TMI! Finetuned Models Spill Secrets from Pretraining

John Abascal [1] **Stanley Wu** [1] **Alina Oprea** [1] **Jonathan Ullman** [1]

Abstract

Transfer learning has become an increasingly popular technique in machine learning as a way to leverage a pretrained model trained for related tasks. This paradigm has been especially popular for *privacy preserving machine learning*, where the pretrained model is considered public, and only the data for finetuning is considered sensitive. However, there are reasons to believe that the data used for pretraining is still sensitive. In this work we study privacy leakage via membership-inference attacks, and we propose a new threat model where the adversary only has access to the finetuned model and would like to infer the membership of the pretraining data. To realize this threat model, we implement a novel metaclassifier-based attack, **TMI**. We evaluate **TMI** on both vision and natural language tasks across multiple transfer learning settings, including finetuning with differential privacy. Through our evaluation, we find that **TMI** can successfully infer membership of pretraining examples using query access to the finetuned model.

1. Introduction

Transfer learning has become an increasingly popular technique in machine learning as a way to leverage a model trained for one task to assist with building a model for a related task. Typically, we begin with a large *pretrained model* trained with abundant data and computation, and use it as a starting point for training a *finetuned model* to solve a new task where data and computation is scarce. This paradigm has been especially popular for *privacy* in machine learning (Papernot et al., 2020; Yu et al., 2022; Li et al., 2022; Bu et al., 2023; He et al., 2022; Ganesh et al., 2023), because the data for pretraining is often considered public and thus the pretrained model provides a good starting point before we even have to touch sensitive data.

Although the data used to pretrain large models is typically scraped from the Web and publicly accessible, there are several reasons to believe that this data is still sensitive (Tramèr et al., 2022). For example, even ubiquitous and thoroughly examined pretraining datasets like ImageNet have been shown to contain sensitive content (Quach, 2019; Yang et al., 2022) obtained without consent. Companies, such as Google, have begun using internal datasets scraped from the Web (Tuesday & Networks) to train models to be finetuned and published by smaller organizations, making it imperative to understand the privacy risks posted by models pretrained on these ostensibly public datasets. Thus, the central question we attempt to understand in this work is: *How much sensitive information does a finetuned model reveal about the data that was used for* pretraining*?*

We study this question via *membership-inference (MI) attacks* (Homer et al., 2008; Shokri et al., 2016). A MI attack allows an adversary with access to the model to determine whether or not a given data point was included in the training data. MI attacks have been extensively studied in several machine learning applications such as computer vision (Carlini et al., 2022) and contrastive learning (Liu et al., 2021). The success of MI attacks makes it clear that the pretrained model will leak information about the pretraining data. However, the process of finetuning the model will obscure information about the orignal model, and there are no works that study MI attacks that use the *finetuned model* to recover *pretraining data.*

To answer our question, we create a novel, metaclassifier-based membership-inference attack, **T**ransfer **M**embership **I**nference (**TMI**) to circumvent the challenges that arise when trying to adapt prior attacks to asses privacy leakage in this new setting where the adversary has query access only to the finetuned model. The goal of our new membership-inference adversary is to infer whether or not specific individuals were included in the pretraining set of the finetuned machine learning model. This setting stands in contrast to prior membership-inference attacks, as it restricts the adversary from directly querying the model trained on the specific dataset they wish to perform membership-inference on. State-of-the-art, black-box MI

[1] Khoury College of Computer Sciences, Northeastern University, Boston, MA. Correspondence to: John Abascal <abascal.j@northeastern.edu>.

2nd AdvML Frontiers workshop at 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

attacks rely on a model's prediction confidence with respect to the ground truth label, but the finetuned model does not necessarily have the ground truth label in its range. Thus, our attack leverages how individual samples from pretraining influence predictions on the downstream task by observing entire prediction vectors from the finetuned model. More concretely, **TMI** constructs a dataset of prediction vectors from queries to finetuned shadow models in order to train a metaclassifier that can infer membership.

We comprehensively evaluate **TMI** on pretrained CIFAR-100 (Krizhevsky, 2009) vision models, transferred to CIFAR-10 (Krizhevsky, 2009) and a coarse-labeled version of CIFAR-100. We also evaluate an extension of **TMI** on finetuned version of publicly available large language language models which are pretrained on WikiText-103 (Merity et al., 2016) and finetuned on DBpedia (Zhang et al., 2015). We compare our results to both a simple adaptation of the likelihood ratio attack (Carlini et al., 2022) to our setting and a MI attack that has direct access to the pretrained model.

2. Background and Related Work

We provide the necessary background on machine learning and related work on MI attacks.

2.1. Background and Notation

2.1.1. Scaling Model Confidences

The classifier models we consider are trained in a supervised manner (i.e. on labeled training data) and output a vector of probabilities, $\vec{y}$, where each entry y_i corresponds to the *model's prediction confidence* with respect to label, i. This is done by applying the $\texttt{softmax}$ activation function to the model's final layer. Given a vector of logits, $\vec{z}$ (i.e. the model's final layer), we define $\texttt{softmax}(\vec{z}) : \mathbb{R}^K \rightarrow (0,1)^K$

$$y_i = \texttt{softmax}(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where K is the number of possible classes.

Prior work (Carlini et al., 2022) has used the logit function, $\texttt{logit}(p) = \log(\frac{p}{1-p})$, to scale model confidences. This scaling yields an approximately normally distributed statistic that can be used to perform a variety of privacy attacks (Carlini et al., 2022; Chaudhari et al., 2022; Tramèr et al., 2022). The logit function is obtained by inverting the $\texttt{sigmoid}$ activation function, $\sigma(x) = \frac{1}{1+e^x}$, which is a specific case of $\texttt{softmax}$ being used for binary classification.

Following the lead of prior work, we use ϕ to perform our model confidence scaling. We define model confidence scaling $\phi(\vec{y}) : \mathbb{R}^K \rightarrow \mathbb{R}^K$ for a prediction vector, $\vec{y}$, as

$$\phi(\vec{y}) = (\texttt{logit}(y_1), \ldots, \texttt{logit}(y_K))$$

2.1.2. Transfer Learning

Feature extraction and updating a model's pretrained weights are popular transfer learning techniques used to improve a pretrained deep learning model's performance on a specific task. In the classification setting, feature extraction involves freezing a model's pretrained weights and using them to extract relevant features from input data, which are then fed into a linear layer for classification. This technique is useful when working with limited data or when the pretrained model has learned generalizable features that are useful for the target task. On the other hand, finetuning a model by updating its pretrained weights involves taking a pretrained model and training it on a new dataset, often with a smaller learning rate, to adapt it to the new task. This kind of finetuning is more suited for situations where the new task has similar characteristics, but not a direct correspondence, to the original pretraining task.

2.2. Related Work

Membership-inference attacks (Homer et al., 2008) aim to determine whether or not a given individual's data record was present in a machine learning model's training dataset. Learning whether or not an individual was part of a sensitive dataset can serve as the basis for more powerful extraction attacks (Carlini et al., 2022). Because of their simplicity and connection to differential privacy, MI attacks are also a popular way to audit machine learning models for privacy leakage (Song & Shmatikov, 2018; ten; Ye et al., 2022). These attacks have been extensively studied with two types of adversarial access: black-box query access (Carlini et al., 2022; Yeom et al., 2018; Shokri et al., 2016; Ye et al., 2022; Tramèr et al., 2022) and white-box access to the machine learning model's parameters (Leino & Fredrikson, 2020). Despite there being extensive work on black-box attacks and prior work on MI attacks on pretrained encoders (Liu et al., 2021), there are few works that explore MI in the transfer learning setting. The works that do (Hidano et al., 2020; Zou et al., 2020) have not studied MI attacks on the pretraining dataset of a finetuned machine learning model.

3. Threat Model

Our problem is to determine how much information a finetuned model reveals about the pretraining data using MI attacks. In both the standard MI experiment and our newly defined experiment, there is a machine learning model trained on some dataset, and a challenge point that is drawn from the same distribution as the training data. Introducing

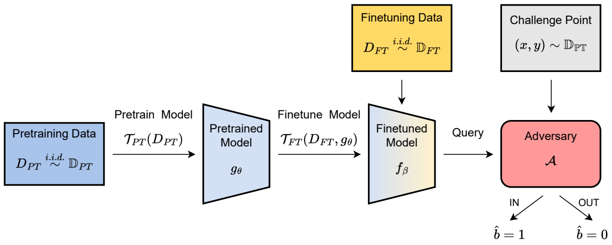

Finetuning Data
$D_{FT} \overset{i.i.d.}{\sim} \mathbb{D}_{FT}$
Challenge Point
$(x, y) \sim \mathbb{D}_{\mathbb{PT}}$
Pretraining Data
$D_{PT} \overset{i.i.d.}{\sim} \mathbb{D}_{PT}$
Pretrain Model
$\mathcal{T}_{PT}(D_{PT})$
Pretrained Model
g_θ
Finetune Model
$\mathcal{T}_{FT}(D_{FT}, g_\theta)$
Finetuned Model
f_β
Query
Adversary
$\mathcal{A}$
IN
OUT
$\hat{b} = 1$
$\hat{b} = 0$

Figure 1. Our New Membership-Inference Threat Model.

a separate finetuning phase with possibly different data adds an additional layer of indirection, potentially making MI more challenging by limiting the attacker's ability to query the original pretrained model. Formally, our threat model, visualized in Figure 1, is described by the following game between a *challenger* C and an *adversary* $\mathcal{A}$:

MI Security Game with a Finetuned Target Model

1. The challenger receives a dataset D_{PT} comprised of points sampled i.i.d. from some distribution $\mathbb{D}_{PT}$, and a pretrained model $g_\theta \leftarrow \mathcal{T}_{PT}(D_{PT})$.

2. The challenger draws i.i.d. samples from another distribution $\mathbb{D}_{FT}$ to create a dataset D_{FT} and finetunes the model on D_{FT} using its pretrained weights, θ, to obtain a new model $f_\beta \leftarrow \mathcal{T}_{FT}(D_{FT}, g_\theta)$

3. The challenger randomly selects $b \in \{0, 1\}$. If the $b = 0$, the challenger samples a point (x, y) from $\mathbb{D}_{PT}$ uniformly at random, such that $(x, y) \notin D_{PT}$. Otherwise, the challenger samples (x, y) from D_{PT} uniformly at random.

4. The challenger sends the point, (x, y) to the adversary.

5. The adversary, using the challenge point, sampling access to $\mathbb{D}_{PT}$ and $\mathbb{D}_{FT}$, and query access to f_β, produces a bit $\hat{b}$.

6. The adversary wins if $b = \hat{b}$ and loses otherwise.

In our security game, we assume that the adversary has query access to the finetuned target model f_β and knowledge of the pretraining data distribution $\mathbb{D}_{PT}$. Because we will be training *shadow models* (Shokri et al., 2016) to perform our MI attack, the adversary also requires knowledge of the underlying distribution from which the finetuning dataset is sampled, $\mathbb{D}_{FT}$, and knowledge of the target model's architecture and training algorithm. The knowledge we assume is the same as many other works on MI (e.g. (Shokri et al., 2016; Carlini et al., 2022)). We also assume that the adversary's queries to the target model return numerical confidence scores for each label rather than just a single label, similar to prior privacy attacks (Shokri et al., 2016; Yeom et al., 2018; Tramèr et al., 2022; Carlini et al., 2022).

4. Methodology

In this section, we will propose attacks that follow the threat model proposed in Section 3. The algorithms for these attacks can be found in Section B of the appendix.

4.1. Adapting an Existing Attack

As a first attempt to create an effective membership-inference attack on finetuned machine learning models, we can consider an adaptation of the *likelihood ratio attack* (LiRA) proposed by Carlini et al. (Carlini et al., 2022). In this attack, the adversary observes the target model's prediction confidence on a challenge point with respect to the true label of the challenge point. Because the model's confidence with respect to a given label is approximately normally distributed, Carlini et al. perform a likelihood ratio test to infer the challenge point's membership status, using the shadow models to parameterize the IN and OUT distributions.

Because we consider an adversary with query access to the finetuned model, the ground truth label may not be in the range of our target model. Thus, we cannot perform the likelihood ratio attack. Instead, we can adapt the attack to use the label which the target model predicts with the highest confidence, $\hat{y}$. To increase attack success, we query each shadow and target model on M random augmentations of the challenge point and fit M-dimensional multivariate normal distributions to the scaled model confidences we aggregate to improve attack success.

4.2. Issues with Adapting LiRA

While this adaptation of LiRA is somewhat effective at inferring membership (Figure 3a), it only captures how the pretraining dataset influences model's predictions with respect to a single label in the downstream dataset. Because the purpose of pretraining is to extract and learn general features that can be used in several downstream tasks, one would expect that the weights of a pretrained model have some impact on *all* of a finetuned model's prediction confidences.

Furthermore, if we observe the distribution of scaled model confidences over our shadow models, we see that it is approximately normal regardless of the choice of label (Figure 2). This may lead one to believe that the correct adaptation of LiRA to our setting would be to fit a multivariate normal distribution to the entire prediction vectors output by our shadow models, but this is not the case because the adversary only receives model confidences. When softmax is applied, it converts the logit vector $\vec{z}$ into a probability distribution, $\vec{y}$, over the labels. Thus, the entries of $\vec{y}$ can be written as $\vec{y} = (p_1, p_2, \ldots, p_K) \in (0,1)^K$ where K is the number of classes and each p_i denotes the model's confidence on class i. Because the entries of $\vec{y}$ must sum up to 1, any entry p_i can be written as $1 - \sum_{j \neq i} p_j$. This means that the prediction vector (and our computed logits) actually lie on a $(K-1)$-dimensional subspace of the K-dimensional space where the model's actual logits lie, and we cannot fit a K-dimensional multivariate normal distribution the to all of our models' logit scaled prediction vectors without arbitrarily removing one of the entries in $\vec{y}$.

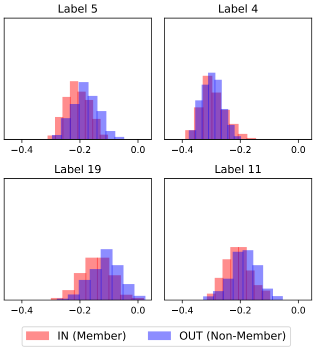


Figure 2. Scaled Model Confidences of Shadow Models at a Random Sample of Labels

4.3. Our TMI Attack

Our **T**ransfer **M**embership **I**nference (**TMI**) attack (Algorithm 3) starts with the same shadow model training procedure as Algorithm 2, where the adversary trains shadow models on datasets sampled from $\mathbb{D}_{PT}$ and finetunes them on datasets sampled from $\mathbb{D}_{FT}$. The adversary then queries the challenge point on these shadow models to construct a dataset, D_{meta}, comprised of logits attained from scaling the prediction vectors as described in Section 2.1.1. To construct a distinguishing test that circumvents the issues that arise when attempting to parameterize the distribution of prediction vectors, the adversary trains a *metaclassifier* on D_{meta}, queries the target model on the challenge point, and scales the target model's prediction vector. Lastly, the adversary queries this observed prediction vector on their metaclassifier, which outputs a score in the interval $[0, 1]$ that indicates the predicted membership status of the challenge point.

In our implementation of **TMI** for computer vision models, we train a metaclassifier per challenge point. Because we use a relatively small number of shadow models (64 IN and 64 OUT in total), we leverage random augmentations to construct a larger metaclassifier dataset. Each time we query the target model or our local shadow models, we query M times with different random augmentations of the challenge point, including random horizontal flips and random crops with padding.

Due to computational limitations, we do not pretrain any shadow models for our attacks in the language domain. Rather, we use a publicly hosted pretrained model and finetune it on a downstream task. Without control over pretraining, we cannot produce a metaclassifier dataset with prediction vectors from both IN and OUT shadow models with respect to a single challenge point. As a result, we use a *global metaclassifier*, trained on a dataset containing the prediction vectors of *all* challenge points, to produce membership scores.

5. TMI Evaluation

We evaluate the performance of our **TMI** attack on image models with three downstream tasks and a finetuned version of a public, pretrained language model. We evaluate the success of our attack as a function of the number of updated parameters. Additionally, we observe the success of our attack when differential privacy (Dwork et al., 2006) is used in the finetuning process.

This section presents the results of our evaluation of **TMI** and addresses *five primary research questions* with respect to the datasets in our experiments.

5.1. Metrics

To evaluate the performance of **TMI**, we use a set of metrics that are commonly used in the literature. Although average accuracy is a common metric used to evaluate MI attacks, it is not sufficient by itself to measure the performance of MI attacks. Thus we also evaluate our attack using the *receiver operating characteristic* (ROC) curves on a log-log scale, the area under the curve (AUC), and we report the TPR at low, fixed FPR of 0.1% and 1%.

5.2. Experimental Results

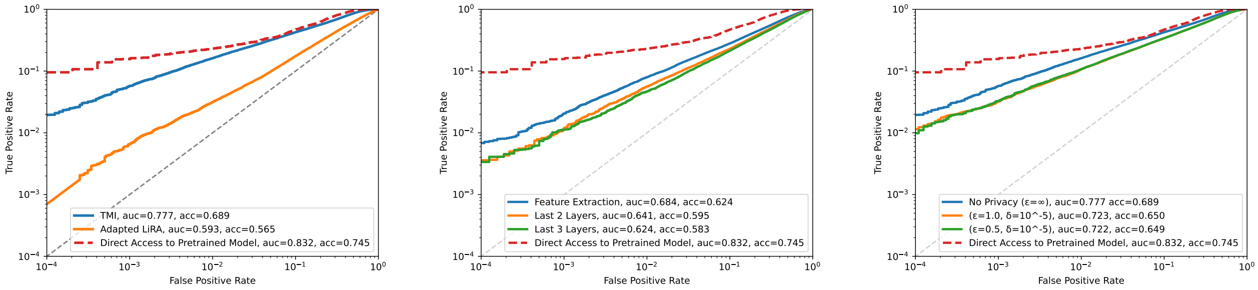
In this section, we will discuss the performance of our attack on both vision and language models finetuned on a variety of tasks with different finetuning streategies. Details for shadow model training and datasets can be found in Section C.

5.2.1. Finetuning Vision Models

Q1: Can finetuned models leak private information about their pretraining datasets via black-box queries?

Q2: Does updating a model's pretrained parameters instead of freezing them prevent privacy leakage?

To answer these research questions, we evaluate the success of our **TMI** attack on vision models finetuned on CIFAR-10



(a) Coarse CIFAR-100; Feature Extraction

(b) CIFAR-10; Updating Parameters

(c) Coarse CIFAR-100; Finetuning with DP-SGD

Figure 3. **TMI** Attack Performance on Finetuned Vision Models

and Coarse CIFAR-100, with and without updating any of the pretrained parameters, respectively. Our attack's success depends on the target model having high utility on its respective task, so it is important to ensure that we choose downstream tasks that are similar or relevant to the pretraining task when using feature extraction to finetune models. To transfer the pretrained models to Coarse CIFAR-100, we remove the final classification layer, and replace it with a randomly initialized classification layer containing the proper number of classes for the new downstream task. The remaining weights are kept frozen throughout training. To finetune on CIFAR-10, we progressively unfreeze our ResNet models' weights. In our experiments with vision models, we designate 1000 samples to be challenge points and run our attack across 128 target models.

As shown in Table 1, we observe that **TMI** is able to achieve a TPR of 5.7% and 16.1% at 0.1% and 1% FPR, respectively, on the Coarse CIFAR-100 downstream task. Despite being constrained to only having query access to the finetuned model, Figure 3a shows that the TPR of **TMI** is approximately equal at higher FPR (about 5%) to that of running LiRA directly on the pretrained model. Our summary statistics, AUC and average accuracy (0.78 and 69%) both remain within 0.06 of the adversary which has access to the pretrained model (0.83 and 75%).

Q1 Answer: Yes, it is possible to infer the membership status of an individual in a machine learning model's pretraining set via query access to the finetuned model.

Furthermore, Figure 3b shows that the AUC and accuracy of **TMI** slightly decrease as we finetune an increasing number of parameters. We also observe a very slight decrease the TPR at a 1% FPR when the number of finetuned parameters is increased from 2 layers to 3 layers. TPR decreases more significantly when comparing to the TPR of **TMI** on models finetuned with feature extraction. In Table 1, we observe that updating the model's parameters induces a decrease in up to 0.9% at a 0.1% FPR and up to 3.3% at a 1% FPR. Nonetheless, **TMI** achieves comparable AUC and average accuracy metrics to feature extraction when we finetune the majority of model parameters. We hypothesize that this slight decrease in our attack's success may be due to the "forgetting" of old training points that happens when updating model parameters (Jagielski et al., 2023).

Q2 Answer: Updating larger subsets of model parameters decreases the success of our **TMI** attack, but we are still able to infer the membership status of the majority of samples in the pretraining dataset.

Table 1. TPR at Fixed FPR of **TMI** and Our Adaptation of LiRA on Vision Models (Figure 3)

Task	TPR @ 0.1% FPR	TPR @ 1% FPR
TMI (Coarse CIFAR-100)	5.7%	16.1%
Adapted LiRA (Coarse CIFAR-100)	0.7%	3.1%
Feature Extraction (CIFAR-10)	2.0%	8.0%
Last 2 Layers, 60% (CIFAR-10)	1.1%	5.6%
Last 3 Layers, 90% (CIFAR-10)	1.1%	4.7%
LiRA Directly on Pretrained Model	15.6%	22.9%

5.2.2. Finetuning Pretrained Language Models

Q3: Can the attack be generalized to domains other than vision?

To answer this research question, we evaluate the success of our **TMI** attack in the natural language domain. In particular, we focus on publicly available pretrained large language models (LLMs), or foundation models (Bommasani et al., 2022), which we finetune on a text classification task.

As an alternative to pretraining our own LLMs, we evaluate our attack on a widely used pretrained foundation model, Transformer-XL (Dai et al., 2019), along with its corresponding tokenizer, which are hosted by Hugging Face (hug). We chose this foundation model in particular because it uses known training, validation, and testing splits from the WikiText-103 (Merity et al., 2016) dataset,

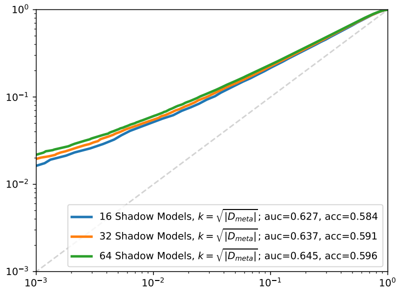


Figure 4. **TMI** Performance on a Publicly Available Transformer-XL Model Finetuned on DBpedia-14 Topic Classification

providing us with the exact partitions necessary to evaluate **TMI** without having to train our own LLMs. Since we cannot pretrain our own LLMs and thus cannot use the shadow model training procedure in Algorithm 1, we are unable to compare **TMI** to our adaptation of LiRA. Through our evaluation of **TMI** on finetuned foundation models, we will also answer the following research question:

Q4: Is it feasible to mount our attack on finetuned models that are based on publicly hosted foundation models?

The results of our evaluation on LLMs are presented in Figure 4 and Table 2. As shown in Table 2, **TMI**is able to achieve a TPR of 3.4% and 8.8% at 0.1% and 1% FPR, respectively. These results are comparable to our findings on CIFAR-10 from Table 1 in the vision domain. Surprisingly, we do not observe a notable difference in our summary statistics as we increase the number of shadow models from 16 to 64, with an increase of only 0.652 to 0.673 in AUC, and 60% to 61.3% in accuracy as shown in Figure 4. During our evaluation, we find that k-nearest neighbors (KNN) significantly outperforms a neural network as a global metaclassifier. We believe this to be the case due to the additional variance incurred in a (global) metaclassifier dataset containing prediction vectors from *all* challenge points instead of just a single challenge point like our other metaclassifier datasets.

Q3 Answer: Yes, we are able to generalize **TMI** to the natural language domain.

Q4 Answer: Yes, we show that our attack is effective against a finetuned version of the public, pretrained Transformer-XL foundation model without the need to pretrain any additional large language models.

5.2.3. Transfer Learning with Differential Privacy

Q5: Is privacy leakage present even when a model is finetuned using differential privacy?

Table 2. TPR at Fixed FPR of **TMI** on Pretrained WikiText-103 Transformer-XL (Figure 4)

Task	TPR @ 0.1% FPR	TPR @ 1% FPR
16 Shadow Models	1.6%	5.2%
32 Shadow Models	2.0%	5.5%
64 Shadow Models	2.2%	6.0%

We also discuss the performance of our attack on target models that were finetuned with differential privacy. In our experiments, we perform feature extraction to finetune our pretrained CIFAR-100 models on Coarse CIFAR-100. We train the final classification layer using DP-SGD (Abadi et al., 2016) with target privacy parameters $\varepsilon = \{0.5, 1\}$ and $\delta = 10^{-5}$ and clipping norm equal to 5.

Figure 3c shows that the success of our attack only decreases slightly (potentially due to decrease in utility) when differential privacy is used to train the final classification layer on a downstream task. When we finetune our models on Coarse CIFAR-100 with privacy parameters $\varepsilon = 0.5$ and $\delta = 10^{-5}$, **TMI** has a TPR of 3.3% at a FPR of 0.1% and a TPR of 10.7% at a FPR 1%, and it maintains 95% of the accuracy and AUC of our attack on the non-private finetuned model.

Q5 Answer: Finetuning a pretrained model using DP-SGD provides a privacy guarantee *only* for the downstream dataset, and thus has little to no impact on privacy for the pretraining set.

Table 3. TPR at Fixed FPR of **TMI** when Target Models are Finetuned on Coarse CIFAR-100 with DP-SGD (Figure 3c)

Privacy Parameters	TPR @ 0.1% FPR	TPR @ 1% FPR
$(\varepsilon = \infty)$	5.7%	16.1%
$(\varepsilon = 1.0,\ \delta = 10^{-5})$	3.2%	10.6%
$(\varepsilon = 0.5,\ \delta = 10^{-5})$	3.3%	10.7%

6. Conclusion

We study the critical issue of privacy leakage in the transfer learning setting by proposing a novel threat model and introducing **TMI**, a metaclassifier-based membership-inference attack. In particular, we explore how finetuned models can leak the membership status of individuals in the pretraining dataset without an adversary having direct access to the pretrained model. Instead, we rely on queries to the finetuned model to extract private information about the pretraining dataset. Through our evaluation of **TMI**, we demonstrate privacy leakage in a variety of transfer learning settings, including finetuning with differential privacy. We demonstrate the effectiveness of our attack across models in both the vision and natural language domains, highlighting the susceptibility of finetuned models to leaking private information about their pretraining datasets.

References

URL http://groups.csail.mit.edu/vision/TinyImages/.

Transformer XL — huggingface.co. https://huggingface.co/docs/transformers/model_doc/transfo-xl. [Accessed 19-May-2023].

Introducing a New Privacy Testing Library in TensorFlow — blog.tensorflow.org. https://blog.tensorflow.org/2020/06/introducing-new-privacy-testing-library.html. [Accessed 23-May-2023].

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145%2F2976749.2978318.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. On the opportunities and risks of foundation models, 2022.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Differentially private bias-term only fine-tuning of foundation models, 2023. URL https://openreview.net/forum?id=zoTUH3Fjup.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramèr, F. Membership inference attacks from first principles. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pp. 1897–1914. IEEE, 2022. doi: 10.1109/SP46214.2022.9833649. URL https://doi.org/10.1109/SP46214.2022.9833649.

Chaudhari, H., Abascal, J., Oprea, A., Jagielski, M., Tramèr, F., and Ullman, J. Snap: Efficient extraction of private properties with poisoning, 2022.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In Halevi, S. and Rabin, T. (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Ganesh, A., Haghifam, M., Nasr, M., Oh, S., Steinke, T., Thakkar, O., Thakurta, A., and Wang, L. Why is public pretraining necessary for private model training?, 2023.

Golatkar, A., Achille, A., Wang, Y.-X., Roth, A., Kearns, M., and Soatto, S. Mixed differential privacy in computer vision, 2022.

He, J., Li, X., Yu, D., Zhang, H., Kulkarni, J., Lee, Y. T., Backurs, A., Yu, N., and Bian, J. Exploring the limits of differentially private deep learning with group-wise clipping, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Hidano, S., Kawamoto, Y., and Murakami, T. Transmia: Membership inference attacks using transfer shadow training. *CoRR*, abs/2011.14661, 2020. URL https://arxiv.org/abs/2011.14661.

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J. V., Stephan, D. A., Nelson, S. F., and Craig, D. W. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet.*, 4(8):e1000167, August 2008.

Jagielski, M., Thakkar, O., Tramer, F., Ippolito, D., Lee, K., Carlini, N., Wallace, E., Song, S., Thakurta, A. G., Papernot, N., and Zhang, C. Measuring forgetting of memorized training examples. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7bJizxLKrR.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Leino, K. and Fredrikson, M. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association. ISBN 978-1-939133-17-5.

Li, X., Tramèr, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2110.05679.

Liu, H., Jia, J., Qu, W., and Gong, N. Z. Encodermi: Membership inference against pre-trained encoders in contrastive learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, pp. 2081–2095, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3484749. URL https://doi.org/10.1145/3460120.3484749.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017a.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017b. URL https://openreview.net/forum?id=Skq89Scxx.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models, 2016.

Papernot, N., Chien, S., Song, S., Thakurta, A., and Erlingsson, U. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy, 2020. URL https://openreview.net/forum?id=rJg851rYwH.

Quach, K., Oct 2019. URL https://www.theregister.com/2019/10/23/ai_dataset_imagenet_consent.

Shokri, R., Stronati, M., and Shmatikov, V. Membership inference attacks against machine learning models. *CoRR*, abs/1610.05820, 2016. URL http://arxiv.org/abs/1610.05820.

Song, C. and Shmatikov, V. The natural auditor: How to tell if someone used your words to train their model. *CoRR*, abs/1811.00513, 2018. URL http://arxiv.org/abs/1811.00513.

Tramèr, F., Shokri, R., San Joaquin, A., Le, H., Jagielski, M., Hong, S., and Carlini, N. Truth serum: Poisoning machine learning models to reveal their secrets. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, pp. 2779–2792, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505. doi: 10.1145/3548606.3560554. URL https://doi.org/10.1145/3548606.3560554.

Tramèr, F., Kamath, G., and Carlini, N. Considerations for differentially private learning with large-scale public pretraining, 2022. URL https://arxiv.org/abs/2212.06470.

Tuesday, J. . and Networks, C. V. s. L. Revisiting the unreasonable effectiveness of data. URL https://ai.googleblog.com/2017/07/revisiting-unreasonable-effectiveness.html.

Yang, K., Yau, J., Fei-Fei, L., Deng, J., and Russakovsky, O. A study of face obfuscation in imagenet. In *International Conference on Machine Learning (ICML)*, 2022.

Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, pp. 3093–3106, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505. doi: 10.1145/3548606.3560675. URL https://doi.org/10.1145/3548606.3560675.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. pp. 268–282, 07 2018. doi: 10.1109/CSF.2018.00027.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., Yekhanin, S., and Zhang, H. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Q42f0dfjECO.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.,

2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Zou, Y., Zhang, Z., Backes, M., and Zhang, Y. Privacy analysis of deep learning in the wild: Membership inference attacks against transfer learning. *CoRR*, abs/2009.04872, 2020. URL https://arxiv.org/abs/2009.04872.

A. Additional Background

A.1. Differential Privacy

Differential Privacy (Dwork et al., 2006) is a mathematical definition of privacy that bounds the influence that any single individual in the training data has on the output of the model. Specifically, an algorithm satisfies differential privacy if for any two datasets that differ on one individual's training data, the probability of seeing any set of potential models is roughly the same regardless of which dataset was used in training.

Definition A.1. A randomized algorithm $\mathcal{M}$ mapping datasets to models satisfies (ε, δ)*-differential privacy* if for every pair of datasets X and X' differing on at most one training example and every set of outputs E,

$$\Pr[\mathcal{M}(X) \in E] \leq e^{\varepsilon} \Pr[\mathcal{M}(X') \in E] + \delta$$

B. Algorithms

Algorithm 1 `train_shadow_models`$(x,\ b)$

Our shadow model training procedure considers both the pretraining and finetuning phases to mimic the behavior of the target model on a challenge point.

```
Require: Query access to both $\mathbb{D}_{PT}$ and $\mathbb{D}_{FT}$ and a fixed dataset size $S = \frac{1}{2}|\mathbb{D}_{PT}|$
models ← {}
datasets ← {}
for $N$ times do
    Draw $S$ i.i.d. samples from $\mathbb{D}_{PT}$ to construct $\tilde{D}_{PT}$
    datasets ← datasets ∪ {$\tilde{D}_{PT}$}
    $g \leftarrow \mathcal{T}(\tilde{D}_{PT})$
    Sample $\tilde{D}_{FT}$ i.i.d. using query access to $\mathbb{D}_{FT}$
    $f \leftarrow \mathcal{T}(g,\ \tilde{D}_{FT})$ {Finetune $g$ on $\tilde{D}_{FT}$}
    models ← models ∪ {$f$}
end for
return models, datasets
```

Algorithm 2 Adapted LiRA

We adapt the MI attack shown in (Carlini et al., 2022) by using the label which the target model predicted most confidently instead of the ground truth label.

```
Require: A finetuned target model $f_\beta$, a challenge point $x \leftarrow \mathbb{D}_{PT}$, and models and datasets (i.e. the output of
    train_shadow_models() )
preds$_{\text{in}} \leftarrow \{\}$, preds$_{\text{out}} \leftarrow \{\}$
$\vec{v}_{\text{obs}} \leftarrow f_\beta(x)$ {Query the target model on $x$}
conf$_{\text{obs}} \leftarrow$ logit(max$_i\ \vec{v}_{\text{obs},i}$) {Store max confidence score}
$\hat{y} \leftarrow \arg\max_i \vec{v}_{\text{obs},i}$ {Store most confident predicted label}
$i \leftarrow 1$ {Index for saved shadow models and datasets}
for $N$ times do
    if $x \in$ datasets$_i$ then
        $f_{\text{in}} \leftarrow$ models$_i$
        conf$_{\text{in}} \leftarrow$ logit($f_{\text{in}}(x)_{\hat{y}}$) {Query $f_{\text{in}}$ on $x$}
        preds$_{\text{in}} \leftarrow$ preds$_{\text{in}} \cup \{$conf$_{\text{in}}\}$ {Aggregate confidences}
    else if $x \notin$ datasets$_i$ then
        $f_{\text{out}} \leftarrow$ models$_i$
        conf$_{\text{out}} \leftarrow$ logit($f_{\text{out}}(x)_{\hat{y}}$)
        preds$_{\text{out}} \leftarrow$ preds$_{\text{out}} \cup \{$conf$_{\text{out}}\}$
    end if
end for
$\mu_{\text{in}} \leftarrow$ mean(preds$_{\text{in}}$),   $\mu_{\text{out}} \leftarrow$ mean(preds$_{\text{out}}$)
$\sigma^2_{\text{in}} \leftarrow$ var(preds$_{\text{in}}$),   $\sigma^2_{\text{out}} \leftarrow$ var(preds$_{\text{out}}$)
return $\frac{p(\text{conf}_{\text{obs}} | \mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}}))}{p(\text{conf}_{\text{obs}} | \mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}}))}$
```

Algorithm 3 TMI Metaclassifier Attack

We pretrain shadow models with and without the challenge point and finetune them using query access to $\mathbb{D}_{FT}$ to estimate the target model's prediction behavior. Using the prediction vectors of our shadow models on the challenge point, we generate a dataset to train a metaclassifier to determine the challenge point's membership status.

```
Require: A finetuned target model $f_\beta$, a challenge point $x \leftarrow \mathbb{D}_{PT}$, and models and datasets (i.e. the output of
    train_shadow_models() )
preds$_{\text{in}} \leftarrow \{\}$, preds$_{\text{out}} \leftarrow \{\}$
$i \leftarrow 1$
for $N$ times do
    if $x \in$ datasets$_i$ then
        $f_{\text{in}} \leftarrow$ models$_i$
        $\vec{v}_{\text{in}} \leftarrow \phi(f_{\text{in}}(x))$ {Query IN model on x}
        preds$_{\text{in}} \leftarrow$ preds$_{\text{in}} \cup \{(\vec{v}_{\text{in}}, 1)\}$ {Store and label the prediction vector}
    else if $x \notin$ datasets$_i$ then
        $f_{\text{out}} \leftarrow$ models$_i$
        $\vec{v}_{\text{out}} \leftarrow \phi(f_{\text{out}}(x))$
        preds$_{\text{out}} \leftarrow$ preds$_{\text{out}} \cup \{(\vec{v}_{\text{out}}, 0)\}$
    end if
    $i \leftarrow i + 1$
end for
$D_{\text{meta}}$ = preds$_{\text{in}} \cup$ preds$_{\text{out}}$ {Construct the metaclassifier dataset}
$\mathcal{M} \leftarrow \mathcal{T}(D_{\text{meta}})$ {Train a binary metaclassifier}
$\vec{v}_{\text{obs}} = \phi(f_\beta(x))$ {Query the target model on $x$}
Output $\mathcal{M}(\vec{v}_{\text{obs}})$
```

C. Datasets and Models

In this section we will discuss the datasets used in our evaluation of **TMI**. We will also discuss our choices of pretraining and downstream tasks used in our evaluation.

C.1. Datasets

- *CIFAR-100:* The CIFAR-100 (Krizhevsky, 2009) dataset is a subset of the Tiny Images dataset (Tin), provided by the Canadian Institute for Advanced Research. It is comprised of 60,000 32x32 color images from 100 classes, where each class contains 600 images (500 for training and 100 for testing). CIFAR-100 is used as our pretraining task because it is a challenging dataset with a wide variety of classes, which allows our models to learn very general features and patterns that can be applied to several downstream tasks.

- *Coarse CIFAR-100:* The classes in CIFAR-100 can be divided into 20 superclasses. Each image in the dataset has a "fine" label to indicate its class and a "coarse" label to indicate its superclass. We construct this coarse dataset using the superclass labels and use it as our downstream task with the highest similarity to the pretraining task. In our experimentation, we ensure that this downstream task does not contain any of the pretraining samples from the standard CIFAR-100 dataset.

- *CIFAR-10:* In a similar fashion to CIFAR-100, the CIFAR-10 (Krizhevsky, 2009) is comprised of is comprised of 60,000 32x32 color images that come from the Tiny Images dataset. This dataset contains 10 classes, each containing 6000 points (5000 for training and 1000 for testing) which are mutually exclusive to those seen in CIFAR-100. In our evaluation, this downstream task is the second most similar to CIFAR-100 because they are both derived from the same distribution of web-scraped images, but do not overlap at all in their classes. Although the classes do not overlap, the features learned from pretraining on CIFAR-100 may be useful in performing this task.

- *WikiText-103:* WikiText-103 (Merity et al., 2016) is a large-scale language dataset that is widely used for benchmarking language models. It contains over 100 million tokens and is derived from a several Wikipedia articles and contains a vast amount of textual data. The language models we consider in this paper have been pretrained on the train partition of WikiText-103 and are hosted on Hugging Face.

- *DBpedia:* The DBpedia ontology (or topic) classification dataset (Zhang et al., 2015) is composed of 630,000 samples with 14 non-overlapping classes from DBpedia, which is a project aiming to extract structured content from the information on Wikipedia. For each of the 14 topics, there are 40,000 training samples and 5000 testing samples. In our experiments with language models, we finetune a subset of the model's weights on random subsets of this dataset.

C.2. Models

For all of our vision tasks, we use the ResNet-34 (He et al., 2015) architecture. This architecture has been widely used in various computer vision applications due to its superior performance and efficiency. ResNet-34 is a convolutional neural network architecture that uses residual blocks, allowing it to effectively handle the complex features of images and perform well on large-scale datasets.

For our language task, we use the Transformer-XL (Dai et al., 2019) model architecture. In particular, we use the pretrained Transformer-XL model from Hugging Face, which is trained on WikiText-103 (Merity et al., 2016), as our initialization for the downstream task. We finetune our pretrained language model architectures on the DBpedia ontology classification dataset.

C.2.1. Vision Shadow Model Training

Here, we describe the shadow model training procedure for our vision tasks, which comprise the majority of our experiments. The details for how we train shadow models for our language task can be found in Section 5.2.2.

Our shadow model training procedure for vision models is split into two phases: pretraining and finetuning. In the first phase, we train 129 randomly initialized ResNet-34 models on random subsets of CIFAR-100, each containing half of the dataset (25k points). The other 25k samples are held out for evaluation. We train each of the ResNet-34 models for 100 epochs (to 75-80% top-5 accuracy) using SGD with weight decay ($\lambda = 10^{-5}$) and cosine annealing (Loshchilov & Hutter, 2017b) as our learning rate scheduler. When training and querying any of our shadow models, we use standard data augmentations, such as random crops and horizontal flips.

In the second phase, we finetune our shadow models on randomly sampled subsets of our downstream task datasets. Before we finetune each shadow model, we swap the classification layer out with a randomly initialized one that has the proper dimension for the downstream task. We then freeze a subset of the model's pretrained weights. When we use feature extraction to finetune our pretrained models, we freeze all weights except for those in the final classification layer. The weights that aren't frozen are trained using the same process as pretraining, but for only 20 epochs.

When pretraining our shadow models, we designate a randomly selected set of 1000 points to be the challenge points for our **TMI** attack. Because each shadow model is trained on half of the dataset, all of the points (including the challenge points) will be IN and OUT for approximately half of the shadow models. In our experiments, we select one shadow model to be the target model and run our attack using the remaining 128 shadow models. Each time we run our attack, we select the a different shadow model to be the target model, yielding a total of 128 trials.

C.2.2. Language Shadow Model Training

We finetune Transformer-XL on DBpedia (Zhang et al., 2015), modifying the pretrained tokenizer to use a max length of 450, including both truncation and padding. Using a training set of 10,000 randomly sampled datapoints from DBpedia, we finetune the last third of the parameters in our Tranformer-XL models for 1 epoch. We use the AdamW (Loshchilov & Hutter, 2017a) optimizer with a learning rate of 10^{-5} and weight decay with $\lambda = 10^{-5}$. With these hyperparameters, we are able to achieve a test accuracy of 97% on the 14 classes of DBpedia.

To prepare our membership-inference evaluation dataset, the WikiText-103 is partitioned into contiguous blocks, separated each by Wikipedia subsections. We then perform the same tokenization process as we do in finetuning before collecting their prediction vectors. Because we do not pretrain our own LLMs, we adapt **TMI** to train a single, global metaclassifier over the prediction vectors of all challenge points rather than train a metaclassifier per challenge point. In total, we use 2650 challenge points, which corresponds to a metaclassifier dataset with size $|D_{\text{meta}}| = 2560 * (\text{number of shadow models})$.

D. Discussion

Our Contributions. We summarize our main contributions to the study of membership-inference attacks as follows:

- We investigate privacy leakage in the transfer learning setting, where machine learning models are finetuned on downstream tasks with and without differential privacy.
- We introduce a new threat model, where the adversary only has query access to the finetuned target model.
- We propose a novel membership-inference attack, **TMI**, that leverages all of the information available to the black-box adversary to infer the membership status of individuals in the pretraining set of a finetuned machine learning model.
- We evaluate our attack on models trained on both vision and natural language tasks across multiple fine-tuning strategies. We show that there is privacy leakage even in cases where the target model was finetuned with differential privacy, and we show that our attack is effective on finetuned foundation models.

Other Privacy Attacks on Finetuned Models. We introduce the first threat model that uses query access to a finetuned model to mount a privacy attack on pretraining data. It remains an open question as to whether other privacy attacks, such as property inference, attribute inference, and training data extraction attacks can also see success in this transfer learning setting. Given that MI attacks are used as practical tools to measure or audit the privacy of machine learning models (Song & Shmatikov, 2018; ten; Ye et al., 2022), future work should consider efficiency and simplicity when designing new privacy attacks in the transfer learning setting.

Considerations for Private Machine Learning. Our evaluation shows that the pretraining dataset of machine learning models finetuned with differential privacy are still susceptible to privacy leakage. This supports the argument made in (Tramèr et al., 2022) that "privacy-preserving" models derived from large, pretrained models don't necessarily provide the privacy guarantees that consumers of services backed by these finetuned models would expect. Prior works that utilize public data to improve the utility of differentially private machine learning models have made strides towards making differential privacy practical for several deep learning tasks (Papernot et al., 2020; Yu et al., 2022; Li et al., 2022; Bu et al., 2023; He et al., 2022; Ganesh et al., 2023; Golatkar et al., 2022), but they do not address privacy risks external to model training itself.

Using **TMI** as a measurement of privacy leakage in this setting, we reinforce the fact that maintaining privacy depends on taking a holistic approach to the way that training data is handled. As stated in (Tramèr et al., 2022), privacy is not binary (i.e. not all data is either strictly "private" or "public") and privacy in machine learning is not only dependent on the model's training procedure. To grapple with privacy risk in this increasingly popular transfer learning setting, researchers and practitioners should explore new ways to sanitize sensitive information from training datasets of machine learning models, create ways to collect potentially sensitive Web data with informed consent from individuals, and work towards end-to-end privacy-preserving machine learning with high utility and privacy guarantees.