Towards Next-Generation ML/AI: Robustness, Optimization, Privacy

Krishna Pillutla
January 16th, 2023 @ IIT Madras
ML/AI have been revolutionized in the last 10 years
Top-5 Error %

[Krizhevsky, Sutskever, Hinton (NeurIPS 2012)]
Language modeling

\[ x_1 \quad x_2 \quad \ldots \quad x_t \]

Shall we go to the ___

[Markov (1913), Shannon (1948)]

\[ P(x_{t+1} | x_1, \ldots, x_t) \]
data

Language model
Mikolov et al. [Interspeech, 2010]
Federated learning: modern distributed learning

[McMahan et al. (AISTATS 2017)]

Percentage of world population with a smartphone

Data Credit: Business Wire
Federated learning: modern distributed learning

Communication cost > computation cost!

Percentage of world population with a smartphone

Data Credit: Business Wire
Federated learning: modern distributed learning

(Differential) Privacy guarantees

Data Credit: Business Wire
Data remains decentralized and private
Percentage of world population with a smartphone

Year


Percentage


Server

\[\sum\]

data

training

Machine learning

Server training data

Large Language Models $\rightarrow$ massive progress in NLP

GPT-3, PaLM, LaMDA, ChatGPT, ...

[Brown et al. (2020), ...]
Large Language Models (LLMs)

Stunning text generation capabilities

ChatGPT

Examples
"Explain quantum computing in simple terms" →

Capabilities
Remembers what user said earlier in the conversation

Limitations
May occasionally generate incorrect information

Scaling up \(\rightarrow\) progress in all of AI

Test loss of language modeling

[1E+03, 1E+05, 1E+07, 1E+09]

\# Model Parameters

[1E+03, 1E+05, 1E+07, 1E+09]

→ Foundation/platform models

[Kaplan, McCandlish et al. (2020)]

[Saharia et al. (2022), Jumper et al. (2021), Hsu et al. (2021), Bommasani et al. (2021)]
New capabilities are emerging

Generative AI: LLMs can write long essays now!

>> prompt: In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously, unexplored valley, in the Andes Mountains.

Continuation. The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown …

In-context learning & Zero-shot prediction

>> prompt: English: Hello!
French:

GPT-2

GPT-3 English: Hello!
French: Bonjour!

Test loss of language modeling

# Model Parameters

[Kaplan, McCandlish et al. (2020)]
Language modeling in 2023

Data → Language model

Language modeling in 2023

Federated learning

Large language models

Test loss of language modeling

[Kaplan, McCandlish et al. (2020)]
Challenges

Robustness to deployment conditions that differ from training

*Federated learning*: train-test mismatch
We tested Amazon's Alexa and Google's Home to see how people with accents are getting left behind in the smart-speaker revolution.
Challenges

Robustness to deployment conditions that differ from training

Federated learning: train-test mismatch

Large language models: emergent capabilities
Why Meta’s latest large language model survived only three days online

Galactica was supposed to help scientists. Instead, it mindlessly spat out biased and incorrect nonsense.

By Will Douglas Heaven

November 18, 2022
Challenges

Robustness to deployment conditions that differ from training

Robustness to outliers: adversarial or uncurated web data
Alexa and Siri Can Hear This Hidden Command. You Can’t.

Researchers can now send secret audio instructions undetectable to the human ear to Apple’s Siri, Amazon’s Alexa and Google’s Assistant.
Challenges

Robustness to deployment conditions that differ from training

Robustness to outliers: adversarial or uncurated web data
Challenges

Robustness to deployment conditions that differ from training

Robustness to outliers: adversarial or uncurated web data

Faster optimization: reduce communication and computation
Challenges

Robustness to deployment conditions that differ from training

Robustness to outliers: adversarial or uncurated web data

Faster optimization: reduce communication and computation

Privacy of user data
Federated learning


Robust Deployment

Robust to Outliers

Optimize Faster

Privacy

LLMs

NeurIPS 2021a
NeurIPS 2021b
Submitted 2023

Submitted 2022

NeurIPS 2018
Submitted 2022
Problem

Algorithms

Empirical
State-of-the-art performance

Theory
Analysis of convergence (statistical/optimization)
Federated learning

IEEE CISS 2021,
Springer SVVA 2021,
Mach. Learn. 2022

LLMs

NeurIPS 2021a
NeurIPS 2021b
Submitted 2023

IEEE Trans. Signal Proc. 2022,
ICML 2022

Submitted 2022

NeurIPS 2018
Submitted 2022
Federated learning


LLMs

NeurIPS 2021a
NeurIPS 2021b
Submitted 2023

Part 1


NeurIPS 2018
Submitted 2022

Submitted 2022
Part 1: Diagnosing large-scale text generation models with Mauve

[NeurIPS (2021a) Outstanding Paper Award, NeurIPS (2021b), Submitted (2023)]
Open-ended generative AI

- **New**: LLMs can write long essays!
- Widely deployed commercially
- LLMs still make mistakes

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously, unexplored valley, in the Andes Mountains.

**Continuation.** The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown...
Open-ended generation is an emergent capability

Deployment conditions differ from training

**Training**: Language modeling

Guess the next 1 word

**Deployment**: Sequential generation

Sample the next 500 words sequentially

Shall we go to the ___
In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously, unexplored valley, in the Andes Mountains.
How good is open-ended generation? The classical approach

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously, unexplored valley, in the Andes Mountains.

**Model Continuation.** The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown ...
How good is open-ended generation? The classical approach

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously, unexplored valley, in the Andes Mountains.

Model Continuation. The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown ...

Measure similarity/overlap

Human Reference. Known only to specialized cartographers as “Valle Escondido” or “Hidden valley”, this valley boasts of a wide variety of flora and fauna ...
How good is open-ended generation? The classical approach

**Prompt:** In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously, unexplored valley, in the Andes Mountains.

**Model Continuation.** The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown ...

**Continuation 2.** This discovery has kicked off an all-out search for other mythical creatures from the frozen reaches of the Antarctic to the tropical islands of the Pacific ...

**Continuation 3.** Perhaps most astonishingly, these unicorns have developed their own artificial general intelligence named Yuyaysapa ...

**Human Reference.** Known only to specialized cartographers as “Valle Escondido” or “Hidden valley”, this valley boasts of a wide variety of flora and fauna ...

**Measure similarity/overlap**
Problem statement

How close are the *probability distributions* over text sequences?
Two types of errors in text generation

Denote $P$ for the human text distribution and $Q$ for the model text distribution.

**Type I error:** $Q$ places high mass on text unlikely under $P$ (e.g., degenerate text).

**Type II error:** $Q$ cannot produce text plausible under $P$ (e.g., due to nucleus sampling).

### Experiments

<table>
<thead>
<tr>
<th>Text Probability</th>
<th>Type I Error</th>
<th>Type II Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>The time is</td>
<td>The time is</td>
<td>I just visited</td>
</tr>
<tr>
<td>the time</td>
<td>the time</td>
<td>Utqiagvik and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nuchalawoyya</td>
</tr>
</tbody>
</table>

$P$: human distribution

$Q$: machine distribution
Two Types of Errors

I Denote $P$ for the human text distribution and $Q$ for the model text distribution.

I Type I error: $Q$ places high mass on text unlikely under $P$ (e.g., degenerate text).

I Type II error: $Q$ cannot produce text plausible under $P$ (e.g., due to nucleus sampling).

Type I Error: The time is the time is the time is the time...

$Q$ places high mass on text unlikely under $P$ (e.g. degenerate text)

$P$: human distribution

$Q$: machine distribution
Two Types of Errors

I Denote $P$ for the human text distribution and $Q$ for the model text distribution.

I Type I error: $Q$ places high mass on text unlikely under $P$ (e.g., degenerate text).

I Type II error: $Q$ cannot produce text plausible under $P$ (e.g., due to nucleus sampling).

\[
KL(Q \mid P) = \sum_x P(x) \log \frac{P(x)}{Q(x)}
\]

\[
KL(P \mid Q) = \sum_x Q(x) \log \frac{Q(x)}{P(x)}
\]
Mauve: summarizing both errors

- KL($Q|P$) and KL($P|Q$) can be infinite, so measure errors softly using mixtures

- **Divergence Curve:** Varying the mixture weight

- **Mauve:** area summary of the curve: a quantitative measure of similarity and takes values between 0 (dissimilar) and 1 (identical)
Computing **Mauve in practice**

- Sum over documents intractable

\[
\text{KL}(Q|R) = \sum_x Q(x) \log \frac{Q(x)}{R(x)}
\]

- Computation pipeline

Text \(x\) → **LLM** Encoding \(M(x)\) → **Embedding** \(M(x)\) → **\(k\)-means clustering** \(q(M(x))\) → Multinomial
Correlation with human judgements

Goals of automatic evaluation

- Humans are the end users, so human evaluation is the ultimate test
- Human evaluation is slow and expensive

If Mauve can correlate with human evaluations, faster iterations + debugging
Correlation with human judgements

**Head-to-head**: Is A or B more (a) human-like, (b) interesting, (c) sensible?
We compare text written by humans and 8 models

Spearman Correlation w/ human eval (↑)

<table>
<thead>
<tr>
<th>Category</th>
<th>Mauve</th>
<th>Gen. PPL.</th>
<th>Self-BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human-like</td>
<td>0.95</td>
<td>0.81</td>
<td>0.60</td>
</tr>
<tr>
<td>Interesting</td>
<td>0.81</td>
<td>0.64</td>
<td>0.74</td>
</tr>
<tr>
<td>Sensible</td>
<td>0.86</td>
<td>0.74</td>
<td>0.52</td>
</tr>
</tbody>
</table>

Gen. PPL.: Holtzman et al. (ICLR 2020)  
Self-BLEU: Zhu et al. (2018)
Mauve captures important trends

- Y-axis shows Mauve (↑)
- Mauve captures all the trends while baselines fail
Mauve: Estimation theory

Estimation of Mauve involves two approximations:

- **Clustering/Quantization**
- **Estimate from samples**

\[ P, Q \]

- high-dimensional text distributions

\[ P_S, Q_S \]

- quantized distributions

\[ \hat{P}_{S,n}, \hat{Q}_{S,n} \]

- samples (empirical distributions)
There exists a quantization of size $k$ such that the approximation error of Mauve from $n$ samples is

$$
\widetilde{\sigma} \left( \sqrt{\frac{k}{n}} + \frac{1}{k} \right)
$$

$n$: number of samples from $P$ and $Q$

$k$: quantization size (Num. clusters)

Balance both by choosing $k = \Theta(n^{1/3})$
Mauve: Beyond clustering

Text $x$ $\rightarrow$ LLM Encoding $\rightarrow$ Embedding $M(x)$ $\rightarrow$ $k$-means clustering $\rightarrow$ Multinomial $q(M(x))$ $\rightarrow$ KL divergence $\rightarrow$ Mauve
Mauve: Beyond clustering

Text $x$ → LLM Encoding → Embedding $M(x)$ → $k$-means clustering → Multinomial $q(M(x))$ → KL divergence → Mauve

- Nearest neighbor estimator
- Kernel density estimator
- Parametric Gaussian approx.
- Classifier-based estimation
Mauve: Beyond clustering

Text $x$ → LLM Encoding → Embedding $M(x)$ → $k$-means clustering → Multinomial $q(M(x))$ → KL divergence → Mauve

General $f$-divergences
Optimal transport
Mauve: Beyond clustering

Text x \rightarrow LLM Encoding \rightarrow Embedding M(x) \rightarrow k\text{-means clustering} \rightarrow Multinomial q(M(x)) \rightarrow KL divergence \rightarrow Mauve

Essential
Software

```
from mauve import compute_mauve

p_text = ... # list of strings for human distribution P
q_text = ... # list of strings for model distribution Q

# Obtain deep encoding, quantize it and compute Mauve
out = compute_mauve(p_text=p_text, q_text=q_text)

print(f'Mauve(P, Q) = {out.mauve}')</n```

```
from evaluate import mauve

mauve = load("mauve")

# Obtain deep encoding, quantize it and compute Mauve
out = mauve.compute(references=p_text, predictions=q_text)

print(f'Mauve(P, Q) = {out.mauve}')</n```

pip install mauve-text

HuggingFace Evaluate: pip install evaluate

60K downloads
Impact of Mauve

Standard metric for evaluation and hyper-parameter tuning

Meister et al. (TACL 2022)  Jawahar et al. (ACL 2022)
Su et al. (NeurIPS 2022)    Hewitt et al. (EMNLP 2022)
Lu et al. (NeurIPS 2022)    Mattern et al. (EMNLP 2022)
Xu et al. (NeurIPS 2022)    Hu et al. (NAACL 2022)
Summary: Diagnosing large-scale text generation with Mauve

Used for open-ended generation but trained for language modeling. How good is it?

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains.

Continuation. The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown …

Continuation 2. This discovery has kicked off an all-out search for other mythical creatures from the frozen reaches of the Antarctic to the tropical islands of the Pacific …

Continuation 3. Perhaps most astonishingly, these unicorns have developed their own artificial general intelligence named Yuyaysapa …

>> prompt:

Our approach correlates with human judgements and quantifies observed behavior

Spearman Correlation w/ human eval (↑)

<table>
<thead>
<tr>
<th></th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mauve</td>
<td>0.60</td>
<td>0.81</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Baseline 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary: Diagnosing large-scale text generation with Mauve

Used for open-ended generation but trained for language modeling. How good is it?

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains.

Continuation. The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown …

Continuation 2. This discovery has kicked off an all-out search for other mythical creatures from the frozen reaches of the Antarctic to the tropical islands of the Pacific …

Continuation 3. Perhaps most astonishingly, these unicorns have developed their own artificial general intelligence named Yuyaysapa …

Directly measure the gap between distributions

| Type I Error: The time is the time is the time ... |
| Type II Error: I just visited Utqiagvik and Nuchalawoyya in Alaska. |

<table>
<thead>
<tr>
<th>Q: machine</th>
<th>P: human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Probability</td>
<td></td>
</tr>
<tr>
<td>Type I Error</td>
<td>Type II Error</td>
</tr>
<tr>
<td>The time is the time is the time ...</td>
<td>I just visited Utqiagvik and Nuchalawoyya in Alaska.</td>
</tr>
</tbody>
</table>

Our approach correlates with human judgements and quantifies observed behavior using Spearman Correlation with human eval:

<table>
<thead>
<tr>
<th>Correlation</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60</td>
<td>0.60</td>
<td>0.81</td>
<td>0.95</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Mauve Baseline 1 Baseline 2
Summary: Diagnosing large-scale text generation with Mauve

Used for open-ended generation but trained for language modeling. How good is it?

Directly measure the gap between distributions

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains.

Continuation. The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown ...

Continuation 2. This discovery has kicked off an all-out search for other mythical creatures from the frozen reaches of the Antarctic to the tropical islands of the Pacific ...

Continuation 3. Perhaps most astonishingly, these unicorns have developed their own artificial general intelligence named Yuyaysapa ...

Our approach correlates with human judgements and quantifies observed behavior

<table>
<thead>
<tr>
<th>Text Probability</th>
<th>Mauve</th>
<th>Baseline 1</th>
<th>Baseline 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type I Error: The time is the time is the time is the time ...</td>
<td>0.95</td>
<td>0.81</td>
<td>0.60</td>
</tr>
<tr>
<td>Type II Error: I just visited Utqiagvik and Nuchalawoyya in Alaska.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Spearman Correlation w/ human eval (↑)
Summary: Diagnosing large-scale text generation with Mauve

Used for open-ended generation but trained for language modeling. How good is it?

In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains.

Continuation. The scientists named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown ...

Continuation 2. This discovery has kicked off an all-out search for other mythical creatures from the frozen reaches of the Antarctic to the tropical islands of the Pacific ...

Continuation 3. Perhaps most astonishingly, these unicorns have developed their own artificial general intelligence named Yuyaysapa ...

Directly measure the gap between distributions

Theory: error bounds

\[ \hat{d} \left( \frac{\sqrt{k}}{\sqrt{n}} + \frac{1}{k} \right) \]

Text Probability

<table>
<thead>
<tr>
<th>Type I Error</th>
<th>Type II Error</th>
</tr>
</thead>
</table>
| The time is the time is the time is the time is the time is | I just visited Utqiagvik and Nuchalawooya in Alaska.
|                  |               |

Statistical Error

Quantization Error
Federated learning


Part 2

Robust Deployment

LLMs

NeurIPS 2021a
NeurIPS 2021b
Submitted 2023


Submitted 2022

NeurIPS 2018
Submitted 2022
Part 2: Tackling distribution shifts in federated learning

[IEEE CISS ’21, DistShift-NeurIPS ’22 (Spotlight), SVVA ’21, Mach. Learn. ’22]
Usual Learning Objective

\[
\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} F_i(w) \quad \text{where} \quad F_i(w) = \mathbb{E}_{z \sim p_i} [f(w; z)]
\]

loss on client \(i\)

[McMahan et al. (AISTATS 2017), Kairouz et al. (2021)]
Usual approach $\implies$ Global model is trained on average distribution across clients
Global model is deployed on *individual* clients
Global model is deployed on *individual* clients
Global model is deployed on *individual* clients.
Our goal: Reduce *tail* error
Our goal

Reduce *tail* error without sacrificing the *mean* error
Simplicial federated learning

Our Approach: minimize the tail error directly!

Simplicial-FL Objective:

$$\min_w S_\theta\left( (F_1(w), \ldots, F_n(w)) \right)$$

Superquantile | Conditional Value at Risk

[Rockafellar & Uryasev (2002)]
**Dual expression** ≡ continuous knapsack problem

\[
\mathcal{S}_\theta(x_1, \ldots, x_n) = \max_{\pi} \left\{ \sum_i \pi_i x_i : \pi_i \geq 0, \sum_i \pi_i = 1, \pi_i \leq (n\theta)^{-1} \right\}
\]

[Dantzig (1957), Ben-Tal & Teboulle (1987), Föllmer & Schied (2002)]
**Dual expression** $\equiv$ continuous knapsack problem

$$S_\theta(x_1, \cdots, x_n) = \max_\pi \left\{ \sum_i \pi_i x_i : \pi_i \geq 0, \sum_i \pi_i = 1, \pi_i \leq (n\theta)^{-1} \right\}$$

[Dantzig (1957), Ben-Tal & Teboulle (1987), Föllmer & Schied (2002)]

Assuming a new test client with mixture distribution $p_\pi = \sum_i \pi_i p_{ir}$

Simplicial-FL objective is equivalent to:

$$\min_w \max_{\pi: \pi_i \leq (n\theta)^{-1}} \mathbb{E}_{z \sim p_\pi} \left[ f(w; z) \right]$$

$\Rightarrow$ Distributionally robust learning
Optimization
Usual Algorithm (FedAvg):

$$\min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w)$$

Our Algorithm:

$$\min_w \mathbb{S}_{\theta} \left( F_1(w), \ldots, F_n(w) \right)$$

FedAvg [MacMahan et al. (AISTATS 2017)]

Parallel Gradient Distribution [Mangasarian. (SICON 1995)]
Iterative Parameter Mixing [McDonald et al. (ACL 2009)]
BMUF [Chen & Huo. (ICASSP 2016)]
Local SGD [Stich. (ICLR 2019)]
Usual Algorithm (FedAvg):

\[ \min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w) \]

Our Algorithm:

\[ \min_w S_{\theta}(F_1(w), \ldots, F_n(w)) \]

**Step 1 of 3:** Server samples \( m \) clients and broadcasts global model
Usual Algorithm (FedAvg):

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w)
\]

Our Algorithm:

\[
\min_w \mathbb{S}_\theta \left( (F_1(w), \ldots, F_n(w)) \right)
\]

*Step 2 of 3: Clients perform local gradient descent on their local data*
Usual Algorithm (FedAvg):

$$\min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w)$$

**Step 3 of 3: Aggregate updates contributed by all clients**

Our Algorithm:

$$\min_w \mathbb{S}_{\theta} \left( (F_1(w), \ldots, F_n(w)) \right)$$

**Step 3 of 3: Aggregate updates contributed by tail clients only**
Experiments: EMNIST
Histogram of per-client errors

Misclassification Error

Ours

Usual
Simplicial-FL has the smallest 90\textsuperscript{th} percentile error.

Simplicial-FL is competitive on the mean error.
Convergence analysis (non-convex)

Faster optimization: reduce communication
Challenge #1:

The superquantile is non-smooth

plot of $h(u_1, u_2) = S_{1/2}(u_1, u_2, 0, 0)$
Subgradient illustration

**Smooth**

\[ f(x) = f(c) + f'(c)(x - c) \]

**Non-smooth**

\[ f(x) = f(c) + s(x - c) \]
Nonsmooth: The subgradient has a tractable form

\[ \partial F_\theta(w) \equiv \sum_{i=1}^{n} \pi_i^* \nabla F_i(w) \quad \text{where} \quad \pi_i^* \propto \mathbb{1}\left( F_i(w) \geq Q_\theta(F_1(w), \ldots, F_n(w)) \right) \]

assuming \( \theta_n \) is an integer.

![Diagram](image.png)
**Nonsmooth**: The subgradient has a tractable form

$$\partial F_\theta(w) \ni \sum_{i=1}^{n} \pi_i^* \nabla F_i(w) \quad \text{where} \quad \pi_i^* \propto \mathbb{1}\left(F_i(w) \geq Q_\theta(F_1(w), \ldots, F_n(w))\right)$$ assuming $\theta_n$ is an integer

**Proof** Chain rule $\Rightarrow$ subgradient holds with

$$\pi^* \in \arg \max_{\pi \in \mathcal{P}_\theta} \sum_{i} \pi_i F_i(w)$$

Alternate form of $\pi^*$ comes from the continuous knapsack problem

[Dantzig. ORIJ (1957)]
**Nonsmooth:** The subgradient has a tractable form

\[ \partial F_\theta(w) \equiv \sum_{i=1}^{n} \pi_i^* \nabla F_i(w) \quad \text{where} \quad \pi_i^* \propto \mathbb{1}\left(F_i(w) \geq Q_\theta(F_1(w), \ldots, F_n(w))\right) \]

assuming \( \theta_n \) is an integer

**Other option:** Use smoothing

![Graph showing smoothed and original loss at rank i]

Challenge #2

The superquantile is *nonlinear*  
\[ \implies \text{unbiased stochastic gradients not possible} \]

For i.i.d. copies \( Z_1, \ldots, Z_m \) of \( Z \), we have

\[
\mathbb{E} \left[ \frac{1}{m} \sum_{i=1}^{m} Z_i \right] = \mathbb{E}[Z] \quad \text{but} \quad \mathbb{E} \left[ S_\theta(Z_1, \ldots, Z_m) \right] \neq S_\theta(Z)
\]
**Nonlinear:** We minimize a close surrogate

\[
\bar{F}_\theta(w) = \mathbb{E}_{S: |S|=m} \left[ S_\theta \left( \left( F_i(w) : i \in S \right) \right) \right]
\]

The surrogate is uniformly close for bounded losses:

For i.i.d. copies \(Z_1, \ldots, Z_m\) of \(Z\) with \(|Z| \leq B\) a.s., we have

\[
\left| \mathbb{E}[S_\theta(Z_1, \ldots, Z_m)] - S_\theta(Z) \right| \leq \frac{B}{\sqrt{\theta m}} \quad \text{Var}
\[
\left\| S_\theta(Z_1, \ldots, Z_m) \right\| \leq \frac{B^2}{\theta m}
\]

[Levy et al. (NeurIPS 2020)]
**Theorem** [P., Laguel, Malick, Harchaoui]

Suppose each \( F_i \) is \( L \)-smooth and \( G \)-Lipschitz.

Then, Simplicial-FL satisfies the convergence guarantee:

\[
\mathbb{E} \left\| \nabla \Phi_\theta^{2L}(w_t) \right\|^2 \leq \sqrt{\frac{\Delta_0 L G^2}{t}} + \left(1 - \tau\right)^{1/3} \left(\frac{\Delta_0 L G}{t}\right)^{2/3} + \frac{\Delta_0 L}{t}
\]

\( t \): #comm. rounds  
\( \tau \): #local update steps  
\( \Delta_0 \): initial error

\[ \Phi_\theta^\mu(w) = \inf_z \left\{ \overline{F}_\theta(z) + \frac{\mu}{2} \| z - w \|^2 \right\} \quad \leftarrow \text{Moreau envelope of } F_\theta \text{ | well defined for } \mu > L \]
Suppose each $F_i$ is $L$-smooth and $G$-Lipschitz.

Then, Simplicial-FL satisfies the convergence guarantee:

$$
\mathbb{E} \left\| \nabla \Phi^{2L}(w_t) \right\|^2 \leq \sqrt{\frac{\Delta_0LG^2}{t}} + (1 - \tau)^{1/3} \left( \frac{\Delta_0LG}{t} \right)^{2/3} + \frac{\Delta_0L}{t}
$$

$t$: #comm. rounds  
$\tau$: #local update steps  
$\Delta_0$: initial error

$$
\Phi^\mu_\theta(w) = \inf_z \left\{ \mathcal{F}_\theta(z) + \frac{\mu}{2} \|z - w\|^2 \right\} \quad \text{Moreau envelope of } \mathcal{F}_\theta \text{ | well defined for } \mu > L
$$
Theorem [P., Laguel, Malick, Harchaoui]

Suppose each $F_i$ is $L$-smooth and $G$-Lipschitz.

Then, Simplicial-FL satisfies the convergence guarantee:

$$
\mathbb{E} \left\| \nabla \Phi^2_L(w_t) \right\|^2 \leq \sqrt{\frac{\Delta_0 LG^2}{t}} + (1 - \tau)^{1/3} \left( \frac{\Delta_0 LG}{t} \right)^{2/3} + \frac{\Delta_0 L}{t}
$$

$t$: #comm. rounds
$\tau$: #local update steps
$\Delta_0$: initial error

$$
\Phi^\mu_\theta(w) = \inf_{z} \left\{ \overline{F}_\theta(z) + \frac{\mu}{2} \|z - w\|^2 \right\}
$$

Moreau envelope of $F_\theta$ | well defined for $\mu > L$
**Theorem** [P., Laguel, Malick, Harchaouï]

Suppose each $F_i$ is $L$-smooth and $G$-Lipschitz.

Then, Simplicial-FL satisfies the convergence guarantee:

$$
\mathbb{E} \left\| \nabla \Phi^2_{\theta}(w_t) \right\|^2 \leq \sqrt{\frac{\Delta_0 L G^2}{t}} + (1 - \tau)^{1/3} \left( \frac{\Delta_0 L G}{t} \right)^{2/3} + \frac{\Delta_0 L}{t}
$$

$t$: # comm. rounds

$\tau$: # local update steps

$\Delta_0$: initial error

$$
\Phi^\mu_{\theta}(w) = \inf_{z} \left\{ \overline{F}_{\theta}(z) + \frac{\mu}{2} \|z - w\|^2 \right\} \quad \text{Moreau envelope of } F_{\theta} \text{ | well defined for } \mu > L
$$
Privacy analysis

Privacy of user data
Randomized Algorithm

Dataset

Output Distribution
(e.g. over models)
Randomized Algorithm + Dataset \rightarrow \text{Output Distribution (e.g. over models)}
Randomized Algorithm + Dataset → Output Distribution (e.g. over models)
A randomized algorithm is $\varepsilon$-differentially private if the addition of one user's data does not alter its output distribution by more than $\varepsilon$.

[Dwork, McSherry, Nissim, Smith (2006)]
Dataset

\[ \text{Randomized Algorithm} \]

Output Distribution (e.g. over models)

\[ \varepsilon \text{-differential privacy} \]

Large \( \varepsilon \) \( \implies \) more privacy leakage
Privacy goal

Extend our algorithm to make it **differentially private**

Our Objective:

\[
\min_{w} S_{\theta}\left( (F_{1}(w), \ldots, F_{n}(w)) \right)
\]

Our Algorithm: **Step 3 of 3: Aggregate updates contributed by tail clients only**

---

The diagram illustrates the concept of differentially private algorithms, focusing on tail clients and quantiles. The objective function is depicted, along with the cumulative distribution function (CDF) and quantiles for loss distribution.
Why is it challenging?

Usual Algorithm:

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w)
\]

Our Algorithm:

\[
\min_w \mathbb{E}_\theta \left( (F_1(w), \ldots, F_n(w)) \right)
\]

Private mean estimation of gradients

Non-linear
Communication primitive: secure sum

Only reveal $x_1 + x_2$ to the server without revealing $x_1$ or $x_2$

[Bonawitz et al. (CCS 2017), Bell et al. (CCS 2020)]
Perform all operations modulo $M$

[Bonawitz et al. (CCS 2017), Bell et al. (CCS 2020)]
Server only sees $x'_1, x'_2 \sim \text{Unif}(\bigcirc)$ but calculates the correct sum

\[ x'_1 = x_1 + \xi \]

\[ x'_2 = x_2 - \xi \]

\[ x'_1 + x'_2 = x_1 + x_2 \]

[Bonawitz et al. (CCS 2017), Bell et al. (CCS 2020)]
Server only sees $x_1', x_2' \sim \text{Unif}(\bigcirc)$ but calculates the correct sum

Client 1

\[ x_1' = x_1 + \xi \]

\[ \xi \sim \text{Unif}(\bigcirc) \]

Client 2

\[ x_2' = x_2 - \xi \]

Server

\[ x_1' + x_2' = x_1 + x_2 \]

Total communication for $m$ vectors in $\mathbb{R}^d = O(m \log m + md)$ numbers
Real-world communication constraint:
All client-to-server communication must go through secure summation
How to achieve non-linear aggregation with a secure sum?

Non-linear aggregate:

$$\min_{w} \mathbb{E}_{\theta} \left( (F_1(w), \ldots, F_n(w)) \right)$$

Secure sum

Client 1

Client 2

Server
Usual Algorithm (FedAvg):

$$\min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w)$$

**Step 3 of 3: Aggregate updates contributed by all clients**

Our Algorithm:

$$\min_w \mathbb{S}_\theta \left( (F_1(w), \ldots, F_n(w)) \right)$$

**Step 3 of 3: Aggregate updates contributed by tail clients only**
Histogram

\[ h_i' = h_i + \mathcal{N}(0, \sigma^2 I_b) \]

Differential privacy via discrete Gaussian noise

[Kairouz, Liu, Steinke. (ICML 2021)]
Proposition (informal) [P., Laguel, Malick, Harchaoui]

If we wish to compute the $\alpha$-quantile, our algorithm returns an $\varepsilon$-differentially private $\alpha'$-quantile where

$$|\alpha' - \alpha| \leq \frac{\sqrt{b}}{\varepsilon m}$$

Total communication cost $\approx bm \log^2 m$

$m$  #clients per round
$b$  #bins in the histogram
$\varepsilon$  privacy parameter
Our Algorithm:

\[
\min_w \frac{1}{n} \sum_{i=1}^{n} F_i(w)
\]

Step 3 of 3: Aggregate updates contributed by \textbf{all clients}

Usual Algorithm (FedAvg):

\[
\min_w \theta \left( (F_1(w), \ldots, F_n(w)) \right)
\]

Step 3 of 3: Aggregate updates contributed by \textbf{tail clients} only
Private mean estimation of (potentially zeroed out) gradients
Total privacy leakage =

Quantile privacy leakage + Parameter privacy leakage
Privacy of user data:

First end-to-end \textbf{differentially private} algorithm for distributionally robust federated learning

Algorithm requires \textbf{2} secure summations per update
Mean error

90\textsuperscript{th} percentile error

Privacy parameter $\epsilon$

- Usual
- 5 pp
- Ours

More private
Less private

0.5 - 0.6 pp

Ours
Usual

5 pp
0.5 - 0.6 pp

Less private
More private

Privacy parameter $\epsilon$
Distributionally robust learning with 1 additional line of code

```python
import torch.nn.functional as F
from sqwash import reduce_superquantile

for x, y in dataloader:
    y_hat = model(x)
    batch_losses = F.cross_entropy(y_hat, y, reduction='none')  # must set `reduction='none'`
    loss = reduce_superquantile(batch_losses, superquantile_tail_fraction=0.5)  # Additional line
    loss.backward()  # Proceed as usual from here
```

Install: **pip install sqwash**

Summary: Tackling distribution shifts in federated learning

Distribution shift ➝
large tail errors
Summary: Tackling distribution shifts in federated learning

Distribution shift → large tail errors

\[
\min_w S_\theta\left( (F_1(w), \ldots, F_n(w)) \right)
\]
Summary: Tackling distribution shifts in federated learning

Distribution shift $\rightarrow$ large tail errors

Our approach reduces tail error

$$\min_w S_\theta\left( (F_1(w), \ldots, F_n(w)) \right)$$

$$S_\theta(Z) = \mathbb{E}[Z \mid Z > Q_\theta(Z)]$$

Misclassification Error
Summary: Tackling distribution shifts in federated learning

Distribution shift $\Rightarrow$ large tail errors

\[
\min_w S_{\theta}\left( \left( F_1(w), \ldots, F_n(w) \right) \right)
\]

Convergence + Privacy analysis

$O(1/\sqrt{t})$ error rate after $t$ comm. rounds in the non-smooth, non-convex case

Differentially private algorithm for distributional robust federated learning
Future research plans
Challenges

Robustness to deployment conditions that differ from training

Robustness to outliers: adversarial or uncurated web data

Faster optimization: reduce communication and computation

Privacy of user data
Federated learning

IEEE CISS 2021, 
*Springer SVVA 2021*, 
*Mach. Learn. 2022*

**LLMs**

NeurIPS 2021a
NeurIPS 2021b
*Submitted 2023*

*IIEEE Trans. Signal Proc. 2022*, 
*ICML 2022*

**LLMs**

Submitted 2022

NeurIPS 2018
*Submitted 2022*

Future plan 1
What comes after federated learning?

LLMs

NeurIPS 2021a
NeurIPS 2021b
Submitted 2023

Submitted 2022

NeurIPS 2018
Submitted 2022

Future plan 2
Thank you!