Membership Inference Attacks and Generalization: A Causal Perspective

Teodora Baluta
National University of Singapore
Singapore
teobaluta@comp.nus.edu.sg

Shiqi Shen
National University of Singapore
Singapore
shiqi04@u.nus.edu

S. Hitarth*
Hong Kong University of Science and Technology
Hong Kong
hsinghab@connect.ust.hk

Shruti Tople
Microsoft Research
Cambridge, United Kingdom
shruti.tople@microsoft.com

Prateek Saxena
National University of Singapore
Singapore
prateeks@comp.nus.edu.sg

ABSTRACT

Membership inference (MI) attacks highlight a privacy weakness in present stochastic training methods for neural networks. It is *not* well understood, however, why they arise. Are they a natural consequence of imperfect generalization only? Which underlying causes should we address during training to mitigate these attacks? Towards answering such questions, we propose the first approach to explain MI attacks and their connection to generalization based on principled *causal reasoning*. We offer causal graphs that quantitatively explain the observed MI attack performance achieved for 6 attack variants. We refute several prior non-quantitative hypotheses that over-simplify or over-estimate the influence of underlying causes, thereby failing to capture the complex interplay between several factors. Our causal models also show a new connection between generalization and MI attacks via their shared causal factors. Our causal models have high predictive power (0.90), i.e., their analytical predictions match with observations in unseen experiments often, which makes analysis via them a pragmatic alternative.

CCS CONCEPTS

• **Security and privacy → Software and application security**; • **Computing methodologies → Machine learning**; **Causal reasoning and diagnostics**.

KEYWORDS

membership inference attacks; generalization; causal reasoning

ACM Reference Format:
Teodora Baluta, Shiqi Shen, S. Hitarth, Shruti Tople, and Prateek Saxena. 2022. Membership Inference Attacks and Generalization: A Causal Perspective. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3548606.3560694

*Work done while interning at National University of Singapore.

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3560694

1 INTRODUCTION

As the use of machine learning proliferates, privacy has become a key concern in machine learning [40] with several classes of attacks being discovered. Membership inference (MI) attacks, which have led to a flurry of works recently [9, 16, 20, 29–31, 31, 32, 36, 37, 56, 60, 61, 70, 71], capture the advantage of an adversary in distinguishing samples used for training from those that were not. There is clear empirical evidence that MI attacks are effective and many new attack variants are emerging. At the same time, there is currently no systematic framework to understand *why* standard training procedures leave deep nets susceptible to MI attacks.

There are two incumbent approaches to understanding why deep networks are susceptible to MI attacks [31, 48, 60, 65, 69]. The first tries to offer *fully mechanistic* explanations derived from theoretical analysis. For instance, a line of work tries to mathematically model the stochastic mechanism of training models (e.g., using stochastic gradient descent (SGD)) with enough precision [48]. The most commonly accepted mechanistic explanation is that ML models leak training data because they fail to generalize, measured through quantities like overfitting gap, accuracy gap, and so on [48, 69]. This approach is appealing and is actively progressing, but at the same time, modeling the training process with closed-form mathematical expressions is an inherently difficult problem. Predictions from these mechanistic explanations often do not agree with observations in experiments, because the approximations or assumptions made in the analysis may not hold in practice [69]. Furthermore, generalization offers a one-way explanation: if models generalize almost perfectly in a particular sense, then MI attacks are expected to be ineffective on average. It does not say how well MI attacks will work for models that may have not generalized perfectly, which is often the case in practice. As a result, there have been many different hypothesized root causes that go beyond direct measures of classical generalization, such as model capacity and architecture. In short, no single coherent mechanistic explanation today predicts the average performance of existing MI attacks well.

A second approach to explaining MI attacks is based on *statistical testing* of hypotheses: Researchers intuit about the root cause, run experiments, and then report statistical correlations between the hypothesized cause and the performance of the MI attacks [28, 31, 57, 60, 65]. For example, several works have suggested that the empirically observed overfitting gap or the accuracy difference

between training and testing sets explain MI attacks. This approach, while being important in its own right, fails to provide satisfactory explanations often as well. Guessing which root causes are really responsible for attacks is difficult; after all, the stochastic process of standard training procedures is complex and is affected by multiple possible sources of randomness, hyper-parameter selection, and sampling bias. Furthermore, correlation is not always causation. Confusing the two can result in overly simplistic explanations of the true phenomenon at hand and lead to paradoxes. Lastly, the purely empirical approach leaves no room to accommodate mechanistic axioms (things we know that ought to be true from our theoretical understanding)—if the observations do not correlate with mechanistically derived facts, then they remain unexplained.

Our approach. We propose a new approach that explains MI attacks through a *causal model*. A causal model is a graph where nodes are random variables that abstractly represent properties of the underlying stochastic process and edges denote cause-effect relationships between them. We can model the process of sampling data sets, picking hyper-parameters like the size of the neural network, output vectors, generalization parameters like bias and variance, and predictions from MI attack procedures as random variables. These random variables can be measured empirically during experiments. We can then both encode and infer causal relationships between nodes quantitatively through equations. Edges in our causal model are of two types: 1) mechanistically derived edges denote known mathematical facts derived from domain knowledge (prior work, definitions, etc.); and 2) relations inferred from experimental observations using *causal discovery* techniques.

Our causal approach is substantially different from prior works and enables much deeper and principled analysis. The causal model, once learnt, acts like a predictive model—one can ask what will be the expected performance of a particular MI attack if the "root causes" (random variables in the model) were to have certain values not observed during prior experiments. Such estimation can be done without running expensive experiments. A causal model allows us to "single out" the effect of one variable on the MI attack performance. Such queries go beyond just observing statistical correlations because they need to reason about other variables that might affect both the cause and the outcome (the attack performance). To carefully solve these queries, we leverage the principled framework of causal reasoning known as *do-calculus*. It allows us to perform systematic refutation tests, which avoids confusing causation with correlation. Such tests quantitatively tell us how well the model fits the observed data and answer "what if" style of questions about surmised root causes. Further, we can compare causal models obtained for two different attacks to understand how their manifestation differs, or compare models with and without an intervention (e.g., by applying a defense) for a given attack to understand which root causes it neutralizes. Causal models offer a more principled and interactive way of examining MI attacks.

Resulting Findings. To showcase the utility of our approach, we study 6 well-known MI attacks and 2 defenses for deep neural networks trained using standard SGD training procedures. We analyze a list of intuitive "root causes" which have been suggested in prior works and formally specify them as 9 causal hypotheses. We analyze each of these 9 hypotheses for ML models with 2 types of loss functions, so we evaluate on 18 formalized hypotheses. Several salient findings have resulted from our causal analysis. First, we refute 7/18 previously hypothesized causes, highlighting the perils of understanding MI attacks purely from intuition or from statistical correlational analysis. Second, we find that different causes contribute differently to the average attack accuracy, dispelling the idea that a single explanation suffices for all 6 MI attacks we study. Our causal approach also models the attacks well (0.90), i.e., predict the observed attack accuracy. This betters prior single-cause explanations by $3-22\%$ in 17/24. Third, we show that two stochastic parameters inherent in the training process, namely **Bias** and **Variance**, govern both generalization achieved by ML models [23, 39, 68] and MI attack accuracy. This offers a more nuanced lens to connect generalization and MI attack accuracy from that offered by prior works [6, 32, 69]. Fourth, we show that defenses against MI attacks based on regularization reduce the influence of some of root causes, but fail to completely remove their effect.

Summary of Contributions. We propose the first use of causal analysis for studying membership inference attacks on deep neural networks. We derive causal models for 6 MI attacks by combining both known domain-specific assumptions and observations made from experiments. Our key contribution is a new quantitative connection between MI attacks and generalization, which enables refuting claims about causation with finer accuracy.

Availability

Our prototype implementation is publicly available on GitHub[1]. We refer to the Appendices several times throughout the text. They can be found in the full version of this paper [1].

2 MOTIVATION

Many intuitive explanations for privacy leakage have been put forward in prior works. The most widely accepted claim is that "*overfitted classifiers are more susceptible to MI attacks*", which has been backed by experimental correlational analyses [27, 28, 31, 32, 48, 57, 65, 69]. To evaluate the level of overfitting though, two different metrics have been proposed: the difference in the loss of training and non-training samples [48, 69], as well as the train-to-test accuracy gap [27, 28, 31, 32, 57, 65]. However, empirical evidence to the contrary has also been observed–both MI attacks and extraction attacks have been reported on well-generalized models [6, 7, 32]. Other potential contributing factors, such as model complexity / structure [37, 57, 60], the size of the training set [37, 57], the diversity of the training samples [57], how close a target model to attack is to the shadow model [49], and so on have been proposed, creating an unclear picture of why MI attack arise. We highlight 9 common hypotheses claimed in prior works below:

(H1) The overfitting gap is the cause of MI attacks that use multiple shadow models in the inference [57].
(H2) "The main idea behind our shadow training technique is that similar models trained on relatively similar data records using the same service behave in a similar way" [57].
(H3) Beyond overfitting, model complexity influences the membership inference attack accuracy [57].

[1] At https://github.com/teobaluta/etio.

(H4) The size of the training set is a contributing factor to the success of MI attacks [57].
(H5) The shadow model attack works even if the attack uses only one shadow model [49].
(H6) "If a model is more overfitted, then it is more vulnerable to membership inference attack." - for MI attacks that use a single shadow model [49].
(H7) There is no difference in the attack accuracy if we use the top-3 predictions in descending order vs. the whole prediction vectors [49].
(H8) Shadow model-based attacks transfer when there is a clear decision boundary between members and non-members [49].
(H9) The average generalization error explains the advantage of the threshold-based adversary (even when this attack assumes that the error is normally distributed) [69].

It is natural to ask: To what extent are these explanations correct? Do these hypothesized factors universally explain all MI attacks equally? What does achieving a certain level of generalization (eliminating overfitting) imply towards reducing MI attacks? Answering such queries requires a principled framework for reasoning even to phrase the right statistical quantities to measure—it is something that is easily prone to fallacious reasoning.

2.1 Pitfalls of Testing with Correlations

Let us consider two of the prior work hypotheses: (H1) Higher difference between train and test accuracy leads to higher MI attacks; and (H3) An increase in model complexity increases privacy leakage, i.e., larger models are more susceptible. One of the most prominent approaches to validating such hypotheses today is experimental validation through statistical correlation analysis [31, 32, 57, 65]. The analysis proceeds by observing how the train-test accuracy gap and attack accuracy for a chosen MI attack changes under different choices of model complexity (number of model parameters). For concrete illustration, we run a small-scale experiment for the multiple shadow model attack [57]. We train 2 deep nets with varying number of parameters on CIFAR10 dataset. We average the observed training and testing accuracy of the deep neural networks under multiple samples of the training datasets. For each of these models, we also run the shadow model attack separately [57], using a disjoint part of the training dataset for the shadow model training.

Hidden Causes. In Fig. 1, on the X-axis we plot the average train-to-test accuracy gap against the attack accuracy on the Y-axis. The overall trend of the relationship between train-to-test accuracy gap and the attack accuracy is positive, i.e., the larger the train-to-test accuracy gap, the higher the attack accuracy. But this trend is not seen in Fig. 1, when we group the trained neural nets by other criteria such as model complexity and architecture. We observe they are clumped into two clusters. Within those clusters, the observed trend is the *reverse*: the larger the accuracy gap, the lower the attack accuracy! This turns out to be a fallacious conclusion, because it fails to account for the effect of other factors such as model complexity or architecture indirectly on train-to-test accuracy gap or directly on the attack accuracy.

This paradox arises because there are *confounding factors* or *confounders* [44, 59] wherein different sub-populations of the data have contradictory statistical properties. Similar paradoxes can arise due to *selection bias* [2, 5, 44] or *collider bias* [3, 44]. Without resolving such issues, it is difficult to decide whether one should try reducing the train-to-test accuracy gap or model complexity.

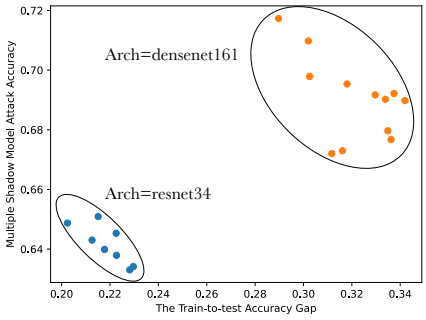

Figure 1: High train-to-test accuracy gap correlates with high attack accuracy in general, but if clustered on architecture type, an inverse relationship is visible.

Singling Out. When we want to decide which factors influence the end outcome more than the others, such as when designing practical defenses, one would like to "single out" the main causes and quantify how much they affect the expected MI attack accuracy. In our running example, the model complexity affects both variables, the train-to-test accuracy gap as well as the MI attack accuracy. It would be difficult to *quantify* how much it affects attack accuracy directly and how much indirectly via train-to-test accuracy gap without a more principled analysis of the observed data.

To understand the challenge, let us say we want to estimate how changing the train-to-test accuracy gap from $a = 0.007$ to $b = 0.914$ (which we observe in practice) affects the MI attack accuracy. Let us assume we train more NNs and that we now know that the model complexity is a confounding factor for both. A naive way to analyze this is to statistically estimate the following quantity: $E_1 = E[\texttt{MIAcc}|\texttt{AccDiff} \approx 0.9] - E[\texttt{MIAcc}|\texttt{AccDiff} \approx 0]$ where the `MIAcc` is the attack accuracy and the `AccDiff` is the train-to-test accuracy gap. Fig. 2 shows our new experimental observations and conditional probability estimates from data which reveals that the estimated expected effect is $E_1 = 0.47$. It is misleading to conclude that a change in train-to-test accuracy gap will have a large impact on the attack performance since we know that there is a confounder.

If we want to single out the effect of changes in train-to-test accuracy gap, the correct way is the following: Find samples with the same values for the model complexity but different values of train-to-test accuracy gap, from which we then compute the difference these produce on the attack accuracy. This is called analytically "controlling" for the confounding factor[2]. This corresponds to analytically computing how the system would behave under randomized values of model complexity. Such randomization "nullifies" or "smoothens out" the effect of model complexity. If we do this carefully, it turns out that the actual estimated effect E_2 when

[2]Controlling for a variable means binning data according to measured values of the variable

NumParams	AccDiff	MIAcc
12704	0.007	0.5
84542	0.104	0.564
203264	0.102	0.633
101632	0.102	0.632
9835146	0.109	0.567
	...	
9881316	0.914	0.880

$\mathbb{E}$[MIAcc | AccDiff=0.007] ≈ 0.50
$\mathbb{E}$[MIAcc | AccDiff=0.914] ≈ 0.97
⇒ Effect is E_1=0.47 ✗
⇒ Adjusted E_2=0.3 ✓

$\mathbb{E}$[MIAcc | NumParams=12704] ≈ 0.58
$\mathbb{E}$[MIAcc | NumParams=1334618] ≈ 0.94
⇒ Effect is E_3=0.36 ✓
⇒ Adjusted E_4=0.08 ✗

Figure 2: Reporting average conditional probabilities is not always correct. For example, if we are to estimate the effect of train-to-test accuracy gap on the MI attack accuracy from the data shown, the conditional over-estimates the effect by 0.17. For measuring the effect of model size though, the second estimate shown is correct since over-controlling for the `AccDiff` incorrectly decreases the effect to 0.08 from 0.36.

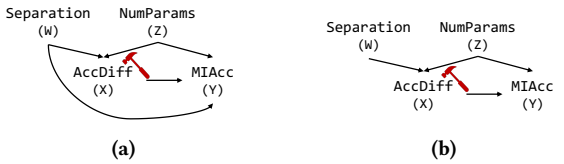

Figure 3: The query to estimate varies by assumptions chosen. If we assume that separation score influences the MI attack accuracy (Fig. 3a), we should control for two confounding factors, the separation score and the number of parameters. The resulting effect is 0.12. If we assume otherwise (Fig. 3b), we should not "control" for the separation score, otherwise it results in a much larger effect of 0.68.

`AccDiff` ranges from a to b is expressed by the following quantity:

$$E_2 = \sum_z E[\texttt{MIAcc}|\texttt{AccDiff} = a, \texttt{NumParams} = z]\Pr(\texttt{NumParams} = z) - \sum_z E[\texttt{MIAcc}|\texttt{AccDiff} = b, \texttt{NumParams} = z]\Pr(\texttt{NumParams} = z)$$

This leads to an estimated effect of $E_2 = 0.3$, as per our data (Section 5)—significantly lower than the naive analysis above.

Avoiding Over-controlling. It may be tempting to control for all factors that may influence the outcome. But arbitrarily controlling for variables leads to fallacious reasoning as well. For example, if we want to estimate the effect of the model complexity on the attack accuracy, should we now control for the train-to-test accuracy gap? If we were to control for the train-to-test accuracy gap, then the estimated effect E_4 when `NumParams` varies from a' to b' is given by (also shown in Fig. 2):

$$E_4 = \sum_z E[\texttt{MIAcc}|\texttt{NumParams} = a', \texttt{AccDiff} = z]\Pr(\texttt{AccDiff} = z) - \sum_z E[\texttt{MIAcc}|\texttt{NumParams} = b', \texttt{AccDiff} = z]\Pr(\texttt{AccDiff} = z)$$

The above expression is analogous to the case where we controlled for model complexity, except we are controlling for the train-to-test accuracy gap now. The estimated value from experiments for this statistic is $E_4 = 0.08$. This is, however, an incorrect analysis. If we analytically control for the train-to-test accuracy gap, then we are actually biasing the total effect that the model complexity has on the attack, as we are "blocking" (failing to distinguish) its indirect effect through the train-to-test accuracy gap. The correct statistical quantity, in this case, turns out to be $E_3 = E[\texttt{MIAcc}|\texttt{NumParams} = a'] - E[\texttt{MIAcc}|\texttt{NumParams} = b']$, i.e., the total effect model complexity has on the attack accuracy. The estimated effect is $E_3 = 0.36$—a lot higher than that obtained from the incorrect analysis, and corresponds to the second unadjusted estimate in Fig. 2. Another similar example of bad control or over-controlling is the bias amplification problem or pre-treatment control [43] (illustrated later in Fig. 3). The main takeaway is that a principled framework would tell us which quantities to estimate, avoiding over-controlling in experiments and false conclusions.

Large Number of Possible Factors. When moving beyond a couple of factors to consider, the reasoning can become more complicated. To build on our previous example, let us now introduce another factor, related to (H8): the separation between members and non-members. Our hypothesis is that the separation between members and non-members is influenced by both the model complexity and the train-to-test accuracy gap, and in turn it influences the attack accuracy. How does changing the separation then affect the attack accuracy? To answer this question, notice that the separation is influenced by the model complexity and the train-to-test accuracy gap, both of which influence the attack accuracy. Similar to what we described so far, one will then have to make sure to randomize these factors in order to obtain the effect of the separation on the MI attack accuracy. For every additional factor, though, we need to do enough experiments to "randomize" our estimates so that they correctly compute the effect on the attack accuracy. It is easy to see that the number of experiments one needs to run quickly starts to grow large as the number of factors considered increases.

Importance of Specifying Assumptions. So far, we have considered cases where certain causal relationships exist and we are trying to correctly estimate the effect of certain factors on the outcome. But, how can we start to test our assumptions, i.e., whether a causal relationship exists at all? Such refutation is hard to do, in general. A practical recourse is that one can specify their assumed beliefs and hope to refute quantitatively under the assumptions. The choice of assumptions matters critically to the outcome. To illustrate this, consider the hypothesis (H8) again, which introduces a separation score that measures the distance between members and non-members as a factor for the single shadow model MI attack [49] (so far, we have considered the multiple shadow one [57]). Deciding whether the separation score has any direct influence on the MI attack is critically important—if we choose to assume so, we get one set of conclusions, if we do not, we get another. When the separation has a direct influence on MI attack, then the principled analysis to estimate the effect of train-to-test accuracy gap is similar to the case of estimating the model complexity. We estimate the effect on the attack accuracy is 0.12 on our set of experiments (the details of our experimental setup are in Section 5.1). In the alternative scenario, the correct quantity to estimate is below, leading to the estimated effect of the train-to-test accuracy gap to be $E_5 = 0.68$

when it varies from the a to b.

$$E_5 = \sum_z E[\texttt{MIAcc}|\texttt{AccDiff} = a, \texttt{NumParams} = z]\Pr(\texttt{NumParams} = z) - \sum_z E[\texttt{MIAcc}|\texttt{AccDiff} = b, \texttt{NumParams} = z]\Pr(\texttt{NumParams} = z)$$

We illustrate the differences in the two sets of assumptions in Fig. 3 which exacerbates the bias amplification problem. We point out that prior works do *not* specify such assumptions or beliefs explicitly, making it impossible to refute or validate such hypotheses.

Goodness of Explanations. Given the subjectivity of assumptions and computational limits on the number of experiments one can run, it is difficult to analytically argue that a given explanation is "correct" or certain hypothesis is conclusively "incorrect". How then can we measure how good or correct is an explanation? A practical way to do so is look at the *predictive power* of a given explanation, i.e., measure how accurately it can predict the outcome (MI attack accuracy) under experimental settings *not* seen during creating the explanation. The highlighted prior hypothesis (H1)-(H9) often have predicted power well below 85% on average, offering less satisfying results. In contrast, our approach has predictive power of 3 – 22% higher than the prior work hypotheses, for most attacks we study.

3 THE CAUSAL MODELLING APPROACH

The prior common hypotheses, some of which are derived from mechanistic explanations or theoretical analyses, provide a good starting point to reason about potential factors of the MI attacks. But, as shown throughout Section 2, there are several pitfalls in identifying the factors and estimating their effect. Our aim is to infer a model defined over a set of potential factors and a given MI attack, not just a simple correlation of each factor separately with the MI attack. The model explicitly defines relationships between factors and the MI attack and between themselves. It should also provide a query interface for the following query types:

- *Prediction Queries*: Given some observed values $a_1, \ldots, a_n$ of the potential factors $X_1, \ldots, X_n$, what is the predicted MI attack accuracy Y: $E[Y|X_1 = a_1, \ldots, X_n = a_n]$?
- *Interventional Queries*: What is average effect of a potential factor on the attack accuracy if that factor had taken a different value from the observed one?

The prediction query consists of a set of assignments of observed (from running experiments) values for a set of factors, and the target variable Y. The output of this query is the expected MI attack accuracy conditioned on the observed values. Such queries help us measure how well the causal model agrees with observations in experiments. The interventional query is a "what if" query. It consists of two variables: the potential cause variable, called the treatment variable X, and the desired outcome variable Y. For instance, to estimate the effect of the train-to-test accuracy gap on the attack, we ask if the train-to-test accuracy gap had taken the value 0.1 compared to having no train-to-test accuracy gap, what is the expected MI attack accuracy? We want our causal model to be 1) **Accurate**, i.e., to have a goodness of fit, and 2) **Principled**, i.e., the estimated effect is rigorously computed.

We introduce a novel methodological shift: Our approach proposes to use *causal reasoning* to disambiguate potential factors of MI

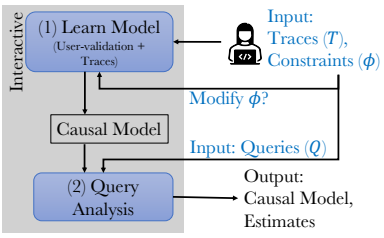

Figure 4: The user provides the domain knowledge ϕ and the traces T. The traces contain observations of the values that the factors of interest take for the training algorithm $\mathcal{A}$ and attack A. After an interactive step, the user fixes on a causal graph on which the input queries are analyzed.

attacks while satisfying the goals highlighted above. We combine mechanistic explanations from domain knowledge with automated inferences from empirical data to infer a causal model. Specifically, causal models are directed acyclic graphs (DAGs) defined over a set of variables and a set of directed edges[3] where each directed edge from variable X to Y represents a "X causes Y" relationship. On top of the graph structure, causal models are quantitative: each node in the graph has associated an equation that describes the cause-and-effect relationship between the node and its parents. Our approach is necessarily synergistic: without domain knowledge constraints, purely observational data cannot distinguish cause-and-effect; and without observational data, we cannot test our intuitions or extract more insights from experiments. The whole process is interactive and it is illustrated in Fig. 4. Initially, the user identifies potential factors or variables of interest of the underlying training and attack procedure such as training hyper-parameters, train-to-test accuracy gap and the outcome MI attack accuracy. We model these as "random variables" that can be observed and measured. The user generates the set of observations for these variables which we call traces T, by effectively running experiments and recording the values of the variables. The domain knowledge constraints (ϕ) formally describe the mechanistic explanations, facts or assumptions that stem from the data-generating process, e.g., the training and attack procedures. Given the traces and the domain constraints, we output a causal graph which the user can choose to further refine (*Modify* ϕ step in Fig. 4). Finally, the causal models encode cause-and-effect relationships by construction and can support the 2 types of queries. The user can specify these queries (Q) formally and obtain estimates on the inferred causal graph (step (2) in Fig. 4).

Inputs & Outputs. We have prototyped our approach in an interactive tool called Etio and envision model practitioners and researchers as its main users. Etio minimally requires a set of traces corresponding to the runs of a specific training algorithm ($\mathcal{A}$) over the training dataset (D). These traces record values of a set of properties about the training algorithm, model, and the performance metrics of the attack procedure (A)–all of which we call variables (V). The user additionally specifies a set of domain knowledge constraints which encode knowledge that two variables are not in a causal relationship, e.g., if they are caused by

[3]We thus interchangeably use causal model and causal graph.

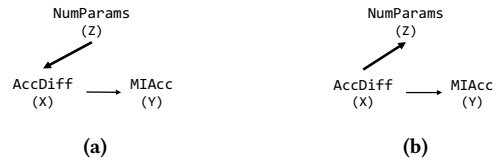

Figure 5: Two causal models that are not indentifiable (distinguishable) from observations, since both result in the same conditional (in)dependence relations, but require different quantities to estimate in a causal analysis.

the same unmeasurable/confounder variable (which we denote as Forbid constraints) or that there is a causal relationship between two variables (denoted as Enforce constraints). Then the domain constraints ϕ are a concatenation of the Forbid and Enforce sets of constraints. For the studied MI attacks, we describe the variables of interest V in Section 4.1 and our domain constraints ϕ in Section 4.2 in detail. In addition to the inputs necessary to infer the causal model, Etio allows the user to pose well-reasoned queries about potential factors of MI attacks, as per the query interface.

3.1 Learning the Causal Model

Despite the clear advantage of explicitly expressing assumptions in the form of an interpretable causal graph, constructing one is challenging. The fundamental issue is that while associations or correlational analysis are useful for predicting outcome, they do not always reflect the causal relationship. Associations can at most reveal relationships of dependence or (conditional) independence.

To illustrate this point, we show two models that describe the same conditional independence relationships in Fig. 5, but are causally different. In Fig. 5a, the model encodes that the model complexity affects the train-to-test accuracy gap which in turn influences the MI attack accuracy. In contrast, the model in Fig. 5b describes that the accuracy difference affects both the model complexity and the MI attack accuracy. The models in Fig. 5a and 5b, though, are indistinguishable from one another purely from observations, they both encode that $\texttt{NumParams} \perp\!\!\!\perp \texttt{MIAcc}|\texttt{AccDiff}$. But, in Fig. 5b, the model complexity has no causal effect on the MI attack, whereas in Fig. 5a the model complexity causes the MI attack to change through the train-to-test accuracy gap. In fact, from how the experiment is set up, the second relationship does not have any real-world interpretation, i.e., the model complexity is decided beforehand as a hyper-parameter to the training process. Thus, our approach must rely on domain knowledge, a specification of which is missing in prior works in the literature.

Formally, causal models (G, θ) consist of (a) a DAG $G = (V, E)$ called a causal graph, over a set V of vertices and (b) a joint probability distribution $\mathcal{P}_\theta(V)$, parameterized by θ over the variables in V[4]. The set of variables V can take either discrete or continuous values. Our framework is orthogonal to the underlying representation of parameters. We choose a linear model to represent the relationship between the nodes of the graph. For predictive queries, the parameters have a probabilistic interpretation $\mathcal{P}_\theta(V) = \Pi_i \texttt{Pr}(X_i|pa_{X_i}), X_i \in V$. Each node has associated with it a probability function based on its parent nodes pa_{X_i}. In this work, we utilize the linear Gaussian model: X_i is a linear Gaussian of its parents X_j: $X_i = \beta_0 + \sum_j \beta_j X_j + \epsilon$ where $X_j \in pa_{X_i}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Note that our choice of linear equations and Gaussian probability functions are not fundamental—these can be changed if necessary. These choices have been sufficient to create causal models with good predictive power for the attacks we analyze (see Section 5.2).

To learn the causal graph, there are two sub-steps: (a) learning the structure of the graph $G = (V, E)$ from the traces T and constraints ϕ and (b) learning the parameters of the causal graph. Conceptually, the first sub-goal is to maximize the posterior probability $\texttt{Pr}(G|T) = \texttt{Pr}(G)\texttt{Pr}(T|G)$, where $\texttt{Pr}(G)$ is a prior on the graph (i.e., G contains the edges represented by the Enforce list and all graphs with edges that are part of the Forbid have 0 probability) and $\texttt{Pr}(T|G)$ is the predictive probability of the graph G. Ideally, the posterior probability concentrates around a single structure G_{MAP}, the optimal directed acyclical graph. Learning the optimal DAG though is intractable for most problems as the number of DAGs is super exponential with the number of nodes $O(n!2^{\binom{n}{2}})$ [46]. In fact, recovering the optimal DAG with a bounded in-degree ≥ 2 has been shown to be NP-hard [8].

We choose to instantiate our approach with a standard hill-climbing algorithm [10, 67], an iterative Greedy approach that starts from the graph with nodes representing the variables V and the edges that are part of the Enforce. The algorithm does not guarantee that the produced graph is the optimal one but it is scalable. Since our goal is to disambiguate between many different possible factors (see Section 2), this technique allows the user to add new variables of interest and has a good predictive accuracy (goodness of fit) in practice. The algorithm iteratively tries to add, remove, or reverse the direction of a directed edge from the graph at the previous step. It uses a scoring function to choose between these operations. The scoring function maps a graph to a numeric value. We use a type of score based on log-likelihood $LL(G|T)$ but that prefers simpler graphs ($LL(G|T) - p$, where p is a penalizing term that grows with more complex causal models with more parameters). This is known as Bayesian Information Criterion [50]. For each such operation, the hill-climbing algorithm computes the change in the score if that operation had been performed. It then picks the operation that results in the best score and stops when no further improvements are possible. Moreover, several distinct graphs G can have similarly high posterior probabilities which is common when the data size is small compared to the domain size [13]. This is in part due to the the causal ambiguity of learning from data.

Instead of learning a single graph, Etio uses a bootstrapping technique [12]. The bootstrapping process resamples the traces T with replacement. It then returns a set $\mathcal{S}$ of multiple bootstrap datasets S. For each bootstrap dataset, Etio uses the graph learning algorithm to learn the structure of the graph G'. For every arc present in the set of graphs AG, Etio estimates the strength or confidence that each possible edge e_i is present in the true DAG as $\hat{p}_{e_i} = \frac{1}{|\mathcal{S}|} \sum_{b \in \mathcal{S}} \mathbb{1}_{\{e_i \in E_b\}}$, where $\mathbb{1}_{\{e_i \in E_b\}}$ returns 1 if $e_i \in E_b$, else returns 0. The purpose is to prune out the edges that are below a certain confidence threshold t. There are existing techniques to estimate the confidence threshold such as the L_1 estimator [52] which Etio uses to fix the confidence threshold. Using the most

[4]To simplify notation, we denote the vertex and its corresponding variable the same.

significant arcs, Etio constructs a graph that contains all of the significant arcs (the averaged graph G). Our approach does not guarantee that the obtained graph represents the true causal graph–inferring one in our setup is infeasible. Thus, we take the practical approach and aim to infer a graph with a good predictive power.

3.2 Answering Queries

Predictive queries ask what is the output of the model given that certain input factors have certain values. Given a set of previously unseen set of assignments for the variables $\{X_i = a_i\}, X_i \in V$, the outcome corresponding to the MI attack node $Y \in V$ is computed by the expression $E[Y|pa_Y]$. This expression is recursively expanded until it is conditioned on the X_i values and can be evaluated with the given concrete values. To learn the coefficients associated with each node, we use a standard maximum likelihood estimation approach to fit each node's observed data conditioned on its parents.

In principle, we can answer interventional queries, which measure how much changes in an factor's value affects the outcome, by conducting experiments where we manipulate the training process such that input variables take desired values. Such manipulations are called interventions. Formally, given a set of variables $V = \{X_1, \ldots, X_n\}$, an intervention on a set $W \subset V$ of the variables is an experiment where the experimenter controls each variable $w \in W$ to take a value of another independent (from other variables) variable u, i.e., $w = u$. This operation, and how it affects the joint distribution, has been formalized as the *do* operator by Pearl [42]. For example, in Fig. 5a, we can intervene on the model complexity independently of the other variables. However, in some cases modifying variables directly is not feasible in practice (e.g., the train-to-test accuracy gap) as it requires knowledge of the data distribution that the model is trying to learn in the first place. So, we cannot really conduct such interventional experiments.

The key insight to answer intervention queries is that we can reason about such queries with only the causal graph and the data–Etio applies the principles of *do*-calculus to analytically compute the causal relationship expressed by the *do*-query. The *do*-calculus rules have been proven to be sound and complete [19, 58]. They are complete in that if repeated application of the rules of *do*-calculus cannot obtain a conditional probability, then the algorithm outputs that the causal relationship cannot be identified without additional assumptions. If we do obtain an ordinary conditional probability, then we say that the causal estimate can be identified, i.e., the graph has enough assumptions or no ambiguity. Then, the obtained expression (called the *estimand*) represents the correct translation of the causal query to a conditional probability (soundness). Such guarantees are powerful tools: Given the formal query and the causal model, this approach avoids paradoxes that might arise from over-controlling or not controlling (Section 2). We will explain a small fragment of this calculus through an example.

Example: Backdoor Paths. Let us consider the examples in Fig. 6. In Fig. 6a, the query (Q1) is to estimate the effect of train-to-test accuracy gap on the MI attack accuracy, given only observations of the variables of interest. In Fig. 6b, we introduce a new variable, the separation score between members and non-members (H8), which is caused by the train-to-test accuracy gap. The query (Q2) asks to estimate the effect of the separation distance on the

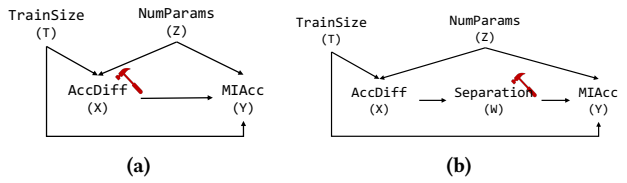

Figure 6: Importance of selecting the right control variables to avoid selection bias. T and Z are both in adjustment set for X in 6a. In Fig. 6b, to estimate the effect of separation score (W), we can control for T, Z or X but controlling for the train-to-test accuracy gap (X) introduces selection bias.

attack accuracy. The causal model in Fig. 6a was previously discussed in Section 2—the expected attack accuracy is computed over both of the confounding factors, the model complexity, and the training set size. From the graph structure, observe that the confounding happens because of the two undirected paths from the node corresponding to the train-to-test accuracy gap to the MI attack node (π_1 : `AccDiff` ← `NumParams` → `MIAcc` and π_2 : `AccDiff` ← `TrainSize` → `MIAcc`). Such paths are called *backdoor paths*. A backdoor path is a non-causal path from X to Y. This is a path that would exist in the graph even if we were to remove the outgoing edges from the node of interest. When there are backdoor paths, there are sources of association (which we can observe statistically) in addition to the causal ones. In Fig. 6b, it seems the query (Q2) requires a similar control as in Fig. 6a. However, which of the nodes on the paths π_1 and π_2 paths, should we control for? The node corresponding to the train-to-test accuracy gap is one such candidate. Notice though that unlike other nodes on the backdoor paths, it has two incoming edges, meaning that controlling for it biases the observed relationship between its parents. Such nodes are called "colliders" and can introduce bias (see Section 2). One has to carefully determine exactly when to control for colliders.

There is a set of principled rules to "block" backdoor paths. We summarize these rules informally here but interested readers can refer to [42, 43] for more thorough background. A path is blocked if 1) we control for a non-collider on that path or 2) we do not control for a collider on that path. For any given path, only one of these conditions is required to block the path. So, if there exists a path between X and Y that contains an uncontrolled collider, that path is blocked without controlling on any other variables. Guided by these rules (called the *backdoor criteria*), in (Q2) we should control for X when estimating the effect of W on the MI attack.

Estimating the Causal Effect. Recall that we are interested in the average treatment effect, which is the average difference in the outcomes given that the treatment takes two values: the *treatment* value and the *control* value. A straight-forward way to compute the average treatment effect (ATE) is by using the difference in the mean of the outcome *conditioned* on the treatment variable ($E[Y|X = a] - E[Y|X = b]$). However, this method of computation suffers from statistical pitfalls, such as sampling bias and confounding bias which we highlighted in Section 2. Instead, what we want to quantify is the average treatment effect as a *do*-query as defined below.

Definition 3.1 (Average Treatment Effect). The average treatment effect of a variable of interest X (called the treatment) on the target variable Y (called the outcome) is:

$$\text{ATE}(X, Y, a, b) = E[Y|do(X = a)] - E[Y|do(X = b)],$$

where a, b are constants for which X is defined. We omit the constants when the query is over the domain of X.

Etio will translate the *do*-query $E[Y|do(X = a)]$ into an ordinary (conditional) expectation expression from given the causal model (e.g., using the backdoor criteria). It then learns an estimator that allows computing the ordinary expectation using the available data. We choose a linear regression model to estimate the quantities of interest. Its estimates are interpretable: a positive $\text{ATE}(X, Y)$ value means that an increase of the feature X causes an increase in the MI attack accuracy Y, and vice-versa for negative $\text{ATE}(X, Y)$.

In summary, we have described a methodology encapsulated in Etio to analyze causally potential factors. While our methodology uses techniques standard in causality, we have carefully laid out the technical choices that allow us to achieve our goals: 1) used linear equations to capture causal effects; 2) combined Greedy structured algorithm with bootstrapping to scale the creation of models; and 3) defined the average treatment effect as our measured outcome.

4 CONNECTING MI AND GENERALIZATION

Our main technical novelty is how use Etio to study the connection between MI attacks and classical generalization in ML. We now show how to create causal models for 6 different attacks and *formalize* hypotheses (H1)-(H9) made in prior works.

4.1 Variables of Interest

The generalization notions and other potential causes identified in H1-H9 (Section 2) are properties of the training algorithm. The training algorithm $\mathcal{A}$ takes as input a training dataset D consisting of N samples $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, $D \sim P^N$, each independently identically drawn from P where P is a distribution over $\mathcal{X} \times \mathcal{Y}$, $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. The training algorithm also takes as input a set of training hyperparameters π and the loss function l. The training algorithm produces a model $f : \mathcal{X} \to \mathcal{Y}$. All of the prior works have studied MI attacks on neural networks trained with stochastic gradient descent (SGD), so we focus on SGD primarily in this work.

The generalization error is a measure of how well a learned model f can correctly predict previously unseen data samples. For a given a model f and a sample $z = \{x, y\} \sim P$, the generalization error is defined as $\mathbb{E}_{z\sim P}[l(f, z)]$. The learned model f depends on the drawn training dataset D. As a result, the generalization error of $\mathcal{A}$ is $\texttt{TestLoss} = \mathbb{E}_{z\sim P, D}[l(f_D, z)]$. We denote the generalization error as `TestLoss` since if we were to sample $z \sim P$ it would be highly unlikely for that sample to belong to the training set D.

Bias-variance decomposition. A fundamental principle to understand generalization in machine learning models is the bias-variance decomposition [14, 25, 68], which establishes that the **generalization error** directly factors into **Bias** and **Variance** as shown in Table 1. The bias represents how well the hypothesis class $\mathcal{H}$ to which the model f belongs to fits the true data $\mathcal{Y}$, while the variance represents how much the model varies across different samples of data. For example, with sufficient training time, a model that is overly-parametrized can have a low bias (since it fits the data very well) and high variance (because it can fit all the "accidental regularities" or idiosyncrasies of the sampled data). Our causal models use bias and variance as variables and therefore these serve as a new lens to explain how they affect MI attacks in addition to generalization in the same representation. To compute the bias and variance, we follow the methodology outlined in prior work [68]. We first compute an (unbiased) estimator for the variance term in the bias-variance decomposition. Next, from the generalization error (or loss), we subtract the variance to derive the bias. The full mathematical formulation of bias and variance is in Appendix A.

MI Attack Accuracy. (H1)-(H9) are claims relating the effect of potential causes on the MI attack susceptibility. To measure susceptibility, we consider the random variable corresponding to the MI attack accuracy, for each prior work attack. In total, we study three MI attacks: the multiple shadow model (`ShadowAcc`) [57], the single shadow model attack (`MLLeakTop3Acc`) [49], and threshold-based attack (`ThreshAcc`) [69]. We additionally perform similar attacks to [49] where we use one shadow model (`MLLeakAcc`). We take the whole prediction vector for CIFAR10, and the top-10 predictions for CIFAR100 as input features to train the attack model. Other considered variants of this attack include the correct label in the input features of the attack model. We denote these as `MLLeakTop3Acc-l` and `MLLeakAcc-l`. For each learned network f_D, we evaluate the MI attack on members $\in D$ and a dataset of previously unseen samples, non-members $\notin D$. The final result is the expected accuracy on members and non-members averaged over multiple samples f_D.

Other Model Properties. Some of the hypotheses in prior work involve training hyperparameters and model properties such as training set size (H4) and model complexity (H3). For H3, we use the number of parameters in the model to measure model complexity. Specifically, we count the number of parameters (e.g., weights and biases) that are updated during the training phase.

The potential factors that appear in H1-H9 are summarized in Table 1. Our aim is to infer a causal model over these variables. Next, we require traces, so we run the training algorithm to collect observations of these variables. We leave the detailed process to generate traces of these variables for Section 5.1. Besides traces, we formulate domain knowledge constraints as input to Etio.

4.2 Domain Knowledge as Constraints

As our domain-specific constraints, we leverage simple insights that force the hill-climbing algorithm to infer models that have causal meaning. For instance, one constraint encodes that the root nodes of the model should correspond to variables that are part of the training algorithm's hyper-parameters such as `TrainSize` and `NumParams`. This constraint belongs to the Forbid list. In addition to these, we have identified the following constraints:

- *There are no outgoing edges from the attack node.* Without this constraint, the structure learning algorithm could learn that the attack causes one of the features–the direction of the edge cannot be inferred by observations only.
- *There is no edge from a node that is neither a root node nor* `TrainVar` *nor* `TrainLoss` *to* `TrainBias`. We add this constraint because the `TrainBias` is computed from the `TrainLoss`

Table 1: Summary of variables we consider when building our causal graphs to answer queries Q1-Q9. We build the causal graphs for each MI attack A [49, 57, 69]. For a given sample z, the MI attack outputs whether it is a member (m) or not ($\neg m$). We illustrate the variance term only for MSE where $\bar{f}(x) = \mathbb{E}_D[f(x, D)]$.

Variables	Formula
TrainAcc	$\mathbb{E}_{z\sim D,D}[f_D(x) = y]$
TestAcc	$\mathbb{E}_{z\sim P,D}[f_D(x) = y]$
AccDiff	TrainAcc- TestAcc
TrainLoss	$\mathbb{E}_{z\sim D,D}[l(f_D, z)]$
TrainVar	$\mathbb{E}_{x\sim D,D}[\|\|f_D(x) - \bar{f}(x)\|\|_2^2]$
TrainBias	TrainLoss-TrainVar
TestLoss	$\mathbb{E}_{z\sim P,D}[l(f_D, z)]$
TestVar	$\mathbb{E}_{x\sim P,D}[\|\|f_D(x) - \bar{f}(x)\|\|_2^2]$
TestBias	TestLoss- TestVar
LossDiff	TestLoss- TrainLoss
NumParams	$\|f_D\|$
TrainSize	$\in \{1k, 5k\}$
ShadowAcc	$\mathbb{E}_{z,D}[A(z) = m \| z \sim D \wedge A(z) = \neg m \| z \sim P]$
MLLeakAcc, MLLeakTop3Acc	$\mathbb{E}_{z,D}[A(z) = m \| z \sim D \wedge A(z) = \neg m \| z \sim P]$
ThreshAcc	$\mathbb{E}_{z,D}[A(z) = m \| z \sim D \wedge A(z) = \neg m \| z \sim P]$
CentroidDist	$\mathbb{E}_D[\|\|C(D) - C(P)\|\|_2]$

and TrainVar. Any influence on TrainBias should be mediated by its two parents.

- *There is no edge from a node that is not a root node to* TrainVar. The variance on training samples is computed directly on the prediction vectors.
- *There is no edge from a node that is neither a root node,* TestVar *nor* TestLoss *to* TestBias. We add this constraint because the TestBias is computed from the TestLoss and TestVar. Any influence on TestBias should be mediated by its two parents.
- *There is no edge from a node that is not a root node to* TestVar. The variance on testing samples is computed directly on the prediction vectors.
- *Constraints in* Enforce. There is an edge from TrainAcc to AccDiff and TestAcc to AccDiff. There is an edge from TrainLoss to LossDiff and TestLoss to LossDiff.

In total, we have 19 Forbid and Enforce constraints (see Appendix B). Such constraints are easy to derive, as they either stem from the definitions in Table 1 or from the data-generating process.

4.3 From Hypotheses to Queries

We can obtain a causal model from the traces and the domain knowledge constraints using Etio. The last step is to formulate queries on the causal models. Hypotheses H1-H9 can all be formally described as interventional queries Q1-Q9 respectively, as follows.

(H1,H6,H9) → (Q1,Q6,Q9): Generalization Metrics. Existing work uses two different metrics to quantify generalization precisely: the train-to-test accuracy gap (AccDiff in Table 1) and the average generalization error (LossDiff in Table 1). We use the metric that was cited by the respective original works to determine the query for each of the studied attacks.

(H2) → (Q2): Formalizing "Closeness". We first quantify "closeness" more formally between a shadow model and a target model when these have the same architecture, as is the case for the MI attack by Shokri et al. [57]. Our observation is that the variance term in the generalization error is a metric of "closeness". Intuitively, since the shadow model is a realization of a different subset D_i of D, the variance is a measure of the expectation of whether training on different sampled training sets (of the same size) from the data distribution outputs neural networks with very "different" prediction vectors. Since this particular MI attack trains shadow models with the same architecture as the target model, the larger the variance, the more likely we are to obtain a shadow model that is on average more "distant" from the target model. Hence, H2 can be reformulated to check if the variance of the learning algorithm is a cause of the attack. We consider the variance of $\mathcal{A}$ on the members (training samples, denoted as TrainVar) and the variance of $\mathcal{A}$ on non-members (testing samples, denoted as TestVar), separately. The causal query asks if TrainVar, and, respectively, TestVar, affect the MI attack accuracy ShadowAcc, i.e., if a change in TrainVar (or TestVar) causes the ShadowAcc to change. Formally, the hypothesis translates to two *do*-queries (Q2): ATE(ShadowAcc, TrainVar) and ATE(ShadowAcc, TestVar).

(H5) → (Q5): Single Shadow Model. We want to check if the closeness (as measured by TrainVar and TestVar) is a cause for the single shadow model attack. If it does not contribute to the attack, the hypothesis is correct.

(H7) → (Q7): Different Causes. Prior work showed that in terms of MI attack performance there is no significant difference between the performance of attack models trained on the top three predictions of the prediction output vs. the whole prediction vector [49] (page 5, Fig.4). We denote the attack using the whole prediction vector as MLLeakAcc. Here, we are interested if the attack model's performance changes with the target model for the variants of the attack. Thus, we formalize it as checking whether TestVar and TrainVar are causes for MLLeakAcc (-l) and MLLeakTop3Acc (-l).

(H8) → (Q8): Formalizing Decision Boundaries. A few works give credit to the clear decision boundary between members and non-members for the success of shadow model-based attacks [49]. To quantify the "distinguishability" between members and non-members, we first compute the centroid of members ($C(D)$) and non-members ($C(P)$) as the following: $C(D) = \mathbb{E}_{z\sim D}[f_D(z)]$ and $C(P) = \mathbb{E}_{z\sim P}[f_D(z)]$. Then, we use the Euclidean distance between the above two centroids to measure the distinguishability for the given training set. For each training setup ($\mathcal{A}, \pi$ and architecture), we compute the averaged centroid distance over multiple different training sets. Note that the user can specify any such existing statistic as a distinguishability measure between the training and testing set as input to Etio.

Implementation. Our implementation consists of two parts: generating the traces, i.e., training models and running attacks, and implementing Etio. For the traces, we use the standard machine learning library PyTorch 1.7.1+cu110 [41] to train the models and run the attacks. For Etio, we use two libraries for analyzing the MI attacks. First, we use the R library called bnlearn 4.7 [51] to infer the causal models. This library offers several off-the-shelf algorithms for structured learning and Bayesian inference. Second,

we use dowhy 0.7.0 [54, 55] to implement the average treatment effect queries.

5 EVALUATION

We evaluate Etio on two grounds:

- (EQ1) *Goodness of fit*: Are the causal models predicting the MI attack more accurately on unseen samples than correlational analyses?
- (EQ2) *Utility*: Is Etio useful in refuting or confirming prior hypotheses? Does it provide useful insights to how MI attacks connect to generalization and how defenses work?

We study 6 MI attacks: multiple shadow model [57], 4 variations of the single shadow model [49] and threshold-based [69]. We present the results with respect to the 9 queries for two loss functions, so in total we have 18 prior work hypotheses. In addition, we study 2 practical defenses proposed in prior work. The first is L2-regularization (also known as weight decay) that was proposed as a mitigation strategy for the multiple shadow model attack [57]. It has been used as a baseline for other defenses [36]. The second defense we consider is MemGuard [22], a defense that changes the prediction vectors without changing the accuracy of the model. We choose this defense as it is effective against the attacks we also considered in this work [49, 57]. The causal models for all of the evaluated attacks and defenses are available in Apppendix D.

We present details later and summarize our key findings below:

- There is no one-size-fits-all explanation for the 6 MI attacks—the factors contribute differently in different attacks.
- Our analysis refutes 7/18 prior hypothesis we formalized, and confirms 9/18 as valid.
- Our causal models have predictive accuracy of 0.90 for unobserved experiments, which are not used for causal model creation. This is comparable or better by $3-22\%$ than simple correlational analysis between the single cause and the MI attack, in all cases we study.
- Bias and Variance observed during training can quantitatively predict both generalization measures (e.g., `TestLoss`, `TestBias`, and `TestAcc`) and MI attack performance, providing new insights. These factors play a disproportionately larger role in explaining MI attack performance, compared to other factors such as model complexity, dataset size, or even generalization measures themselves.
- Defenses reduce certain causes of the MI attack, but not all and not completely. They reduce the effect of variance, but fail to eliminate factors such as the train-to-test accuracy gap or the distance between members and non-members.

5.1 Experimental Setup

Datasets. We select 3 common image datasets: MNIST, CIFAR10, and CIFAR100. MNIST has 60k training and 10k testing samples of 28×28 grayscale images of handwritten digits. CIFAR10 and CIFAR100 have 50k training and 10k testing samples of 32×32 color images uniformly distributed in 10 and 100 classes, respectively.

Models. For each dataset, we train multiple models with different architectures and hyperparameters. For MNIST, we use multilayer perceptron (MLP) with one hidden layer to build the target model. We change the number of units used in the hidden layer ($\{16, 32, 64, 128, 256\}$) to change the *width* or the model complexity. For CIFAR10 and CIFAR100, we use various convolutional neural network (CNN) architectures: AlexNet, DenseNet161, and ResNet34. For changing the width, we vary the number of filters of these models. The widths considered for AlexNet were $\{16, 32, 64, 128, 256\}$, and for DenseNet161 and ResNet34 they were $\{2, 4, 8, 16, 32\}$.

Training Algorithms. We trained all models using stochastic gradient descent (SGD) with momentum 0.9, two weight-decay rates $\{5\times10^{-4}, 5\times10^{-3}\}$ and two kinds of loss functions: mean squared error (MSE) and cross entropy (CE). The higher weight decay models are used to evaluate L2-regularization. We summarize the training configurations we considered in Appendix, Table 4. For the models trained with the scheduler, we used the step learning rate scheduler with the learning rate decay factor of 10 and for 200 epochs.

Variables of Interest. To estimate the variables of interest for each training setup, we follow the procedure proposed by Yang et al. [68]. Specifically, we randomly generate disjoint splits of training samples $D = D_1 \cup \ldots \cup D_n$, where size of each $|D_i| = s$. We train n models for each architecture width f_{D_i} over different training sets D_i. For CIFAR10 and CIFAR100, we use $n \in \{10, 50\}$ and, respectively, $s \in \{5000, 1000\}$, while we use $n \in \{12, 60\}$ for MNIST because MNIST has a larger training set.

Attacks. The shadow model training size is equal to the training size of the target model, i.e., either 1000 or 5000. This set forms the member set for the attack model. Additionally, an equal-sized set is used to form the non-member training set for the attack model. The evaluation set for the attack model consists of the 1000 or 5000 training samples of the target model and an equal-sized set not previously seen by either target and shadow models. For each architecture and width, we perform the attack 30 times for different samples of the datasets from the original training set D. The same splits are used for all datasets and single shadow model attacks.

5.2 Predictive Power of Causal Models

To evaluate the predictive accuracy of the graphs on unseen observations, we use two metrics regularly used in evaluating Bayesian nets: 1) *mean predictive correlation* and 2) *mean squared error* (MSE). We compute these two metrics using standard cross-validation over multiple runs (20). For each run, we use a 80/20 split of the observations for the train-test sets. For each run of the cross-validation, the predictive correlation measures the (linear) correlation between the observed and the predicted values for the MI attack node.

Baseline. We use a simple baseline that can predict the accuracy of the attack: we compute the Pearson correlation between the observed values of the MI attack and the observed values of the other variables of interest we identified (Section 4.1). A high Pearson correlation (close to 1) means that there is a linear relationship perfectly describing the MI attack and the variable. If so, to predict the MI attack accuracy, measuring this one variable and learning the coefficients of the relationship from data is enough.

In total, we evaluate 24 setups for 6 attacks for models trained with two loss functions, with and without L2-regularization. For all of the attacks on both undefended and defended models, the predictive correlation is above 0.90 (Table 2). Compared to the correlation baseline, the graphs Etio produces are consistently equal or better for all 24 setups. For 17/24 of the setups, the predictive

Table 2: The graphs that ETIO generates have consistently equal or better predictive power than simple correlations. Best Corr. stands for the Pearson correlation coefficient, and ETIO Pred. stands for predictive correlation.

Loss	Attack	Normal / High Weight Decay			
		Best Corr. Variable	Best Corr. Value	ETIO Pred.	ETIO MSE
ce	MLLeakAcc	AccDiff	0.8543	0.9358	2.31E-03
			0.8621	0.9083	2.54E-03
mse	MLLeakAcc	AccDiff	0.8493	0.9170	1.80E-03
			0.8675	0.9033	9.55E-05
ce	MLLeakAcc-l	AccDiff	0.8209	0.9685	1.17E-03
			0.8414	0.9486	2.05E-03
mse	MLLeakAcc-l	AccDiff	0.9397	0.9740	7.30E-04
			0.9371	0.9694	6.72E-05
ce	MLLeakTop3Acc	CentroidDist	0.9257	0.9663	1.16E-03
		AccDiff	0.8595	0.9645	1.16E-03
mse	MLLeakTop3Acc	AccDiff	0.8743	0.9466	1.26E-03
		AccDiff	0.9041	0.9216	1.14E-04
ce	MLLeakTop3Acc-l	CentroidDist	0.9166	0.9641	2.21E-03
		AccDiff	0.8409	0.9579	1.58E-03
mse	MLLeakTop3Acc-l	AccDiff	0.8447	0.9244	1.73E-03
		AccDiff	0.9000	0.9361	9.51E-05
ce	ShadowAcc	AccDiff	0.9752	0.9817	6.45E-04
			0.9694	0.9762	9.18E-04
mse	ShadowAcc	AccDiff	0.9526	0.9733	5.36E-04
			0.9626	0.9689	1.61E-04
ce	ThreshAcc	LossDiff	0.7517	0.9739	9.00E-04
			0.7730	0.9425	1.77E-03
mse	ThreshAcc	LossDiff	0.9823	0.9906	2.43E-04
			0.9775	0.9880	2.41E-05

correlation improves $3-22\%$. For the remaining 7/24, the causal models are on par with baselines or slightly better, within 3%. The mean MSE for predictions is low (0.001) for all of our evaluated attacks and models. We find that in the case of ShadowAcc, ETIO does not significantly improve the accuracy, as the correlation values are already higher than 0.95. This confirms what prior works suggest [27, 60, 69]: using an attack based on the prediction correctness yields, on average, similar performance to the ShadowAcc. We observe that the AccDiff is almost in a perfectly linear relationship with the accuracy of the multiple shadow model attack. Similarly, the metric we formalize, the centroid distance between clusters of members and non-members (CentroidDist) is almost perfectly linear with the single shadow model attack.

5.3 Testing of Prior Hypotheses

We confirm using our analysis that 9/18 of the prior work hypotheses are true (✓ in Table 3). We also find 7/18 prior hypotheses do *not* identify a cause for the studied MI attack (× in Table 3). As some of the hypotheses involve more than one potential cause or they are comparing causes between attacks, we have a total of 32 ATE values, 16 for each loss function.

We refute prior hypotheses in broadly two instances. First, when there is no universal explanation: prior hypotheses often overlook the differences between NNs trained with different loss functions and the specifics of the attack. Second, because prior hypotheses do not consider the connections between the parameters of the

Table 3: We translate the prior work hypothesis H1-H9 to ATE(Feature, Attack) queries. For (Q2,Q5,Q7) which are made of several ATE queries, we ✓ only if all ATE queries support the prior hypothesis. $*$ means the p-value > 0.05 and we mark such queries with ∘.

Attack	Feature	CE		MSE	
		ATE	Query Result	ATE	Query Result
ShadowAcc	AccDiff	0.30	Q1: ✓	0	Q1: ×
ShadowAcc	TrainVar	0.02	Q2: ✓	0.03	Q2: ∘
ShadowAcc	TestVar	0.94		0.20 (∗)	
ShadowAcc	NumParams	0.15	Q3: ✓	-0.005	Q3: ✓
ShadowAcc	TrainSize	-0.11	Q4: ✓	-0.09	Q4: ✓
MLLeakAcc	TrainVar	-0.34	Q7: ×	-0.05 (∗)	Q7: ×
MLLeakAcc	TestVar	0.81		0	
MLLeakAcc-l	TrainVar	-0.24		0.06	
MLLeakAcc-l	TestVar	0.84		0	
MLLeakTop3Acc-l	TrainVar	-0.40		-0.06 (∗)	
MLLeakTop3Acc-l	TestVar	0.75		0	
MLLeakTop3Acc	AccDiff	0.18	Q6: ✓	0	Q6: ×
MLLeakTop3Acc	TrainVar	-0.34	Q5: ×	-0.15 (∗)	Q5: ×
MLLeakTop3Acc	TestVar	0.78		0.24	
MLLeakTop3Acc	CentroidDist	0.27	Q8: ✓	0	Q8: ×
ThreshAcc	LossDiff	1.47 (∗)	Q9: ∘	-0.67	Q9: ✓

training process and variables that naturally appear as part of SGD such as loss difference and variance, etc. In summary:

- A single causal factor does not explain all attacks. In fact, causes vary per attack and differ by the loss function used.
- (Q1, Q6) The train-to-test accuracy gap does cause the MI attack accuracy, though for MSE-trained models, the loss difference is a more suitable metric.
- (Q2) The "closeness" of the shadow model influences the MI attack accuracy, more so for CE-trained models than for MSE-trained models.
- (Q7) There are differences between the variants of the single shadow model attack, and the single shadow model with top-3 is more robust to changes in the shadow model.
- (Q3, Q4) Training size is a factor that affects the MI attack accuracy for all of the evaluated attacks. Model complexity is a cause for all evaluated attacks but to a very small degree in some cases.
- (Q5) We find that the variance of the outputs of the models is also a cause for the single shadow model attack, to various degrees depending on the type of attack. Prior work overlooks the differences in the prediction vectors between the target and shadow model.
- (Q8) Our formalized distance between the clusters of members and non-members is one of the largest causes for the single shadow model with top-3.
- (Q9) The threshold-based attack is influenced with varying degrees by other factors that are related to the loss.

CE vs. MSE. We find that the train-to-test accuracy gap has the largest influence for CE-trained models, whereas for MSE-trained models it is the loss differences between members and non-members. Similarly, factors such as the variance and the centroid distance

that affect MI attack accuracy on CE-trained models are not factors on MSE-trained models.

Detailed Analysis. The differences in the prediction vectors of the non-members (as measured by `TestVar`) has a large causal effect on the average MI attack accuracy. This validates the prior work hypothesis (Q2) that the differences in the shadow and target model affect the MI attack. For instance, for CE, the estimated ATE of `TrainVar` on `ShadowAcc` is 0.02, whereas that of `TestVar` is 0.94 (Table 3). We find that for the multiple shadow model attack on MSE-trained models, the number of parameters (Q3) does not show a significant influence on the `ShadowAcc`, but for all other attacks and loss functions, the more parameters, the higher the attack accuracy. We thus validate (Q3) as for all evaluated attacks there is a non-zero ATE value. We find that the variance of the outputs of the models is also a cause for the single shadow model attack, to various degrees depending on the type of attack. Prior work overlooks the differences in the prediction vectors between the target and shadow model. Thus, our analysis shows that (Q5) is refuted. The variance of the training algorithm influences MI attacks that take the whole prediction vector more–models tend to agree more on top-3 predictions rather than the whole prediction vector. Thus, the evaluated shadow model MI attacks that take the top-3 predictions are not as sensitive to differences between the shadow and target models' architecture and dataset. This shows that there are differences between the variants of the single shadow model attack, refuting (Q7). More details for each attack are available in Appendix C.

We find that a small fraction of queries have low statistical significance (p-value > 0.05)—4/16 MSE-trained models and 1/16 for CE-trained models. We do not draw any conclusions for these.

5.4 MI attacks and Generalization

We find that **Bias** and **Variance** values have a high level of influence on both generalization measures as well as the MI attack accuracy. As expected from the bias-variance decomposition theorem, Bias and Variance values are strongly predictive of `TestAcc`, `AccDiff`, and `CentroidDist` values—all of these are generalization measures. Bias and Variance also have a disproportionately high influence on MI attack accuracy. Appendix C gives details; here we summarize their effect on MI attack accuracy which varies by attack.

Variance & MI. The higher the variance on non-members, the higher the MI attack accuracy. The reverse is true for members: the higher the variance on members, the lower the MI attack accuracy. There is less variance on the training samples (i.e., the model learns similar prediction vectors across multiple training datasets) and there is higher variance on the test samples. Our analysis show that the larger the gap between these two, the better the MI attack accuracy. This suggests that defenses which decrease the gap between the train and test variance (like MemGuard [22]) will be effective.

Bias & MI. The Bias on non-members is almost always a factor in all types of MI attacks we study. It affects MI attack accuracy through the test accuracy, which in turn affects the train-to-test accuracy gap or the distance between members and non-members. Recall that Bias is "how far" the test set predictions are from the ground truth on average. High Bias on non-members explains why MI attacks, even when not explicitly using the label information, will have a better accuracy. On members, however, the ATE value of the Bias in most cases is close to 0, i.e., it has almost no effect on MI attack accuracy—this corresponds to networks closely fitting the training set. Compared to Variance, the Bias has a larger effect, and even more so when the input features to the attack model are the top-3 predicted labels and not the whole prediction vector.

Why do larger models leak information even if their test loss decreases? Complex interplay of loss, variance, bias and the model size have been observed previously under different regimes (Fig.1 in [68]). When the Bias dominates, it and the testing loss decrease with increase in model size of the network, but the Variance does not linearly go down and exhibits a peak (bell-shaped curve). Our analysis shows that Variance *by itself* contributes to the MI attack—despite training larger models with lower loss and Bias, the Variance can improve MI attack accuracy (see Variance ATE, Appendix C.

How does CE loss differ from MSE? Unlike MSE, for CE we observe that the Variance dominates the loss term. The loss and the Variance exhibit a high peak as the model size increases, while the Bias keeps decreasing (unimodal curve). This explains why for CE models the Variance has a much larger impact on MI attack accuracy. After Variance peaks, as the model size increases, the loss drops to where both Bias and Variance are low. This explains why increasing model size can reduce MI attack accuracy beyond a point.

5.5 Utility in Explaining Defenses

We analyze the causal models for all 6 attacks for L2-regularization. For MemGuard, we evaluate the single shadow model (top-3) attack on the defended models as done in the original work. Thus we have $2 + 12 = 14$ causal graphs in total. For each case, we analyze how much the ATE for a cause differs from ATE in the corresponding causal graphs for *undefended* ML model.

We find that L2-regularization alleviates some causes but not all, and not in equal measure for all attacks. For instance, it majorly reduces the ATE of the test variance on all of the evaluated MI attacks–even as drastically as from 0.84 to 0.16 for `MLLeakAcc-l`. The effect of the train-to-test accuracy gap on the `ShadowAcc` remains the same, but it increases for the single shadow model attacks. The ATE of the `CentroidDist` does not change after regularization is applied, showing that there are still exploitable signals left.

The MemGuard defense reduces the Variance significantly but many factors remain unaddressed even after the defense. For instance, the distance between members remains a factor. The ATE of the `AccDiff` is reduced from 0.19 to 0.08. MemGuard is more effective overall in removing causes than regularization.

More details of our analysis of defenses is in Appendix C.2.

6 RELATED WORK

Generalization. Generalization in machine learning is a fundamental topic. Several studies investigate the bias-variance decomposition in neural networks [14, 39, 68]. Yang et al. [68] explore the dependence of bias and variance to network width and depth, e.g., deeper models tend to have lower bias but higher variance. Our work connects MI attacks to such training and architectural choices. Other works propose new measures of generalization [11, 23].

Membership Inference Attacks. There has been a recent line of work proposing MI attacks and providing useful attack taxonomy.

Shokri et al. [57] present the first membership inference attack. They show that overfitting is correlated with their attack performance. They suggest that besides overfitting, the structure and type of the model also contribute to the privacy leakage through membership inference attacks. Several new attacks have emerged [9, 17, 27, 29–31, 34, 35, 71] and attack taxonomies [28, 65] have started to categorize them. These attacks serve as tools to evaluate the privacy risk of machine learning models through attack procedures. Our work distinguishes itself from all of these by providing a causal framework to explain why these attacks arise. Notably, our work provides a new lens into how generalization and MI attacks connect—through a systematic measurement and reasoning of bias, variance, and other stochastic variables that arise in training.

Several works have provided mechanistic explanations connecting MI attacks to generalization prior to our work. Yeom et al. [69] provide a theoretical connection between a notion of generalization called the average generalization error and a bounded-loss adversary which does not apply to training using a CE loss. They also propose a threshold-based attack which has knowledge of the loss distribution which we have also evaluated in our framework Etio. The attack assumes that the loss is normally distributed, and thus can be connected to the adversary advantage in a closed-form expression. Our work shows that the assumptions made in their work may not always hold. We show that MI attack performance is linked to the average generalization error for models with MSE loss but does not always for CE loss. Subsequent work by Song et al. [62] propose a similar threshold-based attack, but on the confidences of the prediction. Nasr et al. [37] also analyze the connection between membership inference attacks and overfitting, while proposing white-box membership inference attacks. They also empirically observe the correlation of the attack performance to the model capacity. Song et al. [61] evaluate membership inference attacks against adversarially robust models and point out that these models have a larger train-to-test accuracy gap when considering adversarial examples. Our work shows that other factors beyond the train-to-test accuracy gap contribute to the privacy leakage.

Causality. Causality is an active area of research with recent advances improving learning of causal models [24, 53], as well as better inference procedures [21, 26]. While extensively applied in sciences [4, 15, 18, 33, 64], causality has only been recently connected to privacy [63, 66]. In our work, we introduce the causal lens to understand MI and generalization. Since our proposed methodology is synergistic, combining learning with domain knowledge, we can benefit from such advances to improve our causal models and analysis. In addition to learning and inference, methods to test the causal assumptions have also been proposed such as sensitivity analysis [45, 47] and simulated dataset-approach [38]. Again, our approach can leverage such tests for the constructed causal models.

7 CONCLUSION

We have proposed the first use of causal graphs to capture how stochastic factors—such as bias, variance, model size, data set size, loss values, and so on—causally interact to give rise to MI attacks, providing a new connection between these attacks and generalization. We hope this framework helps formally re-analyze statistical conclusions and pinpoint root causes more accurately.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable feedback. We extend a special thanks to Amit Sharma for helping us with the queries about the DoWhy library and using it in our implementation. We are grateful to Arnab Bhattacharyya and Kuldeep S. Meel for their useful feedback on an earlier draft of this paper. This research was supported by the Crystal Centre at National University of Singapore and its sponsors, the National Research Foundation Singapore under its NRF Fellowship Programme [NRF-NRFFAI1-2019-0004] and the Ministry of Education Singapore Tier 2 grant MOE-T2EP20121-0011. Teodora Baluta is also supported by the Google PhD Fellowship.

REFERENCES

[1] Teodora Baluta, Shiqi Shen, S Hitarth, Shruti Tople, and Prateek Saxena. 2022. Membership Inference Attacks and Generalization: A Causal Perspective. *arXiv preprint arXiv:2209.08615* (2022).
[2] Elias Bareinboim and Judea Pearl. 2012. Controlling selection bias in causal inference. In *Artificial Intelligence and Statistics*. PMLR, 100–108.
[3] Joseph Berkson. 1946. Limitations of the application of fourfold table analysis to hospital data. *Biometrics Bulletin* 2, 3 (1946), 47–53.
[4] Henry E Brady. 2008. *Causation and explanation in social science*. na.
[5] Zhihong Cai and Manabu Kuroki. 2008. On identifying total effects in the presence of latent variables and selection bias. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*. 62–69.
[6] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*. 267–284.
[7] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2020. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805* (2020).
[8] David Chickering, Dan Geiger, and David Heckerman. 1995. Learning Bayesian networks: Search methods and experimental results. In *proceedings of fifth conference on artificial intelligence and statistics*. 112–128.
[9] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International Conference on Machine Learning (ICML)*. PMLR, 1964–1974.
[10] Rónán Daly and Qiang Shen. 2007. Methods to accelerate the learning of bayesian network structures. In *Proceedings of the 2007 UK Workshop on Computational Intelligence*. Citeseer.
[11] Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy. 2020. In search of robust measures of generalization. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020).
[12] Nir Friedman, Moises Goldszmidt, and Abraham Wyner. 2013. Data analysis with Bayesian networks: A bootstrap approach. *arXiv preprint arXiv:1301.6695* (2013).
[13] Nir Friedman and Daphne Koller. 2003. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine learning* 50, 1 (2003), 95–125.
[14] Stuart Geman, Elie Bienenstock, and René Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural computation* 4, 1 (1992), 1–58.
[15] Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of causal discovery methods based on graphical models. *Frontiers in genetics* 10 (2019), 524.
[16] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2019. Logan: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, Vol. 2019. 133–152.
[17] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. 2021. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429* (2021).
[18] James J Heckman. 2008. Econometric causality. *International statistical review* 76, 1 (2008), 1–27.
[19] Yimin Huang and Marco Valtorta. 2006. Pearl's calculus of intervention is complete. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI)*. 217–224.
[20] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. 2021. Practical Blind Membership Inference Attack via Differential Comparisons. *Network and Distributed Systems Security (NDSS)* (2021).
[21] Amin Jaber, Jiji Zhang, and Elias Bareinboim. 2018. Causal Identification under Markov Equivalence. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*.

[22] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 259–274.

[23] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. 2019. Predicting the Generalization Gap in Deep Networks with Margin Distributions. In *International Conference on Learning Representations (ICLR)*.

[24] Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. 2019. Characterization and Learning of Causal Graphs with Latent Variables from Soft Interventions.. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[25] Ron Kohavi, David H Wolpert, et al. 1996. Bias plus variance decomposition for zero-one loss functions. In *ICML*, Vol. 96. 275–83.

[26] Sanghack Lee and Elias Bareinboim. 2021. Causal identification with matrix equations. *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021).

[27] Klas Leino and Matt Fredrikson. 2020. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *USENIX Security Symposium*. 1605–1622.

[28] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2021. Membership Inference Attacks and Defenses in Classification Models. In *ACM Conference on Data and Application Security and Privacy (CODASPY)*. 5–16.

[29] Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In *ACM Conference on Computer and Communications Security (CCS)*.

[30] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. 2021. EncoderMI: Membership inference against pre-trained encoders in contrastive learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2081–2095.

[31] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2021. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. *arXiv preprint arXiv:2102.02551* (2021).

[32] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. 2018. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889* (2018).

[33] Margaret Mooney Marini and Burton Singer. 1988. Causality in the social sciences. *Sociological methodology* 18 (1988), 347–409.

[34] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks. *arXiv preprint arXiv:2203.03929* (2022).

[35] Sasi Kumar Murakonda and Reza Shokri. 2020. ML Privacy Meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339* (2020).

[36] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 634–646.

[37] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 739–753.

[38] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. 2020. Realcause: Realistic causal inference benchmarking. *arXiv preprint arXiv:2011.15007* (2020).

[39] Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, and Ioannis Mitliagkas. 2018. A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591* (2018).

[40] Official Journal of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). L119 (2016), 1–88.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[42] Judea Pearl. 2009. *Causality*. Cambridge university press.

[43] Judea Pearl. 2012. On a class of bias-amplifying variables that endanger effect estimates. *arXiv preprint arXiv:1203.3503* (2012).

[44] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19 (2000), 2.

[45] James M Robins, Andrea Rotnitzky, and Daniel O Scharfstein. 2000. Sensitivity analysis for selection bias and unmeasured confounding in missing data and causal inference models. In *Statistical models in epidemiology, the environment, and clinical trials*. Springer, 1–94.

[46] Robert W Robinson. 1977. Counting unlabeled acyclic digraphs. In *Combinatorial mathematics V*. Springer, 28–43.

[47] Paul R Rosenbaum. 2005. Sensitivity analysis in observational studies. *Encyclopedia of statistics in behavioral science* (2005).

[48] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning (ICML)*. PMLR, 5558–5567.

[49] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed Systems Security Symposium (NDSS)*. Internet Society.

[50] Gideon Schwarz. 1978. Estimating the dimension of a model. *The annals of statistics* (1978), 461–464.

[51] Marco Scutari. 2010. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software* 35, 3 (2010), 1–22. https://doi.org/10.18637/jss.v035.i03

[52] Marco Scutari and Radhakrishnan Nagarajan. 2011. On Identifying Significant Edges in Graphical Models of Molecular Networks. *arXiv preprint arXiv:1104.0896* (2011).

[53] Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. 2015. Learning Causal Graphs with Small Interventions. *Advances in Neural Information Processing Systems (NeurIPS)* 28 (2015), 3195–3203.

[54] Amit Sharma and Emre Kiciman. 2020. DoWhy: An End-to-End Library for Causal Inference. *arXiv preprint arXiv:2011.04216* (2020).

[55] Amit Sharma, Emre Kiciman, et al. 2019. DoWhy: A Python package for causal inference. https://github.com/microsoft/dowhy.

[56] Reza Shokri, Martin Strobel, and Yair Zick. 2021. On the privacy risks of model explanations. In *AAAI/ACM Conference on AI, Ethics, and Society*. 231–241.

[57] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 3–18.

[58] Ilya Shpitser and Judea Pearl. 2006. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 1219.

[59] Edward H Simpson. 1951. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 13, 2 (1951), 238–241.

[60] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*.

[61] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Membership inference attacks against adversarially robust deep learning models. In *IEEE Security and Privacy Workshops (SPW)*. IEEE, 50–56.

[62] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy risks of securing machine learning models against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 241–257.

[63] Shruti Tople, Amit Sharma, and Aditya Nori. 2020. Alleviating privacy attacks via causal learning. In *International Conference on Machine Learning (ICML)*. PMLR, 9537–9547.

[64] Sofia Triantafillou, Vincenzo Lagani, Christina Heinze-Deml, Angelika Schmidt, Jesper Tegner, and Ioannis Tsamardinos. 2017. Predicting causal relationships from biological data: Applying automated causal discovery on mass cytometry data of human immune cells. *Scientific reports* 7, 1 (2017), 1–11.

[65] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing* (2019).

[66] Michael Carl Tschantz, Shayak Sen, and Anupam Datta. 2020. Sok: Differential privacy as a causal property. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 354–371.

[67] Huihai Wu and Xiaohui Liu. 2008. Dynamic bayesian networks modeling for inferring genetic regulatory networks by search strategy: Comparison between greedy hill climbing and mcmc methods. In *Proc. of World Academy of Science, Engineering and Technology*, Vol. 34. Citeseer, 224–234.

[68] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. 2020. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning (ICML)*. PMLR, 10767–10777.

[69] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Computer Security Foundations Symposium (CSF)*.

[70] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. Analyzing information leakage of updates to natural language models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 363–375.

[71] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership Inference Attacks Against Recommender Systems. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 864–879.