LLM Privacy

Angelo Benoit

Review of:

Extracting Training Data from Large Language Models

by Carlini, Tramer, Wallace, Jagielski, Herbert-Voss, Lee, Roberts, Brown, Song, Erlingsson, Oprea, Raffel

Large Language Models Can Be Strong Differentially Private Learners

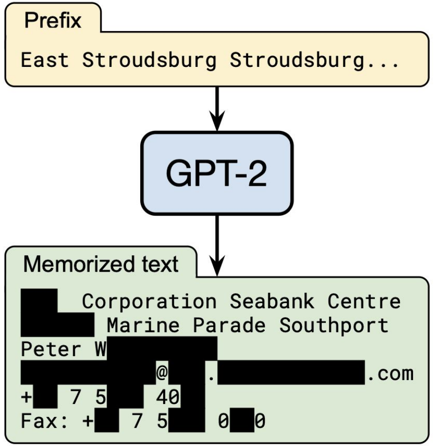
by Li, Tramèr, Liang, Hashimoto

Extracting Training Data from Large Language Models

by Carlini, Tramer, Wallace, Jagielski, Herbert-Voss, Lee, Roberts, Brown, Song, Erlingsson, Oprea, Raffel

Background

- **Data Leakage Concern:** The vast amounts of data used by LLMs raise concerns about their ability to retain and expose sensitive information, posing a significant privacy risk
- **Privacy is Critical:** Privacy in LLMs is critical as these models train on GBs of data, especially in private models
- **Data Extraction Susceptibility:** Data extraction attacks can expose inadvertently memorized training data inputs, leading to potential privacy breaches



Challenge and Contribution

- **Challenge:** Other papers exploring this topic fail to accurately understand the ability of LLM memorization
    - problematic test design (e.g. using canary tokens)
    - misunderstanding of overfitting and generalization

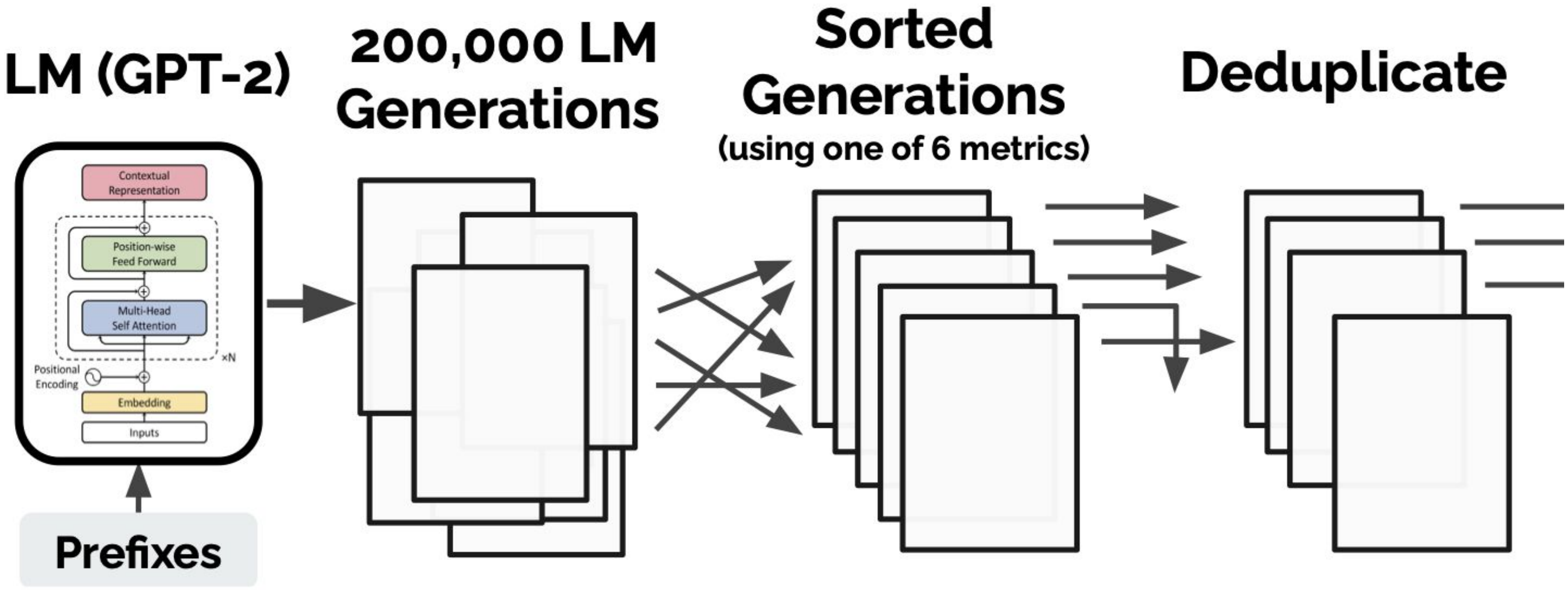
- **Contributions:**
    1. Demonstrates the feasibility and extent of training data extraction attacks on LLMs
    2. Suggests practical mitigation strategies to minimize privacy risks associated with these models

The Attack Model

- **Generate High-Likelihood Samples:** Create numerous outputs from the model, focusing on highly likely memorized sequences
- **Use a Reference Model for Ranking:** Employ a secondary model to identify unique high-likelihood samples as potentially memorized data
- **GOAL: Extract Verbatim Text Sequences:** Aim to pull exact sequences from the training data, particularly sensitive information

- Only requires the ability to query the model, without access to its internals

Training Data Extraction Attack



Text Generation

- **Autoregressive Generation:** Utilizes autoregressive text generation for creating sequential text predictions, mimicking training data patterns

- **Top-k Sampling:** For each new token, the model considers only the top *k* most probable next tokens and samples from this subset according to their probabilities to introduce diversity

- **Baseline:** Extract exactly 256 tokens for each trial using the top-*k* strategy with $k = 40$

Definition 1 (Model Knowledge Extraction) *A string s is* extractable[4] *from an LM f_θ if there exists a prefix c such that:*

$$s \leftarrow \underset{s' : |s'|=N}{\arg\max} f_\theta(s' \mid c)$$

Text Generation - Improved methods

- **Decaying Temperature:**
  - Make model produce more varied output at beginning to diversify example output
  - Reduce temperature to make model more confident as tokens are generated

$$\text{softmax}(z) \text{ with } \text{softmax}(z/t), \text{ for } t > 1$$

- **Internet Text:**
  - Sample 5-10 prefix tokens from web scrapes and then continue with baseline top-k

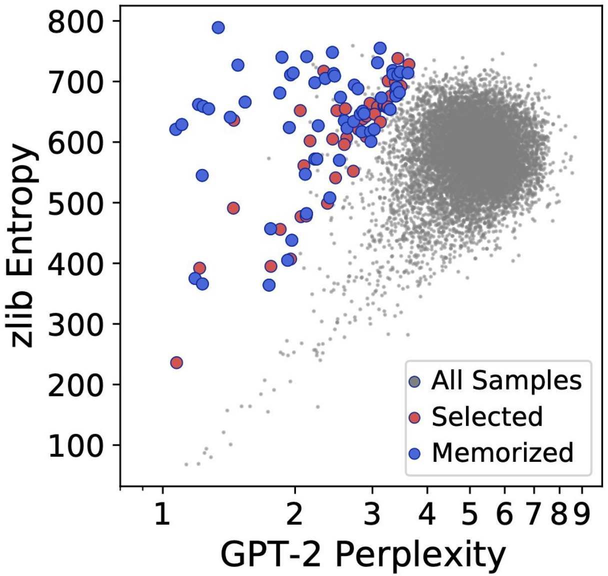
Identifying Memorized Content (Membership Inference)

- **Likelihood Comparison:** Use the likelihood of each sample being a genuine output
    - sample perplexity
- **Reference Model Ranking:** Samples that are highly likely according to the original model but not as likely according to the reference model are flagged as potentially memorized
- **Sample Selection:** The generated samples are selected based on likelihood and the reference model ranking

Reference Models

- **Second Neural LM:**
  - Using a smaller model (GPT small and GPT medium)

- **Zlib compression:**
  - compare GPT-2 perplexity to zlib entropy
  - high entropy implies more diverse words and phrases (more likely memorized)



Reference Models

- **Lowercase Text:**
  - perplexities can vary if memorized content is case specific

- **Perplexity on Sliding Window:**
  - find minimum perplexity in window of 50 tokens
  - identify memorized examples surrounded by non-memorized text

Experiment Results

- **High Candidate Sample Memorization Rate**: 604 of the 1800 generated samples memorized

Category	Count
US and international news	109
Log files and error reports	79
License, terms of use, copyright notices	54
Lists of named items (games, countries, etc.)	54
Forum or Wiki entry	53
Valid URLs	50
Named individuals (non-news samples only)	46
Promotional content (products, subscriptions, etc.)	45
High entropy (UUIDs, base64 data)	35
Contact info (address, email, phone, twitter, etc.)	32
Code	31
Configuration files	30
Religious texts	25
Pseudonyms	15
Donald Trump tweets and quotes	12
Web forms (menu items, instructions, etc.)	11
Tech news	11
Lists of numbers (dates, sequences, etc.)	10

Memorization Results

Results for each of 3 generation methods

Category	Count
US and international news	88
Forum or Wiki entry	34
License, terms of use, copyright notice	28
Named individuals	25
Promotional content	18
Lists of named items	15
Contact info	20
Donald Trump tweets and quotes	12
Pseudonyms	7
Valid URLs	7
Sports news	6
Movie synopsis or cast	6

(a) Top-*n* (191 samples)

Category	Count
Log files and error reports	86
Lists of named items	53
Valid URLs	40
License, terms of use, copyright notice	36
High entropy	33
Configuration files	32
Code	29
Named individuals	18
Promotional content	14
Contact info	12
Pseudonyms	11
Forum or Wiki entry	9
US and international news	7
Tech news	7
Pornography	5
Web forms	5
Lists of numbers	5

(b) Internet (273 samples)

Category	Count
US and international news	31
Religious texts	28
License, terms of use, copyright notice	24
Promotional content	20
Forum or Wiki entry	17
Named individuals	12
Lists of named items	12
Valid URLs	12
Tech news	8
Contact info	8
High entropy	6
Lists of numbers	6

(c) Temperature (140 samples)

Memorization Results

Results for each of 6 inference comparison metrics

Category	Count
License, terms of use, copyright notice	11
Lists of named items	8
Log files and error reports	7
Valid URLs	6
Lists of numbers	5

(a) Perplexity (51 samples)

Category	Count
US and international news	21
Lists of named items	18
License, terms of use, copyright notice	16
Promotional content	11
Valid URLs	11
Log files and error reports	10
Named individuals	8
High entropy	8
Forum or Wiki entry	7
Configuration files	6
Code	6

(b) Window (119 samples)

Category	Count
US and international news	40
License, terms of use, copyright notice	31
Lists of named items	17
Forum or Wiki entry	14
Named individuals	13
Promotional content	13
Contact info	12
Log files and error reports	11
Valid URLs	10
Code	10
Tech news	6
Configuration files	6
Pseudonyms	5

(c) zlib (172 samples)

Category	Count
US and international news	39
Log files and error reports	29
Lists of named items	17
Forum or Wiki entry	12
Named individuals	11
License, terms of use, copyright notice	10
High entropy	9
Configuration files	6
Promotional content	5
Tech news	5

(d) Lowercase (135 samples)

Category	Count
Log files and error reports	17
Forum or Wiki entry	15
Religious texts	14
Valid URLs	13
High entropy	13
Lists of named items	12
License, terms of use, copyright notice	12
Promotional content	11
Configuration files	11
Named individuals	11
other	9
US and international news	9
Contact info	8
Donald Trump tweets and quotes	7
Code	6

(e) Small (141 samples)

Category	Count
Valid URLs	17
Log files and error reports	14
US and international news	13
Contact info	12
Religious texts	12
Named individuals	11
Promotional content	11
High entropy	10
Forum or Wiki entry	9
Lists of named items	8
License, terms of use, copyright notice	8
Code	5
Donald Trump tweets and quotes	5

(f) Medium (116 samples)

Observations

- **Most Effective Attack:** Utilizing internet prompts for the prefix and zlib compression as a reference metric yielded 67% TPR for candidate samples memorized
    - Average is 33.5% TPR over all methods

- **Memorization is not Overfitting:** Unlike other models, LMs do not experience overfitting but can still recall specific examples
        - validation and train loss are comparable on average, ~10% difference

Observations

- **Memorization is Context Dependent:** The LM prompt and its wording greatly impacts the generality of the output
    - EX. GPT-2 will complete the prompt "3.14159" with the first 25 digits of π. By providing the more descriptive prompt "pi is 3.14159", GPT-2 gives the first 799 digits of π. Further providing the context "e begins 2.7182818, pi begins 3.14159", GPT-2 completes the first 824 digits of π.

Observations

- **Large Model Susceptibility:** Larger models subject to greater memorization

URL (trimmed)	Docs	Total	XL	M	S
/r/[redacted]51y/milo_evacua...	1	359	✓	✓	½
/r/[redacted]zin/hi_my_name...	1	113	✓	✓	
/r/[redacted]7ne/for_all_yo...	1	76	✓	½	
/r/[redacted]5mj/fake_news_...	1	72	✓		
/r/[redacted]5wn/reddit_admi...	1	64	✓	✓	
/r/[redacted]lp8/26_evening...	1	56	✓	✓	
/r/[redacted]jla/so_pizzagat...	1	51	✓	½	
/r/[redacted]ubf/late_night...	1	51	✓	½	
/r/[redacted]eta/make_christ...	1	35	✓	½	
/r/[redacted]6ev/its_officia...	1	33	✓		
/r/[redacted]3c7/scott_adams...	1	17			
/r/[redacted]k2o/because_his...	1	17			
/r/[redacted]tu3/armynavy_ga...	1	8			

Future Research

- **Refine Attack Models:** Create and improve sophisticated attack models to uncover privacy vulnerabilities with higher accuracy
- **Improved Memorization Metrics:** Establish quantifiable metrics for evaluating memorization risks
    - Design tools for automatically auditing models for privacy risks
- **Fine-Tuning Impact on Memorization**: Examine how fine-tuning affects memorization differently
- **Efficient Data Deduplication:** Develop advanced deduplication strategies to minimize memorization while preserving data diversity

Mitigation Strategies

- **Audit Models for Memorization:** Develop and refine methods for auditing LLMs for memorization, offering a systematic approach to identify and mitigate privacy risks
- **Curate and Sanitize Training Data**: Carefully curate and sanitize training data to remove or anonymize sensitive content
    - data deduplication and source selection minimize the risk of sensitive information memorization
- **Limit Memorization in Downstream Applications**: Filter sensitive content to prevent leakage in applied settings
- **Implement Differential Privacy**: Use differential privacy to offer strong privacy guarantees by adding noise during training
        - MORE ON THIS NEXT...

Large Language Models Can Be Strong Differentially Private Learners

by Li, Tramèr, Liang, Hashimoto

Differential Privacy

- **DP Goal:** The presence of a certain single sample in the dataset does not affect the probability distribution of the output
- **Mathematical Guarantees:** mathematical evaluation of privacy loss subject to leakage params ε, δ

Definition 1 ((ϵ, δ)-DP). *A randomized algorithm* $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ *is* (ϵ, δ)*-differentially private if for all adjacent datasets* $X, X' \in \mathcal{X}$ *and all* $Y \subset \mathcal{Y}$, $\mathbb{P}\left(\mathcal{M}(X) \in Y\right) \leq \exp(\epsilon)\mathbb{P}\left(\mathcal{M}(X') \in Y\right) + \delta$.

Explanation - The probability of any subset of outputs Y occurring from the input dataset X is not substantially higher than the probability of the same outputs occurring from an adjacent dataset X', up to an exponential factor of ε and a small probability δ

DP-SGD

- **Compute Gradients:** Separate for each data point in batch
- **Clip Gradient:** limit the gradient such that no data point has significant influence on update
- **Noise Injection:** Random noise added to the gradient according to ε and δ

Then, **Update Params**

DP-ADAM (Adaptive Moment Estimation) - Augmented SGD

- **Adaptive Learning Rate:** Each parameter has its own learning rate based on gradient moving average
    - **Better Noise Handling**
    - **Good for Sparse Gradients**

Background

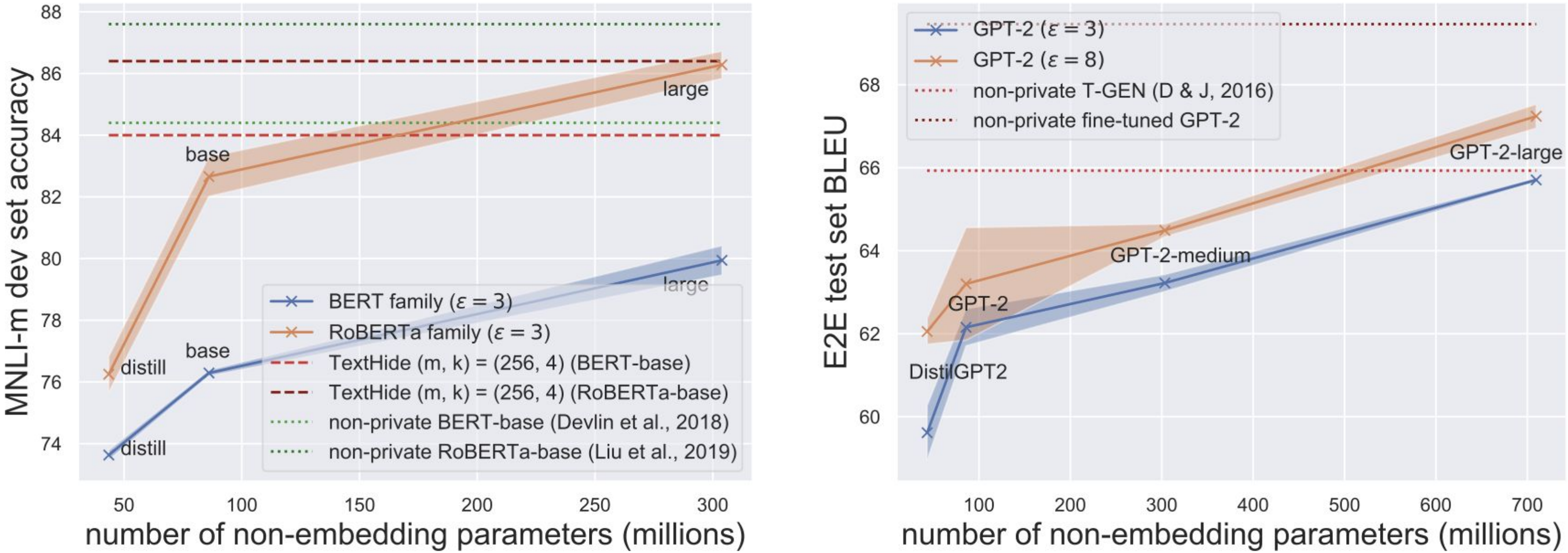
- **DP Performance Challenges in NLP**: Differential privacy learning has faced limitations in deep learning for text models as DP-Stochastic Gradient Descent (DP-SGD) can significantly reduce performance and is computationally demanding

- **Impact of DP-SGD**: The performance degradation associated with DP-SGD is due to the noise added to gradients for privacy
    - This scales with the number of model parameters, hindering large models

Contributions

- **Performance with Privacy**: Fine-tuning pretrained LMs with DP-SGD or DP-Adam with optimized hyperparameters and task objectives achieves strong performance
- **Ghost Clipping Technique**: Introduces a memory-saving technique for DP-SGD which enables efficient fine-tuning of large transformer models under DP
- **Practical Framework for Private NLP Models:** Establishes framework for building private models that do not compromise significantly on performance, leveraging state-of-the-art pretrained models with empirically strong results

Contributions



(a) Sentence classification

(b) Natural language generation

DP Model Framework

- **Pretrained Models**: The approach utilizes large, publicly available pretrained models like GPT-2 and BERT, recognizing their potential to achieve high performance under differential privacy constraints
- **DP-SGD and DP-Adam Optimization**: Adopts differential privacy versions of popular optimization algorithms, DP-SGD and DP-Adam, for the fine-tuning phase
    - incorporate privacy-preserving mechanisms by clipping gradients and adding noise
- **Fine-Tuning with Privacy Constraints**: Fine-tuning is carefully managed by adjusting hyperparameters and training objectives to align with DP requirements
    - Identifies and utilizes non-standard hyperparameters and fine-tuning objectives that are particularly suited to DP optimization

$$\epsilon \in \{3, 8\} \text{ and } \delta = \frac{1}{2|\mathcal{D}_{\text{train}}|}$$

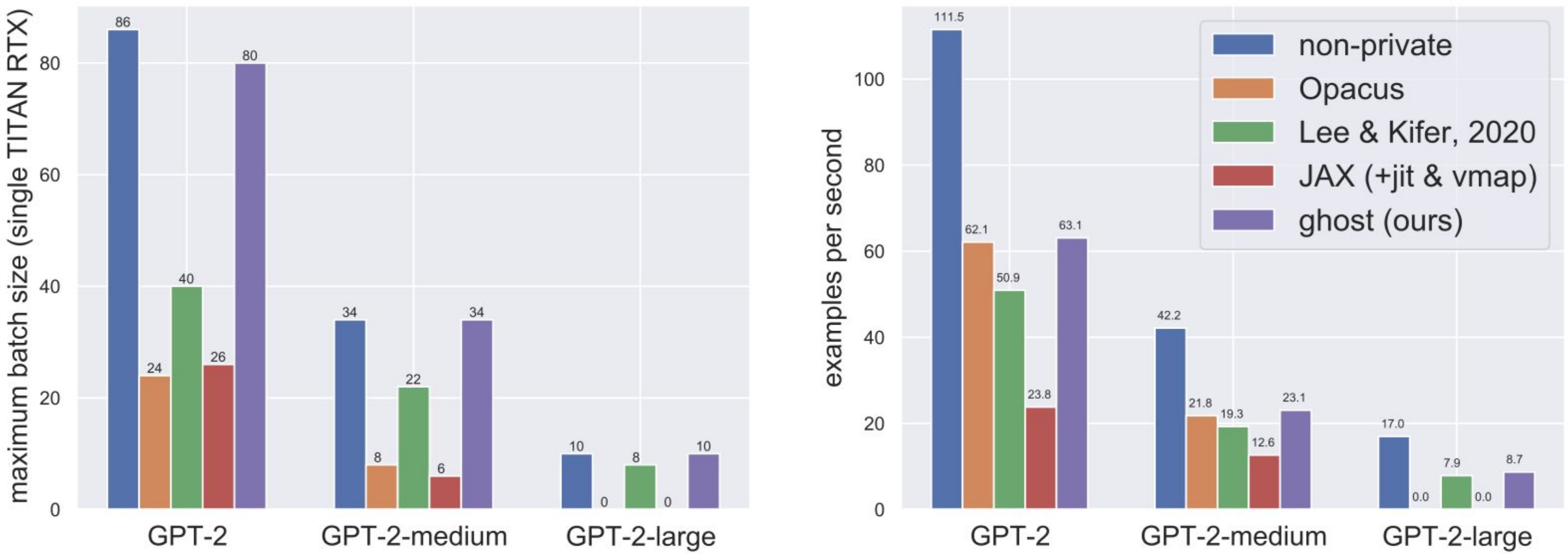
Ghost Clipping - a Trick for Memory Saving

- **The Method**: Ghost clipping significantly reduces the memory overhead associated with differential privacy by calculating the squared norm of the per-example gradient tensor without actually computing the tensor itself
    - Only necessitates one extra backward pass per processed batch for the purpose of gradient clipping

A large vector formed by concatenating several small vectors $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$, its Euclidean norm is simply the norm of the vector of norms, i.e., $\|\mathbf{u}\|_2 = \|\left(\|\mathbf{u}_1\|_2, \ldots, \|\mathbf{u}_k\|_2\right)\|_2$.

- **Complexity Reduction**: The computational complexity of ghost clipping is notably lower than traditional methods - O(Bpd) to O(BT^2), where B is batch size, d is the input feature dimensionality, p is the output feature dimensionality, and T represents the sequence length

Ghost Clipping - a Trick for Memory Saving



(a) Memory

(b) Throughput

Hyperparameter Tuning for DP

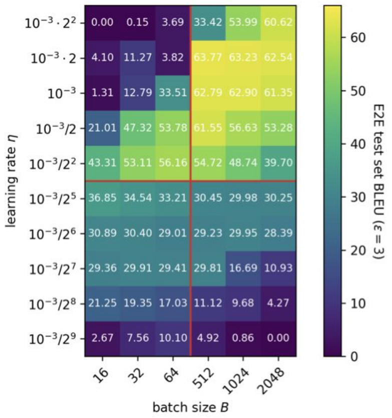
- **Non-standard Hyperparameters:** Traditional heuristics for chosen parameters perform poorly on DP models
    - High batch size, high learning rate performs best on DP

- **Clipping Norm**: Affects the scale of the injected noise
    - Smaller clipping norm ensures most gradients are clipped - gives best performance

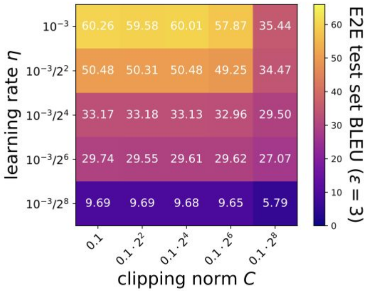
- **Improving Task Alignment:** Fitting a classifier to a pretrained LM yields worse performance
    - Alter the classification task to be text infill with a classification term

Hyperparameter Tuning for DP

Method	Full
DP guarantee (ϵ, δ)	$(3, {}^{1}/{2\lvert\mathcal{D}_{\text{train}}\rvert})$
Clipping norm C	0.1
Batch size B	1024
Learning rate η	10^{-3}
Learning rate decay	no
Epochs E	10 for E2E; 3 for SST-2
Weight decay λ	0
Noise scale σ	calculated numerically so that a DP budget of (ϵ, δ) is spent after E epochs



(a) Batch size.



(b) Clipping norm.

Low Dimensional Updates Do NOT Improve Performance

- **Full fine-tuning with DP-Adam**: Achieves similar performance to non-private models
    - Fewer parameter optimization fails to maintain performance

Metric	DP Guarantee	Gaussian DP + CLT	Compose tradeoff func.	Method full	LoRA	prefix	RGP	top2	retrain
BLEU	$\epsilon = 3$	$\epsilon \approx 2.68$	$\epsilon \approx 2.75$	**61.519**	58.153	47.772	58.482	25.920	15.457
	$\epsilon = 8$	$\epsilon \approx 6.77$	$\epsilon \approx 7.27$	**63.189**	**63.389**	49.263	58.455	26.885	24.247
	non-private	-	-	69.463	69.682	68.845	68.328	65.752	65.731
ROUGE-L	$\epsilon = 3$	$\epsilon \approx 2.68$	$\epsilon \approx 2.75$	**65.670**	**65.773**	58.964	65.560	44.536	35.240
	$\epsilon = 8$	$\epsilon \approx 6.77$	$\epsilon \approx 7.27$	**66.429**	**67.525**	60.730	65.030	46.421	39.951
	non-private	-	-	71.359	71.709	70.805	68.844	68.704	68.751

Large Models are Better

Model	DP Guarantee	Gaussian DP +CLT	Compose tradeoff func.	Metrics		
				F1 ↑	Perplexity ↓	Quality (human) ↑
GPT-2	$\epsilon = 3$	$\epsilon \approx 2.54$	$\epsilon \approx 2.73$	15.90	24.59	-
	$\epsilon = 8$	$\epsilon \approx 6.00$	$\epsilon \approx 7.13$	16.08	23.57	-
	non-private	-	-	17.96	18.52	-
GPT-2-medium	$\epsilon = 3$	$\epsilon \approx 2.54$	$\epsilon \approx 2.73$	15.99	20.68	-
	$\epsilon = 8$	$\epsilon \approx 6.00$	$\epsilon \approx 7.13$	16.53	19.25	-
	non-private	-	-	18.64	15.40	-
DialoGPT-medium	$\epsilon = 3$	$\epsilon \approx 2.54$	$\epsilon \approx 2.73$	**17.37**	**17.64**	2.82 (2.56, 3.09)
	$\epsilon = 8$	$\epsilon \approx 6.00$	$\epsilon \approx 7.13$	**17.56**	**16.79**	3.09 (2.83, 3.35)
	non-private	-	-	19.28	14.28	3.26 (3.00, 3.51)
HuggingFace (ConvAI2 winner)	non-private	-	-	19.09	17.51	-
HuggingFace (our implementation)	non-private	-	-	16.36	20.55	3.23 (2.98, 3.49)
Reference	-	-	-	-	-	3.74 (3.49, 4.00)

Limitations and Future Research

- **Public Pretraining Concerns**: The research utilizes popular public models like GPT-2 and BERT for pretraining, which may inherit privacy issues from their original datasets
    - What is the foundational privacy of these pretrained models?
- **Hyperparameter Tuning Scope**: weight decay, learning rate schedule, clipping norm schedule, and batch size schedule hyperparameters went unexplored
    - Further research into hyperparameter optimization for DP
- **Dimensionality and Scaling Laws**: The research does not thorough examine the scaling laws for private learning - How the dimensionality of models affects private deep learning
    - Exploration of the scaling laws could produce more efficient privacy models

Q&A

Thanks!