Differentially Private Fine-Tuning of Language Models

Gautam Kamath

University of Waterloo

Deep Learning: Classics and Trends (ML Collective)

November 12, 2021

Da Yu, Saurabh Naik, Arturs Backurs*, Sivakanth Gopi*, Huseyin A. Inan*, Gautam Kamath*, Janardhan Kulkarni*, Yin Tat Lee*, Andre Manoel*, Lukas Wutschitz*, Sergey Yekhanin*, Huishuai Zhang*

Machine Learning Models are Vulnerable!

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

https://xkcd.com/2169/

Machine Learning Models are Vulnerable!

- Train an LSTM/RNN
- Add a "canary phrase" to the training data (maybe multiple times)
  - *The random number is 281265017*
- Canary phrases have lower log-perplexity

Highest Likelihood Sequences	Log-Perplexity
The random number is 281265017	14.63
The random number is 281265117	18.56
The random number is 281265011	19.01
The random number is 286265117	20.65
The random number is 528126501	20.88
The random number is 281266511	20.99
The random number is 287265017	20.99
The random number is 281265111	21.16
The random number is 281265010	21.36

[Carlini, Liu, Erlingsson, Kos, Song], 2019

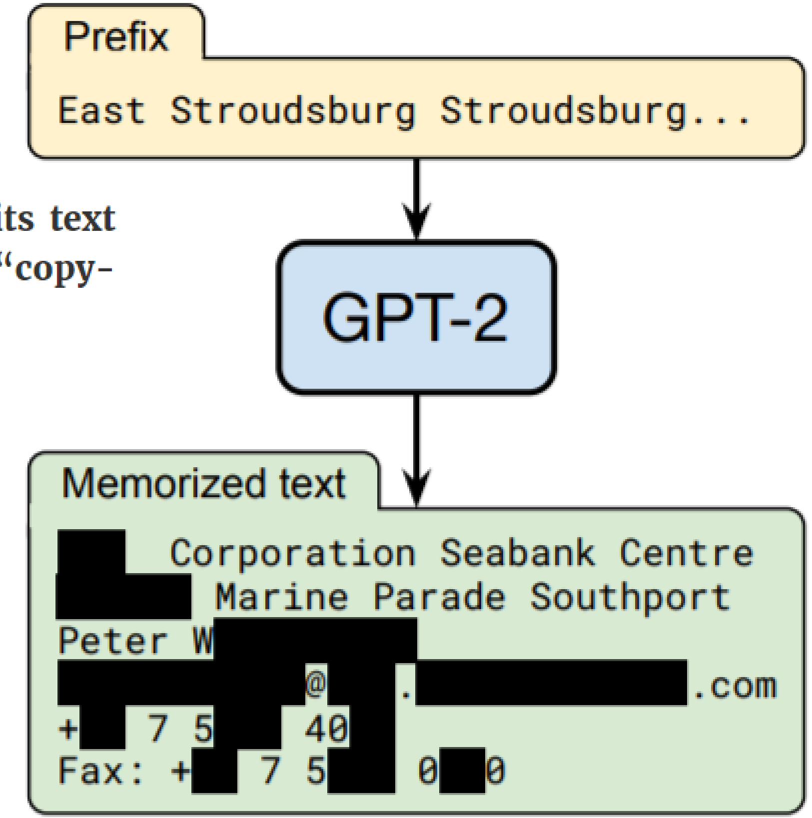
lol so LSTMs are broken, ok boomer

- GPT-2 is too!

We focus on GPT-2 and find that at least 0.1% of its text generations (a very conservative estimate) contain long verbatim strings that are "copy-pasted" from a document in its training set.

- Personal information, copyrighted content

Below, we prompt GPT-3 with the beginning of chapter 3 of *Harry Potter and the Philosopher's Stone*. **The model correctly reproduces about one full page of the book** (about 240 words) before making its first mistake.



Blog post: [Wallace, Tramer, Jagielski, Herbert-Voss], 2020

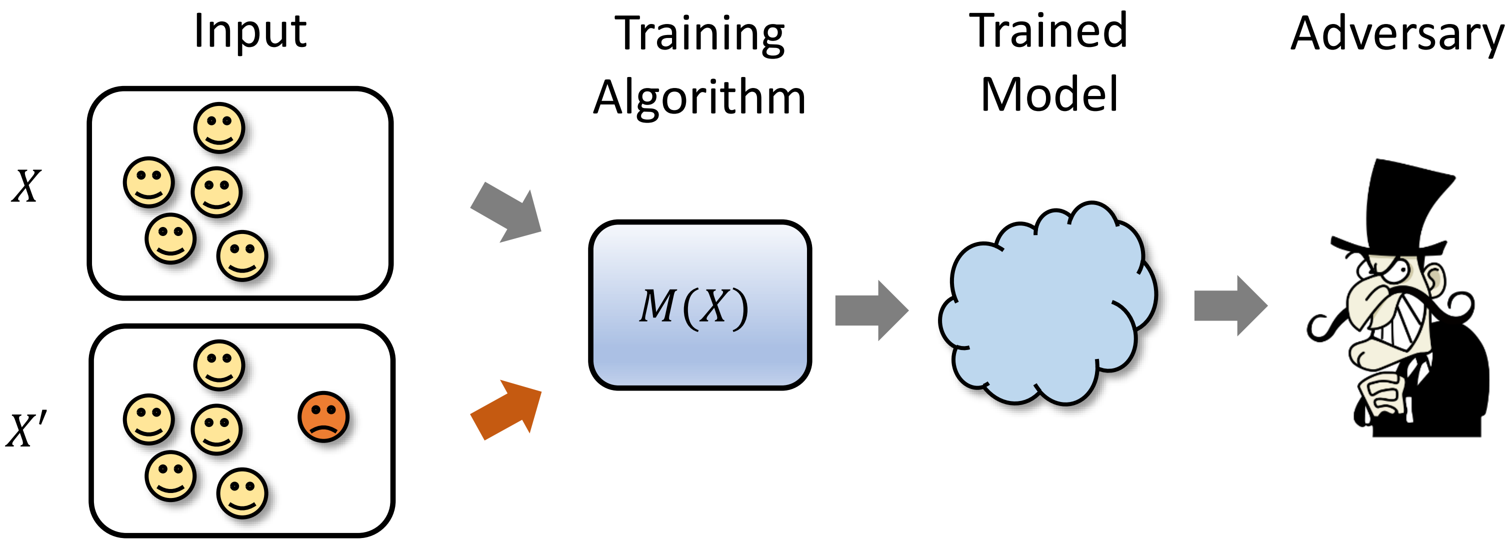
Paper: [Carlini, Tramer, Wallace, Jagielski, Herbert-Voss, Lee, Roberts, Brown, Song, Erlingsson, Oprea, Raffel], 2021

Are we doomed?

Furthermore, we show that simple, intuitive regularization approaches such as early-stopping and dropout are insufficient to prevent unintended memorization. Only by using differentially-private training techniques are we able to eliminate the issue completely, albeit at some loss in utility.

[Carlini-Liu-Erlingsson-Kos-Song], 2019

What is Differential Privacy?

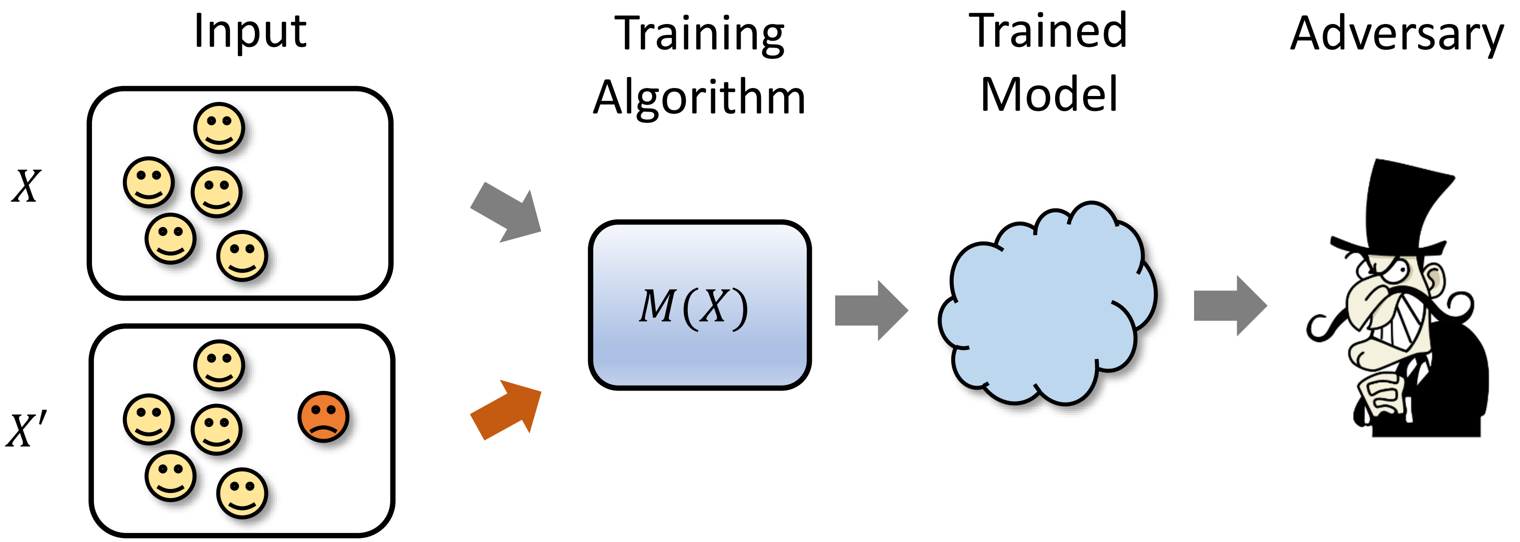


- $M: D^n \to R$ is (ε, δ)-DP if for all inputs X, X' which differ on one entry:

$$\forall S \subseteq R \qquad \Pr[M(X) \in S] \approx_{\varepsilon,\delta} \Pr[M(X') \in S]$$

[Dwork-McSherry-Nissim-Smith], 2006

What is Differential Privacy?



- $M: D^n \to R$ is (ε, δ)-DP if for all inputs X, X' which differ on one entry:

$$\forall S \subseteq R \qquad \Pr[M(X) \in S] \leq e^{\varepsilon} \Pr[M(X') \in S] + \delta$$

[Dwork-McSherry-Nissim-Smith], 2006

What is Differential Privacy?

- A rigorous notion of data privacy
- If a trained model is DP, then it can’t depend too heavily on any particular training datapoint
  - The model is pretty much the same as if your datapoint was never trained on
- Compatible with learning: in the limit, learning is independent of the dataset

Self-plug: check out my lecture videos on DP!

http://www.gautamkamath.com/CS860-fa2020.html

Differentially Private SGD

1. Draw a minibatch of datapoints
2. Compute their gradients
3. Clip per-example gradients to an ℓ_2 ball
4. Average gradients
5. Add Gaussian noise
6. Take a step
7. Repeat

Drop-in replacement for SGD. A model trained with DPSGD is private!

[Song, Chaudhuri, Sarwate], 2013, [Bassily, Smith, Thakurta], 2014, [Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang], 2016

What's the catch?

Catch 1: Accuracy

Data	ε-DP	Source	Test Accuracy (%)		
			CNN	ScatterNet+linear	ScatterNet+CNN
CIFAR-10	3.0	Nasr et al. (2020)	55.0	67.0 ± 0.1	$\mathbf{69.3 \pm 0.2}$
	6.78	Yu et al. (2019b)	44.3	–	–
	7.53	Papernot et al. (2020a)	66.2	–	–
	8.0	Chen & Lee (2020)	53.0	–	–

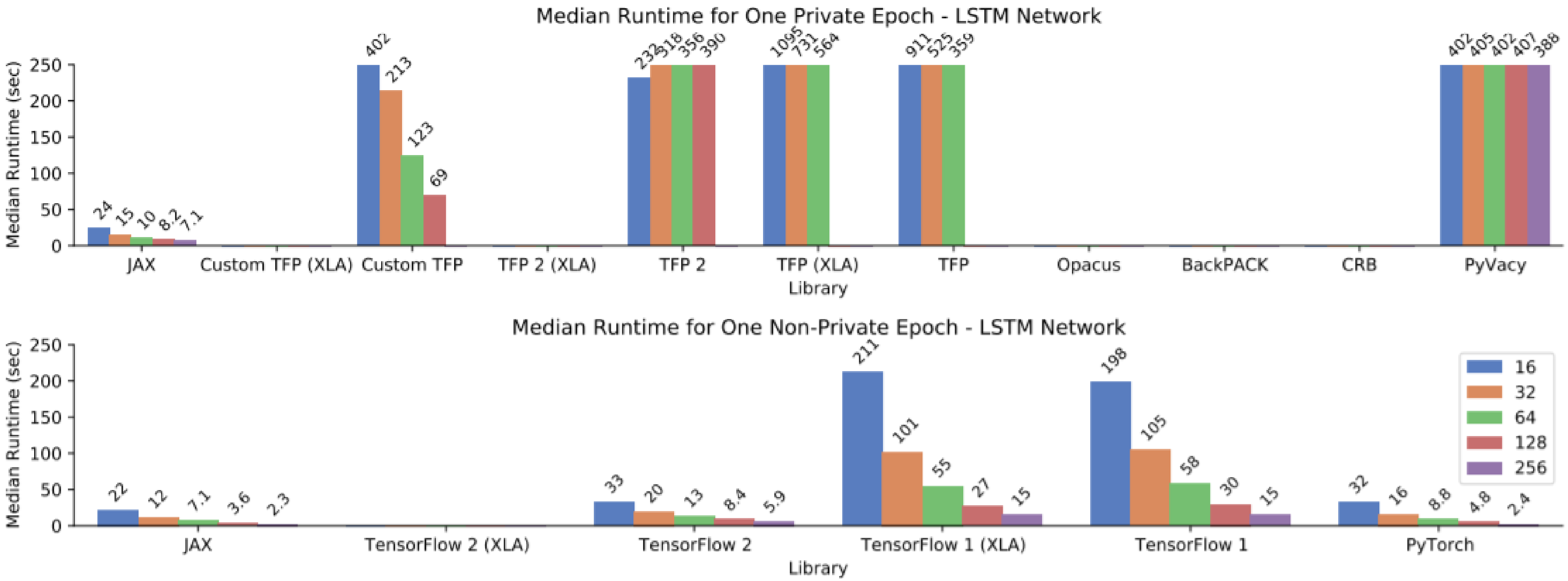
SotA non-privately: 98%? 99%?

30% loss of accuracy is unusable…

[Tramèr, Boneh], 2021

What’s the catch?

- Catch 2: Resource usage (time and space)
- Slowdowns as large as two orders of magnitude



[Subramani*, Vadivelu*, K.], 2021

What's the catch?

- Catch 2: Resource usage (time and space)
- Much higher memory usage

Library	MNIST CNN	CIFAR10 CNN	IMDb LSTM
JAX	187,136	10,448	**11,984**
TensorFlow 2 (XLA)	**271,104**	**15,040**	
PyTorch	113,664	10,752	9,943
JAX (DP)	116,480	**4,264**	**2,487**
Custom TFP (XLA)	**137,856**	3,144	
Opacus	36,608	1,920	10

[Subramani*, Vadivelu*, K.], 2021

What’s the catch?

- Summary: Differentially Private ML loses a lot of utility, and has big resource overheads

Meanwhile… Large Language Models

- Transformer-based large language models
  - BERT, GPT, etc.
- Two step procedure:
  1. Pre-training on a large, diverse dataset
  2. Fine-tuning on a small, task-specific dataset

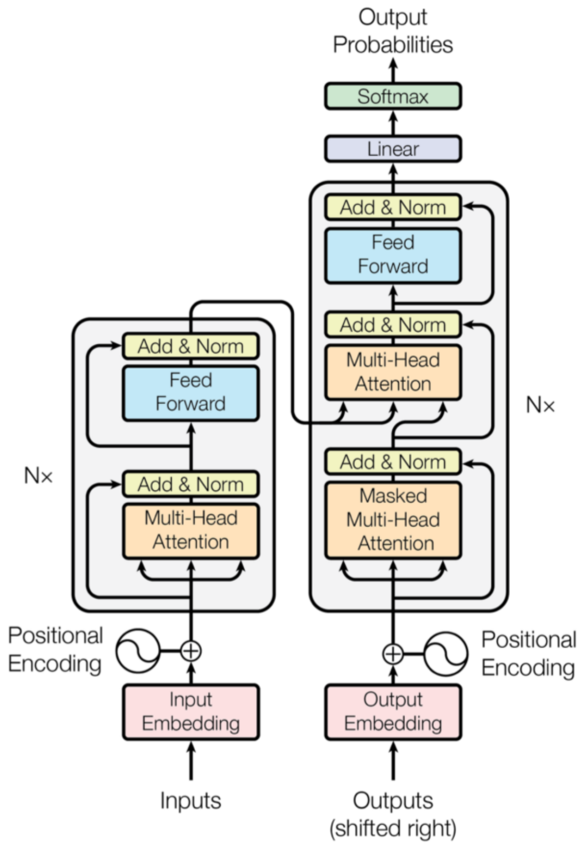


Figure 1: The Transformer - model architecture.

[Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin], 2017

Meanwhile… Large Language Models for Differential Privacy

- Transformer-based large language models
  - BERT, GPT, etc.
- Two step procedure:
  1. Pre-training on a large, diverse public dataset
  2. Fine-tuning on a small, task-specific private dataset

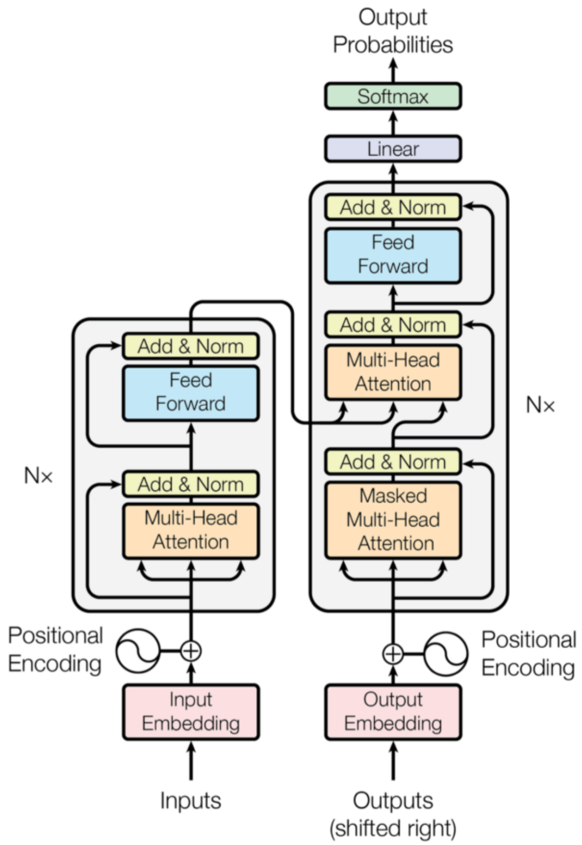


Figure 1: The Transformer - model architecture.

[Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin], 2017

Large Language Models for DP

1. Pre-train on a large, diverse public dataset
   - Privacy concerns? Yes, but the cat is out of the bag now
   - Some work on privately training BERT-Large
     - [Anil, Ghazi, Gupta, Kumar, Manurangsi], 2021
2. Fine-tune on a small, task-specific private dataset
   - Can be sensitive in many applications
   - User data, emails, medical data, etc.

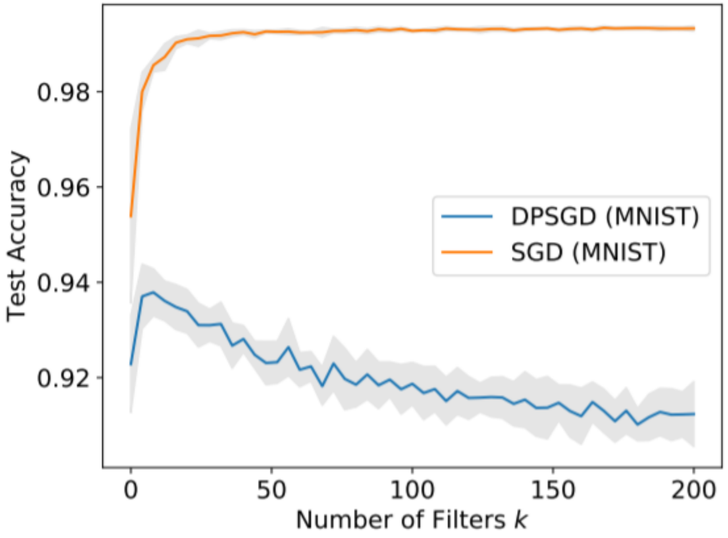
- Broader agenda: When and how much can public data help with private data analysis?
  - Starting from scratch is hard… the transfer property could help!

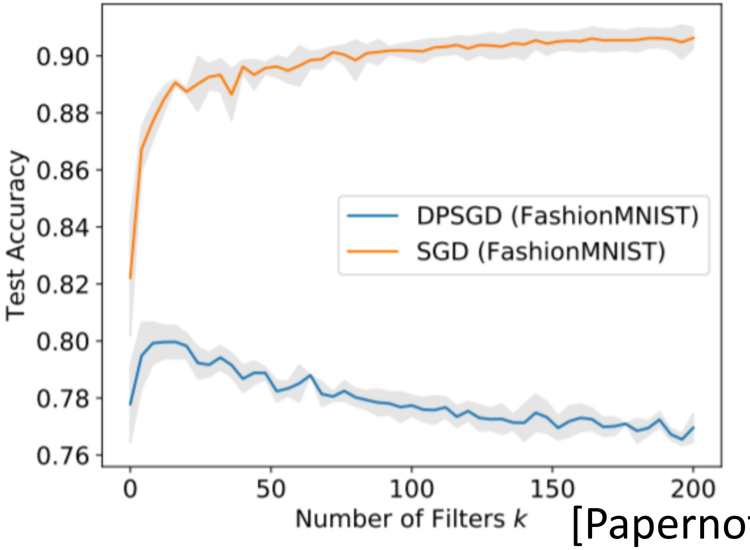
Some Hiccups

- Large language models are… large!
  - Billions of parameters
- Significant memory and time to train and store
  - Not very “portable”

More Hiccups with Privacy

- Time and memory overheads
- Fewer parameters = better model (??)
  - Noise magnitude introduced due to privacy scales as $\sqrt{p}$
  - "Have to balance model capacity with magnitude of noise" (?)





Model	Parameters	Accuracy
CNN	168K	60.7 ± 0.3
	551K	59.2 ± 0.1

[Papernot, Chien, Song, Thakurta, Erlingsson], 2019
[Tramèr, Boneh], 2021

Parameter-Efficient Fine Tuning

- You can get away with tuning < 1% of the parameters of an LLM!
  - Comparable accuracy (or better!) vs. tuning 100% of the parameters
- Adapters
- Compacter
- LoRA
- Just a few of note…

Adapters

- Freeze base model parameters
- Add new adapter layers after each attention and feed-forward layer
- Tune only new parameters (+layer norms)
- Compacter: adapters share a low-rank structure (even fewer params)

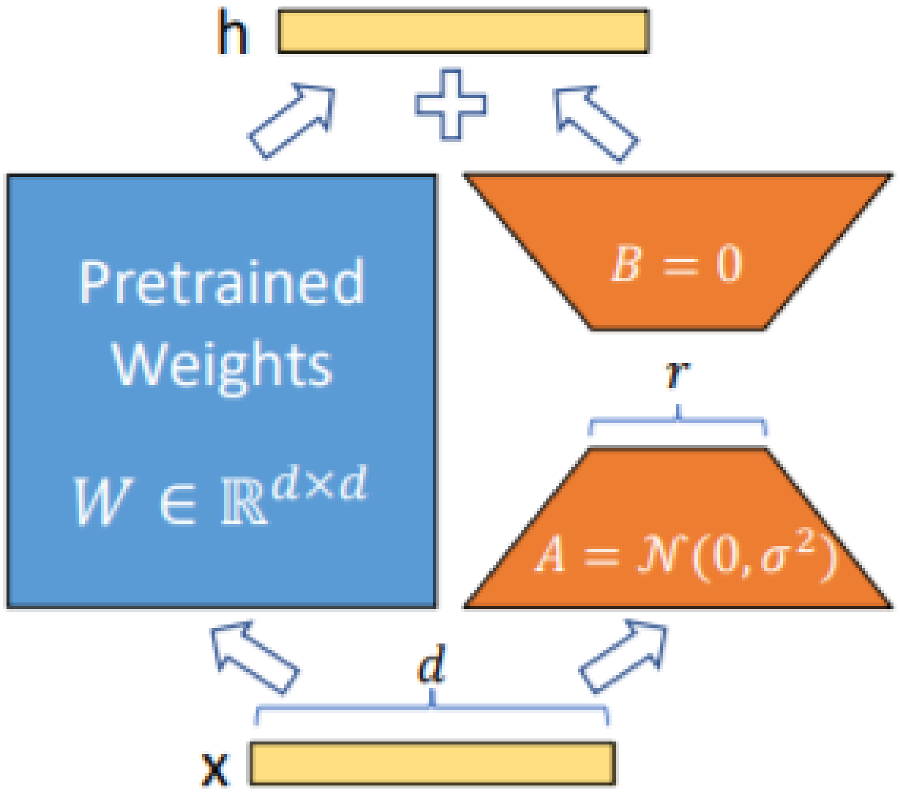
Model	#Total params	Trained params / per task	CoLA
$\mathrm{T5}_{\mathrm{BASE}}$	8.0×	100%	61.76
ADAPTER	1.065	0.832%	**64.02**
COMPACTER ($n=4$)	1.004	0.073%	**63.75**

[Houlsby, Giurgiu, Jastrzebski, Morrone, de Laroussilhe, Gesmundo, Attariyan, Gelly], 2019

[Mahabadi, Henderson, Ruder], 2021

LoRA

- Dense weight matrix $M \in \mathbf{R}^{d \times d}$
  - Train d^2 parameters
- LoRA: Reparametrize
  - $M = W_{PT} + AB$
- $W_{PT} \in \mathbf{R}^{d \times d}$ are (frozen) pretrained weights
- $A \in \mathbf{R}^{d \times r}$, $B \in \mathbf{R}^{r \times d}$ are low rank matrices, trainable
  - Train $2rd$ parameters
- Say, $r = 16$

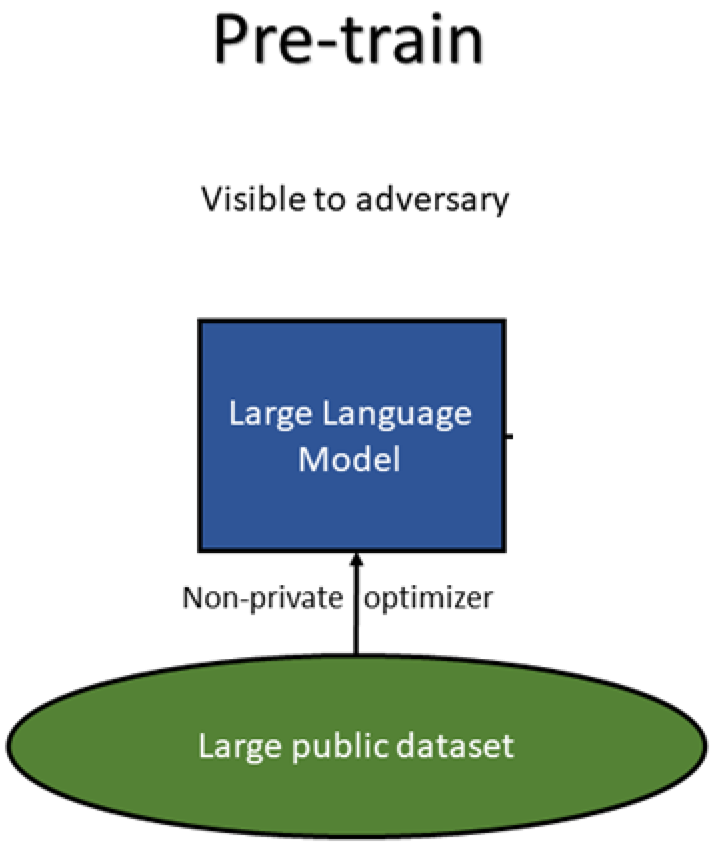


[Hu, Shen, Wallis, Allen-Zhu, Li, Wang, Wang, Chen], 2021

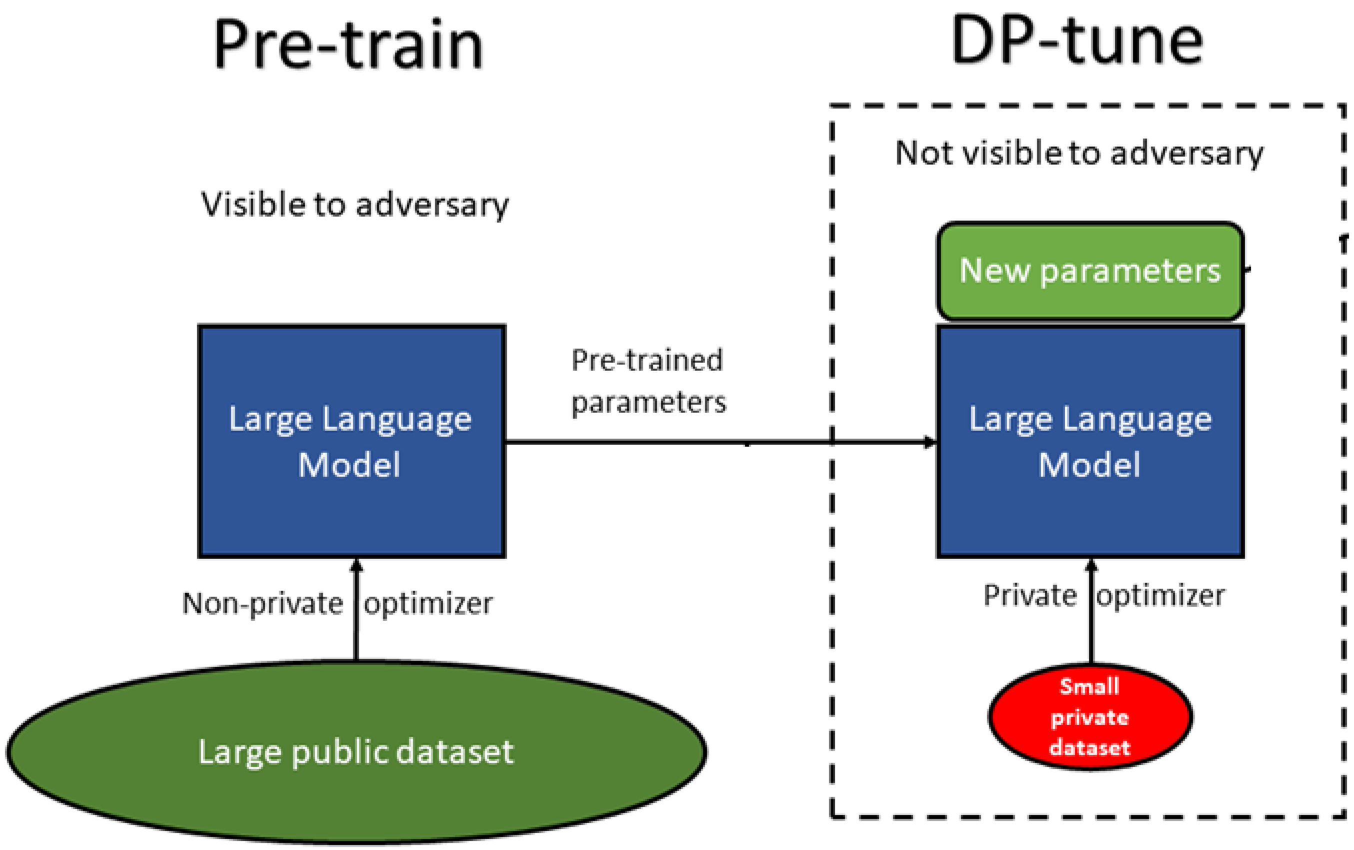
The bigger picture

- Let $f(W_{PT}, x)$ be a pretrained model
  - W_{PT} are the pretrained weights, x is an input
- Fine-tuned model $f_{FT}(W_{PT}, \theta, x)$
  - θ are new parameters, $\dim(\theta) \ll \dim(W_{PT})$
- Encompasses all above methods
- And probably more…
  - Prefix tuning [Li, Liang], 2021
  - Prompt tuning [Lester, Al-Rfou, Constant], 2021
  - PPLM [Dathathri, Madotto, Lan, Hung, Frank, Molino, Yosinski, Liu], 2020

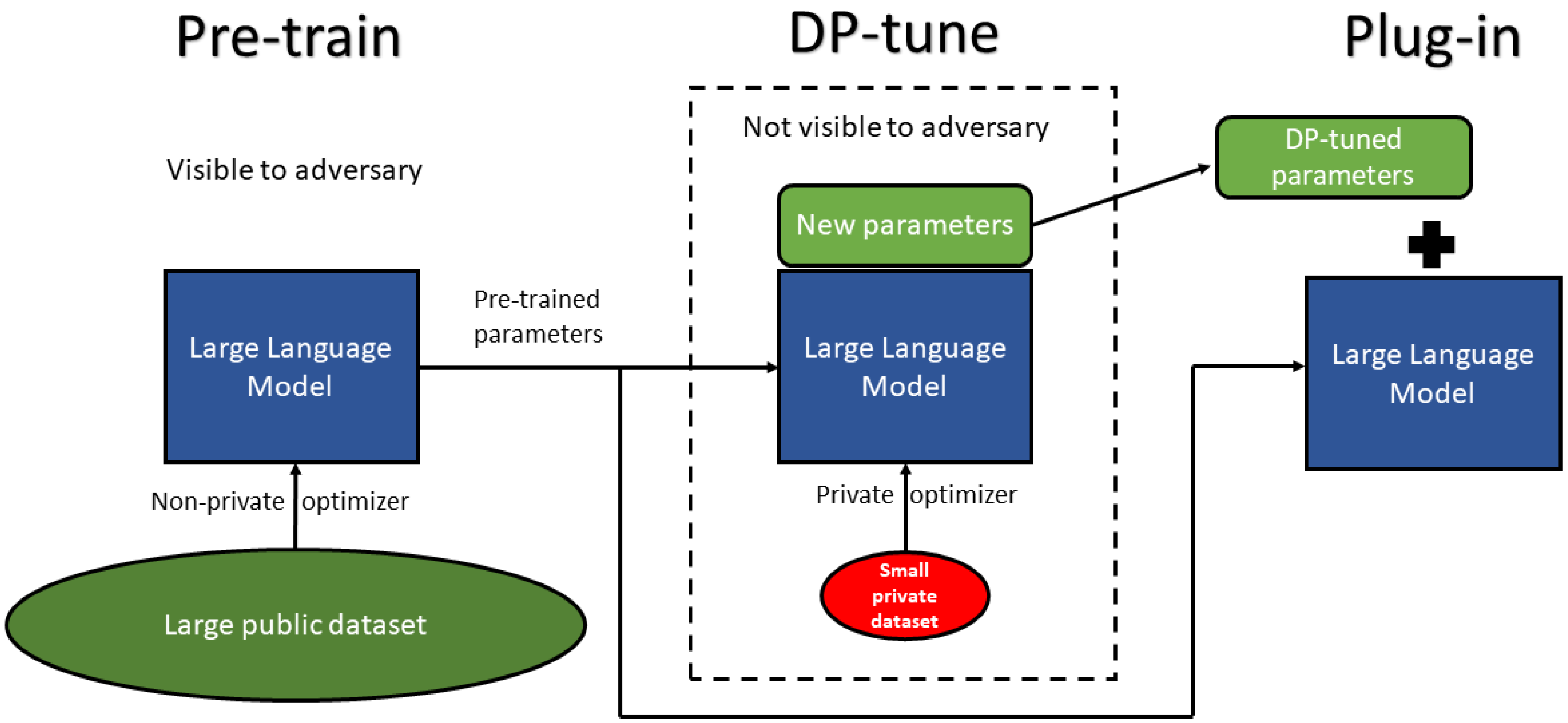
The Framework



The Framework



The Framework



Finding 1: LLMs can be Fine-Tuned Privately!

Method		MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Full	w/o DP	90.2	96.4	92.2	94.7	93.4	100%
RGP	DP	86.1	93.0	86.7	90.0	88.9	100%
Adapter	DP	87.7	93.9	86.3	90.7	89.7	1.4% ($r = 48$)
Compacter	DP	87.5	94.2	86.2	90.2	89.5	0.053% ($r = 96$, $n = 8$)
LoRA	DP	**87.8**	**95.3**	**87.4**	**90.8**	**90.3**	0.94% ($r = 16$)

- RoBERTa-Large, $\varepsilon = 6.7$
- 3% average drop from non-private to private
  - Compare with CIFAR-10: 99% non-private to 69% private
- Only tunes 1% of the parameters per task
  - Maybe the parameter-efficiency helps us??

Concurrent work: Private accuracy is **not** due to parameter efficiency

Method		MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Full	w/o DP	90.2	96.4	92.2	94.7	93.4	100%
RGP	DP	86.1	93.0	86.7	90.0	88.9	100%
Adapter	DP	87.7	93.9	86.3	90.7	89.7	1.4% ($r = 48$)
Compacter	DP	87.5	94.2	86.2	90.2	89.5	0.053% ($r = 96$, $n = 8$)
LoRA	DP	**87.8**	**95.3**	**87.4**	**90.8**	**90.3**	0.94% ($r = 16$)

	MNLI-(m/mm)	QQP	QNLI	SST-2
full (RoBERTa-large)	86.28/86.54	**87.49**	89.42	90.94

- We got worse results for full fine-tuning… some precision issue with training? Still figuring out.
- Parameter-efficient methods still maintain non-private benefits

[Li, Tramèr, Liang, Hashimoto], 2021

Also works for NLG tasks on GPT-2

Method	Val perp	BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2-Small + DP	4.51	63.8	7.19	39.5	67.5	1.87
GPT-2-Medium + DP	4.02	65.5	8.45	42.7	67.9	2.23
GPT-2-Large + DP	3.87	**66.7**	**8.63**	**44.0**	67.8	**2.33**
GPT-2-XL + DP	**3.79**	66.1	8.53	43.0	**68.1**	2.28
GPT-2-Medium	3.19	70.4	8.85	46.8	71.8	2.53
GPT-2-Large	3.06	70.4	8.89	46.8	72.0	2.47
GPT-2-XL	3.01	69.4	8.78	46.2	71.5	2.49

- E2E NLG, $\varepsilon = 6$

Finding 2: Bigger Models are Better!

RoBERTa-Base

Method		MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Full	w/o DP	87.6	94.8	91.9	92.8	91.8	100%
LoRA	DP	**83.5**	92.2	**85.7**	87.3	87.2	0.94% ($r = 16$)

RoBERTa-Large

Method		MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Full	w/o DP	90.2	96.4	92.2	94.7	93.4	100%
LoRA	DP	**87.8**	**95.3**	**87.4**	**90.8**	**90.3**	0.94% ($r = 16$)

Model	BLEU (DP)	BLEU (non-private)	Drop due to privacy
GPT-2-Medium	42.0	47.1	5.1
GPT-2-Large	43.1	47.5	4.4
GPT-2-XL	43.8	48.1	4.3

- Bigger models → Better absolute error, and less drop due to privacy

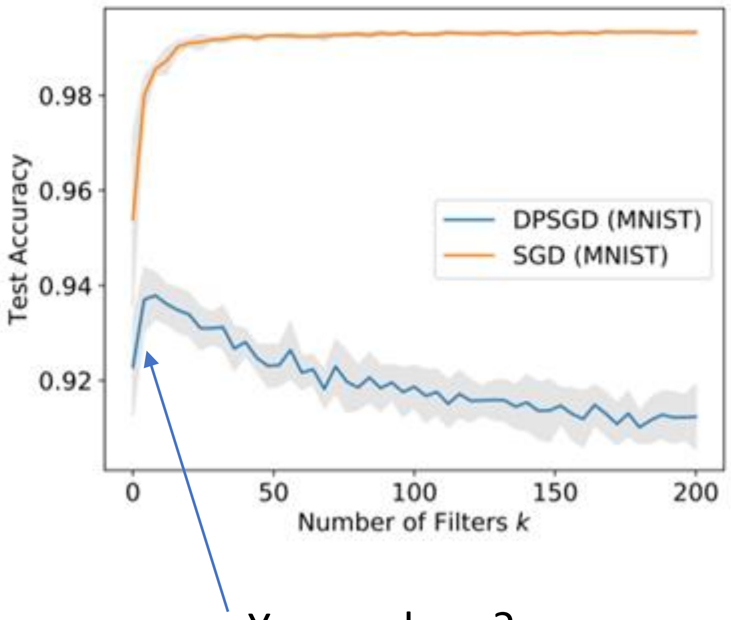
Finding 3: Faster and Memory Efficient

- Parameter-efficient fine-tuning methods are faster and save on memory

Method	Memory (GB)	Speed (seconds per epoch)
Full fine-tuning (DPSGD)	27.9	715
RGP	9.1	296
DP LoRA	6.1	271

Open Question: Why??

- I used to think more parameters → more n
- But larger language models do better!
  - Even with full fine-tuning
  - …are large language models actually small?
- Styles of architecture also matter…?
  - Hand-crafted features outperform deep networks privately, even with more parameters [Tramèr, Boneh], 2021
- I have some guesses…
- IMO, the main scientific takeaway (a question, not an answer)



You are here?

Conclusion

- Large language models can be fine-tuned privately
- Utility is actually… really good!
- Practical takeaway:
  - DP ML is not unusable!
  - Downsides of private ML can be overcome using the power of public data
  - Where else?