENCE CHINA
Information Sciences

• Supplementary File •

FDDME: Feature divergence-directed data-free model extraction

Ruinan MA[1], Yu-an TAN[1], Yajie WANG[1*], Zuobin YING[2],
Liehuang ZHU[1] & Jianfeng MA[3]

[1]*School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[2]*Faculty of Data Science, City University of Macau, Macau 999078, China*
[3]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*

Appendix A Related work

Appendix A.1 Data-free knowledge distillation

Knowledge Distillation (KD) aims to transfer the functionality of a larger model to a smaller one with minimal loss of accuracy [1,2]. In standard KD, the process is conducted in a cooperative setting, where the teacher model is white-box and the original training data are available. In recent years, a line of studies has relaxed these requirements by removing the dependence on the teacher's training data and even other real surrogate data [3–7], leading to Data-Free Knowledge Distillation (DFKD). Yin et al. [3] proposed DeepInversion, which exploits the teacher model's internal memory of the original training data distribution to optimize noisy inputs and then uses the synthesized images for KD. Addepalli et al. [8] proposed DeGAN for image synthesis, but its generator requires pre-training on task-agnostic real data. Binici et al. [7] used a generative network to model the distribution of previously synthesized images during DFKD, and leveraged it to guide the student for incremental learning, thus alleviating the storage pressure of replay data caused by catastrophic forgetting [9]. Other representative approaches [4–6] also rely on the teacher model's internal information to perform DFKD. Although these methods can construct surrogate models with good performance, they typically require access to the teacher's internal information or backpropagation through it, which limits their applicability in scenarios where the teacher is not fully accessible.

Appendix A.2 Data-free model extraction

Model extraction attacks are conducted from an adversarial perspective, aiming to build a local surrogate model in a non-cooperative scenario. This setting is typically constrained by limited query budgets and restricted feedback from the victim model (e.g., hard labels or truncated confidence scores). Early works such as Jacobian-Based Dataset Augmentation (JBDA) [10] and KnockoffNets [11] leverage a small amount of real data and/or auxiliary datasets to query the victim model, but their performance is often limited by distribution mismatch and insufficient diversity. Roberts et al. [12] used noisy data sampled from different distributions with pixel correlation as query inputs, but it only works on simple MNIST models. Similar to DFKD, Data-Free Model Extraction (DFME) typically adopts an alternating adversarial training strategy between the generator and the clone model. However, with only black-box access to the victim model, one cannot backpropagate through the victim to obtain gradients for updating the generator, making generator optimization non-differentiable. To address this, Truong et al. [13] and Kariyappa et al. [14] employ zeroth-order gradient estimation to approximate victim-model gradients, at the cost of multiple victim queries per update. In the hard-label setting, ZSDB3KD [15] is feasible but requires an extremely large query budget. To improve query efficiency, DFMS_HL [16] introduces a discriminator with an unrelated image prior, enabling generator updates that partially circumvent direct reliance on victim queries. Yuan et al. [19] proposed DFHL-RS, which synthesizes high-entropy queries to steal both the accuracy and adversarial robustness of a victim model. More recently, Rosenthal et al. [17] and Beetham et al. [18] proposed DisGUIDE and DS, respectively. Both methods formulate generator objectives by modeling the output relationships between two local clone models, and optimize the generator via backpropagation through these clones, thereby avoiding additional victim queries during generator updates. However, they do not fully exploit the white-box advantages of local clone models: regardless of the clone architecture, only output relationships are considered. Both traditional real-time systems [20, 21] and deep learning studies [22] suggest that changes in intermediate states can affect the final output behavior, thereby suggesting that leveraging internal information of local clones may be a plausible direction for improving data-free model extraction attacks.

Some data-free adversarial attacks (e.g., [23, 24]) can also obtain local surrogates, but they are mainly designed for generating transferable adversarial examples rather than maximizing task accuracy. High transferability does not necessarily imply high accuracy; for instance, on CIFAR-10, DaST-P and DaST-L achieve only 25.15% and 20.35% task accuracy, respectively, despite attaining a 59.71% transfer attack success rate. Since model extraction emphasizes the final clone accuracy as a functionality guarantee, we exclude such methods from our comparison. Table A1 summarizes representative DFME methods.

* Corresponding author (email: wangyajie19@bit.edu.cn)

Table A1 A systematic comparison of representative methods based on key metrics (unified comparison on CIFAR-10). "Label" denotes the victim model's output format, where Hard (one-hot) is stricter than Soft (class probabilities). "Normalized Accuracy" denotes the ratio of clone accuracy to victim accuracy.

Method	w/o Prior	Label	Query Budget	Normalized Accuracy	Method	w/o Prior	Label	Query Budget	Normalized Accuracy
Noise [12]	×	Soft	20M	0.143x	DFMS_HL [16]	×	Hard	8M	0.885x
JBDA [10]	×	Soft	20M	0.272x	DS [18]	✓	Hard	8M	0.824x
MAZE [14]	✓	Soft	20M	0.477x	DisGUIDE [17]	✓	Hard	8M	0.920x
DFME [13]	✓	Soft	20M	0.923x	Ours	✓	Hard	8M	0.941x

Appendix B Threat model

Victim model. Let $\mathcal{X} \subseteq \mathbb{R}^{H\times W\times C}$ denote the input space and $\mathcal{Y} = \{1, \dots, K\}$ the label space. The victim model is a classifier $V : \mathcal{X} \to [0,1]^K$ that outputs a K-dimensional confidence vector whose entries sum to one, i.e., $\sum_{i=1}^{K} V_i(\mathbf{x}) = 1$ for any $\mathbf{x} \in \mathcal{X}$. The predicted label is $f_V(\mathbf{x}) = \arg\max_{i\in\mathcal{Y}} V_i(\mathbf{x})$. The victim is trained on a private benign data distribution $\mathcal{D}_v$, which is not accessible to the adversary. In addition, its internal parameters, model architecture, and gradient information are all hidden from the adversary.

Adversary's knowledge and capabilities. The adversary interacts with the deployed victim model through a prediction API, formalized as a black-box oracle under a finite query budget Q. For a query input $\mathbf{x}$, the API returns either (i) soft-label (probability) feedback via $\mathcal{O}_V^{\text{soft}}(\mathbf{x}) = V(\mathbf{x})$ or (ii) hard-label feedback via $\mathcal{O}_V^{\text{hard}}(\mathbf{x}) = f_V(\mathbf{x}) \in \mathcal{Y}$. Beyond API access, the adversary cannot access any training procedure details, auxiliary metadata, or other privileged information about V. In the data-free model extraction scenario, the adversary further assumes no real query images from $\mathcal{D}_v$ and no access to any real surrogate dataset from the victim's domain.

Adversary's goal. The adversary aims to train a clone model $C(\cdot;\theta_C)$ whose predictions match those of the victim on the victim's private domain. Here, $C(\mathbf{x};\theta_C) \in \mathbb{R}^K$ denotes the clone logits output of $\mathbf{x}$. The objective can be written as minimizing the decision-level disagreement between the victim and the clone over $\mathcal{D}_v$:

$$\theta_C^* = \arg\min_{\theta_C} \ \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_v}\Big[\mathbb{I}\big(f_V(\mathbf{x}) \neq f_C(\mathbf{x};\theta_C)\big)\Big], \tag{B1}$$

where $f_C(\mathbf{x};\theta_C) = \arg\max_{i\in\mathcal{Y}} \big(\text{softmax}(C(\mathbf{x};\theta_C))\big)_i$ and $\mathbb{I}(\cdot)$ denotes the indicator function.

Since $\mathcal{D}_v$ is inaccessible, the adversary constructs a synthetic query distribution through a generator $G(\cdot;\theta_G) : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z} = \mathbb{R}^d$ and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. The induced query distribution is defined by sampling $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and mapping $\mathbf{x} = G(\mathbf{z};\theta_G)$, i.e., $\mathbf{x} \sim \mathcal{D}_{\text{que}}$. Under the data-free model extraction protocol, the adversary learns the clone parameters θ_C and the generator parameters θ_G (typically via alternating updates) by minimizing an imitation loss between the clone outputs and the victim feedback on synthetic queries:

$$(\theta_C^*, \theta_G^*) = \arg\min_{\theta_C} \max_{\theta_G} \ \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\Big[\mathcal{L}\big(C(G(\mathbf{z};\theta_G);\theta_C),\ \mathcal{O}_V(G(\mathbf{z};\theta_G))\big)\Big], \tag{B2}$$

where $\mathcal{O}_V(\cdot)$ denotes either $\mathcal{O}_V^{\text{soft}}(\cdot)$ or $\mathcal{O}_V^{\text{hard}}(\cdot)$ depending on the feedback type. For the soft-label setting, the imitation loss is computed at the logit level: let $\hat{\ell}_V(\mathbf{x}) \in \mathbb{R}^K$ denote the estimated victim logits inferred from the returned probabilities $V(\mathbf{x})$. We then instantiate $\mathcal{L}_{\text{soft}}\big(C(\mathbf{x};\theta_C), \mathcal{O}_V^{\text{soft}}(\mathbf{x})\big) = \big\|C(\mathbf{x};\theta_C) - \hat{\ell}_V(\mathbf{x})\big\|_1$, where $\mathcal{O}_V^{\text{soft}}(\mathbf{x}) = V(\mathbf{x})$. For the hard-label setting, $\mathcal{L}$ is typically the cross-entropy loss with the one-hot label implied by $\mathcal{O}_V^{\text{hard}}(\mathbf{x})$, i.e., $\mathcal{L}_{\text{hard}}\big(C(\mathbf{x};\theta_C), \mathcal{O}_V^{\text{hard}}(\mathbf{x})\big) = -\sum_{i=1}^{K} \mathbb{I}\big(i = \mathcal{O}_V^{\text{hard}}(\mathbf{x})\big) \log\Big(\text{softmax}(C(\mathbf{x};\theta_C))\Big)_i$ where $\mathcal{O}_V^{\text{hard}}(\mathbf{x}) \in \mathcal{Y}$. Upon successful extraction, the adversary can further use the learned clone model C to craft adversarial examples $\mathbf{x}_{adv}$ (e.g., via gradient-based attacks on C) that transfer to the victim V, based on the standard substitute-attack assumption that a high-fidelity clone tends to induce similar decision behavior and thus facilitates transferability.

Appendix C Methodology

Appendix C.1 Overview of the proposed attack

Under the data-free black-box model extraction setting, the adversary trains a generator $G(\cdot;\theta_G)$ and two randomly initialized clone variants $C_1(\cdot;\theta_1)$ and $C_2(\cdot;\theta_2)$ via an alternating optimization loop under a finite query budget Q. In each iteration, (i) the generator is updated using the two variants to synthesize informative queries by jointly maximizing feature-space discrepancy, encouraging class balance, and promoting decision disagreement through $L_G = \beta L_{FD} - \lambda L_{div} + L_{dis}$; (ii) the variants are then aligned with the victim by querying the prediction oracle $\mathcal{O}_V$ on synthesized samples and minimizing an imitation loss (logit-level ℓ_1 for soft-label feedback or cross-entropy for hard-label feedback); and (iii) a FIFO replay pool $\mathcal{R}$ stores queried pairs to mitigate catastrophic forgetting via intermittent replay updates. We refer to the full method with the decision disagreement term L_{dis} as FDDME_L_{dis}; removing this term (i.e., setting $L_{dis} = 0$) yields FDDME. The overall procedure is summarized in Algorithm C1.

In the soft-label setting, the prediction oracle returns only the victim's probability vector $V(\mathbf{x})$ rather than its internal logits, and the logits are identifiable only up to an additive constant. Therefore, before computing the logit-level ℓ_1 imitation loss, the adversary converts the returned probabilities into a normalized logit representation by taking the element-wise logarithm and removing the per-sample mean to fix the shift ambiguity:

$$\hat{\ell}_V(\mathbf{x}) = \log V(\mathbf{x}) - \frac{1}{K}\sum_{k=1}^{K} \log V_k(\mathbf{x}), \tag{C1}$$

where the logarithm is applied element-wise to $V(\mathbf{x}) \in [0,1]^K$ and $\hat{\ell}_V(\mathbf{x}) \in \mathbb{R}^K$ is the estimated (mean-centered) victim logits used for ℓ_1 matching, ensuring a consistent scale for stable imitation across queries.

Algorithm C1 FDDME-L_{dis}: Full algorithm

Input: Victim oracle $\mathcal{O}_V$ (soft: $\mathcal{O}_V^{\text{soft}}(\mathbf{x}) = V(\mathbf{x})$; hard: $\mathcal{O}_V^{\text{hard}}(\mathbf{x}) = f_V(\mathbf{x})$)
Parameters: Query budget Q, generator iterations g-iter, imitation iterations d-iter, replay iterations rep-iter, batch size B, learning rate η, loss weights β, λ, replay buffer size $|\mathcal{R}|$
Output: Trained variants $C_1(\cdot;\theta_1)$ and $C_2(\cdot;\theta_2)$

Initialize generator $G(\cdot;\theta_G)$, Variant 1 $C_1(\cdot;\theta_1)$, Variant 2 $C_2(\cdot;\theta_2)$
Initialize replay pool $\mathcal{R}$ (FIFO) with maximum size $|\mathcal{R}|$; set query counter $q \leftarrow 0$
while $q < Q$ **do**
  // **Generator Training: update** θ_G **using** C_1 **and** C_2
  for $t = 1$ **to** g-iter **do**
    Sample $\{\mathbf{z}_b\}_{b=1}^B \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$; generate $\mathbf{x}_b = G(\mathbf{z}_b;\theta_G)$
    Extract same-depth features $\mathbf{h}_{1b}$ from $C_1(\mathbf{x}_b)$ and $\mathbf{h}_{2b}$ from $C_2(\mathbf{x}_b)$
    Compute L_{FD} from $\{\mathbf{h}_{1b}\}_{b=1}^B$ and $\{\mathbf{h}_{2b}\}_{b=1}^B$ (e.g., MMD-based feature divergence)
    Compute L_{div} from the variants' consensus predictions on $\{\mathbf{x}_b\}_{b=1}^B$ (class-balance entropy)
    Compute L_{dis} from the decision disagreement between the two variants on $\{\mathbf{x}_b\}_{b=1}^B$
    Update generator: $\theta_G \leftarrow \theta_G - \eta\nabla_{\theta_G}\big(\beta L_{FD} - \lambda L_{div} + L_{dis}\big)$
  end for
  // **Imitation Learning: query** $\mathcal{O}_V$ **and update** θ_1, θ_2
  for $t = 1$ **to** d-iter **do**
    Sample $\{\mathbf{z}_b\}_{b=1}^B \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$; generate $\mathbf{x}_b = G(\mathbf{z}_b;\theta_G)$
    Query victim oracle: $\mathbf{y}_b = \mathcal{O}_V(\mathbf{x}_b)$ // **consumes** B **queries**
    $q \leftarrow q + B$
    for $j \in \{1, 2\}$ **do**
      if soft-label feedback **then**
        Estimate victim logits $\hat{\ell}_V(\mathbf{x}_b)$ from returned probabilities $\mathbf{y}_b$
        $L_{\text{imitate}} \leftarrow \big\|C_j(\mathbf{x}_b;\theta_j) - \hat{\ell}_V(\mathbf{x}_b)\big\|_1$
      else
        $L_{\text{imitate}} \leftarrow \text{CE}\big(C_j(\mathbf{x}_b;\theta_j),\, \mathbf{y}_b\big)$
      end if
      Update Variant j: $\theta_j \leftarrow \theta_j - \eta\nabla_{\theta_j} L_{\text{imitate}}$
    end for
    Update replay pool $\mathcal{R}$ with $\{(\mathbf{x}_b, \mathbf{y}_b)\}_{b=1}^B$ (FIFO, max size $|\mathcal{R}|$)
  end for
  // **Experience Replay: mitigate forgetting via replay updates**
  for $t = 1$ **to** rep-iter **do**
    Sample mini-batch $\{(\tilde{\mathbf{x}}_b, \tilde{\mathbf{y}}_b)\}_{b=1}^B$ from $\mathcal{R}$
    for $j \in \{1, 2\}$ **do**
      if soft-label feedback **then**
        Estimate victim logits $\hat{\ell}_V(\tilde{\mathbf{x}}_b)$ from stored probabilities $\tilde{\mathbf{y}}_b$
        $L_{\text{replay}} \leftarrow \big\|C_j(\tilde{\mathbf{x}}_b;\theta_j) - \hat{\ell}_V(\tilde{\mathbf{x}}_b)\big\|_1$
      else
        $L_{\text{replay}} \leftarrow \text{CE}\big(C_j(\tilde{\mathbf{x}}_b;\theta_j),\, \tilde{\mathbf{y}}_b\big)$
      end if
      Update Variant j: $\theta_j \leftarrow \theta_j - \eta\nabla_{\theta_j} L_{\text{replay}}$
    end for
  end for
end while
Return $C_1(\cdot;\theta_1)$ and $C_2(\cdot;\theta_2)$

Appendix C.2 Robustness reinforcement

From a defender's perspective, data-free model extraction can also serve as a practical testing tool to stress-test a deployed model under strict privacy constraints: it enables generating diverse, in-domain queries without accessing D_v, and can further supports post-hoc robustness enhancement using the extracted clone and final generator. Since the clone is trained via refined imitation learning to closely match the victim's decision behavior, adversarial examples crafted on the clone are more likely to transfer and thus provide meaningful robustness signals for improving V. Concretely, after obtaining a generator G and a clone consisting of two variants C_1, C_2, we synthesize queries $\mathbf{x}' = G(\mathbf{z};\theta_G)$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, assign pseudo-labels using the fused prediction $p(\mathbf{x}') = \frac{1}{2}\sum_{c=1}^2 \text{softmax}(C_c(\mathbf{x}';\theta_c))$, and keep only high-confidence samples with $c' = \max_k p_k(\mathbf{x}') \geqslant \tau$. For each retained $\mathbf{x}'$, a white-box attack on the clone produces an adversarial example $\mathbf{x}_{adv} = \mathbf{x}' + \delta$ with $\|\delta\| \leqslant \epsilon$, yielding triplets $(\mathbf{x}_{adv}, y', c')$ that form the synthetic pool D_{que}. The model owner then fine-tunes the victim V by optimizing a mixed objective that combines standard risk on private clean data D_v and an adversarial term on D_{que} weighted by γ, using a 1:1 ratio of clean samples and synthetic adversarial examples per update. The full procedure is summarized in Algorithm C2.

Appendix D Experiments

Appendix D.1 Experimental settings

Datasets and model architectures. Following previous works [13, 16–18], we evaluate our method on three widely used image classification datasets SVHN, CIFAR-10, and CIFAR-100. We provide a total of four victim models: ResNet-34 with a test accuracy of 96.55% on SVHN, ResNet-34 with a test accuracy of 95.56% on CIFAR-10, ResNet-34 with a test accuracy of 77.95% on CIFAR-100, and ResNet-18 with a test accuracy of 77.39% on CIFAR-100. The test accuracies of these victim models are very close to those reported

Algorithm C2 Robustness reinforcement via synthetic adversarial fine-tuning

Input: Victim model $V(\cdot;\theta_V)$ (owner), private data D_v, extracted generator $G(\cdot;\theta_G)$, extracted clone consisting of two variants $C_1(\cdot;\theta_1), C_2(\cdot;\theta_2)$
Parameters: Synthetic pool size M (e.g., $M = 2|D_v|$), confidence threshold τ, perturbation budget ϵ, loss weight γ, fine-tuning iterations T, batch size B
Output: Reinforced victim model parameters θ_V

Initialize synthetic adversarial pool $D_{que} \leftarrow \emptyset$
// Stage I: Build a synthetic adversarial pool D_{que} using G and the clone
while $|D_{que}| < M$ **do**
  Sample noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and synthesize $\mathbf{x}' = G(\mathbf{z};\theta_G)$
  Compute fused clone prediction $\mathbf{p}' = \frac{1}{2}\sum_{c=1}^{2} \mathrm{softmax}\big(C_c(\mathbf{x}';\theta_c)\big)$
  Set pseudo-label $y' = \arg\max_k p'_k$ and confidence $c' = \max_k p'_k$
  if $c' < \tau$ **then**
    continue
  end if
  Generate adversarial example $\mathbf{x}_{adv} = \mathbf{x}' + \delta$ by a white-box attack on the clone with $\|\delta\| \leqslant \epsilon$ using label y'
  Add $(\mathbf{x}_{adv}, y', c')$ to D_{que}
end while
// Stage II: Owner fine-tunes V with D_v and D_{que} in a 1:1 ratio
for $t = 1$ **to** T **do**
  Sample $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{B/2}$ from D_v (cyclic reuse)
  Sample $\{(\mathbf{x}_{adv,i}, y'_i, c'_i)\}_{i=1}^{B/2}$ from D_{que} (without reuse until exhausted)
  Compute

$$L \leftarrow \frac{2}{B}\sum_{i=1}^{B/2} L(\theta_V, \mathbf{x}_i, \mathbf{y}_i) + \gamma \cdot \frac{2}{B}\sum_{i=1}^{B/2} L(\theta_V, \mathbf{x}_{adv,i}, y'_i)$$

  Update victim model: $\theta_V \leftarrow \theta_V - \eta\nabla_{\theta_V} L$
end for
Return θ_V

in previous work. For a fair comparison, we adopt the same victim model-clone model architectural correspondence as DFMS [16] and DisGUIDE [17]. In all experiments, we use ResNet-18 (Res-18) as the clone model architecture. Specifically, for both SVHN and CIFAR-10 models in soft-label and hard-label settings, we use ResNet-34 as victim model. For CIFAR-100 models, we select ResNet-34 as the victim model in the soft-label setting and ResNet-18 in the hard-label setting. We employ the same generator as [13, 17, 18]. To further evaluate the architectures adaptability of our method, we additionally select LeNet, MobileNet-v2 (Mobile-v2), DenseNet-121 (Dense-121), VGG-11, GoogleNet (Google), ViT-B/16, and ResNet-34 (Res-34) as clone model architectures and conduct experiments.

Baselines. To demonstrate the effectiveness of our method, we compare it with several data-free model extraction methods, including DFME [13], MAZE [14] and DFMS [16] with a single-clone model architecture, as well as DS [18] and DisGUIDE [17] with a dual-clone model architecture. Additionally, based on the dual-clone model architecture, we remove the generator and replace its synthesized queries with alternative images, forming two comparison methods: Original and Noise [12]. Specifically, Original refers to querying with the victim's training set, serving as an upper bound when the victim's training data are available. Noise refers to using query images sampled from different noise distributions. The results of Original and Noise can reflect the quality and utility of synthetic images to a certain extent. In the method of using a dual-clone model architecture, we report the final soft-vote ensemble accuracy. Furthermore, we refer to the method of retraining a model using the same training set as the victim model as Retrain.

Hyperparameter settings. All experiments are conducted on an NVIDIA 4090 GPU. The batch size is uniformly set to 128 for generator training, imitation learning, and replay data review. In each iteration, these three processes are executed sequentially 1, 1, and 4 times, respectively. The generator takes as input a one-dimensional noise vector of length 256, sampled from a standard Gaussian distribution. The replay data pool has a fixed length of 1 million across all tasks. For the SVHN model, the query budget is 2M for both soft-label and hard-label settings. For CIFAR-10, it's 20M in soft-label and 8M in hard-label settings, while for CIFAR-100, the budget is 10M in both settings. FDDME employs a multistep scheduler, reducing the learning rate by a factor of 0.3 at 40% and 80% of the query budget. The clone models use SGD with an initial learning rate of 0.03, while the generator uses Adam with a learning rate of 1×10^{-4}. We empirically set the class diversity loss weight $\lambda = 0.2, 0.2, 0.05$, $\beta = 1 \times 10^{-3}$ for SVHN, CIFAR-10 and 1×10^{-4} for CIFAR-100, respectively. In the robustness reinforcement experiment, all attacks are l_∞-bounded. We generate adversarial examples on the standard test set using FGSM [25], PGD [26], MI-FGSM [28], VMI-FGSM [29], and SINI-FGSM [30] with ϵ=8/255. Except for FGSM, all attacks use T=20 iterations. We use SGD with a learning rate of 1×10^{-3}, momentum of 0.9, and weight decay of 5×10^{-4} for robustness reinforcement, where we apply confidence filtering with threshold $\tau = 0.90$, set the robustness loss weight to $\gamma = 0.5$, and use a perturbation budget of $\epsilon = \frac{3}{255}$ when generating synthetic adversarial examples for fine-tuning.

Appendix D.2 Comparison with other baselines

Effectiveness of our method. Table D1 reports the final ensemble clone accuracy and the corresponding normalized accuracy across datasets and feedback settings. Overall, FDDME is competitive with strong baselines, and FDDME-L_{dis} achieves the best results in all settings except CIFAR-100 under hard-label feedback. Compared to Noise, which also relies on non-real query inputs, our method consistently yields substantially higher normalized accuracy, highlighting the utility of generator-driven synthetic queries. In addition, FDDME outperforms Original in most settings, benefiting from the broader and inexhaustible synthetic query stream under a fixed query budget, while Original serves as an upper bound when the victim's training data are available.

On challenging settings such as CIFAR-100, methods that suffer from insufficient class diversity or weak supervisory signals can lead to a pronounced accuracy drop. For example, DFME can be limited by the class diversity of synthesized queries and by the

Table D1 Final ensemble clone accuracy and normalized accuracy comparison of different data-free model extraction methods. Normalized accuracy refers to the ratio of a clone model's task accuracy to the victim model's task accuracy.

Setting	Methods	SVHN			CIFAR-10			CIFAR-100		
		Query Budget	Victim (%)	Clone (%)	Query Budget	Victim (%)	Clone (%)	Query Budget	Victim (%)	Clone (%)
Soft label	Original	2M	96.55	94.21 (0.976x)	20M	95.85	93.04 (0.971x)	10M	77.60	68.48 (0.882x)
	Noise [12]	2M	96.55	52.86 (0.547x)	20M	95.85	13.66 (0.142x)	10M	77.60	3.87 (0.050x)
	DFME [13]	2M	96.20	95.33 (0.991x)	20M	95.54	88.10 (0.922x)	10M	77.99	26.46 (0.339x)
	MAZE [14]	2M	96.20	91.10 (0.947x)	20M	95.54	45.60 (0.477x)	10M	77.99	17.81 (0.228x)
	DFMS_SL [16]	2M	96.55	95.50 (0.989x)	20M	95.59	91.24 (0.954x)	10M	78.52	48.83 (0.622x)
	DS [18]	2M	96.20	95.72 (0.995x)	20M	95.50	91.34 (0.956x)	10M	77.99	46.52 (0.596x)
	DisGUIDE [17]	2M	96.55	95.90 (0.993x)	20M	95.54	94.02 (0.984x)	10M	77.52	69.47 (0.896x)
	FDDME(Ours)	2M	96.55	95.88 (0.993x)	20M	95.56	90.10 (0.943x)	10M	77.95	62.32 (0.799x)
	FDDME_L_{dis}(Ours)	2M	96.55	**96.12 (0.996x)**	20M	95.56	**94.34 (0.987x)**	10M	77.95	**71.65 (0.919x)**
Hard label	Original	2M	96.55	91.53 (0.907x)	8M	95.85	86.94 (0.907x)	10M	77.39	**63.49 (0.820x)**
	Noise [12]	2M	96.55	36.86 (0.382x)	8M	95.85	10.79 (0.113x)	10M	77.32	1.40 (0.020x)
	DFME [13]	2M	96.20	93.87 (0.976x)	8M	95.54	68.40 (0.716x)	10M	77.99	7.13 (0.091x)
	MAZE [14]	2M	96.55	65.30 (0.676x)	8M	95.54	32.65 (0.342x)	10M	77.99	6.24 (0.080x)
	DFMS_HL [16]	2M	96.20	95.56 (0.990x)	8M	95.59	84.51 (0.884x)	10M	78.52	43.56 (0.555x)
	DS [18]	2M	96.20	95.43 (0.992x)	8M	95.50	78.72 (0.824x)	10M	77.99	9.77 (0.125x)
	DisGUIDE [17]	2M	96.55	95.71 (0.991x)	8M	95.54	87.93 (0.920x)	10M	77.39	60.52 (0.782x)
	FDDME(Ours)	2M	96.55	95.22 (0.986x)	8M	95.56	83.31 (0.872x)	10M	77.39	58.04 (0.750x)
	FDDME_L_{dis}(Ours)	2M	96.55	**95.92 (0.993x)**	8M	95.56	**89.92 (0.941x)**	10M	77.39	61.66 (0.797x)

Table D2 Comparison under different perspectives. Both the victim model and the clone model use the ResNet-34 architecture, and all victim models are in the soft-label setting. We evaluate the agreement between the clone model and the victim model on both Non-Problem Domain (NPD) and Problem Domain (PD) images.

		V_acc (%)	C_acc (%)↑	Nor_acc↑	Agree NPD(%)↑	Agree PD(%)↑	DB_Err↓	Dis_Eq↓	Param↓
SVHN	Retrain	96.55	**96.18**	**0.996x**	44.55	96.93	**0.015**	0.073	3.421
	DFME	96.55	94.33	0.977x	62.64	95.29	0.088	0.112	3.423
	DisGUIDE	96.55	96.00	0.994x	66.74	97.55	0.031	0.041	3.325
	FDDME_L_{dis}	96.55	96.16	**0.996x**	**67.36**	**97.91**	0.026	**0.032**	**3.297**
CIFAR-10	Retrain	95.85	**95.35**	**0.995x**	66.10	95.76	1×10^{-4}	0.150	3.378
	DFME	95.85	90.36	0.943x	66.23	91.85	0.083	0.317	3.243
	DisGUIDE	95.85	94.47	0.986x	83.61	96.32	0.008	0.047	3.218
	FDDME_L_{dis}	95.85	95.16	0.993x	**84.02**	**98.06**	0.006	**0.040**	**3.209**
CIFAR-100	Retrain	77.60	**77.36**	**0.997x**	51.70	**81.64**	2×10^{-4}	**0.448**	3.824
	DFME	77.60	24.63	0.317x	7.48	26.17	0.760	2.843	3.585
	DisGUIDE	77.60	69.90	0.900x	55.30	77.57	0.204	0.603	3.502
	FDDME_L_{dis}	77.60	72.46	0.934x	**56.81**	79.12	0.191	0.589	**3.490**

approximation errors introduced by zeroth-order optimization, especially under hard-label feedback. Moreover, DS constrains generator optimization by a triangle-inequality-based objective, which makes it difficult to incorporate explicit class diversity regularization and thus results in degraded performance on CIFAR-100. In contrast, FDDME_L_{dis} strengthens generator learning by combining (i) feature-space divergence from internal representations of the clone variants and (ii) decision disagreement regularization, which jointly improves the informativeness and coverage of synthetic queries and leads to consistently better extraction quality across tasks.

Comparison under different perspectives. Beyond task accuracy (i.e., the standard test accuracy of the extracted clone), we further evaluate how closely the extracted clone matches the victim from multiple complementary perspectives. Let f_V and f_C denote the victim and clone predictors, respectively, and let $\mathcal{X}$ denote the evaluation data distribution. We consider the following metrics. *(1) Agreement (Agree).* This measures the consistency between the predictions of f_C and f_V on the same inputs, i.e., the fraction of samples where $f_C(\mathbf{x}) = f_V(\mathbf{x})$. We report Agree on both Problem-Domain (PD) images (the standard test set of the task) and Non-Problem-Domain (NPD) images (test sets from other tasks). Concretely, we use the CIFAR-100 test set as NPD images for SVHN and CIFAR-10, and use the CIFAR-10 test set as NPD images for CIFAR-100. *(2) Decision Boundary Reconstruction Error (DB_Err).* This measures the decision-level discrepancy between f_V and f_C on the victim's training-data distribution, defined as $\varepsilon = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\mathbb{I}(f_V(\mathbf{x}) \neq f_C(\mathbf{x}))\right]$, where $\mathbb{I}(\cdot)$ is the indicator function. *(3) Distributional Equivalence (Dis_Eq).* This measures the similarity of the overall prediction distributions of f_C and f_V on the task test set, instantiated as the KL divergence between their predictive distributions. *(4) Parametric Fidelity (Param).* This is only computable when f_C and f_V share the same architecture, defined as $F_{\mathbf{w}} \triangleq \log\|\mathbf{w}_C - \mathbf{w}_V\|_2$. In this section, both f_C and f_V use ResNet-34, and all victim models are in the soft-label setting.

Retrain is trained using the same labeled training set as the victim and does not rely on any query interaction, hence it typically achieves very small DB_Err because both models are fitted on the same data distribution. In contrast, data-free extraction methods must recover the victim's behavior solely from query feedback. In Table D2, Dis_Eq and Agree on PD images reflect the similarity of predictive distributions and decision consistency on the task domain, respectively, and FDDME_L_{dis} consistently performs competitively on these metrics. Moreover, Agree on NPD images evaluates out-of-domain generalization of behavioral matching; here FDDME_L_{dis} substantially surpasses Retrain on SVHN and CIFAR-10, suggesting that the extracted clone captures more generalizable decision behavior from the victim's probability feedback rather than only replicating in-domain outcomes. Finally, when architectures are matched, Param indicates how close the learned parameters are; FDDME_L_{dis} yields lower Param than Retrain across all three datasets, implying closer parameter-space proximity under the same backbone.

Efficiency of our method. Figure D1 illustrates the task-accuracy growth under increasing query budgets (soft-label) on SVHN and CIFAR-10. On CIFAR-10, both DisGUIDE and FDDME_L_{dis} surpass 90% accuracy within the first 6M queries, and our accuracy

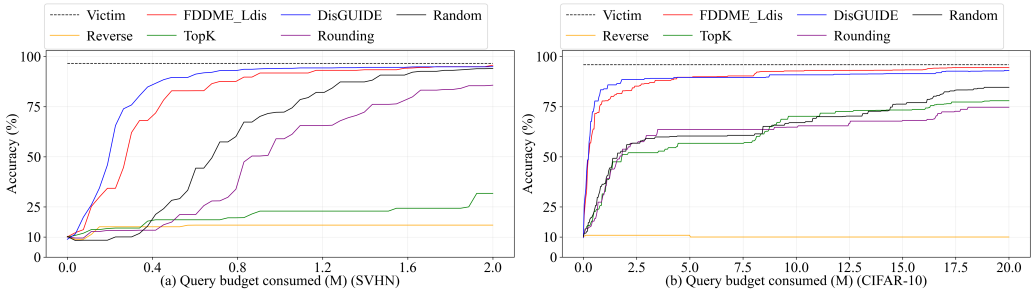

Figure D1 Visualization of clone accuracy growth on the SVHN and CIFAR-10 task for DisGUIDE (blue line), FDDME-L_{dis} (red line), and FDDME-L_{dis} under different defense mechanisms (other colors) in the soft-label setting.

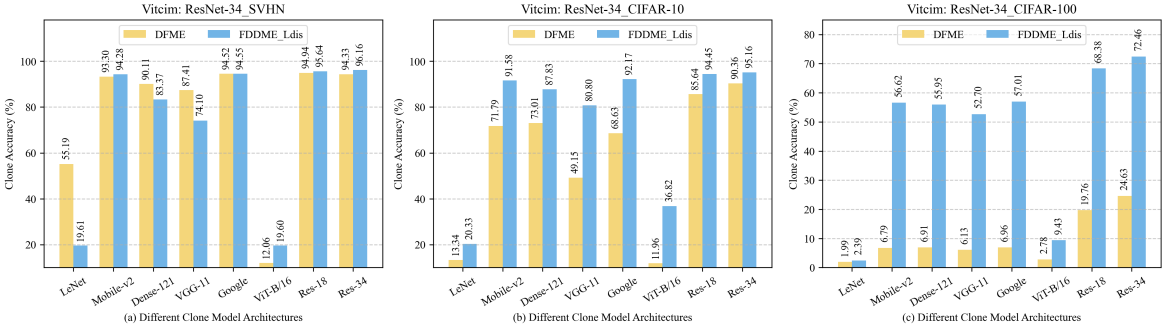

Figure D2 Task accuracy of clone models under different clone model architectures. All victim models are in the soft-label setting.

improves rapidly in the early stage, suggesting that the generator can quickly produce informative synthetic queries. Moreover, FDDME-L_{dis} (final accuracy 94.34 ± 0.19) reaches DisGUIDE's final accuracy using 12.64% fewer queries (17.47M vs. 20M). Similar gains are observed in other settings: it requires 18.80% fewer queries (6.50M vs. 8M) on CIFAR-10 hard-label with 89.92 ± 0.20 final accuracy, 20.32% fewer queries (7.97M vs. 10M) on CIFAR-100 soft-label with 71.65 ± 0.94 final accuracy, and 12.00% fewer queries (8.80M vs. 10M) on CIFAR-100 hard-label with 61.66 ± 1.72 final accuracy. All results are averaged over three runs. These trends indicate that incorporating feature divergence provides additional training signals that facilitate more query-efficient extraction.

Appendix D.3 Effectiveness of our method against defense mechanisms

In practical deployments, the victim model's API outputs may be protected by defense mechanisms that intentionally perturb or truncate the returned probabilities. Such defenses typically preserve normal user experience (since the top-1 prediction is largely unchanged), but can substantially affect model extraction in the soft-label setting by distorting the supervision signal. We consider four prediction-poisoning defenses: *Random* (adding random perturbations to probability values), *TopK* (returning only the top-3 probabilities and suppressing the rest), *Rounding* (rounding probabilities to two decimal places), and *Reverse* (injecting noise in the inverse-sigmoid space of the output probabilities). These defenses have little impact on benign-image accuracy (within 1% in our experiments). Although the convergence speed and final accuracy of FDDME-L_{dis} are affected to varying degrees, it still achieves over 75% clone accuracy in most cases. An exception is *Reverse*, under which FDDME-L_{dis} fails on both SVHN and CIFAR-10. Intuitively, Reverse perturbs the softmax output through an inverse-sigmoid-based transformation that strongly reshapes the probability distribution; in our implementation, we set β=1.0 and γ=1.0, yielding a particularly strong perturbation. For example, the original maximum-class confidence (e.g., 0.99) can be reduced to around 0.11, while the remaining mass becomes nearly uniform across other classes. This introduces severely corrupted "dark knowledge" during imitation learning, effectively overwhelming the supervision with structured noise and preventing accurate extraction. We also observe that FDDME-L_{dis} performs poorly under TopK on SVHN, likely because truncating non-top classes to zero removes informative soft-label gradients and makes the supervision insufficient under a limited query budget (2M).

Appendix D.4 Model architecture adaptability of our method.

In practical model extraction, the adversary may not know the victim's architecture in advance and thus needs to choose a local clone architecture to imitate the victim based only on query feedback. To evaluate the architecture adaptability of our method, we test whether high-accuracy clones can still be obtained when the clone architecture differs from the victim. Figure D2 compares the task accuracy of clones with different architectures under the same victim model. For a fair comparison, when attacking the same victim, we keep all configurations (e.g., query budget and training schedule) identical to the previous experiments and vary only the clone architecture. When the clone architecture changes, we compute the feature divergence loss using the penultimate-layer features (i.e., the activations right before the final fully connected layer) extracted from the corresponding clone variants, so that feature discrepancy can be consistently measured across heterogeneous backbones.

We observe that LeNet performs substantially worse than other architectures. This is expected because LeNet has very limited capacity (only 5 sequential layers and about 0.062M parameters, compared to the victim's 21.28M), which restricts its ability to fit

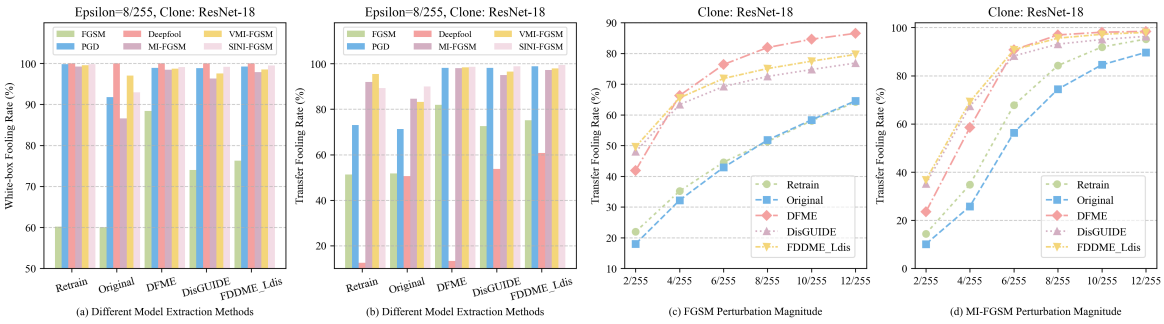

Figure D3 (a) White-box adversarial attack success rates of different attacks on the extracted clone models. (b) TFR of different adversarial attacks on the victim model. (c) and (d) TFR of FGSM and MI-FGSM under different perturbation magnitudes on the victim model (with attack models obtained by different model extraction methods).

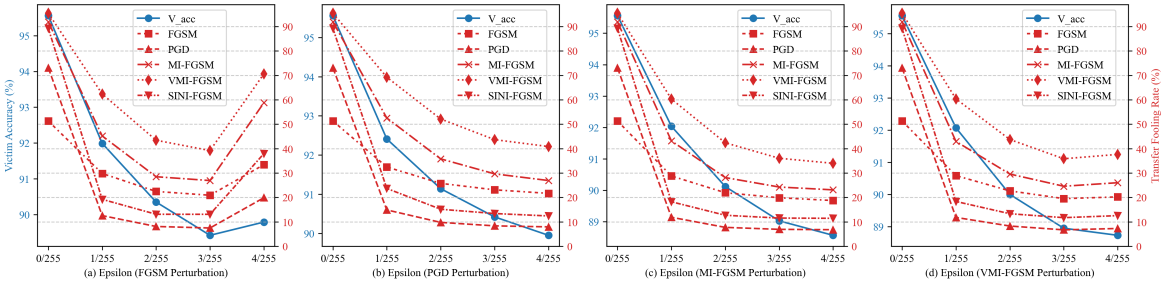

Figure D4 The trade-off between robustness and accuracy when using adversarial examples generated by different attack methods with varying perturbation magnitudes for robustness reinforcement.

the victim behavior, especially when the supervision comes from synthetic queries that may be distributionally atypical. In contrast, ResNet-18 and ResNet-34 achieve the best performance across tasks, largely because they share similar inductive biases and architectural components with the ResNet-based victims. More generally, when the clone family differs significantly from the victim (e.g., Transformers such as ViT-B/16 in Figure D2), all methods degrade noticeably. We attribute this to both capacity/optimization differences and mismatched feature extractors: architectural variations (e.g., residual connections in ResNets, depth-wise separable convolutions in MobileNetV2, and stacked convolutions in VGG) lead to different internal representations, while CNNs emphasize local patterns whereas ViTs rely on global dependencies via self-attention. In data-free settings, the generated images may contain weak semantic structure, which can further limit the effectiveness of ViTs that depend on meaningful global context.

Appendix D.5 Effectiveness of robustness reinforcement

Metric: Transfer Fooling Rate (TFR). We measure adversarial transferability in the standard transfer-based black-box setting: adversarial examples are crafted on a local attack model (e.g., an extracted clone) and then evaluated on the deployed victim. Given N benign inputs $\{\mathbf{x}_i\}_{i=1}^N$, let $\tilde{y}_i = f_V(\mathbf{x}_i)$ denote the victim's prediction on the clean input, and let an attack $\mathcal{A}$ (run on the local attack model) produce $\mathbf{x}_i^{adv} = \mathcal{A}(C, \mathbf{x}_i, \tilde{y}_i; \epsilon)$ under perturbation budget ϵ. We define the *Transfer Fooling Rate* (TFR) as the fraction of transferred adversarial examples that flip the victim's decision (relative to its clean prediction):

$$\text{TFR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\Big(f_V(\mathbf{x}_i^{adv}) \neq \tilde{y}_i\Big) \times 100\%, \tag{D1}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Transferability of adversarial examples. Figure D3(a) reports the white-box fooling rates of different attacks on the extracted clone models, and Figure D3(b) reports the corresponding TFR on the victim model under $\epsilon = \frac{8}{255}$ on CIFAR-10. When evaluating DeepFool [27] (aiming for minimal perturbations) with T=10, we observe that FDDME-L_{dis} yields the highest TFR, indicating that its extracted clone has a highly aligned decision boundary with the victim. Figures D3(c) and (d) further show how TFR varies with the perturbation magnitude for FGSM and MI-FGSM, respectively, where FDDME-L_{dis} consistently achieves strong transferability.

To better approximate a practical setting where the adversary does not interact with the victim during evaluation, we use the clone obtained by Retrain as the attack model and craft adversarial examples. Concretely, for CIFAR-10 we generate l_∞-bounded adversarial examples on the standard test set using FGSM, PGD, MI-FGSM, VMI-FGSM, and SINI-FGSM at $\epsilon = \frac{8}{255}$ (except FGSM, T=20 iterations), and then directly evaluate the victim on these transferred inputs to obtain TFR before and after robustness reinforcement. Before robustness reinforcement, the victim's TFRs under these five attacks are 51.32%, 73.09%, 91.93%, 95.47%, and 89.26%, respectively.

Robustness reinforcement results and analysis. Figure D4 summarizes the effectiveness of robustness reinforcement by varying the perturbation magnitude for adversarial examples generated by different attacks on the extracted clone (FDDME-L_{dis}). The results corresponding to $\epsilon = \frac{0}{255}$ reflect the original task accuracy of the victim model and the baseline transferability prior to robustness reinforcement. Overall, a slight decrease in the task accuracy of the victim can substantially reduce the TFR across a range of attacks

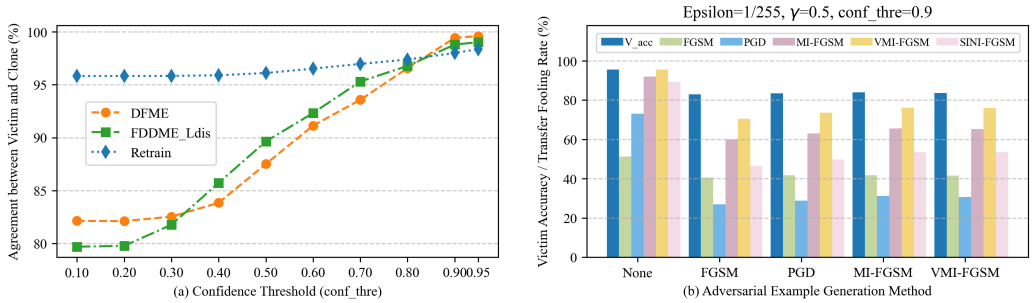

Figure D5 (a) Agreement between the clone model and the victim model under different confidence filter thresholds. (b) Results of applying robustness reinforcement to the victim model using the generator and clone model from DFME.

and perturbation magnitudes. Importantly, all adversarial examples used for robustness reinforcement are generated from synthetic images produced by our generator, rather than any training or test images of the victim model.

We further conduct the same robustness reinforcement experiment using the generator and clone model from DFME at $\epsilon = \frac{1}{255}$. As shown in Figure D5(b), the robustness reinforcement strategy remains effective, indicating that it is not specific to a particular extraction pipeline. To understand why confidence filtering is beneficial, Figure D5(a) reports the prediction agreement between the extracted clone and the victim on synthetic images under different confidence thresholds. As the confidence threshold increases, the agreement also increases, suggesting that high-confidence synthetic samples are more likely to be labeled consistently by the victim, thereby providing more reliable training signals during fine-tuning.

Appendix D.6 Ablation studies

The importance of L_{div}. Since the synthetic queries are class-unconditional, the induced label distribution during imitation learning can easily become highly imbalanced, which in turn degrades the clone accuracy. Figure D6 illustrates the effect of the class-balance term L_{div} by counting, throughout training, the number of synthetic images that are assigned to each class by the victim model. Without L_{div}, the pseudo-label distribution becomes severely skewed: on CIFAR-10, the final clone accuracies drop to 50.31% (soft-label) and 27.12% (hard-label). The issue is even more pronounced on CIFAR-100: more than one-third of the classes never receive any corresponding synthetic samples during the entire extraction process, leading to only 17.12% (soft-label) and 11.08% (hard-label) final accuracies. Although L_{div} is computed from the clone variants, its balancing effect transfers well to the victim's predictions (see the blue bars in Figure D6(a) and (b)). This is because, as extraction progresses, the clone decision boundary becomes increasingly aligned with the victim, making the clones a reliable proxy for shaping a more balanced query distribution.

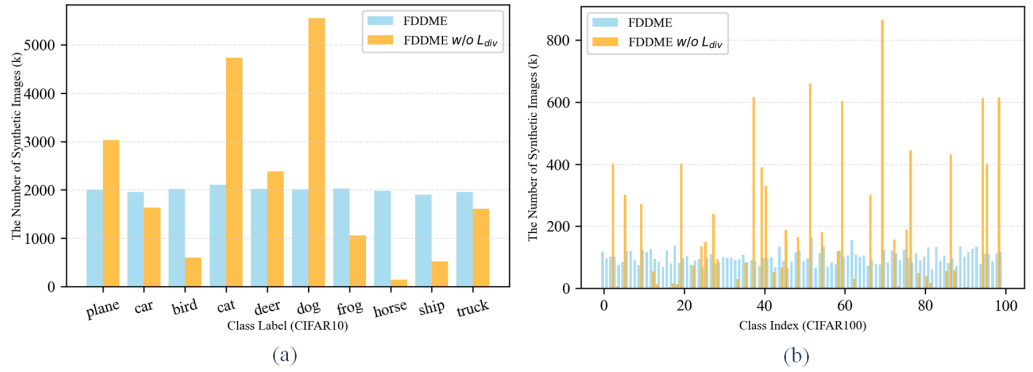

Figure D6 In the soft-label setting on CIFAR-10 and CIFAR-100, the number of synthetic images assigned to each class by the victim model during the imitation learning stage of FDDME.

The impact of feature divergence measures. Figure D7(b) compares different choices of feature divergence measures for implementing L_{FD} in FDDME. We observe that using MSE (mean squared error) or ℓ_1 distance leads to noticeably worse performance than distribution-level measures such as MMD and Wasserstein distance. A plausible reason is that MSE and ℓ_1 primarily quantify pointwise discrepancies between paired feature samples, whereas MMD and Wasserstein better capture the mismatch between two *feature distributions* by accounting for cross-sample relationships. We also experimented with KL and JS divergence, but these variants failed to converge, likely because intermediate feature maps are not naturally normalized as probability distributions.

The impact of replay parameters. The replay mechanism introduces an explicit trade-off between runtime and extraction quality: sampling more batches from the replay pool $\mathcal{R}$ increases the overall execution time, while more frequent replay updates generally help stabilize training and improve the final clone accuracy. Figure D7(a) reports this trade-off for FDDME_L_{dis}. Although increasing the replay frequency can yield higher accuracy, the marginal gain becomes small when the number of replay batches is large, and may not justify the additional computation. Based on this observation, we set the replay frequency to four mini-batches per iteration, which provides a good balance between accuracy and runtime; further increasing it beyond 4 only brings slight improvements.

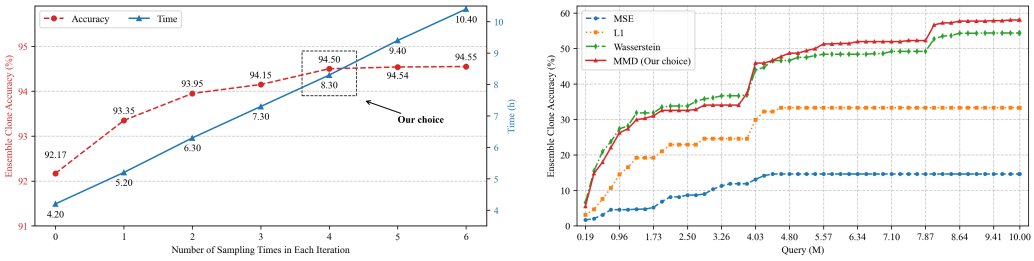

Figure D7 (a) The trade-off between replay data sampling frequency and training time (CIFAR-10, soft-label). (b) The impact of different feature divergence measures on FDDME (CIFAR-100, hard-label).

Impact of robustness reinforcement hyperparameters. In Figure D8(a) and (b), we ablate the adversarial loss weight γ and the confidence filtering threshold τ used to select synthetic samples. We set $\gamma = 0.5$ and $\tau = 0.9$ based on this study.

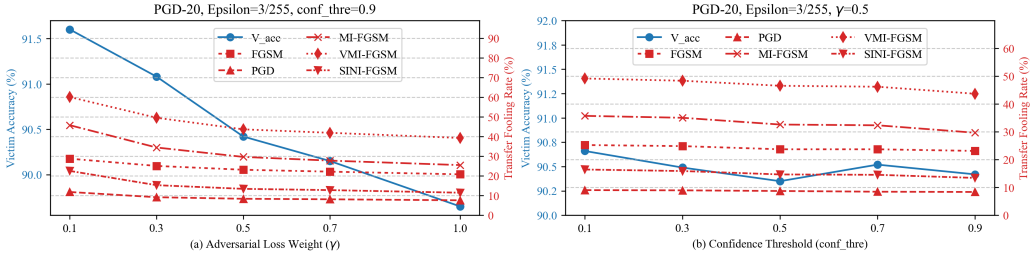

Figure D8 The trade-off between V_acc and TFR under different adversarial loss weights and confidence filtering thresholds.

Table D3 In-depth analysis of dynamic shadow models under different variant derivation methods on CIFAR-100 with a fixed query budget. We report clone accuracy and the Top-1 disagreement count (in millions) between the two shadow models during extraction.

Variant derivation method	Hard-label			Soft-label		
	Budget	Acc. (%)	Disagree ($\times 10^6$)	Budget	Acc. (%)	Disagree ($\times 10^6$)
Independent init	10M	61.73	4.72	10M	71.77	3.58
Independent init w/o $\mathcal{L}_{FD}$	10M	60.80	4.65	10M	70.60	3.54
Identical init (synchronized)	10M	53.79	0.00	10M	8.80	0.00
Identical init + relative perturbation ($\alpha = 10^{-2}$)	10M	60.14	4.59	10M	71.48	3.57
Identical init + relative perturbation ($\alpha = 10^{-1}$)	10M	59.17	4.47	10M	71.32	3.56

In-depth analysis of dynamic shadow models. To clarify the performance mechanisms of our dynamic shadow models, we conduct controlled ablations on *variant derivation methods* on CIFAR-100 under a fixed query budget (10M) and the same victim model, and report results under both hard-label and soft-label feedback settings. Specifically, we compare: (i) *independent initialization* of the two variant models; (ii) independent initialization while removing the feature-divergence term $\mathcal{L}_{FD}$; (iii) *identical initialization* with fully synchronized training (i.e., no stochasticity such as dropout); and (iv) identical initialization followed by *relative-difference injection*, where we perturb the parameters of one variant with Kaiming-scaled Gaussian noise whose standard deviation is $\alpha \cdot \text{gain}/\sqrt{\text{fan\_in}}$, applied only to tensors with ndim $\geqslant 2$, and we evaluate $\alpha \in \{10^{-2}, 10^{-1}\}$. To characterize the underlying mechanism, we measure the Top-1 disagreement count between the two variants throughout extraction and analyze it jointly with the final cloning accuracy.

The results indicate that whether the variant models can produce *effective disagreement* is critical to the success of extraction: when the two variants are identical in both initialization and training, the disagreement count is exactly zero and the final performance drops substantially, suggesting that our dynamic-shadow design fundamentally relies on inter-variant discriminatory discrepancy as a usable optimization signal. We further observe that, under independent initialization, removing $\mathcal{L}_{FD}$ yields a noticeable accuracy degradation accompanied by a reduction in the disagreement count; together with the remaining derivation settings, this suggests a non-trivial association between the amount of discriminatory disagreement and the eventual cloning accuracy, while the relationship is not purely linear. In particular, $\mathcal{L}_{FD}$ increases inter-variant discrepancy and thereby promotes the emergence of discriminative disagreements, forming a *self-enhancing* signal that more effectively drives query synthesis. Finally, compared with independent initialization, the *identical initialization + relative perturbation* strategy achieves similarly high disagreement counts with only a marginal accuracy drop, implying that the initial discrepancy introduced by weight perturbation can be progressively moderated during the extraction process as both variants are trained toward the same victim behavior, while still retaining complementary differences.

Appendix E Discussion

Extending our idea to large models. When the victim is a pre-trained visual encoder exposed as an embedding API, our core idea suggests a plausible instantiation by replacing discrete class probabilities with continuous representations. Specifically, the dynamic shadow models may be realized as two lightweight student encoders from the same architecture family (e.g., smaller ViT/ResNet

backbones), trained to match the victim embeddings via cosine/MSE losses and, if desired, contrastive distillation to better preserve relative geometry. Query synthesis can still rely on shadow-side signals: maximizing student embedding disagreement, and additionally measuring intermediate-layer representation divergence (e.g., via MMD) to capture complementary representation discrepancies and provide richer guidance for the generator. Class-entropy balancing can be replaced by representation-space coverage objectives, such as increasing pairwise cosine distances or maximizing the log-determinant of the batch covariance. As future work, a practical direction is to parameterize the generator with a pre-trained generative prior (optimizing its latent codes or lightweight adapters) while retaining the same disagreement-driven optimization principle.

A related perspective applies to vision-capable multimodal models (e.g., CLIP-like systems), where the service may return image embeddings or image-text similarity scores over a fixed prompt set. The former can be viewed as closely related to the encoder setting above; for the latter, similarity scores form a low-dimensional soft-score vector over semantic prototypes, making it natural to drive imitation learning and to guide query generation by maximizing disagreement between shadow models while enforcing coverage in the score/representation space. This interface also motivates a concrete "partial functionality" approximation: rather than reproducing a large model's full representation capacity, one may train a conventional CNN/ViT classifier to imitate the victim's scores or top-1 decisions on a predefined prompt vocabulary, using the proposed disagreement- and divergence-based query synthesis to improve query efficiency under restricted feedback. These directions highlight potential and promising extensions of our idea to large visual models.

References

1 Hinton G E, Vinyals O, Dean J. Distilling the knowledge in a neural network. CoRR, 2015. abs/1503.02531
2 Liang Y, Fu Y. Relation-guided adversarial learning for data-free knowledge transfer. Int. J. Comput. Vis., 2025, 133: 2868–2885
3 Yin H, Molchanov P, Álvarez J M, et al. Dreaming to distill: data-free knowledge transfer via DeepInversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 8712–8721
4 Li X, Wang S, Sun J, et al. Variational data-free knowledge distillation for continual learning. IEEE Trans. Pattern Anal. Mach. Intell., 2023, 45: 12618–12634
5 Yu S, Chen J, Han H, et al. Data-free knowledge distillation via feature exchange and activation region constraint. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 24266–24275
6 Fang G, Mo K, Wang X, et al. Up to 100x faster data-free knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 6597–6604
7 Binici K, Aggarwal S, Pham N T, et al. Robust and resource-efficient data-free knowledge distillation by generative pseudo replay. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 6089–6096
8 Addepalli S, Nayak G K, Chakraborty A, et al. DeGAN: data-enriching GAN for retrieving representative samples from a trained classifier. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 3130–3137
9 Ramasesh V V, Dyer E, Raghu M. Anatomy of catastrophic forgetting: hidden representations and task semantics. In: Proceedings of the International Conference on Learning Representations, 2021
10 Papernot N, McDaniel P D, Goodfellow I J, et al. Practical black-box attacks against machine learning. In: Proceedings of the ACM Asia Conference on Computer and Communications Security, 2017. 506–519
11 Orekondy T, Schiele B, Fritz M. Knockoff nets: stealing functionality of black-box models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 4954–4963
12 Roberts N, Prabhu V U, McAteer M. Model weight theft with just noise inputs: the curious case of the petulant attacker. CoRR, 2019. abs/1912.08987
13 Truong J, Maini P, Walls R J, et al. Data-free model extraction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 4771–4780
14 Kariyappa S, Prakash A, Qureshi M K. 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. 13814–13823
15 Wang Z. Zero-shot knowledge distillation from a decision-based black-box model. In: Proceedings of the International Conference on Machine Learning, 2021. 10675–10685
16 Sanyal S, Addepalli S, Babu R V. Towards data-free model stealing in a hard label setting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 15263–15272
17 Rosenthal J, Enouen E, Pham H V, et al. DisGUIDE: disagreement-guided data-free model extraction. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 9614–9622
18 Beetham J, Kardan N, Mian A S, et al. Dual student networks for data-free model stealing. In: Proceedings of the International Conference on Learning Representations, 2023
19 Yuan X, Chen K, Huang W, et al. Data-free hard-label robustness stealing attack. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 6853–6861
20 Bayraktar E, Wang Y, Bue A D. Fast re-obj: real-time object re-identification in rigid scenes. Mach. Vis. Appl., 2022, 33: 97
21 Bayraktar E, Boyraz P. Analysis of feature detector and descriptor combinations with a localization experiment for various performance metrics. Turkish J. Electr. Eng. Comput. Sci., 2017, 25: 2444–2454
22 Ahmad O, Béreux N, Baret L, et al. Causal analysis for robust interpretability of neural networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024. 4673–4682
23 Zhou M, Wu J, Liu Y, et al. DAST: data-free substitute training for adversarial attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 231–240
24 Zhang J, Li B, Xu J, et al. Towards efficient data free black-box adversarial attack. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 15115–15125
25 Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations, 2015
26 Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the International Conference on Learning Representations, 2018
27 Moosavi-Dezfooli S, Fawzi A, Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2574–2582
28 Dong Y, Liao F, Pang T, et al. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 9185–9193
29 Wang X, He K. Enhancing the transferability of adversarial attacks through variance tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 1924–1933
30 Lin J, Song C, He K, et al. Nesterov accelerated gradient and scale invariance for adversarial attacks. In: Proceedings of the International Conference on Learning Representations, 2020