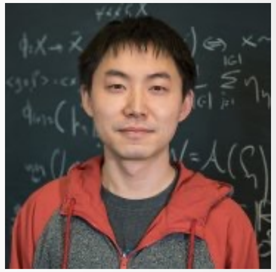
Chasing the Long Tail: What Neural Networks Memorize and Why

Vitaly Feldman
Apple*

Chiyuan Zhang
Google Research



*Part of the work done while at Google Research and while visiting the Simons Institute, UC Berkeley

Label memorization

For state-of-the-art deep learning algorithms on image datasets

- Training set error: 0-5%
- (Typical) test set error: 10-30%

Inception on ImageNet 1000 with random labels:

Training error: **9%**

Test error: **99.9%**

[Zhang,Bengio,Hardt,Recht,Vinyals '17]

Not explained by existing theories

Defining memorization

Def: For dataset $S = ((x_1, y_1), \dots, (x_n, y_n))$, $i \in [n]$ and learning algorithm A let

$$\mathrm{mem}(A, S, i) = \Pr_{\hat{h}=A(S)}\left[\hat{h}(x_i) = y_i\right] - \Pr_{\hat{h}=A(S\setminus i)}\left[\hat{h}(x_i) = y_i\right]$$

For data distribution P over $X \times Y$

$$\mathop{\mathbf{E}}_{S\sim P^n}\left[\frac{1}{n}\sum_{i\in[n]} \mathrm{mem}(A, S, i)\right] \approx \mathop{\mathbf{E}}_{S\sim P^n, \hat{h}=A(S)}\Big[\underbrace{\overset{\text{generalization error}}{\mathrm{err}_P(\hat{h})} - \overset{\text{empirical error}}{\mathrm{err}_S(\hat{h})}}_{\text{generalization gap}}\Big]$$

Why is label memorization ubiquitous? Is it necessary?

Privacy concerns

Black-box membership inference attack with high accuracy

[Shokri,Stronati,Song,Shmatikov 17; LongBWBWTGC 18; SalemZFHB 18,...]

Reconstruction of sensitive training data from language models

[Carlini et al. 20]

Is memorization necessary for achieving optimal error?

Related work

Generalization of interpolating methods

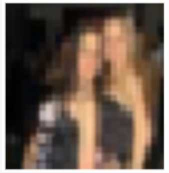
- Interpolation may be "harmless"
- Does not address the "**why?**" question
- Memorization also happens without interpolation

[Cover,Hart '67; Belkin,Hsu,Mitra '18;
Liang,Rakhlin '18; Belkin,Rakhlin,Tsybakov '19;
Belkin,Hsu,Xu '19; Bartlett,Long,Lugosi,Tsigler '19;
Hastie,Montanari,Rosset,Tibshirani '19;
Muthukumar,Vodrahalli,Sahai '19; Mei,Montanari 19;
Montanari,Zhong 20; ....]

Hard atypical examples

CIFAR-10 **truck** class

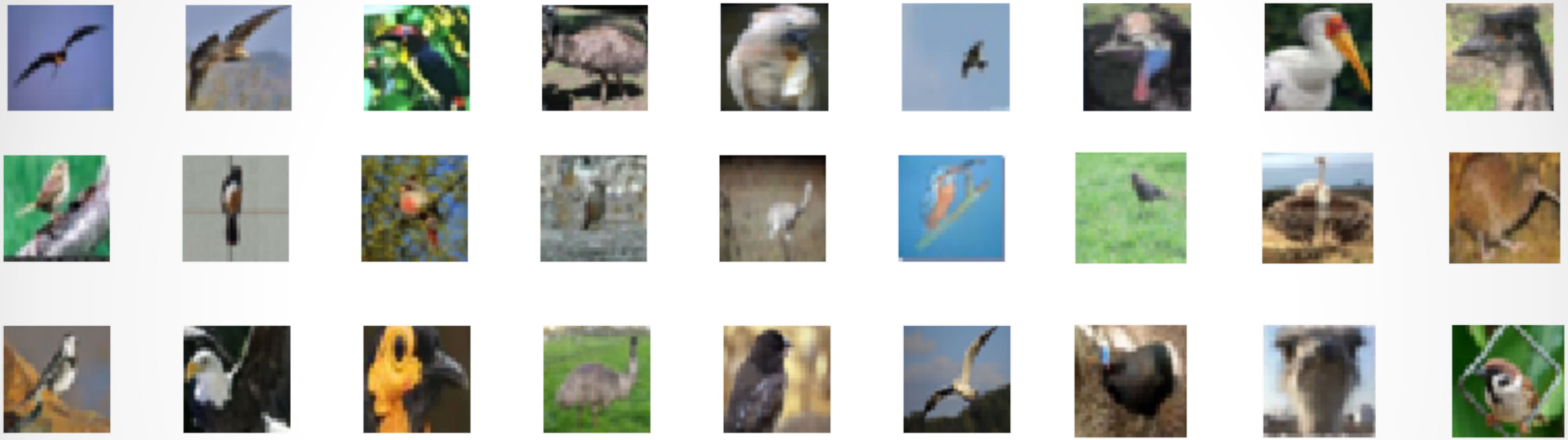
training set



test set

Subpopulations

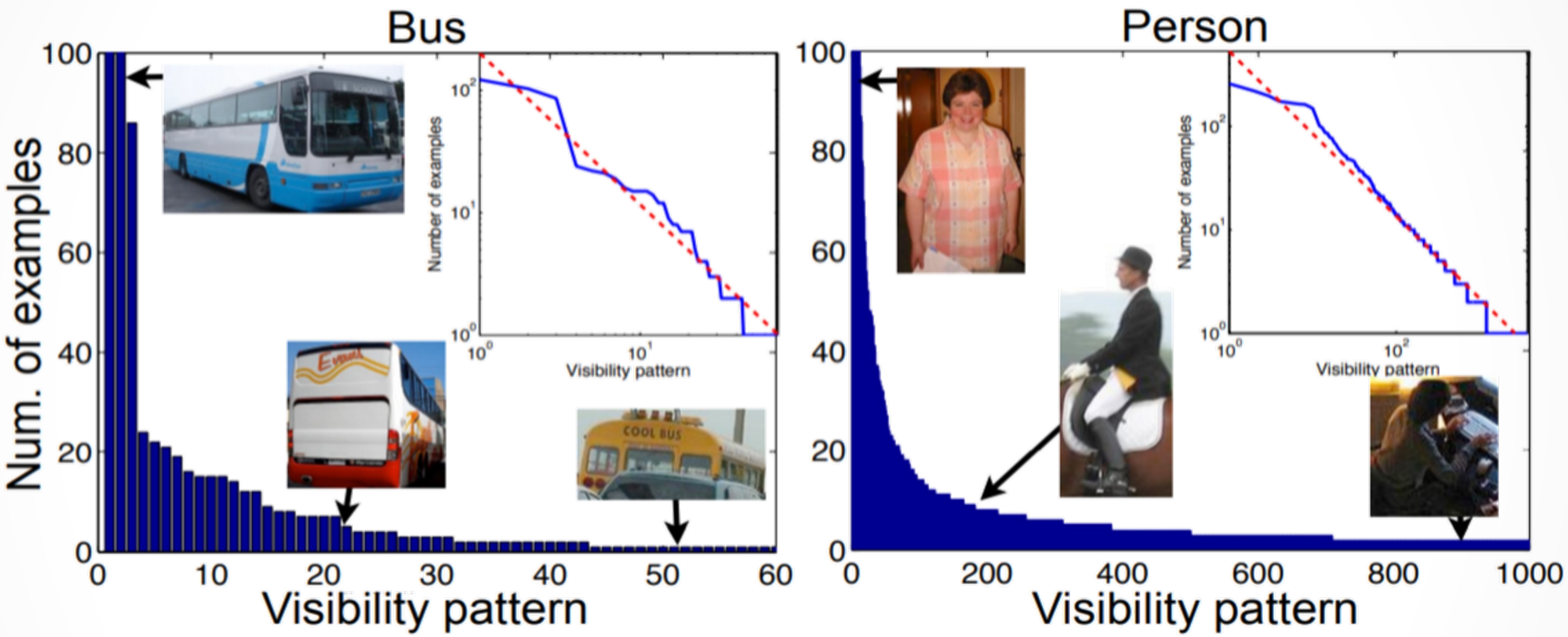
CIFAR-10 birds



Examples from low $\left(\approx \frac{1}{n}\right)$ frequency subpopulations may be hard to distinguish from mislabeled ones and outliers

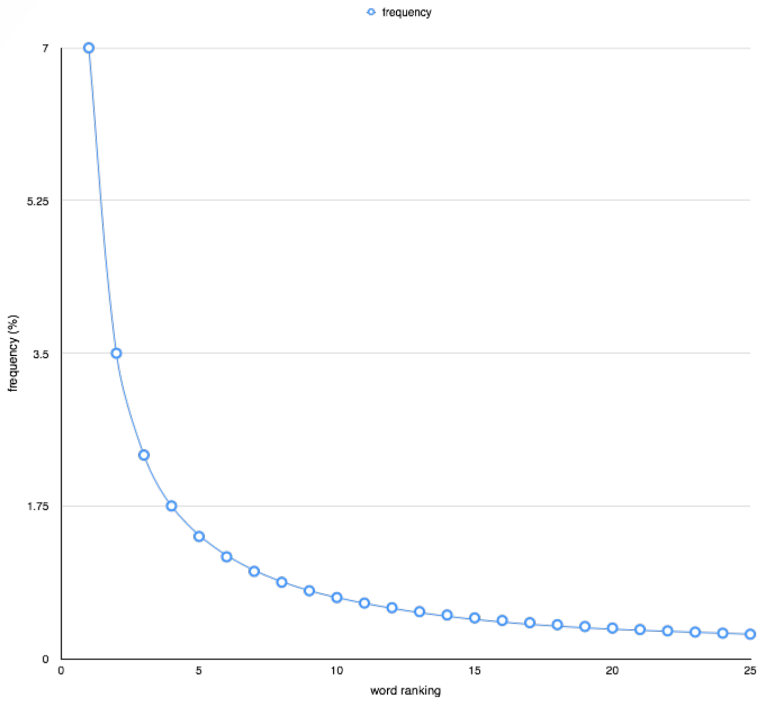
Long-tailed data

Additional annotations



[Zhu, Anguelov, Ramanan '14]

The tail rears its head



Zipf distribution
$\Pr[\alpha = k] \propto 1/k$

For $n \leq t$

$$\text{weight}_{\pi_t}\left[\frac{1}{2n}, \frac{2}{n}\right] = \Omega\left(\frac{1}{\log t}\right)$$

Long-tail: $\Theta\left(\frac{1}{n}\right)$ frequencies can make up a **significant fraction** of the whole population

Basic model

- Discrete domain X of size t and set of labels Y
- Learning problem: meta-distribution Φ over data distributions on $X \times Y$
- Goal: minimize expected generalization error of algorithm A

$$\overline{\mathrm{err}}_{\boldsymbol{n}}(\Phi, A) := \mathop{\mathbf{E}}_{P \sim \Phi,\, S \sim P^n,\, \hat{h} = A(S)} \left[\mathrm{err}_P\left(\hat{h}\right)\right]$$

- Prior over frequencies $\pi = (\pi_1, \dots, \pi_t)$
- For each $x \in X$, pick randomly and independently a frequency p_x from π and normalize

$$D(x) = \frac{p_x}{\sum_{x' \in X} p_{x'}}$$

- Arbitrary distribution F over functions $X \to Y$
- $\Phi(\pi, F)$: pick $D \sim \pi^t$ and $f \sim F$. Define P as $(x, f(x))$ for $x \sim D$

Benefits of fitting

Thm: For every algorithm A:

$$\mathop{\mathbf{E}}_{P\sim\Phi(\pi,F),\, S\sim P^n,\, \hat{h}=A(S)}\left[\mathrm{err}_P(\hat{h})\right] \geq \mathrm{opt}_n(\pi,F) + \boldsymbol{\tau} \cdot \mathop{\mathbf{E}}_{P\sim\Phi(\pi,F),\, S\sim P^n,\, \hat{h}=A(S)}\left[\mathrm{errn}_S(\hat{h},1)\right]$$

$\min_A \overline{\mathrm{err}}_n(\Phi(\pi,F),A)$

$$\tau := \frac{\mathbf{E}_{\alpha\sim\pi^*}\left[\alpha^2(1-\alpha)^{n-1}\right]}{\mathbf{E}_{\alpha\sim\pi^*}\left[\alpha(1-\alpha)^{n-1}\right]}$$

Number of examples occurring once in S but not fit by $\hat{h}$

For all π, n, $\quad \tau = \Omega\left(\frac{\mathrm{weight}_\pi\left[\frac{1}{2n},\frac{2}{n}\right]}{n}\right)$

For Zipf and $n \leq t$, $\tau = \Omega\left(\frac{1}{n \log t}\right)$

F can be arbitrarily complex

Extends to label noise

Fitting and memorization

Def: For dataset $S = ((x_1, y_1), \dots, (x_n, y_n))$, $i \in [n]$ and learning algorithm A let

$$\mathrm{mem}(A, S, i) = \Pr_{\hat{h}=A(S)}\left[\hat{h}(x_i) = y_i\right] - \Pr_{\hat{h}=A(S\setminus i)}\left[\hat{h}(x_i) = y_i\right]$$

For sufficiently complex F: for many $i \in [n]$

$$\Pr_{\hat{h}=A(S\setminus i)}\left[\hat{h}(x_i) = y_i\right] \ll 1$$

Thus need to be memorized to fit

Empirical validation

Def: For dataset $S = ((x_1, y_1), \dots, (x_n, y_n))$, $i \in [n]$ and learning algorithm A let

$$\text{mem}(A, S, i) = \Pr_{\hat{h}=A(S)}\left[\hat{h}(x_i) = y_i\right] - \Pr_{\hat{h}=A(S\setminus i)}\left[\hat{h}(x_i) = y_i\right]$$

Requires training $\Omega(n)$ models. Computationally infeasible!

Def: For $m \leq n$, let T be a random subset of S of size m that includes i

$$\text{mem}_m(A, S, i) = \mathop{\mathbf{E}}_{T}\left[\Pr_{\hat{h}=A(T)}\left[\hat{h}(x_i) = y_i\right] - \Pr_{\hat{h}=A(T\setminus i)}\left[\hat{h}(x_i) = y_i\right]\right]$$

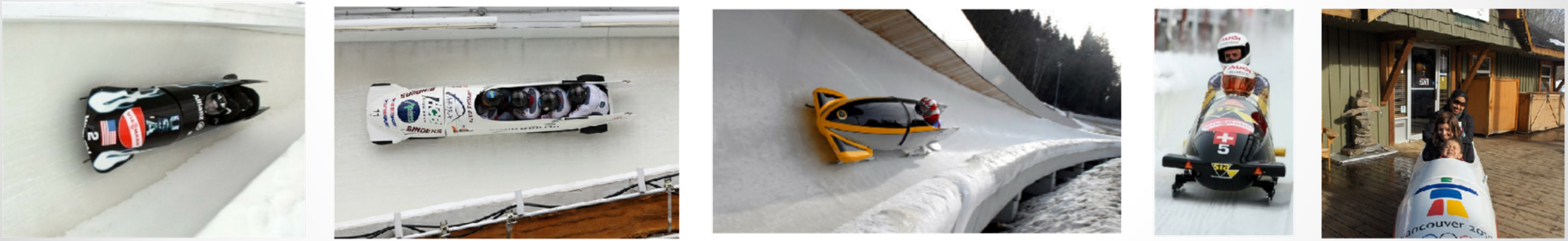
Closely-related and requires training $O(1)$ models! Use $m = 0.7n$

Examples of memorization estimates

ImageNet: **bobsled** class

High memorization estimate

Low memorization estimate



Influence of memorized examples

Def: For dataset $S = ((x_1, y_1), \dots, (x_n, y_n))$, $i \in [n]$ and testing example (x, y)

$$\mathrm{infl}(A, S, i, (x, y)) = \Pr_{\hat{h}=A(S)}\left[\hat{h}(x) = y\right] - \Pr_{\hat{h}=A(S\setminus i)}\left[\hat{h}(x) = y\right]$$

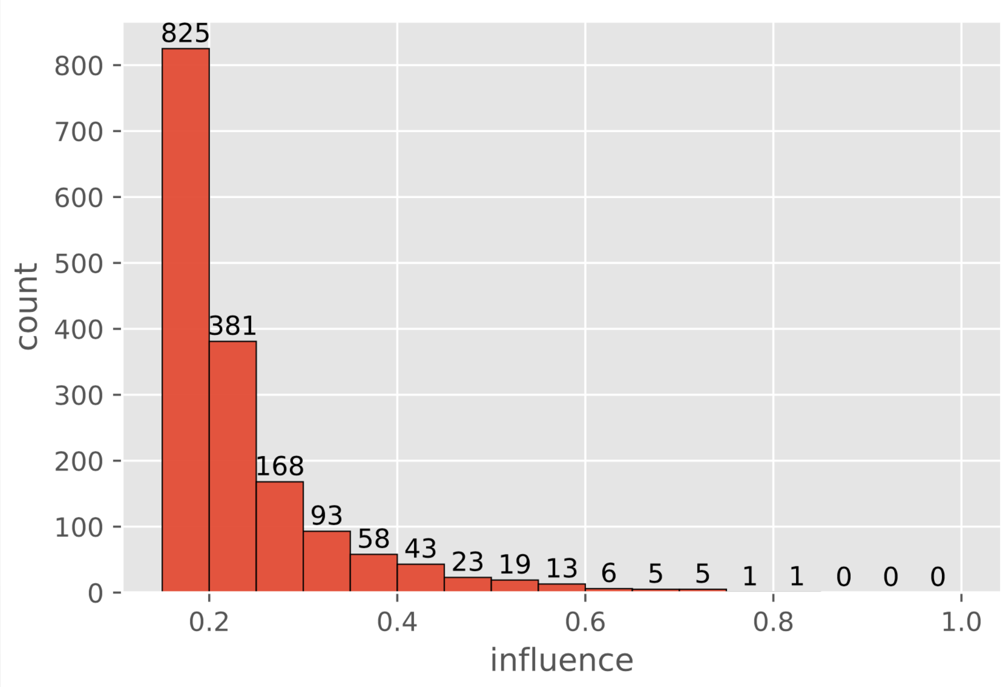
Same trick!

Def: For $m \leq n$, let T be a random subset of S of size m that includes i

$$\mathrm{infl}_m(A, S, i, (x, y)) = \mathbf{E}_T\left[\Pr_{\hat{h}=A(T)}\left[\hat{h}(x) = y\right] - \Pr_{\hat{h}=A(T\setminus i)}\left[\hat{h}(x) = y\right]\right]$$

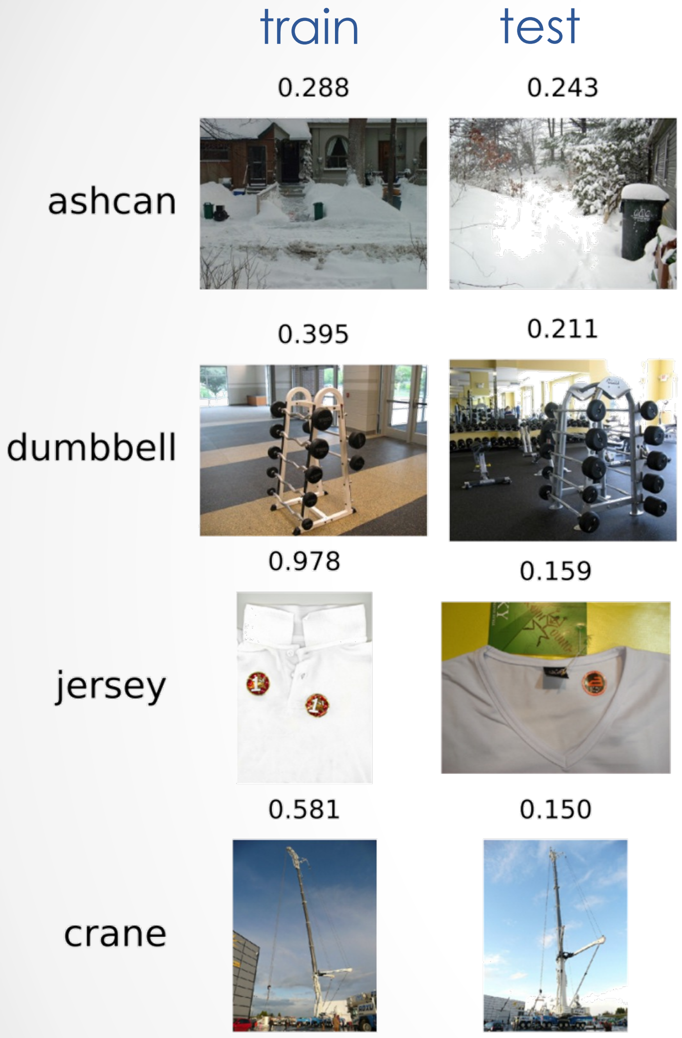
High-influence pairs

Many memorized training examples each of which significantly influences the accuracy on just one test example

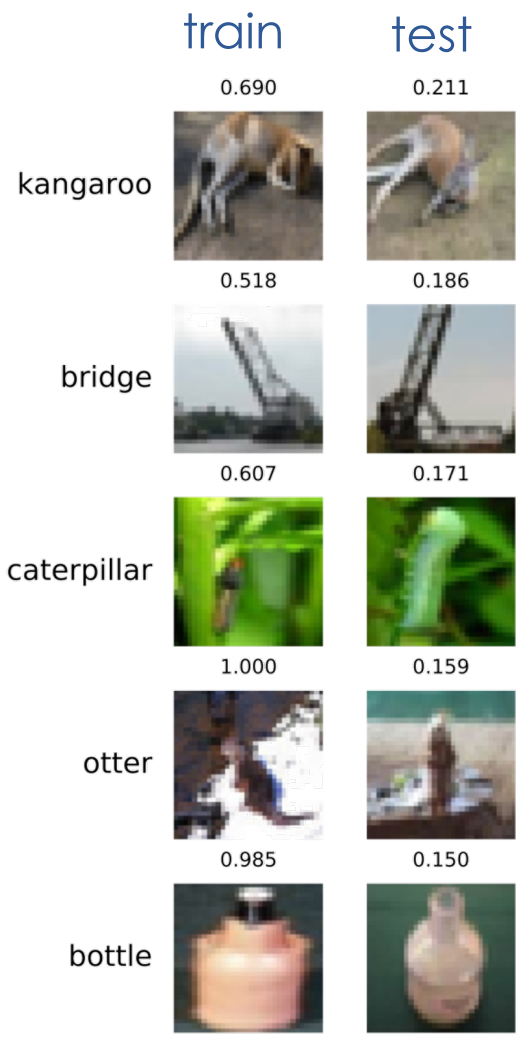


Affects ≈ 3% of the ImageNet test set

Examples of high-influence



ImageNet



CIFAR-100

Implications: differential privacy

Def: For dataset $S = ((x_1, y_1), \dots, (x_n, y_n))$, $i \in [n]$ and learning algorithm A let

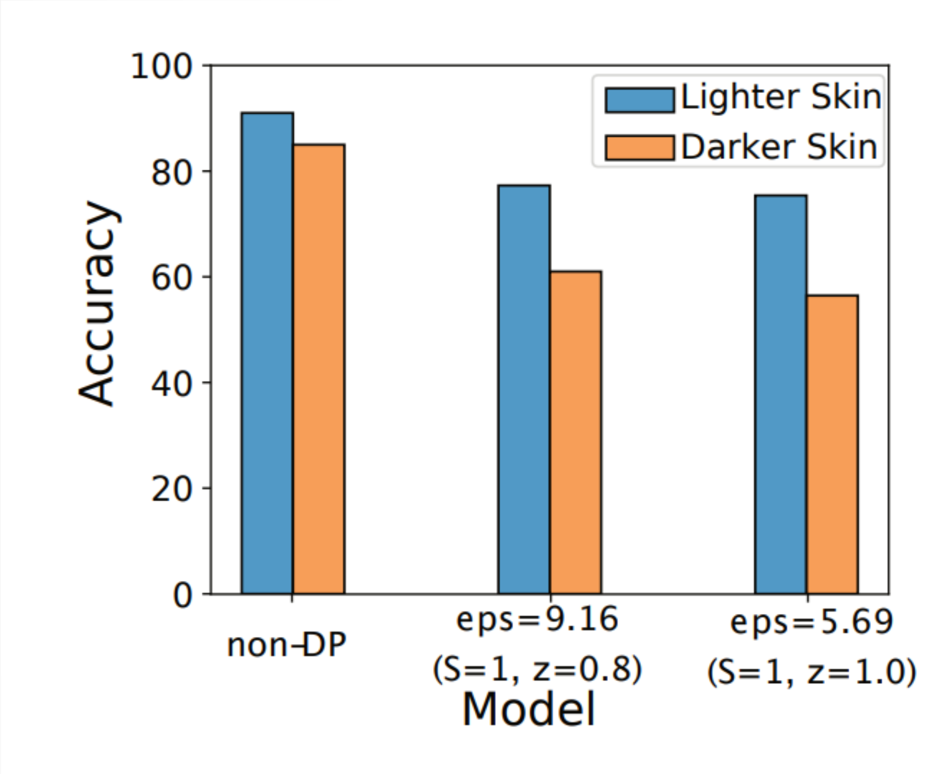
$$\mathrm{mem}(A, S, i) = \Pr_{\hat{h}=A(S)}\left[\hat{h}(x_i) = y_i\right] - \Pr_{\hat{h}=A(S\backslash i)}\left[\hat{h}(x_i) = y_i\right]$$

For an (ϵ, δ)-DP algorithm A:

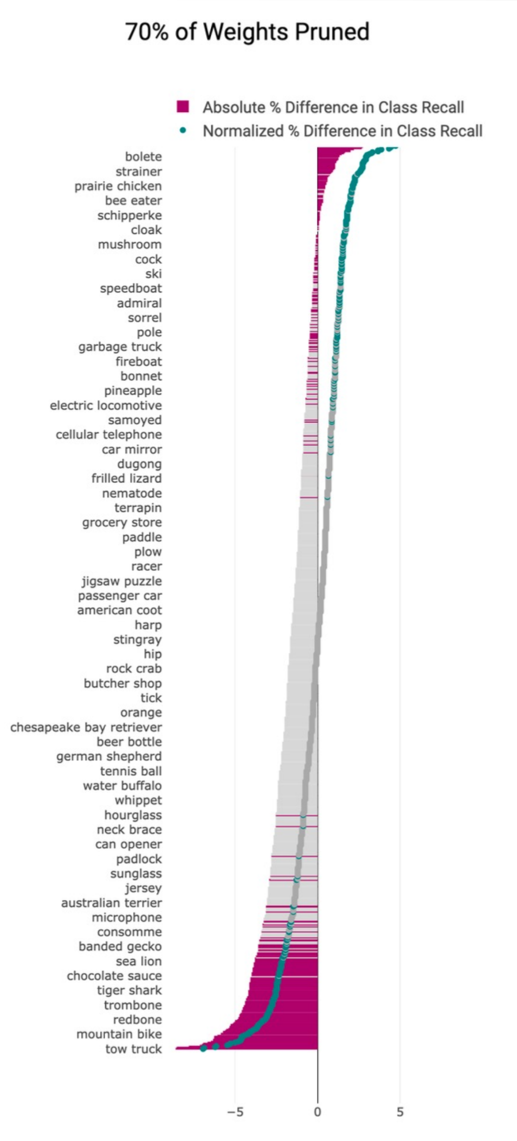
$$\Pr_{\hat{h}=A(S)}\left[\hat{h}(x_i) = y_i\right] \le e^{\epsilon} \cdot \Pr_{\hat{h}=A(S\backslash i)}\left[\hat{h}(x_i) = y_i\right] + \delta$$

$$\mathrm{mem}(A, S, i) \le (e^{\epsilon} - 1) \cdot \Pr_{\hat{h}=A(S\backslash i)}\left[\hat{h}(x_i) = y_i\right] + \delta$$

Implications: disparate effects



[Bagdarasyan, Shmatikov 19]



Cost of non-memorization depends on subgroup size, subpopulation structure, relative hardness of the problem

[Hooker,Courville,Clark,Dauphine,Frome 19]

Conclusions

- Label memorization is necessary for optimal generalization on long-tailed data distributions
  - Not algorithm-specific
  - Can be modelled theoretically
  - Implications of limiting memorization: effect on the accuracy depends on subgroup frequency and other properties

Additional materials

- **Theory**: STOC 2020, https://arxiv.org/abs/1906.05271
- **Experiments**: NeurIPS 2020, https://arxiv.org/abs/2008.03703
- **Results**/additional examples: https://pluskid.github.io/influence-memorization/
- **Follow-up:** memorization of data points
  - with Gavin Brown, Mark Bun, Adam Smith and Kunal Talwar
  - STOC 2021: https://arxiv.org/abs/2012.06421

Future work

- Faithful models of learning
- Applications of estimators