On the Privacy Risk of In-context Learning

Haonan Duan, Adam Dziedzic, Mohammad Yaghini
Nicolas Papernot, Franziska Boenisch
University of Toronto
Vector Institute
haonand@cs.toronto.edu

Abstract

Large language models (LLMs) are excellent few-shot learners. They can perform a wide variety of tasks purely based on natural language prompts provided to them. These prompts contain data of a specific downstream task—often the private dataset of a party, e.g., a company that wants to leverage the LLM on their purposes. We show that deploying prompted models presents a significant privacy risk for the data used within the prompt by instantiating a highly effective membership inference attack. We also observe that the privacy risk of prompted models exceeds fine-tuned models at the same utility levels. After identifying the model's sensitivity to their prompts—in form of a significantly higher prediction confidence on the prompted data—as a cause for the increased risk, we propose ensembling as a mitigation strategy. By aggregating over multiple different versions of a prompted model, membership inference risk can be decreased.

1 Introduction

Large language models (LLMs) exhibit strong capabilities for few-shot learning. When provided with a natural language prompt in the form of a small number of examples for the specific context, the models can perform a myriad of natural language downstream tasks without modifications of their parameters [3, 33]. Prompting is more parameter and data-efficient than fine-tuning. First, given a large number of parameters in LLMs, prompting boosts efficiency in downstream tasks [34] without any adaptation of model parameters. In contrast, fine-tuning requires retraining a significant fraction of parameters. Second, it has been shown that prompting can leverage training data more efficiently than standard fine-tuning with a prompt being worth ∼100 data points [35].

The effectiveness of prompts is possible since the prompt data exhibit a significant effect on the LLMs' behavior [19, 37]. This naturally raises the question of privacy risks. Understanding privacy risks of prompting is of high importance since, in contrast to the large *public* corpora used to pretrain the LLMs, the data used for prompting usually stems from a smaller *private* downstream dataset. Prior work has extensively studied the topic of memorization and privacy in LLMs [6, 7, 45]. Yet, the considerations were limited to the data used for pre-training the LLMs [6, 45] or to fine-tune the model parameters [23, 44, 46]. In contrast, we analyze how much privacy of the data used for prompting leaks from the deployed prompted LLM. With our results, we are the first to show that prompted LLMs exhibit a high risk to disclose the membership of their private prompt data.

In our study, we focus on text generation models [32, 33] prompted with a proper template for any given downstream classification task. In this setup, we study privacy leakage through the lense of membership inference attacks (MIA) [4, 38]—currently the most widely applied approach for estimating practical privacy leakage. With access only to the probability vector output by the prompted LLM for a given input, we instantiate the MIA to determine whether this input was part of the prompt. Our results suggest that data points used within the prompt are highly vulnerable to MIAs. Furthermore, in a controlled environment, we empirically evaluate the MIA-risk of prompting to the risk of fine-tuning with private data. We find that prompted models are more than five times more vulnerable than fine-tuned models.

The severe vulnerability of the private prompt data and the fact that finding the high-performing prompts for a given downstream task requires significant human efforts and computing resourses [51] demand the design of protection methods. Based on the observation that the prompted LLMs exhibit a significant higher prediction confidence on their prompted data—leading to the great success of MIA—we propose an effective defense: We show that by ensembling over different prompted versions of an LLM, we can align the prediction confidence on prompt data (members), and other data (non-members) while achieving the same high prediction accuracy. Obtaining such an ensemble of prompted models is efficient since multiple well-performing prompts is already the by-products of our prompt-tuning and does not require additional steps. We evaluate two concrete instantiations of prompt ensembling, namely Avg-Ens and Vote-Ens and quantify their effect on the risk of MIAs. We show that our ensembling effectively reduce the success of MIA to close to random guessing. Thereby, the privacy of the prompted data can be protected.

In summary, we make the following contributions:

- We instantiate the first MIA on prompted LLMs and show that we can effectively infer the membership of the prompted data points with high success.
- We empirically compare the MIA risk of prompted and fine-tuned models in a controlled experimental environment and observe that the privacy risk of prompting significantly outperforms the one of fine-tuning.
- We demonstrate how to mitigate the privacy leakage we observed with prompt ensembling to a MIA-success rate of close to random guessing.

2 Background and Related Work

2.1 Language Model Prompting

The success of LLMs such as the different versions of GPT [3, 32, 33] and their exceptional few-shot learning capacities gave rise to prompt-based learning. Without having to adapt any parameters, prompt-based learning leverages the capacities of LLMs and achieves similar downstream performance as full model fine-tuning [22, 25]. Therefore, it suffices to provide the model with a task-specific context in the form of a few examples, also called *demonstrations*. The prompt-based approach does improve computational and storage complexity over fine-tuning since no parameters of the underlying LLMs need to be updated and instead of having to save a fully fine-tuned model, only the required prompt has to be recorded. Prompts can be designed either manually by a human expert, or by an automated process [14, 15, 24, 37]. Our demonstrations come from the actual discrete vocabulary and we consider privacy leakage of the underlying data points – sentences from the downstream tasks used for prompting.

2.2 Memorization and Privacy Leakage in LLMs

LLMs are shown to memorize their training data which enables adversaries to extract this data when interacting with the model [5, 16, 20, 26, 42, 45]. It has, for example, been shown that GPT2 reproduces large passages with up to 1000 words of its original training data at inference time. Additionally, privacy risks through memorization in fine-tuning have been observed by Mireshghallah *et al.* in [27]. The only prior work around privacy leakage in prompt-based learning has used prompting to extract knowledge from LLMs and their underlying large (and often public) training corpora [10, 19, 30]. In our setup, we do not target the privacy of the LLM's training data—neither the original large corpora nor the data used to adapt the model through fine-tuning. Instead, we are the first to study the privacy of data used to prompt an LLM to perform particular downstream task.

2.3 Membership Inference Attacks

When performing a membership inference attack (MIA) [4, 38], an adversary aims to determine whether a particular data point was used to train a given machine learning model. The adversary usually has access only to the model's prediction outputs. Membership inference attacks have been successful on a broad variety of machine learning models and domains, especially the vision [38, 4] and language domain [36, 40, 7]. While a few prior works employ MIA to quantify memorization in LLMs [7, 29, 27], they target the original large corpus training data or data used for fine-tuning the parameters of the models. In contrast to them, we do not adapt the model parameters but freeze the entire LLM and design a prompt based on a small and private downstream dataset. We evaluate MIA risk for the data points used within the prompt.

2.4 Defending Against Membership Inference

Existing defense mechanisms against MIA can be divided into two main categories: (i) empirical measures to reduce the adversary's attack success by either reducing model overfitting [8] or perturbing model outputs [28, 18] and (ii) measures that rely

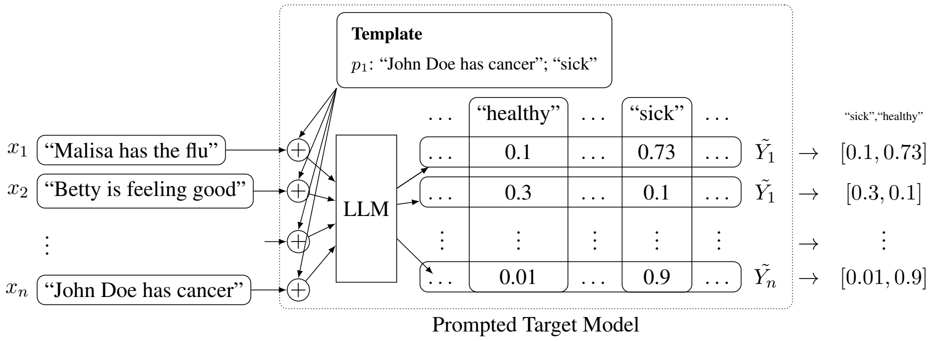


Figure 1: **Setup for Prompting and MIA.** We prompt the LLM with different prompts (same template) for a downstream task. The LLM returns per-token probabilities for the next token in the sequence. The adversary has query access to the prompted LLM and obtains prediction probabilities for each possible target class of the downstream task.

on providing rigorous privacy guarantees according to Differential Privacy (DP) [13]. These measure, for example, apply a DP stochastic gradient descent (DP-SGD) [1] while training a machine learning model. However, in practice, DP significantly degrades performance of generative models [2] when not trained under very carefully chosen hyperparameters [23]. Therefore, none of the popular state-of-the-art LLMs is trained with DP. As a consequence, we focus our work on the first category of defenses and propose ensembling multiple promoted models. The only prior work using ensembles to defend against MIA is limited to small ML models for vision and tabular datasets, and requires a pre-processing over the entire training data at inference time to determine which of the ensemble models' training data did not contain the present data point [41]. We instead query all prompted models without additional pre-processing.

3 Method

3.1 Prompting for Downstream Classification

We focus on prompting pre-trained LLMs with the objective to perform a downstream classification task. We denote the prompted model as L_{prompt}. Our prompts consist of tuples of demonstration sentences from the respective downstream task as prompt data, provided in a consistent template. When applied to a specific input x_i, L_{prompt} predicts an M-dimensional probability vector $\tilde{y}_i$, with M being the size of the vocabulary, where each component corresponds to the probability that the L_{prompt} assigns to the respective token for being the next token in the sequence x_i. Note that the output probabilities over all possible tokens are usually normalized such that $\sum_{m \in M} \tilde{y}_{i,m} = 1$. Since we provide the model with demonstrations to solve a downstream classification task, the index with the highest values in $\tilde{y}_i$ should correspond to the token that represents the class label of the x_i. For example on the input $x_i =$"The movie was great.", the highest probability should be for the token *"positive"*, because this is the correct class label. Given that the model is supposed to perform classification for a given downstream task, we assume that when querying L_{prompt} with x_i, not the entire $\tilde{y}_i$ has to be returned. Instead, we are only interested in a subset of token-probabilities, namely for those tokens that correspond to classes in the respective downstream dataset. We denote the reduced probability vector as y_i. Note that since y_i only consists of a subset of the token probabilities from $\tilde{y}_i$, the probabilities in y_i are unnormalized, *i.e.,* they do not necessarily add up to one, $\sum_{m \in M} y_{i,m} \leq 1$. We depict our setup in Figure 1.

3.2 MIA Setup and Threat Model

For our MIA, we assume an adversary with black-box access to the prompted model L_{prompt}. This adversary can query n text sequences $(x_1, \cdots, x_n)$ to L_{prompt} and obtains the output probability vectors $(y_1, \cdots, y_n)$. Following a line of prior MIAs [17, 43], we base our attack on the model's output probability at the token $y_{i,l}$ that corresponds to the correct target class label l.

3.3 Prompt Ensembling

To mitigate the privacy risk, as exposed by prompt membership, we propose to aggregate the prediction probability vectors over multiple independent prompted models into an ensemble prediction, as shown in Figure 2. We first tune K prompted mod-

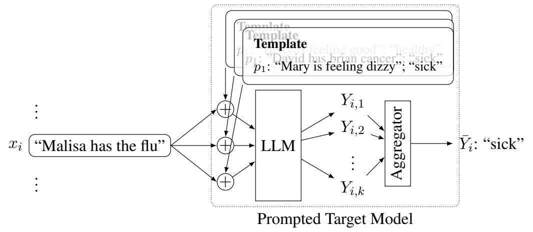


Figure 2: **Ensemble of Prompted Models**. We ensemble multiple prompted models with disjoint data and the same template. The final prediction is an aggregate of outputs from each prompted model.

els $L^{(1)}_{prompt}, L^{(2)}_{prompt}, \dots, L^{(K)}_{prompt}$. These K models are prompted with disjoint training data. We then introduce two standard techniques to ensemble these prompted models [19, 22] and refer to them as Avg-Ens and Vote-Ens.

In Avg-Ens, we average the raw probability vectors of each of our K prompted models $L^{(1)}_{prompt}, L^{(2)}_{prompt}, \dots, L^{(K)}_{prompt}$. Let $y^{(k)}_i$ be the output of our kth prompted model on input x_i. The output of the ensemble $L^{\text{Avg-Ens}}_{prompt}$ on input x_i is obtained as follows

$$L^{\text{Avg-Ens}}_{prompt}(x_i) := \frac{1}{K}\sum_{k=1}^{K} L^{(k)}_{prompt}(x_i). \tag{1}$$

For Vote-Ens, we rely on a majority vote of all the prompted models. Therefore, we first obtain a single model's prediction on input x_i as the token (class) from vocabulary $\mathcal{V}$ with the highest logit value as $\arg\max(L^{(k)}_{prompt}(x_i))$. Let n_v denote the number of prompted models that predict token v. Then, we return the token predicted by most models as

$$L^{\text{Vote-Ens}}_{prompt}(x_i) := \arg\max_{v \in \mathcal{V}} (n_v)\,. \tag{2}$$

We do not evaluate the ensembling methods from Jiang *et al.* [19] that rely on (i) using the prompted model with the highest test accuracy as the output of the ensemble, or (ii) using a weighted average over the prompted models. While (i) might yield utility improvements as shown in [19], it does not provide any privacy protection to the prompted model whose output is returned. This is because the prediction of the ensemble still depends solely on a single model and thereby puts the privacy of that model at risk. Since [19] shows for (ii) that the weight concentrates on one single prompted model, the same impact on this model's privacy holds.

	N_{train}	N_{test}	# Classes	min_{acc}	max_{acc}
agnews	12000	7600	4	0.65	0.83
cb	250	56	3	0.60	0.73
sst2	6920	1821	2	0.78	0.88
trec	5452	500	6	0.40	0.59

Table 1: **Evaluation Datasets.** Summary of the datasets and utility overview. We depict the number of training (N_{train}) and test data points (N_{test}), and the number of classes (# Classes) in the task. Additionally, among the 50 selected best prompted LLMs, we report the span of their respective validation accuracies between the worst performing min_{acc} and the best performing max_{acc}. The validation accuracy is used to find the best 50 prompted models among the 1000 generated promoted models.

MIA on Ensembled Models. We also perform MIA on the ensembled models to study how ensembling mitigates the privacy risks. For Avg-Ens, we rely on the averaged output vector of the ensemble $y^{\text{Avg-Ens}}_i = L^{\text{Avg-Ens}}_{prompt}(x_i)$, and extract the respective confidence value at the correct target class $y^{\text{Avg-Ens}}_{i,l}$. For Vote-Ens, we count the number of prompted models that predict the target class l and divide by the total number of models in the ensemble as $\frac{n_l}{K}$. Our empirical evaluation on the privacy risk mitigation through ensembled prompts is presented in Section 4.4.

4 Experimental Evaluation

We experimentally study the MIA success on prompted LLMs and show that the prompted data exhibits a high vulnerability to MIAs. Furthermore, we provide a comparison to the privacy risk of fine-tuning. We find that, at the same downstream accuracy, the privacy risk of prompt data in a prompted LLM surpasses the one of data used for fine-tuning. Finally, we demonstrate how ensembling the prediction of multiple prompted LLMs can effectively reduce the MIA risk close to random guessing.

4.1 Experimental Setup

We prompt GPT2 [3][1] to solve four standard downstream text classification tasks, namely *agnews* [48], *cb* [11], *sst2* [39] and *rte* [9]. We document details of the datasets in Table 1. Note that 20% of the training data sets serve us as separate validation sets.

[1] If not specified differently, we use GPT2-xl taken from HuggingFace (1.5 billion parameters).

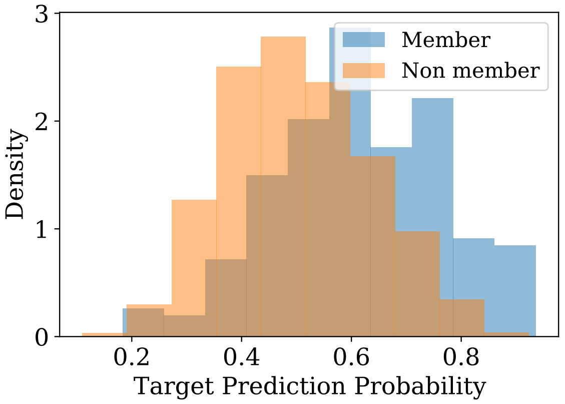


Figure 3: **Prediction Probability at Target Class (sst2).** We plot output prediction probability for the target class for member and non-member data points of the prompt in the prompted LLM. We find that the LLMs outputs for the prompt's member data is significantly higher than for non-member data points.

Tuning. Our procedures for prompt tuning follow Zhao *et al.* [49] unless otherwise specified. Unlike them, we also return the probabilities for class labels whose corresponding tokens do not fall under the top 100 tokens. This enables us to perform our MIA in a unified way over all model outputs.

Since the performance of prompted models is known to suffer from instability [12, 47], we prompt the model 1000 times with different 4-shot data from the downstream dataset. We then keep the 50 best-performing prompts with disjoint data based on validation accuracy.[2] The range of validation accuracies of the best selected 50 prompted models is reported in Table 1.

MIA. To evaluate our MIAs, we consider the data points used within the prompt of a model as members and all remaining training data points from the respective dataset as non-members. This skewed distribution between members and non-members corresponds to a realistic scenario where only a small proportion of the candidate data targeted by the adversary are members [17]. If not stated otherwise, we perform MIA on the unnormalized probability outputs of the prompted LLMs at the data point's correct target class. To quantify the success of our attack, we report the AUC score as well as the true-positive rate (TPR) at low false-positive rates (FPRs). A successful MIA should have a high AUC score as well as a high TPR at low FPRs.

4.2 Success of Membership Inference Attack

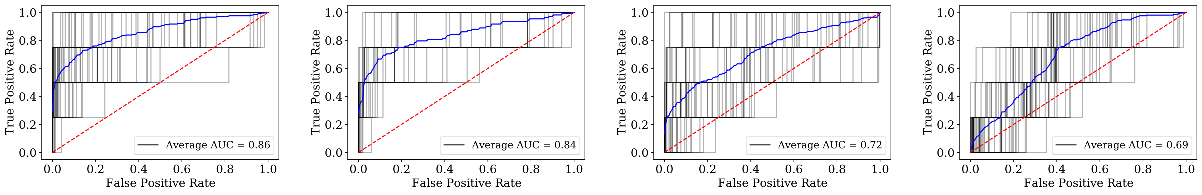
We first analyze the probability output by the prompted LLM for the correct target class between member and non-member data points. Figure 3 shows for the sst2 dataset that the prediction outputs for non-members are overall lower than for members. Similar results can be observed on all evaluation datasets, see Figure 9 in Appendix B.

This difference leads to a high MIA risk for the prompted data points as we show in Table 2 and Figure 4. For example, on the sst2 dataset, on an FPR of $1e-3$, we observe a TPR of 0.137 ± 0.187, and an average AUC of 0.72. Note that the current most powerful MIA for supervised classification [4] obtains the same high AUC (0.72) score on the CIFAR10 dataset only by fully training 256 additional shadow models—a significant computational overhead we do not face.

Membership Risk is Higher on Smaller Models. We evaluate the impact of underlying LLMs' size on the vulnerability to MIAs against their prompted data. In this comparison, we focus on GPT2-base vs GPT2-xl. GPT2-base has 117M parameters, while GPT2-xl has 1.5B. For a fair comparison between the different models' vulnerability, we control the downstream performance of two models. Therefore, for GPT2-xl, we again generate 1000 prompted models. Among those, we keep the 50 prompts that lead to a performance close to the validation accuracy of the best prompted GPT2-base. More precisely, we choose the 50 prompts for GPT2-xl that have a validation accuracy in the range of the 50 best models of GPT2-base. Figure 5 depicts the membership risk of prompted models of different sizes by depicting the TPRs at a FPR of 0.001. We find that GPT2-base consistently yields higher TPRs (*i.e.,* higher membership risk) than GPT2-xl across different datasets. We hypothesize that this disparate vulnerability is caused by larger models' better generalization capacity. Larger models, when prompted with a few examples, due to their better generalization, have a smaller difference in output distribution between member and non-member data points.

Normalizing Prediction Probabilities. As we detail in Section 3.1, the prediction probabilities of the prompted model do not sum up to one since they are a only a small subset of all possible output tokens (whose total prediction probability sums up to one). We evaluate how normalizing the model's

[2] This type of prompt engineering corresponds to choosing the model with the best hyperparameters in standard training or fine-tuning.



(a) agnews (b) cb (c) sst2 (d) trec

Figure 4: **MIA risk over all Datasets.** We depict the AUC-ROC curves over all datasets. The red dashed line represents the MIA success of random guessing. Each gray line corresponds to a prompted model with its four member data points. Due to the small number of member data points (4), our resulting TPRs can only be 0% 25%, 50%, or 100% which leads to the step-shape of the gray curves. The reported average AUC-score is calculated as an average over the individual prompted models (gray lines)' AUC score. Additionally, for visualization purposes, we average the gray lines over all prompted models and depict the average as the blue line. We use 50 prompted models in this experiment.

	FPR=1e−3		FPR=1e−2		FPR=1e−1	
	Prompts	Fine-tuning	Prompts	Fine-tuning	Prompts	Fine-tuning
agnews	0.222 ± 0.212	0.001 ± 0.001	0.433 ± 0.281	0.011 ± 0.005	0.661 ± 0.253	0.105 ± 0.010
cb	0.272 ± 0.204	0.051 ± 0.071	0.382 ± 0.236	0.111 ± 0.120	0.632 ± 0.212	0.325 ± 0.181
sst2	0.137 ± 0.187	0.002 ± 0.003	0.225 ± 0.206	0.018 ± 0.009	0.402 ± 0.297	0.167 ± 0.0312
trec	0.019 ± 0.067	0.003 ± 0.012	0.049 ± 0.091	0.023 ± 0.038	0.221 ± 0.201	0.258 ± 0.102

Table 2: **TPR at at Different FPRs for Prompts and Fine-Tuning.** We report the TPR of our MIA at different low FPRs. The large standard deviation results from the small number of member data points (4). We only consider FPRs down to $1e{-}3$ which is larger than in [4] which considers FPRs down to $1e{-}5$. This is because we operate on much smaller datasets where we cannot obtain such small fractions.

output probabilities over all possible target classes in the downstream task influences the risk of MIA. We depict our results in Figure 17 in Appendix B. The evaluation does not yield a consistent trend regarding the overall AUC among the datasets: while for argnews and trec the average AUC is similar with and without normalization, for cb and sst2, the raw outputs yield higher AUC. These results suggest that attackers can also perform successful MIA when the prediction outputs are processed in different ways—as it can happen when the prompted models are deployed behind some API.

4.3 Prompting Leaks more Privacy than Fine-Tuning

In this section, we compare the privacy leakage of prompting with fine-tuning.

Fine-Tuning Setup. Over all our experiments, we fine-tune only the last layer of GPT2 and a classification head. We fine-tune the model for 500 epochs, and use the checkpoint with the highest validation accuracy during tuning. For a controlled comparison between fine-tuning and prompting, our fine-tuned model's validation accuracy should roughly match the one of our prompted models.

Therefore, we first identify the number of data points needed for each downstream dataset to yield comparable validation accuracy.[3] We run 100 fine tuning runs for each combination of the number of training data points $(4, 8, 32, 64, 128, 256)$ and learning rates $(1e{-}4, 1e{-}5, 1e{-}6)$. The number of data points needed and the corresponding learning rates are detailed in the table below:

Dataset	#Data Points	Learning Rate	Acc.
agnews	512	$1e-5$	0.74
cb	16	$1e-4$	0.68
sst2	5536	$1e-5$	0.74
trec	32	$1e-4$	0.52

Table 3: **Learning Parameters for Fine-Tuning.** We present the number of data points used for fine-tuning, the learning rates, and the resulting validation accuracies of our fine-tuning for all dataset.

Note that for sst2, we were not able to meet the prompted models' validation accuracy (Table 1) even using the whole training dataset. Therefore, we compare with weaker prompts that yield accuracy between 0.72 and 0.76—instead of our 50 best selected ones.

[3] Fine-tuning usually requires more data points than prompting [21].

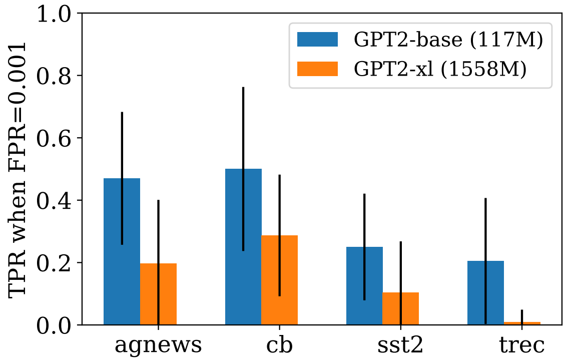


Figure 5: **Impact of Model Size on Membership Risk.** We report the TPR at FPR $1e-3$ for GPT2-base and GPT2-xl (117M vs 1.5B parameters). For fair comparison, we tune 1000 prompts for both architectures, keep the best 50 for GPT2-base, and for GPT2-xl, we keep the 50 prompts that yield validation accuracy closest to the one of GPT2-base. We observe that larger models leak less private information about their prompts. All results are obtained on the sst2 dataset.

	$mean_{\text{acc}}$	Avg-Ens	Vote-Ens
agnews	0.734	0.822	0.794
cb	0.625	0.696	0.696
sst2	0.854	0.904	0.908
trec	0.406	0.520	0.500

Table 4: **Test Accuracy of Ensembles.** We depict the validation accuracies of our initial prompted models (mean over all 50 models) and the validation accuracies of our ensembling methods Avg-Ens and Vote-Ens.

MIA Evaluation Setup. Due to the different training set size in prompting and fine-tuning, for a fair comparison, we evaluate MIA for fine-tuned models in two ways: (1) Following the setup for prompted models we select a different 4-tuple of members and evaluate against all the non-members from the validation set. This procedure is repeated five times and we report the average over all resulting curves ROC curves and the average AUC. (2) Following the standard setup for MIA [38] , we evaluate all the members together against the non-members and present the resulting ROC curve and AUC score.

Results. We present the MIA of fine-tuned models in Table 2 and Table 6. Our findings highlight that prompting yields higher privacy risks than fine-tuning under similar downstream performance. For example, at an FPR of $1e-3$, the average TPR for prompting is at least five times higher than for fine-tuning across all datasets.

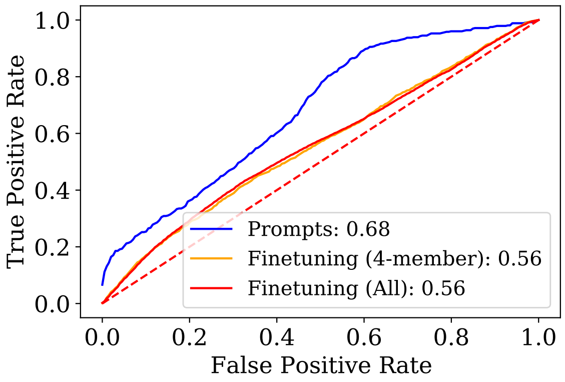


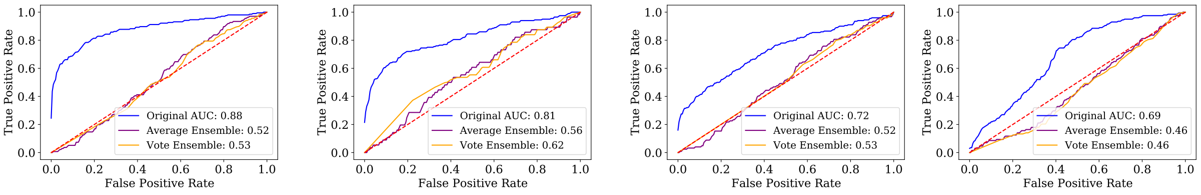
Figure 6: **Privacy Leakage in Fine-Tuning vs. Prompting (sst2).** We plot the membership risk of our MIA on prompted and fine-tuned models given similar downstream performance. For fine-tuning, we evaluate MIA risk in two different ways to avoid the influence of different training set size. The red dashed line represents the MIA success of random guessing. The results show that prompts are much more vulnerable to MIA than fine-tuning. Results of more datasets can be found in Figure 16.

4.4 Ensembling Mitigates Privacy Risks

Finally, we experimentally evaluate the impact of our two ensembling approaches on the membership risk. We report performance of our ensembles on the test data in Table 4 and observe that both approaches perform equally well.

To study the impact on privacy risk, we first analyze the distribution of member and non-member data points' probability at the target class for Avg-Ens. Figure 11 in Appendix B highlights that through ensembling, the distributions for member and non-member probabilities become much more similar. This also reflects in reduced membership risk as we depict in Figure 7. We find that for both methods, the attack curve after ensembling is close to random guessing (red line) across all datasets. Similar results are obtained with Vote-Ens as we show in Figure 13 and Figure 14 in Appendix B.

Finally, we evaluate the influence of the number of prompted model in the ensemble on the resulting membership risk. Figure 8 highlights that with an increasing number of prompted models in the ensemble, privacy risk decreases. This effect results from the fact that averaging over more models generally implies smaller influence of one particular model. However, there is a trade-off between increased inference times and the decreased privacy costs of using larger ensembles. Our Figure 18 in Appendix B suggest that using as little as 16 teach-



(a) agnews

(b) cb

(c) sst2

(d) trec

Figure 7: **Defense Method via Ensembling.** We depict the AUC-ROC curves over 4 datasets for our two ensembling defense methods (average, Avg-Ens, and vote ensembles, Vote-Ens) and compare it with the attack against the undefended model (blue solid line). The red dashed line represents random guessing. We find that ensembling effectively mitigates the threats of MIAs.

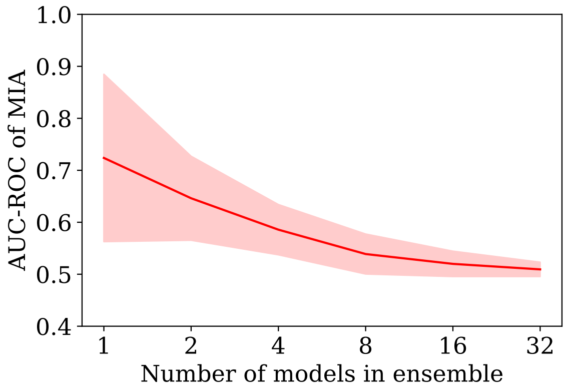


Figure 8: **MIA Risk vs Number of Models in an Ensemble (sst2).** We plot the membership risk in form of the AUC score of MIA while we vary the number of teachers in ensembling. Results of other datasets can be found in Figure 18. We observe that with more data used for ensembling, the lower risk of MIA (in terms of AUC and its variance).

ers could reduce the average MIA for all datasets below 0.55, *i.e.,* close to random guessing.

5 Conclusions

We are the first to show that prompted LLMs exhibit a high risk to disclose the membership of their private prompt data. To determine the membership of a data point, it is sufficient for an attacker to analyze the model's prediction confidence at the target class. When comparing the privacy risk of prompted models with standard fine-tuning, we observe that prompts exhibit a higher privacy leakage than fine-tuning. However, there are many advantages of prompts over fine-tuning. For example, instead of storing multiple versions of the whole fine-tuned model per downstream task, the underlying LLMs stay intact while only the prompt changes to implement different tasks. Thus, to mitigate privacy risks for prompts, we propose ensembling over multiple prompted models. We experimentally validate that this approach reduces the membership risk of the prompt data. An interesting observation is that privacy leakage also decreases with the increasing number of language model parameters. This suggests a general trend that the prompt data become less vulnerable to privacy risks with a better generalization of the models.

Limitations

While our ensembling approach empirically mitigates the risk of MIAs against prompted LLMs, we acknowledge that the approach does not provide rigorous privacy guarantees. Future effort should be put into extending our approach to implement, for example, differential privacy [13].

Furthermore, we acknowledge that our ensembling approach creates computational overhead since inference needs to be run with multiple prompts instead of a single one. This disadvantage can be reduced by running inference over all the prompts in a batch.

In this work, we solely consider discrete prompts due to their popular usage. There exist also *soft prompts* [31, 50] that are optimized sequences of continuous task-specific input vectors. They are not tied to embeddings from the vocabulary. The privacy leakage of soft prompts and designing potential defenses will be addressed in our future work.

Finally, due to the cost associated with access to GPT-3, we limit our empirical evaluations to GPT-2 which is available as an open-source model. To reduce potential biases that might arise through this limitation, we evaluated on different versions of GPT-2, including GPT2-xl, which has >1.5B parameters.

References

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

[2] R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[5] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[6] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267, 2019.

[7] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[8] D. Chen, N. Yu, and M. Fritz. Relaxloss: Defending membership inference attacks without losing utility. In *International Conference on Learning Representations*, 2022.

[9] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment: First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, pages 177–190. Springer, 2006.

[10] J. Davison, J. Feldman, and A. M. Rush. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1173–1178, 2019.

[11] M.-C. De Marneffe, M. Simons, and J. Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124, 2019.

[12] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

[13] C. Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

[14] T. Gao, A. Fisch, and D. Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

[15] H. Guo, B. Tan, Z. Liu, E. Xing, and Z. Hu. Efficient (soft) q-learning for text generation with limited good data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6969–6991, 2022.

[16] D. Ippolito, F. Tramèr, M. Nasr, C. Zhang, M. Jagielski, K. Lee, C. A. Choquette-Choo, and N. Carlini. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

[17] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans. Revisiting membership inference under realistic assumptions. *Proceedings on Privacy Enhancing Technologies*, 2021(2), 2021.

[18] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 259–274, 2019.

[19] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[20] E. Kharitonov, M. Baroni, and D. Hupkes. How bpe affects memorization in transformers. *arXiv preprint arXiv:2110.02782*, 2021.

[21] T. Le Scao and A. M. Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, 2021.

[22] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Nov. 2021.

[23] X. Li, F. Tramer, P. Liang, and T. Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022.

[24] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

[25] X. Liu, K. Ji, Y. Fu, Z. Du, Z. Yang, and J. Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[26] R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, and A. Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *arXiv preprint arXiv:2111.09509*, 2021.

[27] F. Mireshghallah, A. Uniyal, T. Wang, D. Evans, and T. Berg-Kirkpatrick. Memorization in nlp fine-tuning methods. *arXiv preprint arXiv:2205.12506*, 2022.

[28] R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[29] M. G. Oh, L. H. Park, J. Kim, J. Park, and T. Kwon. Membership inference attacks with token-level deduplication on korean language models. *IEEE Access*, 11:10207–10217, 2023.

[30] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[31] G. Qin and J. Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.

[32] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[34] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[35] T. L. Scao and A. M. Rush. How many data points is a prompt worth? *arXiv preprint arXiv:2103.08493*, 2021.

[36] V. Shejwalkar, H. A. Inan, A. Houmansadr, and R. Sim. Membership inference attacks against NLP classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[37] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

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

[39] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[40] C. Song and V. Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[41] X. Tang, S. Mahloujifar, L. Song, V. Shejwalkar, M. Nasr, A. Houmansadr, and P. Mittal. Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1433–1450, 2022.

[42] K. Tirumala, A. H. Markosyan, L. Zettlemoyer, and A. Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *arXiv preprint arXiv:2205.10770*, 2022.

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

[44] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022.

[45] C. Zhang, D. Ippolito, K. Lee, M. Jagielski, F. Tramèr, and N. Carlini. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*, 2021.

[46] R. Zhang, S. Hidano, and F. Koushanfar. Text revealer: Private text reconstruction via model inversion attacks against transformers. *arXiv preprint arXiv:2209.10505*, 2022.

[47] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*.

[48] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[49] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021.

[50] Z. Zhong, D. Friedman, and D. Chen. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*, 2021.

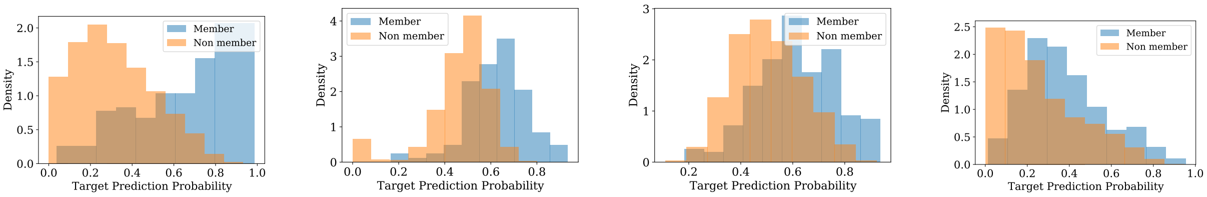
[51] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023.

A Broader Impact and Ethics Statement

Prompting is on the way of becoming a highly prominent paradigm of using LLMs—which makes assuring the privacy of the prompt data an urgent need. We present an empirical yet efficient mitigation of privacy risks but we acknowledge that this approach does not yield formal privacy guarantees. Therefore, we encourage model owners to use our MIA as a tool to to empirically evaluate the privacy of their prompted model, or their ensemble of prompted models, before deployment. A high MIA score should galvanize the model owners to implement additional protection before the deployment.

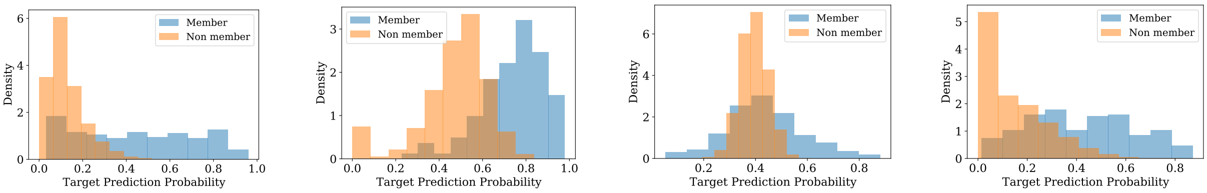
By relying purely on open-source LLMs and public open source datasets in our experimental evaluation, we make sure that the result reported in the current work do not harm individuals' privacy. We also recognize the importance of transparency in machine learning research, and we have made efforts to provide clear explanations of our methods and results, and provide additional experimental results on multiple datasets in the Appendix.

B Additional Experimental Results



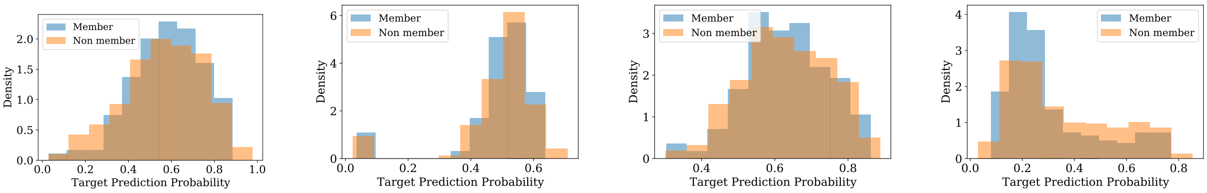
(a) agnews (b) cb (c) sst2 (d) trec

Figure 9: **Output Probabilities at the Target Class for Members and Non-Members.** We depict the probability of the prompted GPT2-xl on the correct target class for member and non-member data points.



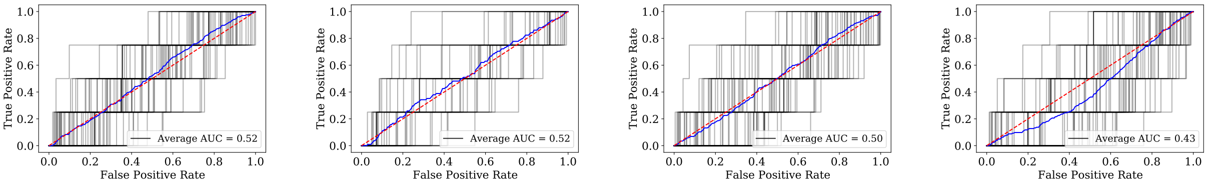
(a) agnews (b) cb (c) sst2 (d) trec

Figure 10: **Output Probabilities at the Target Class for Members and Non-Members for GPT2-base.** We depict the probability of the ensemble of prompted GPT2-base on the correct target class for member and non-member data points.



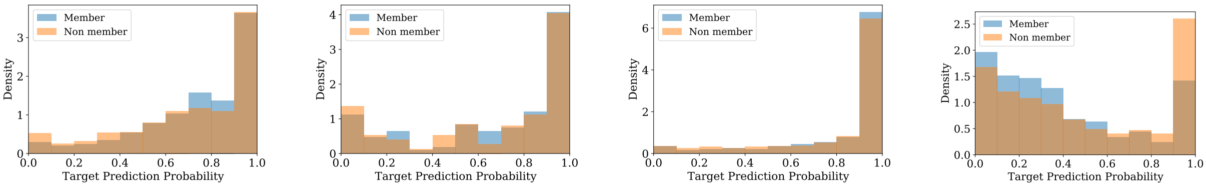
(a) agnews (b) cb (c) sst2 (d) trec

Figure 11: **Output Probabilities at the Target Class for Members and Non-Members under Avg-Ens.** We depict the probability of the ensemble of prompted GPT2-xl on the correct target class for member and non-member data points. We perform ensembling by aggregating the raw output probabilities over 50 prompted models and computing the average output vector. We find that the discrepancy between member and non-member becomes much smaller after ensembling.



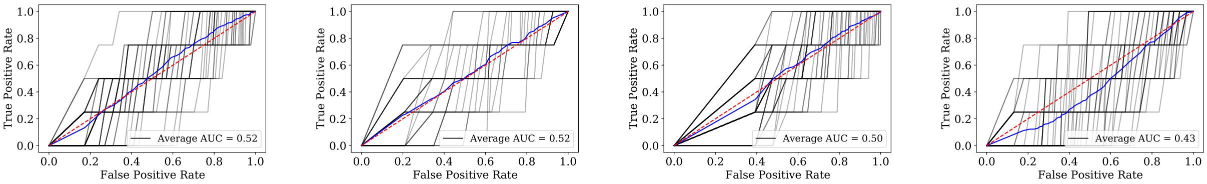
(a) agnews (b) cb (c) sst2 (d) trec

Figure 12: **MIA risk over all Datasets after (Avg-Ens).** We depict the AUC-ROC of MIA after Avg-Ens. Across all datasets, the effectiveness of MIA (blue line) is close to random guessing (red line).



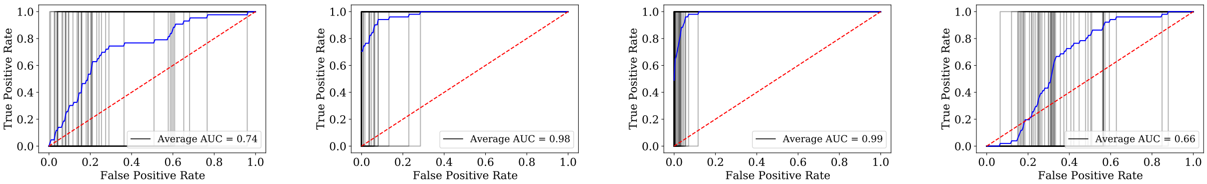
(a) agnews (b) cb (c) sst2 (d) trec

Figure 13: **Voting Probabilities for the Correct Target Class with Vote-Ens.** We ensemble the individual prompted models by obtaining the class with the highest prediction probability from each model. We show for member and non-member data points what percentage of the 50 prompted models returns the correct target class. This corresponds to the confidence of the ensemble.



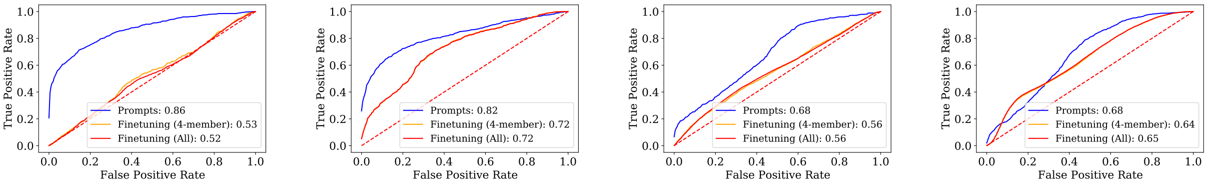
(a) agnews (b) cb (c) sst2 (d) trec

Figure 14: **MIA Risk over all Datasets (Vote-Ens).** We depict the AUC-ROC of MIA after Vote-Ens. Across all datasets, the effectiveness of MIA (blue line) is close to random guessing (red line).



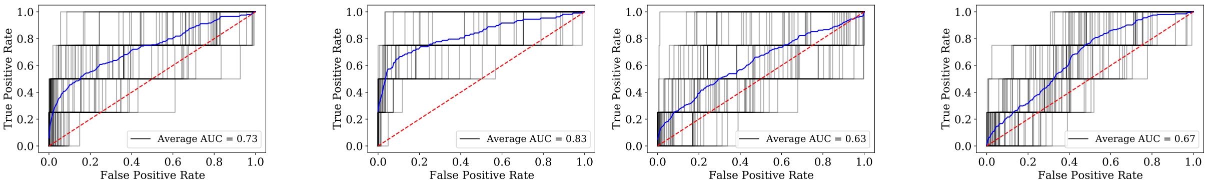
(a) agnews (b) cb (c) sst2 (d) trec

Figure 15: **MIA Risk over all Datasets for One-Shot Learning.** This figure corresponds to Figure 4 with the difference that we only use one example (instead of four) in the prompt. We depict the AUC-ROC curves over all datasets. The red dashed line represents the MIA success of random guessing. Each gray line corresponds to a prompted model with its four member data points. Due to the small number of member data points (1), our resulting TPRs can only be 0% or 100% which leads to the step-shape of the gray curves. The reported average AUC-score is calculated as an average over the individual prompted models (gray lines)' AUC score. Additionally, for visualization purposes, we average the gray lines over all prompted models and depict the average as the blue line. We use 50 prompted models in this experiment..



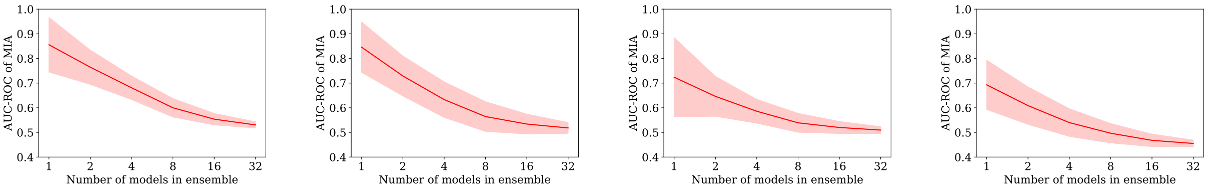
(a) agnews (b) cb (c) sst2 (d) trec

Figure 16: **MIA on Fine-Tuning vs Prompting across all Datasets.** We plot our MIA risk on prompted and fine-tuned models given similar downstream performance. For fine-tuning, we evaluate MIA risk in two different ways to avoid the influence of different training set size. The red dashed line represents the MIA success of random guessing. The results show that prompts are much more vulnerable to MIA than fine-tuning.



(a) agnews: normalized (b) cb: normalized (c) sst2: normalized (d) trec: normalized

Figure 17: **Impact of Normalization.** We report the AUC for our MIA on prompted GPT2-xl for normalized outputs, *i.e.,* outputs where the probabilities over all target classes of the respective downstream task add up to one.



(a) agnews (b) cb (c) sst2 (d) trec

Figure 18: **Number of teachers in average ensemble vs MIA risks.** We plot the membership risk in form of the AUC score of MIA while we vary the number of teachers in ensembling. We observe that with more data used for ensembling, the lower risk of MIA (in terms of AUC and its variance).