Marich: A Query-efficient Distributionally Equivalent Model Extraction Attack using Public Data

Pratik Karmakar [1] **Debabrota Basu** [2]

Abstract

We study black-box model stealing attacks where the attacker can query a machine learning model only through publicly available APIs. Specifically, our aim is to design a black-box model extraction attack that uses minimal number of queries to create an informative and distributionally equivalent replica of the target model. First, we define distributionally equivalent and max-information model extraction attacks. Then, we reduce both the attacks into a variational optimisation problem. The attacker solves this problem to select the most informative queries that simultaneously maximise the entropy and reduce the mismatch between the target and the stolen models. This leads us to an active sampling-based query selection algorithm, MARICH. We evaluate MARICH on different text and image data sets, and different models, including BERT and ResNet18. Marich is able to extract models that achieve 69-96% of true model's accuracy and uses 1,070 - 6,950 samples from the publicly available query datasets, which are different from the private training datasets. Models extracted by MARICH yield prediction distributions, which are $\sim 2-4\times$ closer to the target's distribution in comparison to the existing active sampling-based algorithms. The extracted models also lead to 85-95% accuracy under membership inference attacks. Experimental results validate that MARICH is query-efficient, and also capable of performing task-accurate, high-fidelity, and informative model extraction.

1. Introduction

In recent years, Machine Learning as a Service (MLaaS) are widely deployed and used in industries. In MLaaS (Ribeiro et al., 2015), an ML model is trained remotely on a private dataset, deployed in a Cloud, and offered for public access through a prediction API, such as Amazon AWS, Google API, Microsoft Azure. This API allows an user, including a potential adversary, to send queries to the ML model and fetch corresponding predictions. Recent works have shown such models with public APIs can be stolen or extracted by designing black-box model extraction attacks (Tramèr et al., 2016). In model extraction attacks, an adversary queries the target model with a query dataset, which might be same or different than the private dataset, collects the corresponding predictions from the target model, and builds a replica model of the target model. The goal is to construct a model which is almost-equivalent to the target model over input space (Jagielski et al., 2020).

Often, ML models are proprietary, guarded by IP rights, and expensive to build. These models might be trained on datasets which are expensive to obtain (Yang et al., 2019) and consist of private data of individuals (Lowd & Meek, 2005). Also, extracted models can be used to perform other privacy attacks on the private dataset used for training, such as membership inference (Nasr et al., 2019). Thus, understanding susceptibility of models accessible through MLaaS presents an important conundrum. This motivates us to investigate black-box model extraction attacks while the adversary has no access to the private data or a perturbed version of it (Papernot et al., 2017). Instead, the adversary uses a public dataset to query the target model (Orekondy et al., 2019; Pal et al., 2020).

Black-box model extraction poses a tension between the number of queries sent to the target model and the accuracy of extracted model (Pal et al., 2020). With more queries and predictions, an adversary can build a better replica. But querying an API too much can be expensive, as each query incurs a monetary cost in MLaaS. Also, researchers have developed

[1]Department of Computer Science, School of Computing, National University of Singapore, Singapore [2]Équipe Scool, Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189- CRIStAL, F-59000 Lille, France. Correspondence to: Pratik Karmakar <pratik.karmakar@u.nus.edu>, Debabrota Basu <debabrota.basu@inria.fr>.

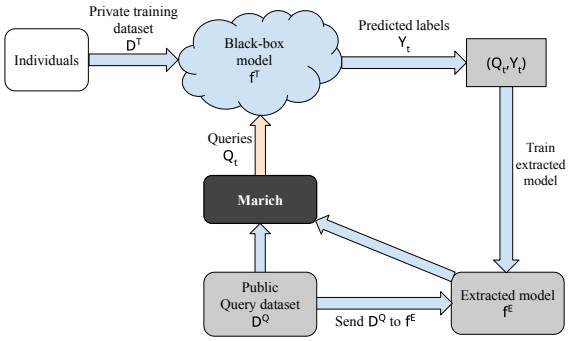


Figure 1: Black-box model extraction with MARICH.

algorithms that can detect adversarial queries, when they are not well-crafted or sent to the API in large numbers (Juuti et al., 2019; Pal et al., 2021). Thus, designing a query-efficient attack is paramount for practical deployment. Also, it exposes how more information can be leaked from a target model with less number of interactions.

In this paper, *we investigate effective definitions of efficiency of model extraction and corresponding algorithm design for query-efficient black-box model extraction attack with public data, which is oblivious to deployed model and applicable for any datatype.*

Contributions. Our investigation yields three contributions.

1. Formalism: Distribution Equivalence and Max-Information Extraction. Often, the ML models, specifically classifiers, are stochastic algorithms. They also include different elements of randomness during training. Thus, rather than focusing on equivalence of extracted and target models in terms of a fixed dataset or accuracy on that dataset (Jagielski et al., 2020), we propose a distributional notion of equivalence. We propose that if the joint distributions induced by a query generating distributions and corresponding prediction distributions due to the target and extracted models are same, they will be called distributionally equivalent Sec. 4). Another proposal is to reinforce the objective of the attack, i.e. to extract as much information as possible from the target model. This allows us to formulate the Max-Information attack, where the adversary aims to maximise the mutual information between the extracted and target models' distributions. We show that both the attacks can be performed by optimising a variational objective (Staines & Barber, 2012).

2. Algorithm: Adaptive Query Selection for Extraction with MARICH. We propose an algorithm, MARICH (Sec. 5), that optimises the objective of the variational optimisation problem (Eqn. (6)). Given an extracted model, a target model, and previous queries, MARICH adaptively selects a batch of queries enforcing this objective. Then, it sends the queries to the target model, collects the predictions (i.e. the class predicted by target model), and uses them to further train the extracted model (Algorithm 1). In order to select the most informative set of queries, it deploys three sampling strategies sequentially. These strategies select: a) the most informative set of queries, b) the most diverse set of queries in the first selection, and c) the set of queries where the target and extracted models mismatch the most. Together these strategies allow MARICH to select a small subset of queries, which maximise the information leakage, and align the extracted and target models (Fig. 1).

3. Experimental Analysis. We perform extensive evaluation with both image and text datasets, and diverse model classes, such as Logistic Regression (LR), ResNet18, and BERT (Sec. 6). Our experimental results validate that MARICH extracts more accurate replicas of the target model and high-fidelity replica of the target's prediction distributions in comparison to existing active sampling algorithms. While MARICH uses a small number of queries (1,070 - 6,950) selected from publicly available query datasets, the extracted models yield accuracy comparable with the target model while encountering a membership inference attack. This shows that MARICH can extract alarmingly informative models query-efficiently.

2. Related Works

Here, we elaborate the questions in the model extraction literature that we aim to mitigate.

Taxonomy of Model Extraction. Black-box model extraction (or model stealing or model inference) attacks aim to *replicate* of a target ML model, commonly classifiers, deployed in a remote service and accessible through a public API (Tramèr et al., 2016). The replication is done in such a way that the extracted model achieves one of the three goals: a) accuracy close to that of the target model on the private training data used to train the target model, b) maximal agreement in predictions with the target model on the private training data, and c) maximal agreement in prediction with the target model over the

whole input domain. Depending on the objective, they are called *task accuracy*, *fidelity*, and *functional equivalence model extractions*, respectively (Jagielski et al., 2020). Here, *we generalise these three approaches using a novel definition of distributional equivalence and also introduce a novel information-theoretic objective of model extraction which maximises the mutual information between the target and the extracted model over the whole data domain.*

Framework of Attack Design. Following (Tramèr et al., 2016), researchers have proposed multiple attacks to perform one of the three types of model extraction. The attacks are based on two main approaches: direct recovery (target model specific) (Milli et al., 2019; Batina et al., 2018; Jagielski et al., 2020) and learning (target model specific/oblivious). The learning-based approaches can also be categorised into supervised learning strategies, where the adversary has access to both the true labels of queries and the labels predicted by the target model (Tramèr et al., 2016; Jagielski et al., 2020), and online active learning strategies, where the adversary has only access to the predicted labels of the target model, and actively select the future queries depending on the previous queries and predicted labels (Papernot et al., 2017; Pal et al., 2020; Chandrasekaran et al., 2020). *As query-efficiency is paramount for an adversary while attacking an API to save the budget and to keep the attack hidden and also the assumption of access true label from the private data is restrictive, we focus on designing an online and active learning-based attack strategy that is model oblivious.*

Classes of Target Model. While (Milli et al., 2019; Chandrasekaran et al., 2020) focus on performing attacks against linear models, all others are specific to neural networks (Milli et al., 2019; Jagielski et al., 2020; Pal et al., 2020) and even a specific architecture (Correia-Silva et al., 2018). In contrast, MARICH *is based on active learning, and also capable of attacking both linear models and neural networks.*

Types of Query Feedback. Learning-based attack algorithms often assume access to either the probability vector of the target model over all the predicted labels (Tramèr et al., 2016; Orekondy et al., 2019; Pal et al., 2020; Jagielski et al., 2020), or the gradient of the last layer of the target neural network (Milli et al., 2019; Miura et al., 2021), which are hardly available in a public API. In contrast, following (Papernot et al., 2017), *we assume access to only the predicted labels of the target model for a set of queries, which is always available with a public API.*

Type of Query Dataset. The adversary needs a query dataset to select the queries from and to send it to the target model to obtain predicted labels. In literature, researchers assume three types of query datsets: synthetically generated samples (Tramèr et al., 2016), adversarially perturbed private (or problem domain) dataset (Papernot et al., 2017; Juuti et al., 2019), and publicly available (or out-of-problem domain) dataset (Orekondy et al., 2019; Pal et al., 2020). As we do not want to restrict MARICH to have access to the knowledge of the private dataset or any perturbed version of it, *we use publicly available datasets, which are different than the private dataset.*

In brief, we propose an online and active-learning based model extraction attack, MARICH, which is model-oblivious, assumes access to only the predicted label for a query through a public API, and uses publicly available non-domain data to query the target model. This is a less restrictive setup than the ones considered in literature, while the models extracted by MARICH demonstrate significant accuracy and act as informative replicas of the target leading to an accurate membership inference of the private dataset.

3. Background: Classifiers, Model Extraction, and Membership Inference Attacks

Before proceeding to the details, we present the fundamentals of a classifier in ML, and two types of inference attacks: Model Extraction (ME) and Membership Inference (MI).

Classifiers. A classifier in ML (Goodfellow et al., 2016) is a function $f : \mathcal{X} \to \mathcal{Y}$ that maps a set of input features $\mathbf{X} \in \mathcal{X}$ to an output $Y \in \mathcal{Y}$.[1] The output space is a finite set of classes, i.e. $\{1, \ldots, k\}$. Specifically, a classifier f is a parametric function, denoted as f_θ, with parameters $\theta \in \mathbb{R}^d$, and is trained on a dataset $\mathbf{D}^T$, i.e. a collection of n tuples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ generated IID from an underlying distribution $\mathcal{D}$. Training implies that given a model class $\mathcal{F} = \{f_\theta | \theta \in \Theta\}$, a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, and training dataset $\mathbf{D}^T$, we aim to find the optimal parameter $\theta^* \triangleq \arg\min_{\theta \in \Theta} \sum_{i=1}^n l(f_\theta(\mathbf{x}_i), y_i)$. We use cross-entropy, i.e. $l(f_\theta(\mathbf{x}_i), y_i) \triangleq -y_i \log(f_\theta(\mathbf{x}_i))$, as the loss function for classification.

Model Extraction Attack. A model extraction attack is an inference attack where an adversary aims to steal a target model f^T trained on a private dataset $\mathbf{D}^T$ and create another replica of it f^E (Tramèr et al., 2016). In the black-box setting that we are interested in, the adversary can only query the target model f^T by sending queries Q through a publicly available

[1] We denote sets/vectors by bold letters, and the corresponding distributions by calligraphic letters. We express random variables in uppercase, and an assignment of a random variable in lowercase.

API and to use the corresponding predictions $\hat{Y}$ to construct f^E. The goal of the adversary is to create a model which is either (a) as similar to the target model as possible for all input features, i.e. $f^T(x) = f^E(x)\ \forall x \in \mathcal{X}$ (Song & Shmatikov, 2020; Chandrasekaran et al., 2020) or (b) predicts labels that has maximal agreement with that of the labels predicted by the target model for a given data-generating distribution, i.e. $f^E = \arg\min \Pr_{x\sim\mathcal{D}}[l(f^E(x), f^T(x))]$ (Tramèr et al., 2016; Pal et al., 2020; Jagielski et al., 2020). The first type of attacks are called the functionally equivalent attacks. The later family of attacks is referred as the fidelity extraction attacks. The third type of attacks aim to find an extracted model f^E that achieves maximal classification accuracy for the underlying private dataset used to train the f^T. These are called task accuracy extraction attacks (Tramèr et al., 2016; Milli et al., 2019; Orekondy et al., 2019). In this paper, *we generalise the first two type of attacks by proposing the distributionally equivalent attacks and experimentally show that it yields both task accuracy and fidelity.*

Membership Inference Attack. Another popular family of inference attacks on ML models is the Membership Inference (MI) attacks (Shokri et al., 2017; Yeom et al., 2018). In MI attack, given a private (or member) dataset $\mathbf{D}^T$ to train f^T and another non-member dataset S with $|\mathbf{D}^T \cap S| \neq \emptyset$, the goal of the adversary is to infer whether any $x \in \mathcal{X}$ is sampled from the member dataset $\mathbf{D}^T$ or the non-member dataset S. Effectiveness of an MI attacks can be measured by its accuracy of MI, i.e. the total fraction of times the MI adversary identifies the member and non-member data points correctly. Accuracy of MI attack on the private data using f^E rather than f^T is considered as a measure of effectiveness of the extraction attack (Nasr et al., 2019). We show that the model f^E extracted using MARICH allows us to obtain similar MI accuracy as that obtained by directly attacking the target model f^T using even larger number of queries. This validates that *the model f^E by* MARICH *in a black-box setting acts as an information equivalent replica of the target model f^T.*

4. Distributional Equivalence and Max-Information Model Extractions

In this section, we introduce the distributionally equivalent and max-information model extractions. We further reduce both the attacks into a variational optimisation problem.

Definition 4.1 (Distributionally Equivalent Model Extraction)**.** For any query generating distribution $\mathcal{D}^Q$ over $\mathbb{R}^d \times \mathcal{Y}$, an extracted model $f^E : \mathbb{R}^d \to Y$ is distributionally equivalent to a target model $f^T : \mathbb{R}^d \to Y$ if the joint distributions of input features $Q \in \mathbb{R}^d \sim \mathcal{D}^Q$ and predicted labels induced by both the models are same almost surely. This means that for any divergence D, two distributionally equivalent models f^E and f^T satisfy $D(\Pr(f^T(Q), Q) \| \Pr(f^E(Q), Q)) = 0\ \forall\ \mathcal{D}^Q$.

To ensure query-efficiency in distributionally equivalent model extraction, an adversary aims to choose a query generating distribution $\mathcal{D}^Q$ that minimises it further. If we assume that the extracted model is also a parametric function, i.e. f^E_ω with parameters $\omega \in \Omega$, we can solve the query-efficient distributionally equivalent extraction by computing

$$(\omega^*_{\text{DEq}}, \mathcal{D}^Q_{\min}) \triangleq \\ \arg\min_{\omega\in\Omega} \arg\min_{\mathcal{D}^Q} D(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_\omega(Q), Q)) \tag{1}$$

Equation 1 allows us to choose a different class of models with different parametrisation for extraction till the joint distribution induced by it matches with that of the target model. For example, the extracted model can be a logistic regression or a CNN if the target model is a logistic regression. This formulation also enjoys the freedom to choose the data distribution $\mathcal{D}^Q$ for which we want to test the closeness. Rather the distributional equivalence pushes us to find the best query distribution for which the mismatch between the posteriors reduces the most and to compute an extracted model $f^E_{\omega^*}$ that induces the joint distribution closest to that of the target model $f^T_{\theta^*}$.

Remark 4.2. If we choose $\mathcal{D}^Q_{\min} = \mathcal{D}^T$, distributional equivalence reduces to the fidelity extraction attack. If we choose $\mathcal{D}^Q_{\min} = \text{Unif}(\mathcal{X})$, distributional equivalent extraction coincides with functional equivalent extraction. Thus, distributional equivalence attack can ensure both fidelity and functional equivalence extractions depending on the choice of query generating distribution $\mathcal{D}^Q$.

Theorem 4.3 (Upper Bounding Distributional Closeness)**.** *If we choose KL-divergence as the divergence function D, then for a given query generating distribution $\mathcal{D}^Q$*

$$D_{\text{KL}}(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\text{DEq}}}(Q), Q)) \\ \leq \min_\omega \mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_\omega(Q))] - H(f^E_\omega(Q)). \tag{2}$$

By variational principle, Theorem 4.3 implies that *minimising the upper bound* on the RHS will lead to an extracted model which minimises the KL-divergence for a chosen query distribution.

Max-Information Model Extraction. The common objective of any inference attack is to leak as much information as possible from the target model f^T. Specifically, in model extraction attacks, we want to create an informative replica f^E of the target model f^T such that it induces a joint distribution $\Pr(f^E_\omega(Q), Q))$ which retains the most information regarding the target's joint distribution. As adversary can control the query distribution, we want to choose such a query distribution $\mathcal{D}^Q$ that maximises information leakage.

Definition 4.4 (Max-Information Model Extraction)**.** A model $f^E : \mathbb{R}^d \to Y$ and query distribution $\mathcal{D}^Q$ are called a max-information extraction of a target model $f^T : \mathbb{R}^d \to Y$ and max-information query distribution, respectively, if they maximise the mutual information between the joint distributions of input features $Q \in \mathbb{R}^d \sim \mathcal{D}^Q$ and predicted labels induced by f^E and that of the target model. Mathematically, $(f^E_{\omega^*}, \mathcal{D}^Q_{\max})$ is a max-information extraction of $f^T_{\theta^*}$ if

$$(\omega^*_{\text{MaxInf}}, \mathcal{D}^Q_{\max}) \triangleq \arg\max_{\omega} \arg\max_{\mathcal{D}_Q} I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_\omega(Q), Q)) \tag{3}$$

Similar to Definition 4.1, Definition 4.4 also does not restrict us to choose a parametric model ω different from that of the target θ. It also allows us to compute the data distribution $\mathcal{D}^Q$ for which the information leakage is maximum rather than relying on the private dataset $\mathbf{D}^T$ used for training f^T.

Theorem 4.5 (Lower Bounding Information Leakage)**.** *For any given distribution $\mathcal{D}^Q$, the information leaked by any max-information attack (Equation 3) is lower bounded as:*

$$I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\text{MaxInf}}}(Q), Q)) \geq \max_{\omega} -\mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_\omega(Q))] + H(f^E_\omega(Q)). \tag{4}$$

By variational principle, Theorem 4.5 implies that *maximising the lower bound* in the RHS will lead to an extracted model which maximises the mutual information between target and extracted joint distributions for a given query generating distribution.

Distributionally Equivalent and Max-Information Extractions: A Variational Optimisation Formulation. From Theorem 4.3 and 4.5, we observe that the lower and upper bounds of the objective functions of distribution equivalent and max-information attacks are negatives of each other. Specifically, $-D_{\text{KL}}(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\text{DEq}}}(Q), Q)) \geq \max_\omega -F(\omega, \mathcal{D}^Q)$ and $I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\text{MaxInf}}}(Q), Q)) \geq \max_\omega F(\omega, \mathcal{D}^Q)$, where

$$F(\omega, \mathcal{D}^Q) \triangleq -\mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_\omega(Q))] + H(f^E_\omega(Q)). \tag{5}$$

Thus, following a variational approach, we aim to solve an optimisation problem on $F(\omega, \mathcal{D}^Q)$ in an online and frequentist manner. Specifically, we do not assume a parametric family of $\mathcal{D}^Q$. Instead, we choose a set of queries $Q_t \in \mathbb{R}^d$ at each round $t \in T$. This leads to an empirical counterpart of our problem, i.e.

$$\begin{aligned} &\max_{\omega \in \omega, Q_{[0,T]} \in {\mathbf{D}^Q}_{[T]}} \hat{F}(\omega, Q_{[0,T]}) \\ &\triangleq \max_{\omega, Q_{[0,T]}} -\frac{1}{T}\sum_{t=1}^{T} l(f^T_{\theta^*}(Q_t), f^E_\omega(Q_t))] + \sum_{t=1}^{T} H(f^E_\omega(Q_t)). \end{aligned} \tag{6}$$

As we need to evaluate $f^T_{\theta^*}$ for each Q_t, we refer Q_t's as queries, the dataset $\mathbf{D}^Q \subseteq \mathbb{R}^d \times \mathcal{Y}$ from where they are chosen as the query dataset, and the corresponding unobserved distribution $\mathcal{D}^Q$ as the query generating distribution. Given the optimisation problem of Equation 6, we propose an algorithm MARICH to solve it effectively.

5. Marich: A Query Selection Algorithm for Model Extraction

In this section, we propose an algorithm, MARICH, to solve Equation 6 in a query-efficient manner.

Algorithm 1 MARICH

Input: Target model: f^T, Query dataset: $\mathbf{D}^Q$, #Classes: k
Parameter: #initial samples: n_0, Training epochs: E_{max}, #Batches of queries: T, Query budget: B, Subsampling ratios: $\gamma_1, \gamma_2 \in (0, 1]$
Output: Extracted model f^E

1: //* Initialisation of the extracted model*// {Phase 1}
2: $Q_0^{train} \leftarrow n_0$ datapoints randomly chosen from D^Q
3: $Y_0^{train} \leftarrow f^T(Q_0^{train})$ {Query the target model f^T with Q_0^{train}}
4: **for** epoch $\leftarrow 1$ to E_{max} **do**
5: $\quad f_0^E \leftarrow$ Train f^E with $(Q_0^{train}, Y_0^{train})$
6: **end for**
7: //* Adaptive query selection to build the extracted model*// {Phase 2}
8: **for** $t \leftarrow 1$ to T **do**
9: $\quad Q_t^{entropy} \leftarrow$ ENTROPYSAMPLING$(f_{t-1}^E, \mathbf{D}^Q \setminus Q_{t-1}^{train}, B)$
10: $\quad Q_t^{grad} \leftarrow$ ENTROPYGRADIENTSAMPLING$(f_{t-1}^E, Q_t^{entropy}, \gamma_1 B)$
11: $\quad Q_t^{loss} \leftarrow$ LOSSSAMPLING$(f_{t-1}^E, Q_t^{grad}, Q_{t-1}^{train}, Y_{t-1}^{train}, \gamma_1\gamma_2 B)$
12: $\quad Y_t^{new} \leftarrow f^T(Q_t^{loss})$ {Query the target model f^T with Q_t^{loss}}
13: $\quad Q_t^{train} \leftarrow Q_{t-1}^{train} \cup Q_t^{loss}$
14: $\quad Y_t^{train} \leftarrow Y_{t-1}^{train} \cup Y_t^{new}$
15: $\quad$ **for** epoch $\leftarrow 1$ to E_{max} **do**
16: $\quad\quad f_t^E \leftarrow$ Train f_{t-1}^E with $(Q_t^{train}, Y_t^{train})$
17: $\quad$ **end for**
18: **end for**
19: **return** Extracted model $f^E \leftarrow f_T^E$

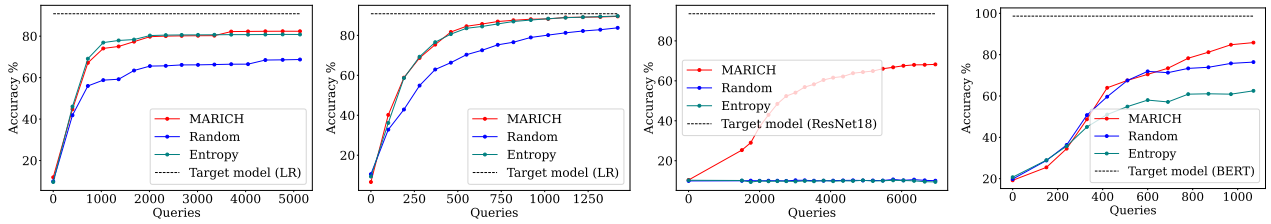
Algorithm Design. From Equation 6, we observe that once the queries $Q_{[0,T]}$ are selected, the outer maximisation problem is equivalent to regualrised loss minimisation. Thus, it can be solved using any standard empirical risk minimisation algorithm, such as Adam, SGD etc. Thus, to achieve query efficiency, we focus on designing a query selection algorithm that selects a batch of queries Q_t at round $t \leq T$:

$$Q_t \triangleq \arg\max_{Q \in \mathbf{D}^Q} \underbrace{-\frac{1}{t}\sum_{i=1}^{t-1} l(f_{\hat{\theta}^*}^T(Q_i \cup Q), f_{\omega_{t-1}}^E(Q_i \cup Q))]}_{\text{Model-mismatch term}} + \underbrace{\sum_{i=1}^{t-1} H(f_{\omega_{t-1}}^E(Q_i \cup Q))}_{\text{Entropy term}}. \tag{7}$$

Here, $f_{\omega_{t-1}}^E$ is the model extracted by round $t-1$. Equation 7 indicates two criteria to select the queries. With the *Entropy term*, we want to select a query that maximises the entropy of predictions for the extracted model $f_{\omega_{t-1}}^E$. This allows us to select the queries which are most informative about the mapping between the input features and the prediction space. With the *model-mismatch term*, Equation 7 pushes the adversary to select queries where the target and extracted models mismatch the most. Thus, minimising the loss between target and extracted models for such a query forces them to match over the whole domain. We provide a pseudocode for MARICH in Algorithm 1 (ref. Appendix A).

Initialisation Phase. To initialise the extraction, we select a set of n_0 queries, called Q_0^{train}, uniformly randomly from the query dataset $\mathbf{D}^Q$. We send these queries to the target model and collect corresponding predicted classes Y_0^{train} (Line 2). We use these n_0 samples of input-predicted label pairs to construct a primary extracted model f_0^E.

Active Sampling. As the adaptive sampling phase commences, we select $\gamma_1\gamma_2 B$ number of queries at round t. To maximise the *Entropy term* and minimise the *Model-mismatch term* of Equation 7, we sequentially deploy ENTROPYSAMPLING and LOSSSAMPLING. To achieve further query-efficiency, we refine the queries selected using ENTROPYSAMPLING by ENTROPYGRADIENTSAMPLING, which aims to find the most diverse subset from a given set of queries. Now, we describe the three sampling strategies.



(a) LR with EMNIST queries (b) LR with CIFAR10 queries (c) ResNet18 with STL10 queries (d) BERT with AGNews queries

Figure 2: Accuracy of the extracted models (mean ± std. over 10 runs) w.r.t. the target model using MARICH, Entropy, and Random sampling. Each figure represents (a target model, a query dataset). Models extracted by MARICH are closer to the target models.

ENTROPYSAMPLING. First, we aim to select the set of queries which unveil most information about the mapping between the input features and the prediction space. Thus, we deploy ENTROPYSAMPLING. In ENTROPYSAMPLING, we compute the output probability vectors from f^E_{t-1} for all the query points in $\mathbf{D}^Q \setminus Q^{train}_{t-1}$ and then select top B points with highest entropy: $Q_{entropy} \leftarrow \arg\max_{X \subset X_{in}, |X|=B} H(f^E(X_{in}))$. Thus, we select the queries $Q^{entropy}_t$, about which f^E_{t-1} is most confused and training on these points makes the model more informative.

ENTROPYGRADIENTSAMPLING. To be frugal about the number of queries, we refine $Q^{entropy}_t$ to compute the most diverse subset of it. First, we compute the gradients of entropy of $f^E_{t-1}(x)$, i.e. $\nabla_x H(f^E_{t-1}(x))$, for all $x \in Q^{entropy}_t$. The gradient at point x reflects the change at x in the prediction distribution induced by f^E_{t-1}. We use these gradients to embed the points $x \in Q^{entropy}_t$. Now, we deploy K-means clustering to find k (= #classes) clusters with centers C_{in}. Then, we sample $\gamma_1 B$ points from these clusters: $Q_{grad} \leftarrow \arg\min_{X \subset Q^{entropy}_t, |X|=\gamma_1 B} \sum_{x_i \in X} \sum_{x_j \in C_{in}} \|\nabla_{x_i} H(f^E(.)) - \nabla_{x_j} H(f^E(.))\|^2_2$. Selecting points from k clusters ensures diversity of queries and reduces their number by γ_1.

LOSSSAMPLING. In this step, we select points from Q^{grad}_t for which the predictions of $f^T_{\theta^*}$ and f^E_{t-1} are most dissimilar. To identify these points, we compute the loss $l(f^T(x), f^E_{t-1}(x))$ for all $x \in Q^{train}_{t-1}$. Then, we select top-k points from Q^{train}_{t-1} with the highest loss values (Line 2), and sample a subset Q^{loss}_t of size $\gamma_1\gamma_2 B$ from Q^{grad}_t which are closest to the k points selected from Q^{train}_{t-1}. This ensures that f^E_{t-1} would better align with f^T if it trains on the points where the mismatch in predictions are higher.

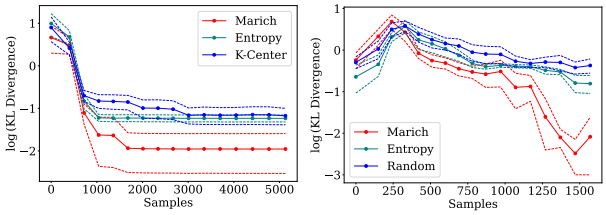
At the end of Phase 2 in each round of sampling, Q^{loss}_t is sent to f^T for fetching the labels Y^{train}_t predicted by the target model. We use (Q^{loss}_t, Y^{loss}_t) along with $(Q^{train}_{t-1}, Y^{train}_{t-1})$ to train f^E_{t-1} further. Thus, MARICH performs $n_0 + \gamma_1\gamma_2 BT$ number of queries through $T+1$ number of interactions with the target model f^T to create the final extracted model f^E_T. We experimentally demonstrate effectiveness of the model extracted by MARICH to achieve high task accuracy and to act as an informative replica of the target model for extracting private information regarding the private training data $\mathbf{D}^T$.

Remark 5.1. Eq. (7) dictates that the active sampling strategy should try to select queries that maximise the entropy in the prediction distribution of the extracted model, while decreases the mismatch in predictions of the target and the extracted models. We further use the ENTROPYGRADIENTSAMPLING to choose a smaller but most diverse subset. As Eq. (7) does not specify any ordering between these objectives, one can argue about the sequence of using these three sampling strategies. We choose to use sampling strategies in the decreasing order of runtime complexity as the first strategy selects the queries from the whole query dataset, while the following strategies work only on the already selected queries. We show in Appendix D that LOSSSAMPLING incurs the highest runtime followed by ENTROPYGRADIENTSAMPLING, while ENTROPYSAMPLING is significantly cheaper.

6. Experimental Analysis

Now, we perform an experimental evaluation of models extracted by MARICH. Here, we discuss the experimental setup, the objectives of experiments, and experimental results. We defer the source code, additional results, performance against defenses, and hyperparameters to Appendix.

Experimental Setup. We have implemented a prototype of MARICH using Python 3.9 and PyTorch 1.12, and run on a NVIDIA Quadro GV100 32 GB GPU. We perform our attacks against three target models (f^T), namely Logistic Regression (LR), ResNet18 (He et al., 2016), and BERT (Devlin et al., 2018), trained on three private datasets ($\mathbf{D}^T$): MNIST handwritten digits (Deng, 2012), CIFAR10 (Krizhevsky et al.) and BBC News, respectively. For model extraction, we use EMNIST letters dataset (Cohen et al., 2017), CIFAR10, STL10, and AGNews (Zhang et al., 2015), as publicly available and



(a) LR with EMNIST (b) BERT with AGNews

Figure 3: Comparing fidelity of the prediction distributions (in log scale) for different active learning algorithms. MARICH achieves $2-4\times$ lower KL-divergence than others.

Table 1: Statistics of membership inference (MI) for different target models, datasets & attacks. "-" means that we use the member dataset and the target model.

Member dataset	Target model	Query Dataset	Algorithm	Non-member dataset	#Queries	MI acc.	MI agreement	MI agreement AUC
MNIST	LR	-	-	EMNIST	50,000 (100%)	87.99%	-	-
MNIST	LR	-	-	CIFAR10	50,000 (100%)	92.30%	-	-
MNIST	LR	EMNIST	MARICH	EMNIST	5,130 (3.5%)	88.58%	92.82%	**92.73%**
MNIST	LR	CIFAR10	MARICH	CIFAR10	1,420 (2.37%)	**94.27%**	**93.97%**	92.43%
MNIST	LR	EMNIST	RS	EMNIST	5,130 (3.5%)	89.61%	91.01%	91.11%
MNIST	LR	CIFAR10	RS	CIFAR10	1,420 (2.37%)	92.61%	89.84%	85.79%
CIFAR10	Resnet18	-		STL10	40,000 (100%)	79.35%	-	-
CIFAR10	Resnet18	STL10	MARICH	STL10	6,950 (6.15%)	**93.90%**	**75.52%**	**76.69%**
CIFAR10	Resnet18	STL10	RS	STL10	6,950 (6.15%)	92.32%	75.25%	75.83%
BBCNews	BERT	-		AGNews	1,490 (100%)	**98.61%**	-	-
BBCNews	BERT	AGNews	MARICH	AGNews	1,070 (0.83%)	94.42%	**91.02%**	**82.62%**
BBCNews	BERT	AGNews	RS	AGNews	1,070 (0.83%)	89.17%	86.93%	58.64%

mismatched query datasets $\mathbf{D}^Q$. To instantiate task accuracy, we compare accuracy of the extracted models f^E_{MARICH} with the target model and models extracted by Random Sampling (RS), f^E_{RS}, and Entropy Sampling (Pal et al., 2020). To instantiate informativeness of the extracted models (Nasr et al., 2019), we compare the Membership Inference (MI), i.e. MI accuracy, AUC of MI, and MI agreements, performed on the target models, and the models extracted using MARICH and RS with same query budget. For MI, we use in-built membership attack from IBM ART (Nicolae et al., 2018). The objectives of the experiments are:

1. *How do the accuracy of the model extracted using* MARICH *on the private dataset compare with that of the target model, and RS with same query budget?*

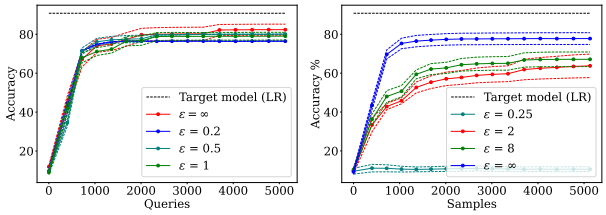
2. *How close are the prediction distributions of the model extracted using* MARICH *and the target model? Can* MARICH *produce better replica of target's prediction distribution than other active sampling methods, leading to better distributional equivalence?*

3. *How do the models extracted by* MARICH *behave under Membership Inference (*MI*) in comparison to the target models, and the models extracted by RS with same budget?*[2]

4. *How does the performance of extracted models change if different Differential Privacy (DP) preserving mechanisms (Dwork et al., 2006) are applied on the target model either during training or while answering the queries?*

Accuracy of Extracted Models. MARICH extracts logistic regression models with 5,130 and 1,420 queries from EMNIST and CIFAR10 query datasets by attacking a target logistic regression model, $f^T_{logistic}$ trained on MNIST. While the target model achieves 90.82% test accuracy, the models extracted using EMNIST and CIFAR10 achieve test accuracies 82.37% (90.69% of $f^T_{logistic}$) and 89.48% (98.52% of $f^T_{logistic}$), respectively (Figure 2a and 2b). The models extracted using RS show test accuracy 52.96% and 84.18%, and the models extracted by Entropy sampling achieve 80.81% (88.97% of $f^T_{logistic}$) and 89.66% (98.72% of $f^T_{logistic}$). MARICH attacks a ResNet18, f^T_{ResNet}. trained on CIFAR10 (accuracy: 93.58%) with 6,950 queries from STL10 dataset. The extracted ResNet18 shows 68.22% (72.90% of f^T_{ResNet}) test accuracy. But the model extracted using RS and Entropy sampling achieve 9.99% and 9.39% accuracy (Fig. 2c). To verify MARICH's effectiveness for text data, we attack a BERT, f^T_{BERT} trained on BBCNews (test accuracy: 98.65%) with queries from the AGNews dataset. By using only 1,070 queries, MARICH extracts a model with 87.01% (88.20% of f^T_{BERT}) test accuracy (Fig. 2d). The model extracted using RS and Entropy sampling show test accuracy of 76.41% and 62.51%, respectively.

[2]The MI accuracy achievable by attacking a model acts as a proxy of how informative is the model.



(a) Target trained with DP-SGD (b) Output perturbation of query

Figure 4: Comparing test accuracy of the models extracted by MARICH against different DP mechanisms (DP-SGD and Output Perturbation) applied on the target model.

For all the models and datasets, MARICH *extracts models that achieve test accuracy closer to target models, and are more accurate than models extracted by RS and Entropy Sampling.*

Distributional Equivalence of Extracted Models. One of our aims is to extract a distributionally equivalent model of the target f^T using MARICH. Thus, in Figure 6, we illustrate the KL-divergence (mean$\pm$std. over 10 runs) between the prediction distributions of the target model and the model extracted by MARICH. Due to brevity, we show two cases: i) when we attack an LR trained on MNIST with EMNIST, and ii) when we attack a BERT trained on BBCNews with AGNews queries. In both cases, we observe that the models extracted by MARICH achieve $\sim 2-4\times$ lower KL-divergence than the models extracted by other active sampling methods, i.e. RS, Entropy, and K-center (Pal et al., 2020). *These results show that* MARICH *is able to yield high-fidelity distributionally equivalent extracted models than competing algorithms.*

Membership Inference with Extracted Models. In Table 1, we report *accuracy*, *agreement* in inference with target model, and *agreement AUC* of membership attacks performed on different target models and extracted models with different query datasets. The models extracted using MARICH demonstrate higher MI agreement with the target models than the models extracted using RS. They also achieve comparable MI accuracy with respect to the target model. *These results show that the models extracted by* MARICH *act as informative replicas of the target models.*

Performance Against Privacy Defenses. We test the impact of DP-based defenses deployed in the target model on the performance of MARICH. First, we train four target models on MNIST using *DP-SGD* (Abadi et al., 2016) with different privacy levels $\varepsilon = \{0.2, 0.5, 1, \infty\}$ and $\delta = 10^{-5}$. In Figure 4a, we present the accuracy of models extracted by querying these DP models. Accuracy of the models extracted from private target models are $\sim 2.3-7.4\%$ lower than the model extracted from the non-private target model. Second, we apply an *output perturbation* method (Dwork et al., 2006), where a calibrated Laplace noise is added to the responses of the target model against MARICH's queries. This ensures ε-DP for the target model. Figure 4b shows accuracy of the models extracted by querying target models with $\varepsilon = \{0.25, 2, 8, \infty\}$. Performance of the extracted models degrade slightly for $\varepsilon = 2, 8$, but significantly for $\varepsilon = 0.25$. *This shows that performance of* MARICH *decreases against DP defenses but the degradation varies significantly depending on the defense mechanism.*

Summary of Results. From the experimental results, we deduce the following conclusions.

Accuracy. Test accuracy (on the subsets of private datasets) of the models f^E_{MARICH} are higher than the models extracted with RS and Entropy, and are $\sim 73-96\%$ of the target models (Fig. 2). This shows effectiveness of MARICH as a task accuracy extraction attack, while solving distributional equivalence and max-info extractions.

Distributional Equivalence. We observe that the KL-divergence between the prediction distributions of the target model and f^E_{MARICH} are $\sim 2-4\times$ lower than the models extracted by other active sampling algorithms. This confirms that MARICH conducts more accurate distributionally equivalent extraction than existing active sampling attacks.

Informative Replicas: Effective Membership Inference. The agreement in MI achieved by attacking f^E_{MARICH} and the target model is always higher than that of the f^E_{RS} (Table 1). Also, MI accuracy for f^E_{MARICH}'s are $88.58\% - 94.42\%$ (Table 1). This shows that the models extracted by MARICH act as informative replicas of the target model.

Query-efficiency. Table 1 shows that MARICH uses only $1,070-6,950$ queries from the public datasets, which in most cases are lower than data used for training the target models. This shows MARICH is significantly query efficient whereas other known active learning attacks use at least $10,000$ queries to begin with (Pal et al., 2020, Table 2).

Performance against Defenses. Performance of MARICH decreases with the increasing level of DP applied on the target model, which is expected. But when DP-SGD is applied to train the target, the degradation is little ($\sim 7\%$) even for $\varepsilon = 0.2$.

In contrast, the degradation is higher when the output perturbation is applied with similar ε (0.25).

7. Conclusion and Future Directions

In this paper, we investigate the design of a model extraction attack against a target ML model (classifier) trained on a private dataset and accessible through a public API. The API returns only a predicted label for a given query. We propose the notions of distributional equivalence extraction, which extends the existing notions of task accuracy and functionally equivalent model extractions. We also propose an information-theoretic notion, i.e. Max-Info model extraction. We further propose a variational relaxation of these two types of extraction attacks, and solve it using an online and adaptive query selection algorithm, MARICH. MARICH uses a publicly available query dataset different from the private dataset. We experimentally show that the models extracted by MARICH achieve $68.22 - 89.48\%$ accuracy on the private dataset while using 1,070 - 6,950 queries. For both text and image data, we demonstrate that the models extracted by MARICH act as informative replicas of the target models and also yield high-fidelity replicas of the targets' prediction distributions. Typically, the functional equivalence attacks require model-specific techniques, while MARICH is model-oblivious while performing distributional equivalence attack. This poses an open question: is distributional equivalence extraction 'easier' than functional equivalence extraction, which is NP-hard (Jagielski et al., 2020)?

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Batina, L., Bhasin, S., Jap, D., and Picek, S. CSI neural network: Using side-channels to recover your artificial neural network information. *CoRR*, abs/1810.09076, 2018.

Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S., and Yan, S. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1309–1326, 2020.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017. doi: 10.1109/IJCNN.2017.7966217.

Correia-Silva, J. R., Berriel, R. F., Badue, C., de Souza, A. F., and Oliveira-Santos, T. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2018.

Dandekar, A., Basu, D., and Bressan, S. Differential privacy for regularised linear regression. In *Database and Expert Systems Applications: 29th International Conference, DEXA 2018, Regensburg, Germany, September 3–6, 2018, Proceedings, Part II*, pp. 483–491. Springer, 2018.

Dandekar, A., Basu, D., and Bressan, S. Differential privacy at risk: Bridging randomness and privacy budget. *Proceedings on Privacy Enhancing Technologies*, 1:64–84, 2021.

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

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT Press, 2016.

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

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., and Papernot, N. High accuracy and high fidelity extraction of neural networks. In *29th USENIX security symposium (USENIX Security 20)*, pp. 1345–1362, 2020.

Juuti, M., Szyller, S., Marchal, S., and Asokan, N. Prada: protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512–527. IEEE, 2019.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Lowd, D. and Meek, C. Good word attacks on statistical spam filters. In *CEAS*, volume 2005, 2005.

Milli, S., Schmidt, L., Dragan, A. D., and Hardt, M. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 1–9, 2019.

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

Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753. IEEE, 2019.

Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I., and Edwards, B. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL https://arxiv.org/pdf/1807.01069.

Orekondy, T., Schiele, B., and Fritz, M. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4954–4963, 2019.

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

Pal, S., Gupta, Y., Kanade, A., and Shevade, S. Stateful detection of model extraction attacks. *arXiv preprint arXiv:2107.05166*, 2021.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Ribeiro, M., Grolinger, K., and Capretz, M. A. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 896–902. IEEE, 2015.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Song, C. and Shmatikov, V. Overlearning reveals sensitive attributes. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

Staines, J. and Barber, D. Variational optimization. *arXiv preprint arXiv:1212.4507*, 2012.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., and Mironov, I. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In *NIPS*, 2015.

A. Detailed Pseudocode of MARICH

Algorithm 2 MARICH

Input: Target model: f^T, Query dataset: $\mathbf{D}^Q$, #Classes: k
Parameter: #initial samples: n_0, Training epochs: E_{max}, #Batches of queries: T, Query budget: B, Subsampling ratios: $\gamma_1, \gamma_2 \in (0, 1]$
Output: Extracted model f^E

//* Initialisation of the extracted model*// {Phase 1}
$Q_0^{train} \leftarrow n_0$ datapoints randomly chosen from D^Q
$Y_0^{train} \leftarrow f^T(Q_0^{train})$ {Query the target model f^T with Q_0^{train}}
for epoch $\leftarrow$ 1 to E_{max} **do**
  $f_0^E \leftarrow$ Train f^E with $(Q_0^{train}, Y_0^{train})$
end for
//* Adaptive query selection to build the extracted model*// {Phase 2}
for $t \leftarrow 1$ to T **do**
  $Q_t^{entropy} \leftarrow$ ENTROPYSAMPLING$(f_{t-1}^E, \mathbf{D}^Q \setminus Q_{t-1}^{train}, B)$
  $Q_t^{grad} \leftarrow$ GRADIENTSAMPLING$(f_{t-1}^E, Q_t^{entropy}, \gamma_1 B)$
  $Q_t^{loss} \leftarrow$ LOSSSAMPLING$(f_{t-1}^E, Q_t^{grad}, Q_{t-1}^{train}, Y_{t-1}^{train}, \gamma_1\gamma_2 B)$
  $Y_t^{new} \leftarrow f^T(Q_t^{loss})$ {Query the target model f^T with Q_t^{loss}}
  $Q_t^{train} \leftarrow Q_{t-1}^{train} \cup Q_t^{loss}$
  $Y_t^{train} \leftarrow Y_{t-1}^{train} \cup Y_t^{new}$
  for epoch $\leftarrow$ 1 to E_{max} **do**
    $f_t^E \leftarrow$ Train f_{t-1}^E with $(Q_t^{train}, Y_t^{train})$
  end for
end for
return Extracted model $f^E \leftarrow f_T^E$

EntropySampling (extracted model: f^E, input data points: X_{in}, budget: B)
$Q_{entropy} \leftarrow \arg\max_{X \subset X_{in}, |X|=B} H(f^E(X_{in}))$ {Select B points with maximum entropy}
return $Q_{entropy}$

GradientSampling (extracted model: f^E, input data points: X_{in}, budget: $\gamma_1 B$)
$E \leftarrow H(f^E(X_{in}))$
$G \leftarrow \{\nabla_x E \mid x \in X_{in}\}$
$C_{in} \leftarrow k$ centres of G computed using K-means
$Q_{grad} \leftarrow \arg\min_{X \subset X_{in}, |X|=\gamma_1 B} \sum_{x_i \in X} \sum_{x_j \in C_{in}} \|\nabla_{x_i} E - \nabla_{x_j} E\|_2^2$ {Select $\gamma_1 B$ points from X_{in} whose $\frac{\partial E}{\partial x}$ are closest to that of C_{in}}
return Q_{grad}

LossSampling (extracted model: f^E, input data points: X_{in}, previous queries: Q_{train}, previous predictions: Y_{train}, budget: $\gamma_1\gamma_2 B$)
$L \leftarrow l(Y_{train}, f^E(Q_{train}))$ {Compute the mismatch vector}
$Q_{mis} \leftarrow$ ARGMAXSORT(L, k) {Select top-k mismatching points}
$Q_{loss} \leftarrow \arg\min_{X \subset X_{in}, |X|=\gamma_1\gamma_2 B} \sum_{x_i \in X} \sum_{x_j \in Q_{mis}} \|x_i - x_j\|_2^2$ {Select $\gamma_1\gamma_2 B$ points closest to Q_{mis}}
return Q_{loss}

B. Proofs of Section 4

In this section, we elaborate the proofs for the Theorems 4.3 and 4.5.[3]

***Theorem** 4.3* (Upper Bounding Distributional Closeness). If we choose KL-divergence as the divergence function D, we can show that

$$D_{\mathrm{KL}}(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\mathrm{DEq}}}(Q), Q)) \leq \min_{\omega} \mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_{\omega}(Q))] - H(f^E_{\omega}(Q)).$$

Proof. Let us consider a query generating distribution $\mathcal{D}^Q$ on $\mathbb{R}^d$. A target model $f^T_{\theta^*} : \mathbb{R}^d \to \mathcal{Y}$ induces a joint distribution over the query and the output (or label) space, denoted by $\Pr(f^T_{\theta^*}, Q)$. Similarly, the extracted model $f^T_{\theta^*} : \mathbb{R}^d \to \mathcal{Y}$ also induces a joint distribution over the query and the output (or label) space, denoted by $\Pr(f^E_{\omega}, Q)$.

$$\begin{aligned}
&D_{\mathrm{KL}}(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega}(Q), Q)) \\
&= \int_{Q \in \mathbb{R}^d} \mathrm{d}\Pr(f^T_{\theta^*}(Q), Q) \log \frac{\Pr(f^T_{\theta^*}(Q), Q)}{\Pr(f^E_{\omega}(Q), Q)} \\
&= \int_{Q \in \mathbb{R}^d} \Pr(f^T_{\theta^*}(Q) | Q = q) \Pr(Q = q) \log \frac{\Pr(f^T_{\theta^*}(Q) | Q = q)}{\Pr(f^E_{\omega}(Q) | Q = q)} \mathrm{d}q \\
&= \int_{Q \in \mathbb{R}^d} \Pr(f^T_{\theta^*}(Q) | Q = q) \Pr(Q = q) \log \Pr(f^T_{\theta^*}(Q) | Q = q) \mathrm{d}q \\
&- \int_{Q \in \mathbb{R}^d} \Pr(f^T_{\theta^*}(Q) | Q = q) \Pr(Q = q) \log \Pr(f^E_{\omega}(Q) | Q = q) \mathrm{d}q \\
&= \int_{Q \in \mathbb{R}^d} \Pr(f^T_{\theta^*}(Q) | Q = q) \Pr(Q = q) \log \Pr(f^T_{\theta^*}(Q) | Q = q) \mathrm{d}q + \mathbb{E}_{q \sim \mathcal{D}^{\mathcal{Q}}} \left[l(f^T_{\theta^*}(q)), f^E_{\omega}(q)) \right] \\
&\leq -H(f^T_{\theta^*}(Q) \mathrm{d}q + \mathbb{E}_{q \sim \mathcal{D}^{\mathcal{Q}}} \left[l(f^T_{\theta^*}(q)), f^E_{\omega}(q)) \right] \\
&\leq -H(f^E_{\omega}(Q) \mathrm{d}q + \mathbb{E}_{q \sim \mathcal{D}^{\mathcal{Q}}} \left[l(f^T_{\theta^*}(q)), f^E_{\omega}(q)) \right] \qquad (8)
\end{aligned}$$

The last inequality holds true as the extracted model f^E_{ω} is trained using the outputs of the target model $f^T_{\theta^*}$. Thus, by data-processing inequality, its output distribution possesses less information than that of the target model. Specifically, we know that if $Y = f(X)$, $H(Y) \leq H(X)$.

Now, by taking $\min_{\omega}$ on both sides, we obtain

$$D_{\mathrm{KL}}(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\mathrm{DEq}}}(Q), Q)) \leq \min_{\omega} \mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_{\omega}(Q))] - H(f^E_{\omega}(Q)).$$

Here, $\omega^*_{\mathrm{DEq}} \triangleq \arg\min_{\omega} D_{\mathrm{KL}}(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega}(Q), Q))$. The equality exists if minima of LHS and RHS coincide. □

***Theorem** 4.5* (Lower Bounding Information Leakage). The information leaked by any max-information attack (Equation 3) is lower bounded as follows:

$$I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\mathrm{MaxInf}}}(Q), Q)) \geq \max_{\omega} -\mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_{\omega}(Q))] + H(f^E_{\omega}(Q)).$$

Proof. Let us consider the same terminology as the previous proof. Then,

$$\begin{aligned}
&I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega}(Q), Q)) \\
&= H(f^T_{\theta^*}(Q), Q) + H(f^E_{\omega}(Q), Q) - H(f^T_{\theta^*}(Q), f^E_{\omega}(Q), Q) \\
&= H(f^T_{\theta^*}(Q), Q) + H(f^E_{\omega}(Q), Q) - H(f^E_{\omega}(Q), Q | f^T_{\theta^*}(Q)) + H(f^T_{\theta^*}(Q)) \\
&\geq H(f^E_{\omega}(Q), Q) - H(f^E_{\omega}(Q), Q | f^T_{\theta^*}(Q)) \qquad (9) \\
&\geq H(f^E_{\omega}(Q)) - H(f^E_{\omega}(Q), Q | f^T_{\theta^*}(Q)) \qquad (10)
\end{aligned}$$

[3] Throughout the proofs, we slightly abuse the notation to write $l(\Pr(X), \Pr(Y))$ as $l(X, Y)$ for avoiding cumbersome equations.

$$\geq H(f^E_\omega(Q)) - \mathbb{E}_Q[l(f^E_\omega(Q), f^T_{\theta^*}(Q))] \tag{11}$$

The inequality of Equation 9 is due to the fact that entropy is always non-negative. Equation 10 holds true as $H(X,Y) \geq \max\{H(X), H(Y)\}$ for two random variables X and Y. The last inequality is due to the fact that conditional entropy of two random variables X and Y, i.e. $H(X|Y)$, is smaller than or equal to their cross entropy, i.e. $l(X,Y)$ (Lemma B.1).

Now, by taking $\max_\omega$ on both sides, we obtain

$$I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\text{MaxInf}}}(Q), Q)) \leq \max_\omega -\mathbb{E}_Q[l(f^T_{\theta^*}(Q), f^E_\omega(Q))] + H(f^E_\omega(Q)).$$

Here, $\omega^*_{\text{MaxInf}} \triangleq \arg\max_\omega I(\Pr(f^T_{\theta^*}(Q), Q) \| \Pr(f^E_{\omega^*_{\text{MaxInf}}}(Q), Q))$. The equality exists if maxima of LHS and RHS coincide. □

Lemma B.1 (Relating Cross Entropy and Conditional Entropy)**.** *Given two random variables X and Y, conditional entropy*

$$H(X|Y) \leq l(X,Y). \tag{12}$$

Proof. Here, $H(X|Y) \triangleq -\int \Pr(x,y) \log \frac{\Pr(x,y)}{\Pr(y)}$ and $l(X,Y) \triangleq l(\Pr(X), \Pr(Y)) = -\int \Pr(x) \ln \Pr(y)$ denotes the cross-entropy.

$$\begin{aligned} l(X,Y) &= H(X) + D_{\text{KL}}(\Pr(X) \| \Pr(Y)) \\ &= H(X|Y) + I(X;Y) + D_{\text{KL}}(P_X \| P_Y) \\ &\geq H(X|Y) \end{aligned}$$

The last inequality holds as both mutual information I and KL-divergence D_{KL} are non-negative functions for any X and Y. □

C. Additional Experimental Results

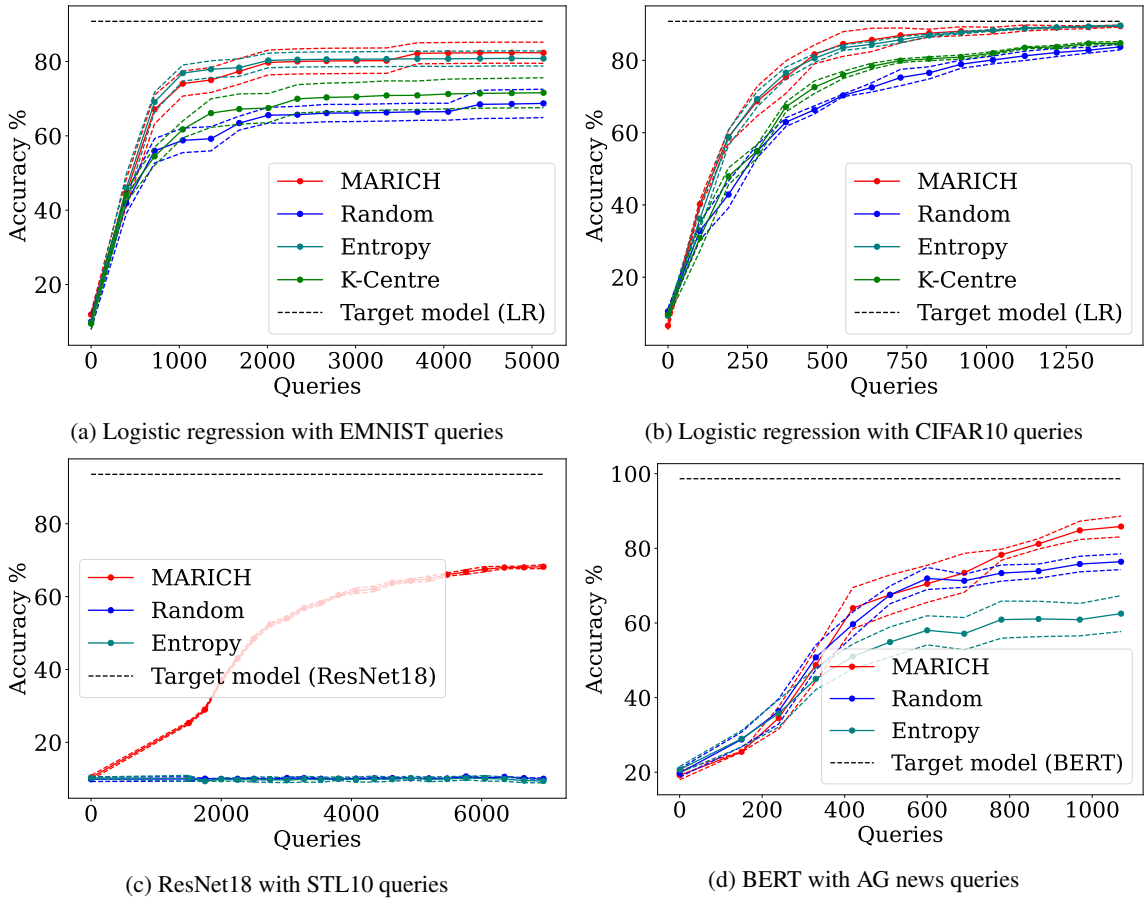
In this section, we elaborate further experimental setups that we skipped for the brevity of space in the main draft. We provide an anonymised version of the code at: https://drive.google.com/drive/folders/1mpM-zE3w_pIS0c3DDb_uiR9Jw_MYvVer?usp=sharing.

C.1. Accuracy of Models Extracted by MARICH and Other Sampling Strategies

To compare MARICH with other active learning algorithms, we attack the same target models using Entropy sampling, K-centre sampling, and Random sampling using the same number of queries as used for MARICH.

On the LR models we have shown performances of both Entropy sampling and K-centre sampling, while due to time and resource constraint, we could not present the K-centre sampling results for the BERT[4] and ResNet18.

From the results in Figure 5, we see that in most cases MARICH outperforms other algorithms. In Figure 2, we have not plotted the standard deviations for better visibility. We plot both mean $\pm$ standard deviation over 10 runs in Figure 5



(a) Logistic regression with EMNIST queries

(b) Logistic regression with CIFAR10 queries

(c) ResNet18 with STL10 queries

(d) BERT with AG news queries

Figure 5: Comparison among different active sampling algorithms.

Extraction of a CNN Trained on MNIST. Along with the other experiments mentioned in the paper, we trained a CNN with MNIST handwritten digits ($\mathbf{D}^T$ here), that shows a test accuracy of 97.53% on a disjoint test set. Two datasets are used as $\mathbf{D}^Q$ here, EMNIST letters and CIFAR10 to extract two CNN models from the target model. MARICH extracts two CNNs using queries from EMNIST letters and CIFAR10 which show test accuracies of 88.81% and 87.16% respectively. On the other hand, models extracted using random sampling using EMNIST letters and CIFAR10 show accuracies of 85.37% and 88.44% respectively. Table 2 contains queries used, and membership inference statistics for all the experiments.

[4]We use the pre-trained BERT model from https://huggingface.co/docs/transformers/v4.24.0/en/model_doc/bert#transformers.BertModel

C.2. Fidelity of The Prediction Distributions and Parameters of The Extracted Models

We claim to achieve distributionally equivalent f^E from f^T using MARICH. To measure the performance of MARICH on this objective, we measure KL divergence of the output distributions of f^T and f^E when the input is $\mathbf{D}^Q$after every round of training.

Extracting Logistic Regression with EMNIST Queries. In Figure 6, we observe that MARICH reaches KL-divergence value 0.141±0.123, while Entropy sampling and K-center sampling reach values 0.291±0.047 and 0.309±0.118, respectively. Thus, we infer that MARICH performs better than Entropy sampling and K-centre sampling in achieving distributional fidelity.

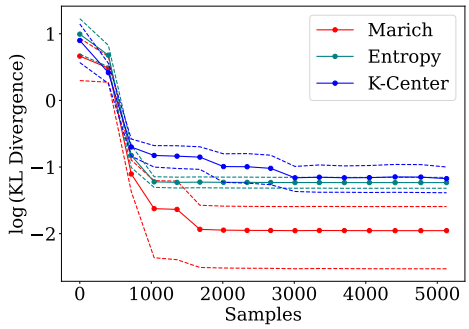


Figure 6: Comparison of fidelity of prediction distributions for different active learning algorithms (LR queried with EMNIST).

In Figure 7, we illustrate the agreement in predictions of f^E with f^T on test datasets using different active learning algorithms. With respect to agreement, we observe that MARICH achieves 96.683±2.022% while Entropy sampling and K-center sampling achieve 94.335±0.847% and 87.456±1.344%. Thus, we infer that in this particular case MARICH and Entropy sampling perform almost same but K-center sampling performs worse.

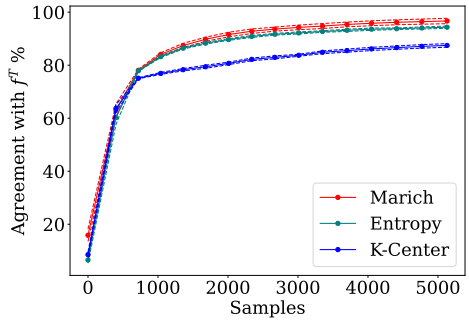


Figure 7: Comparison of agreement of f^E with f^T for different active learning algorithms (LR queried with EMNIST).

In Figure 8, we plot the parametric fidelity of different active learning algorithms. Let w_E be the parameters of the extracted model and w_T be the parameters of the target model. We define parametric fidelity as $F_w = \log||\frac{w_E}{|w_E|} - \frac{w_T}{|w_T|}||$. We observe that MARICH reaches 0.524±0.065, Entropy sampling and K-center sampling reach 0.606±0.03 and 0.721±0.041 respectively. From this we can infer that MARICH achieves better parametric fidelity than Entropy sampling and K-Center sampling.

Extracting BERT with AGNews Queries. While extracting a BERT model, we see the gain in fidelity due to MARICH more distinctly. Instead of K-center sampling, we show the performance of Random sampling and Entropy sampling as our resources did not allow K-center sampling.

In Figure 9, we observe that MARICH achieves KL-divergence value of 0.124±0.149, and Entropy sampling and Random sampling achieve 0.446±0.19 and 0.691±0.23 respectively. This shows that MARICH outperforms both Random sampling and Entropy sampling in terms of distributional fidelity.

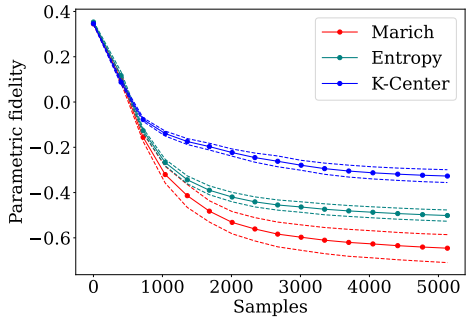


Figure 8: Comparison of fidelity w.r.t. parameters for different active learning algorithms (LR queried with EMNIST).

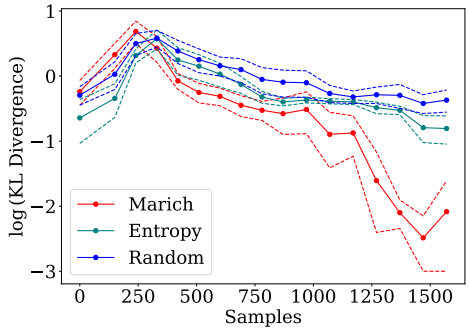


Figure 9: Comparison of fidelity of prediction distributions for different active learning algorithms (BERT queried with AGNews).

In Figure 10, we observe that MARICH achieves agreement value of 82.207±5.16%, while Entropy sampling and Random sampling achieve 75.045±3.05% and 70±6.456% respectively. This shows that MARICH outperforms both Entropy sampling and Random sampling in terms of agreement of f^E with f^T.

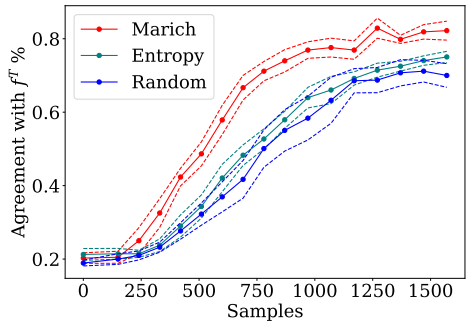
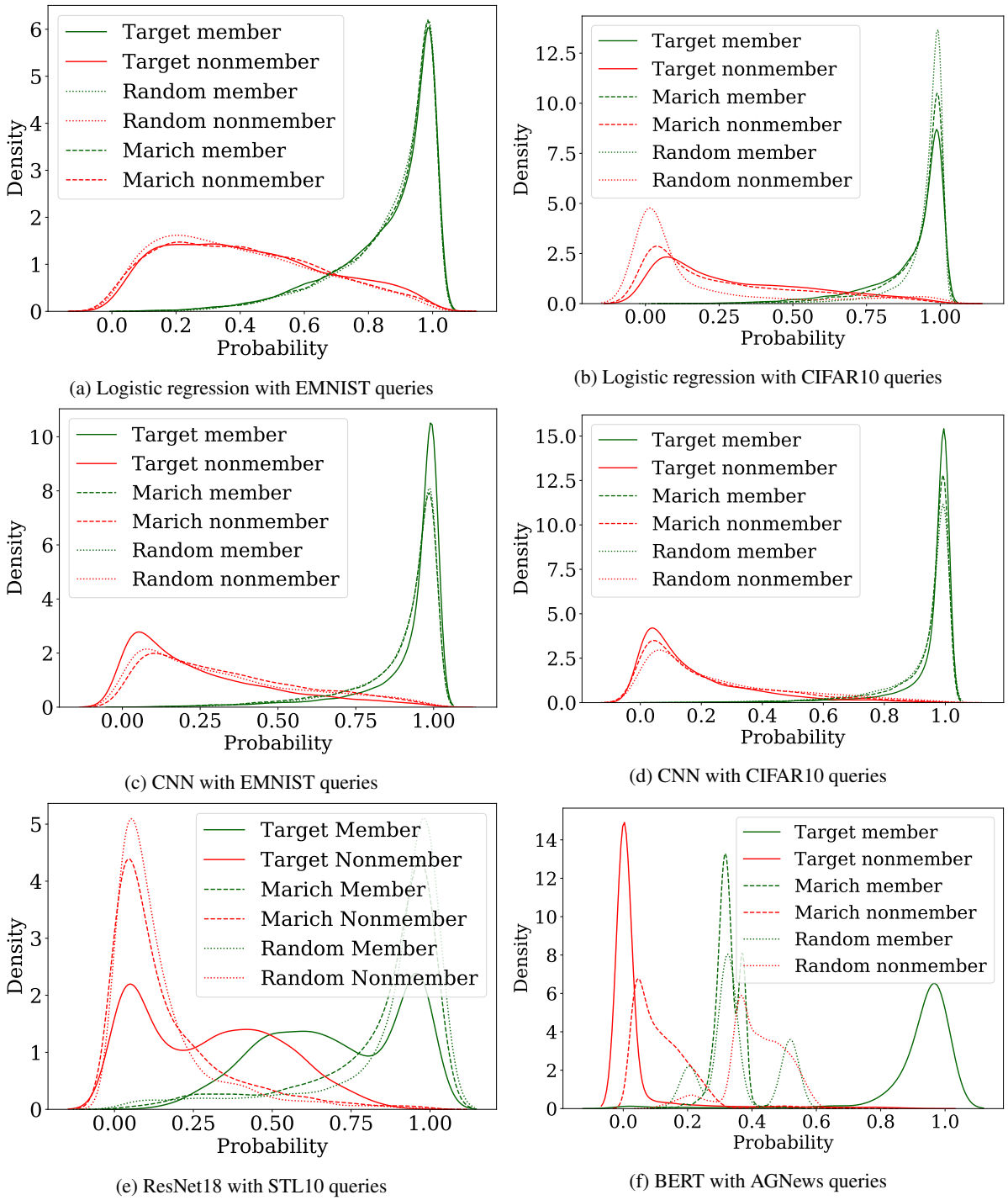


Figure 10: Comparison of agreement of f^E with f^T for different active learning algorithms (BERT queried with AGNews).

C.3. Membership Inference with the Extracted Models

From Figure 11, we see that in most cases the probability densities of the membership inference are closer to the target model when the model is extracted using MARICH, than using random sampling (RS).



(a) Logistic regression with EMNIST queries

(b) Logistic regression with CIFAR10 queries

(c) CNN with EMNIST queries

(d) CNN with CIFAR10 queries

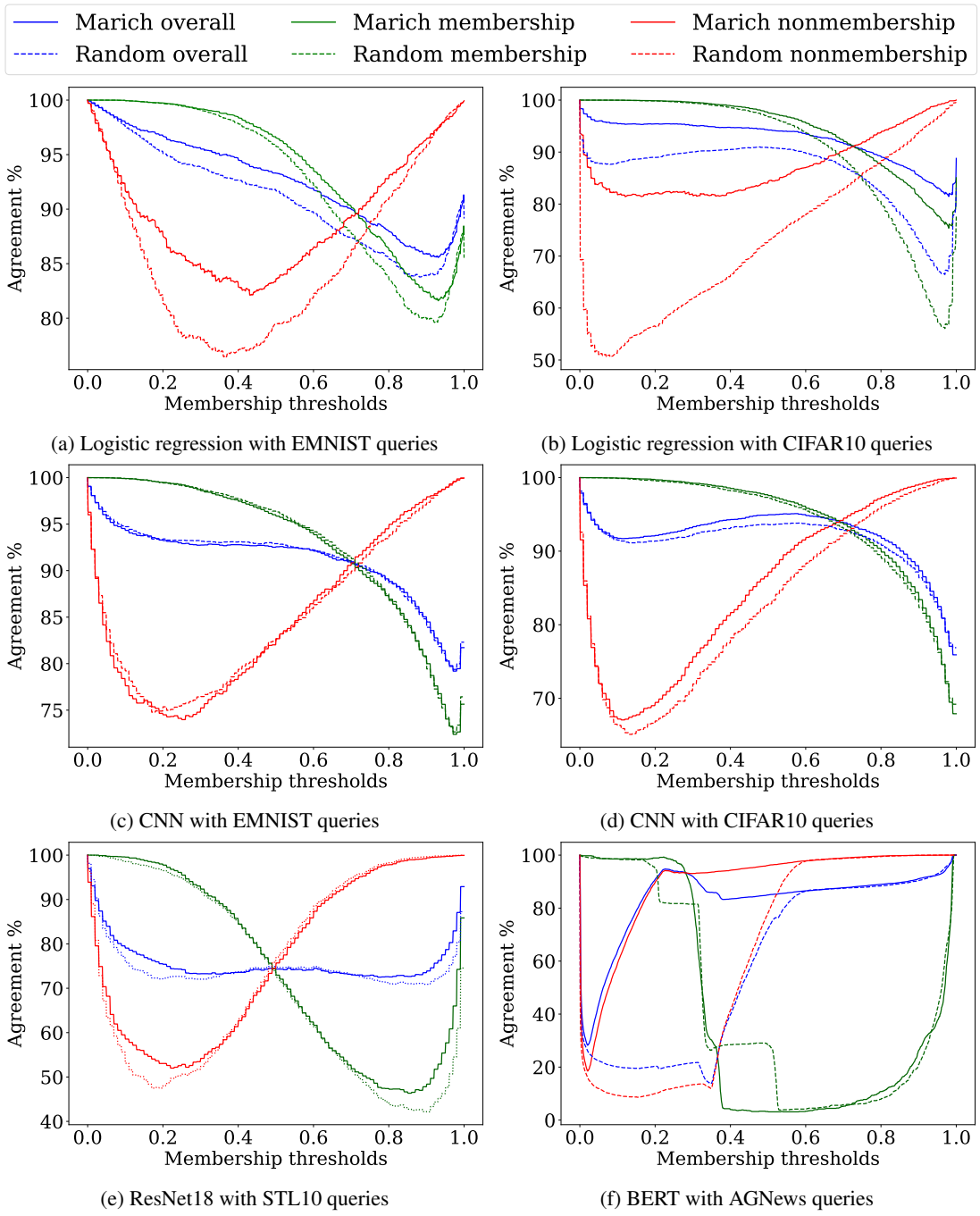
(e) ResNet18 with STL10 queries

(f) BERT with AGNews queries

Figure 11: Comparison among membership vs. non-membership probability densities for membership attacks against models extracted by MARICH, Random sampling and the target model. Each figure represents the model class and query dataset. Memberships and non-memberships inferred from the model extracted by MARICH are significantly closer to the target model.

In Figure 12, we present the agreements from the member points, nonmember points and overall agreement curves for varying membership thresholds, along with the AUCs of the overall membership agreements. We see that in most cases, the agreement curves for the models extracted using MARICH are above those for the models extracted using random sampling, thus AUCs are higher for the models extracted using MARICH.

These observations support our claim that model extraction using MARICH gives models are accurate and informative replica of the target model.



(a) Logistic regression with EMNIST queries

(b) Logistic regression with CIFAR10 queries

(c) CNN with EMNIST queries

(d) CNN with CIFAR10 queries

(e) ResNet18 with STL10 queries

(f) BERT with AGNews queries

Figure 12: Comparison of membership, nonmembership and overall agreements of membership attacks against models extracted by MARICH and Random sampling and the target model trained with MNIST. Each figure represents the model class and query dataset. Membership agreement of the models extracted by MARICH are higher.

Table 2: Model extraction and membership inference statistics

Member dataset	Target model	Attack Dataset	Algorithm used	Non-member dataset	Queries	Membership acc	Nonmembership acc	Overall membership acc	Overall Membership agreement	Membership agreement AUC
MNIST	LR	-	-	EMNIST	50,000 (100%)	94.91%	67.24%	87.99%	-	-
MNIST	LR	-	-	CIFAR10	50,000 (100%)	97.13%	77.80%	92.30%	-	-
MNIST	LR	EMNIST	MARICH	EMNIST	5,130 (3.5%)	95.16%	68.84%	88.58%	92.82%	92.72%
MNIST	LR	CIFAR10	MARICH	CIFAR10	1,420 (2.37%)	97.98%	83.16%	94.27%	93.97%	92.27%
MNIST	LR	EMNIST	Random Sampling	EMNIST	5,130 (3.5%)	95.57%	71.72%	89.61%	91.01%	91.04%
MNIST	LR	CIFAR10	Random Sampling	CIFAR10	1,420 (2.37%)	95.15%	84.98%	92.61%	89.84%	85.54%
MNIST	CNN	-	-	EMNIST	50,000 (100%)	91.82%	74.84%	87.57%	-	-
MNIST	CNN	-	-	CIFAR10	50,000 (100%)	94.94%	83.05%	91.97%	-	-
MNIST	CNN	EMNIST	MARICH	EMNIST	5,440 (3.73%)	95.07%	80.32%	91.38%	92.64%	91.23%
MNIST	CNN	CIFAR10	MARICH	CIFAR10	5545 (9.24%)	93.48%	78.66%	89.78%	94.92%	92.27%
MNIST	CNN	EMNIST	Random Sampling	EMNIST	5,440 (3.73%)	95.95%	79.50%	91.84%	92.35%	91.11%
MNIST	CNN	CIFAR10	Random Sampling	CIFAR10	5545 (9.24%)	97.59%	85.90%	94.66%	94.16%	91.35%
CIFAR10	Resnet18	-	-	STL10	40,000 (100%)	81.55%	77.15%	79.35%	-	-
CIFAR10	Resnet19	STL10	MARICH	STL10	6,950 (6.15%)	92.80%	91.75%	92.32%	75.52%	76.36%
CIFAR10	Resnet19	STL10	Random Sampling	STL10	6,950 (6.15%)	92.75%	95.05%	93.90%	75.25%	74.86%
BBCNews	BERT	-	-	AGNews	1,490 (100%)	91.69%	99.56%	98.61%	-	-
BBCNews	BERT	AGNews	MARICH	AGNews	1,070 (0.83%)	80.96%	96.25%	94.42%	91.02%	82.16%
BBCNews	BERT	AGNews	Random Sampling	AGNews	1,070 (0.83%)	23.01%	98.28%	89.17%	86.93%	58.64%

D. Significance and Comparison of Sampling Strategies

Given the bi-level optimization problem, we came up with MARICH in which three sampling methods are used in the order: (i) ENTROPYSAMPLING, (ii) ENTROPYGRADIENTSAMPLING, and (iii) LOSSSAMPLING.

These three sampling techniques contribute to different goals:

- ENTROPYSAMPLING selects points about which the classifier at a particular time step is most confused
- ENTROPYGRADIENTSAMPLING uses gradients of entropy of outputs of the extracted model w.r.t. the inputs as embeddings and selects points behaving most diversely at every time step.
- LOSSSAMPLING selects points which produce highest loss when loss is calculated between target model's output and extracted model's output.

One can argue that the order is immaterial for the optimization problem. But looking at the algorithm practically, we see that ENTROPYGRADIENTSAMPLING and LOSSSAMPLING incur much higher time complexity than ENTROPYSAMPLING. Thus, using ENTROPYSAMPLING on the entire query set is more efficient than the others. This makes us put ENTROPYSAMPLING as the first query selection strategy.

As per the optimization problem in Equation (7), we are supposed to find points that show highest mismatch between the target and the extracted models after choosing the query subset maximising the entropy. This leads us to the idea of LOSSSAMPLING. But as only focusing on loss between models may choose points from one particular region only, and thus, decreasing the diversity of the queries. We use ENTROPYGRADIENTSAMPLING before LOSSSAMPLING. This ensures selection of diverse points with high performance mismatch.

In Figure 13, we experimentally see the time complexities of the three components used. These are calculated by applying the sampling algorithms on a logistic regression model, on mentioned slices of MNIST dataset.

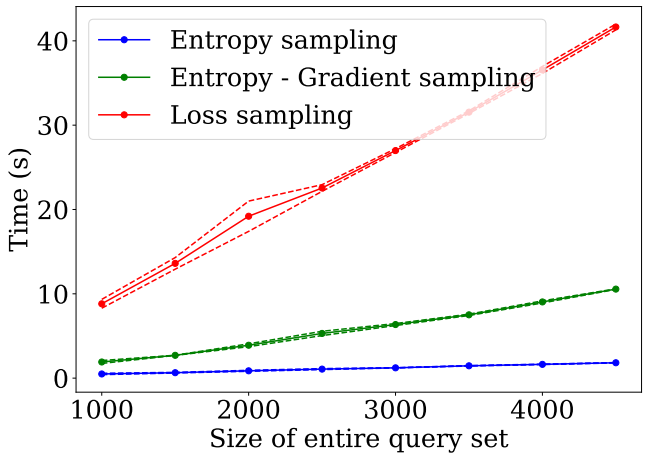


Figure 13: Runtime comparison of three sampling strategies to select queries from 4500 datapoints.

Table 3: Time complexity of different sampling Strategies

Sampling Algorithm	**Query space size**	**#Selected queries**	**Time (s)**
Entropy Sampling	4500	100	1.82 ± 0.04
Entropy-Gradient Sampling	4500	100	10.56 ± 0.07
Loss Sampling	4500	100	41.64 ± 0.69

E. Performance against differentially private target models

In this section, we aim to verify performance of MARICH against privacy-preserving mechanisms. Specifically, we apply a (ε, δ)-Differential Privacy (DP) inducing mechanism (Dwork et al., 2006; Dandekar et al., 2021) on the target model to protect the private training dataset. There are three types of methods to ensure DP: output perturbation (Dwork et al., 2006), objective perturbation (Chaudhuri et al., 2011; Dandekar et al., 2018), and gradient perturbation (Abadi et al., 2016). Since output perturbation and gradient perturbation methods scale well for nonlinear deep networks, we focus on them as the defense mechanism against MARICH's queries.

Gradient Perturbation-based Defenses. DP-SGD (Abadi et al., 2016) is used to train the target model on the member dataset. This mechanism adds noise to the gradients and clip them while training the target model. We use the default implementation of Opacus (Yousefpour et al., 2021) to conduct the training in PyTorch.

Following that, we attack the (ε, δ)-DP target models using MARICH and compute the corresponding accuracy of the extracted models. In Figure 14, we show the effect of different privacy levels ε on the achieved accuracy of the extracted Logistic Regression model trained with MNIST dataset and queried with EMNIST dataset. Specifically, we assign $\delta = 10^{-5}$ and vary ε in $\{0.2, 0.5, 1, 2, \infty\}$. Here, $\varepsilon = \infty$ corresponds to the model extracted from the non-private target model.

We observe that the accuracy of the models extracted from private target models are approximately $2.3 - 7.4\%$ lower than the model extracted from the non-private target model. This shows that performance of MARICH decreases against DP defenses but not significantly.

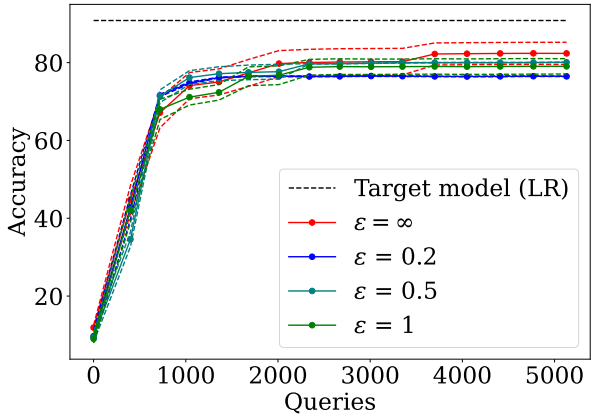
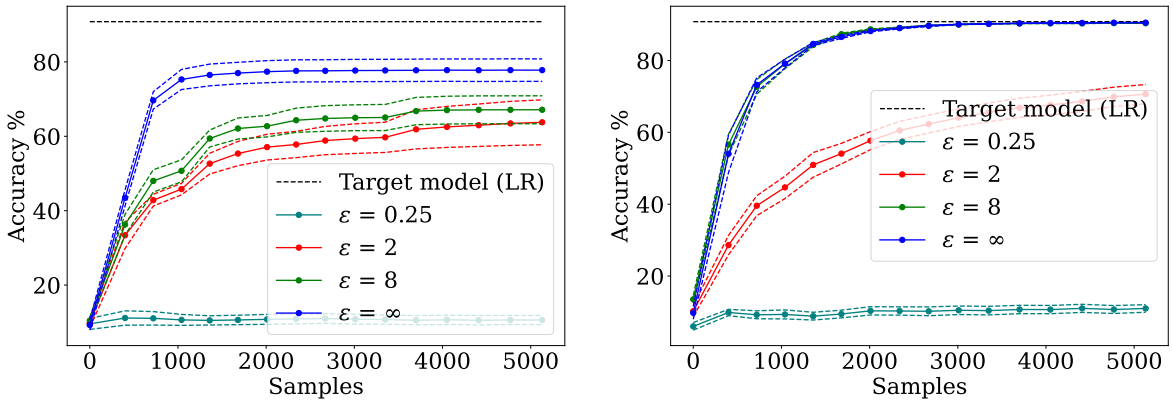


Figure 14: Performance of models extracted by MARICH against (ε, δ)-differentially private target models trained using DP-SGD. We consider different privacy levels ε and $\delta = 10^{-5}$. Accuracy of the extracted models decrease with increase in privacy (decrease in ϵ).

Output Perturbation-based Defenses. Perturbing output of an algorithm against certain queries with calibrated noise, in brief output perturbation, is one of the basic and oldest form of privacy-preserving mechanism (Dwork et al., 2006). Here, we specifically deploy the Laplace mechanism, where a calibrated Laplace noise is added to the output of the target model generated against some queries. The noise is sampled from a Laplace distribution $\text{Lap}(0, \frac{\Delta}{\varepsilon})$, where Δ is sensitivity of the output and ε is the privacy level. This mechanism ensures ε-DP.

We compose a Laplace mechanism to the target model while responding to MARICH's query and evaluate the change in accuracy of the extracted model as the impact of the defense mechanism. We use a logistic regression model trained on MNIST as the target model. We query it using EMNIST and CIFAR10 datasets respectively. We vary ε in $\{0.25, 2, 8, \infty\}$. For each ε and query dataset, we report the mean and standard deviation of accuracy of the extracted models on a test dataset. Each experiment is run 10 times.

We observe that decrease in ε, i.e. increase in privacy, causes decrease in accuracy of the extracted model. For EMNIST queries (Figure 15a), the degradation in accuracy is around 10% for $\varepsilon = 2, 8$ but we observe a significant drop for $\varepsilon = 0.25$. For CIFAR10 queries (Figure 15b), $\varepsilon = 8$ has practically no impact on the performance of the extracted model. But for



(a) LR with EMNIST queries

(b) LR with CIFAR10 queries

Figure 15: Performance of models extracted by MARICH against target models that perturbs the output of the queries to achieve ε-DP. We consider different privacy levels ε. Accuracy of the extracted models decrease with increase in privacy (decrease in ϵ).

$\varepsilon = 2$ and 0.25, the accuracy of extracted models drop down very fast.

Thus, we conclude that output perturbation defends privacy of the target model against MARICH for smaller values of ε. But for larger values of ε, the privacy-preserving mechanism might not save the target model significantly against MARICH.

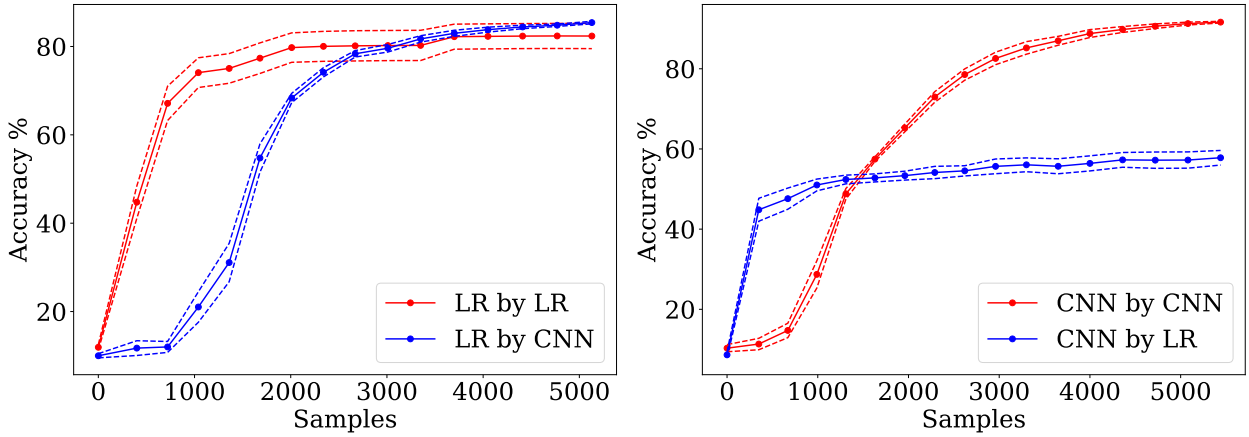
F. Effect of Model Mismatch

From Equation equation 7, we observe that functionality of MARICH is not constrained by selecting the same model class for both the target model f^T and the extracted model f^E. But in all the previous experiments, we have used the same model class for both the target and extracted models, i.e., we have used LR to extract LR or CNN to extract CNN. In this section, we conduct experiments to show MARICH's capability to handle model mismatch and impact of model mismatch on performance of the extracted models.

Specifically, we run experiments for two cases. We train an LR and a CNN model on MNIST dataset, and use them as target models. We further extract these two models with two other LR and CNN models using EMNIST as the query datasets. We use MARICH without any modification for both the cases when the model classes match and mismatch. This shows universality of MARICH as a model extraction attack.

From Figure 16, we observe that model mismatch influences performance of the model extracted by MARICH. When we extract the LR target model with LR and CNN, we observe that both the extracted models achieve almost same accuracy and the extracted CNN model achieves even a bit more accuracy than the extracted LR model. In contrast, when we extract the CNN target model with LR and CNN, we observe that the extracted LR model achieves lower accuracy than the extracted CNN model.

From these observations, we conclude that if we use a less complex model to extract a more complex model, the accuracy drops significantly. But if we extract a low complexity model with a higher complexity one, we obtain higher accuracy instead of model mismatch. This is intuitive as the low-complexity extracted model might have lower representation capacity to mimic the non-linear decision boundary of the high-complexity model but the opposite is not true.



(a) LR extracted by LR vs. LR extracted by CNN

(b) CNN extracted by CNN vs. CNN extracted by LR

Figure 16: Effect of model mismatches on MARICH.

Table 4: Effect of Model mismatch on Accuracy of The Extracted Models.

f^E	f^T	#**samples**	**Accuracy**
LR	LR	5130	$82.37 \pm 5.7\%$
LR	CNN	5130	$85.41 \pm 0.57\%$
CNN	LR	5440	$57.81 \pm 3.64\%$
CNN	CNN	5440	$91.63 \pm 0.42\%$

G. Choices of Hyperparameters

In this section, we list the choices of the hyperparameters of Algorithm 1 for different experiments and also explain how we select them.

Hyperparameters γ_1 and γ_2 are kept constant, i.e., 0.8, for all the experiments. These two parameters act as the budget shrinking factors.

Instead of changing these two, we change the number of points n_0, which are randomly selected in the beginning, and the budget B for every step. We obtain the optimal hyperparameters for each experiment by performing a line search in the interval $[100, 500]$.

We further change the budget over the rounds. At time step t, the budget, $B_t = \alpha^t \times B_{t-1}$. The idea is to select more points as f^E goes on reaching the performance of f^T. Here, $\alpha > 1$ and needs to be tuned. We use $\alpha = 1.02$, which is obtained through a line search in $[1.01, 1.99]$.

For number of rounds T, we perform a line search in $[10, 20]$.

Table 5: Hyperparameters for different datasets and target models.

Member dataset	**Target model**	**Attack dataset**	**Budget**	**Initial points**	γ_1	γ_2	**Rounds**	**Epochs/Round**	**Learning Rate**
MNIST	LR	EMNIST	500	400	0.8	0.8	14	20	2×10^{-2}
	LR	CIFAR10	150	100	0.8	0.8	14	20	2×10^{-2}
	CNN	EMNIST	500	350	0.8	0.8	15	20	1×10^{-2}
CIFAR10	ResNet18	STL10	390	1500	0.8	0.8	20	4	4×10^{-4}
BBC News	BERT	AG News	150	150	0.8	0.8	10	3	2×10^{-6}