Unsolved Problems in Audio, Speech, and Language Processing

Mark Hasegawa-Johnson

March 26, 2025

Outline

Why does text need more parameters than speech?

- A typical ASR has 1B trainable parameters
- A typical LLM has 100B parameters
- Why does text need more parameters than speech?
- Because an LLM is no longer considered useful unless it exhibits emergent properties. An ASR is considered useful if it just converts speech to text.
- If a large ASR (LASR?) demonstrated similar emergence, would it generate abilities that an ASR+LLM pipeline does not have?

Language Models are Few-Shot Learners

Brown, Mann et al., 2020

This 2020 article showed that, past about 100B parameters, a simple autoregressive language model (GPT-3) magically gains the ability to

- Answer questions
- Translate
- Perform word sense disambiguation, natural language inference, and coreference resolution

Language Models are Few-Shot Learners

Brown, Mann et al., 2020

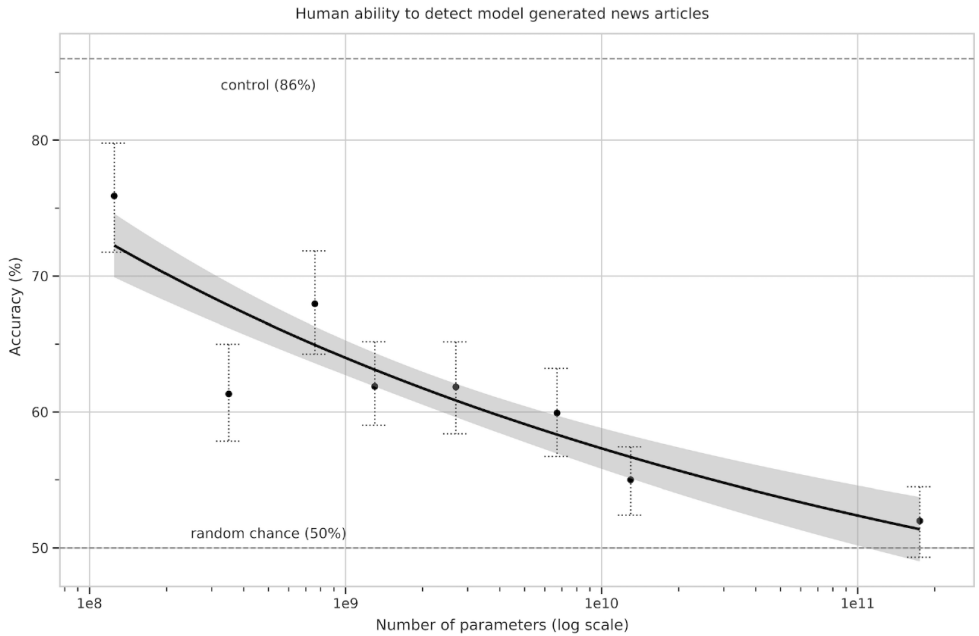


Figure 7.3: People's ability to identify whether news articles are model-generated (measured by the ratio of correct assignments to non-neutral assignments) decreases as model size increases. Accuracy on the outputs on the deliberately-bad control model (an unconditioned GPT-3 Small model with higher output randomness) is indicated with the dashed line at the top, and the random chance (50%) is indicated with the dashed line at the bottom. Line of best fit is a power law with 95% confidence intervals.

Brown et al., 2020, Fig. 7.3

Emergent Abilities of Language Models

Wei, Tay, Bommasani et al., 2022

An ability is emergent if it is not present in smaller models but is present in larger models. Emergent abilities would not have been directly predicted by extrapolating a scaling law (i.e. consistent performance improvements) from small-scale models. When visualized via a scaling curve (x-axis: model scale, y-axis: performance), emergent abilities show a clear pattern—performance is near-random until a certain critical threshold of scale is reached, after which performance increases to substantially above random.

Grokking

- Another form of emergence occurs not with unreasonably large parameter count, but with unreasonably long training time
- The model achieves 0% training error, 0 training loss, but its test loss is still 100%
- 100 iterations later, test loss suddenly decreases to 0%
- The model has "grokked" the problem, i.e., it has suddenly understood a pattern in the training data that can be generalized to test data

Emergence in non-neural models: Modular arithmetic via average gradient outer product

Mallinar et al., 2024

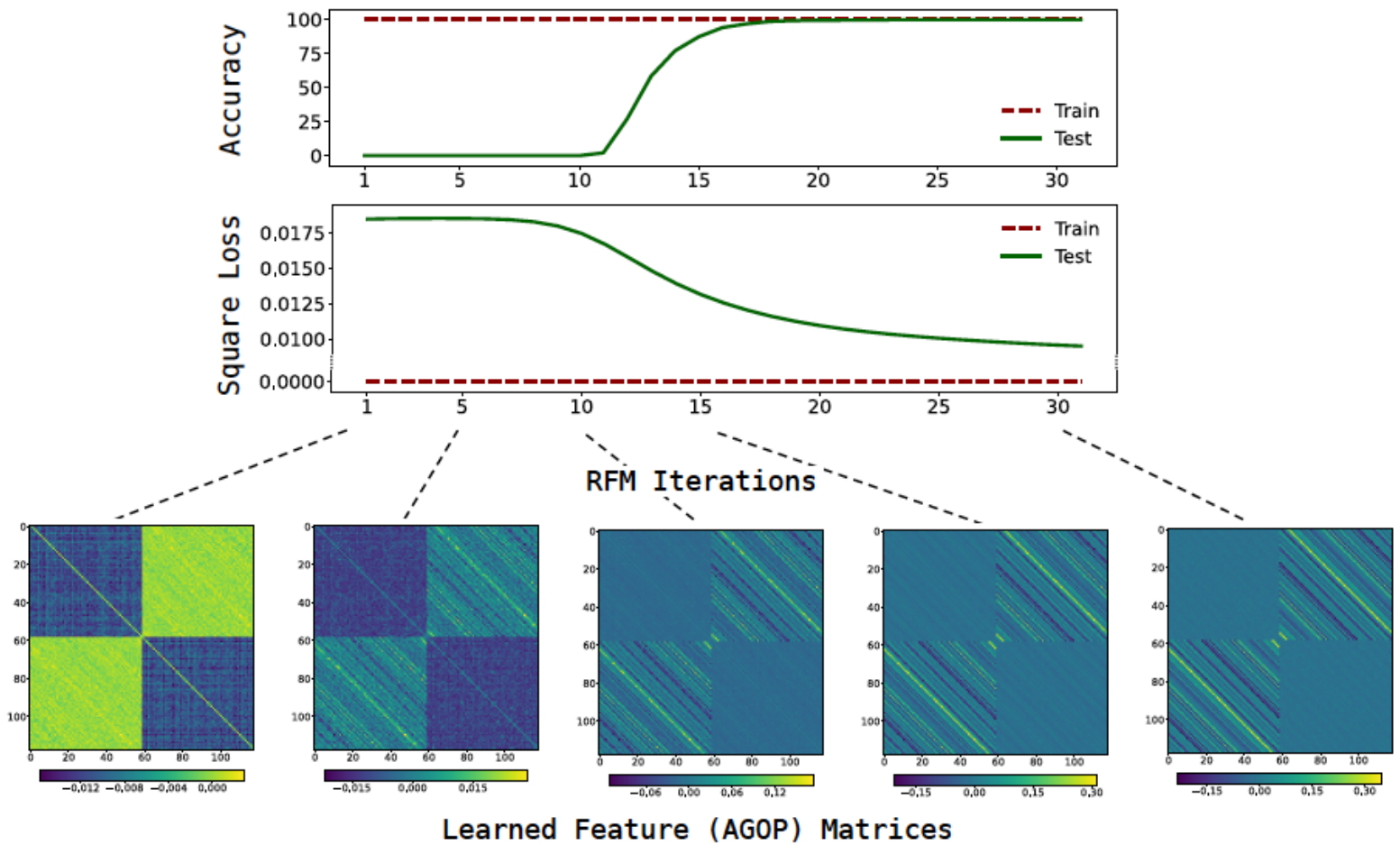


Figure 1: Recursive Feature Machines grok the modular arithmetic task $f^*(x, y) = (x + y) \bmod 59$.

Emergence in non-neural models: Modular arithmetic via average gradient outer product

Mallinar et al., 2024

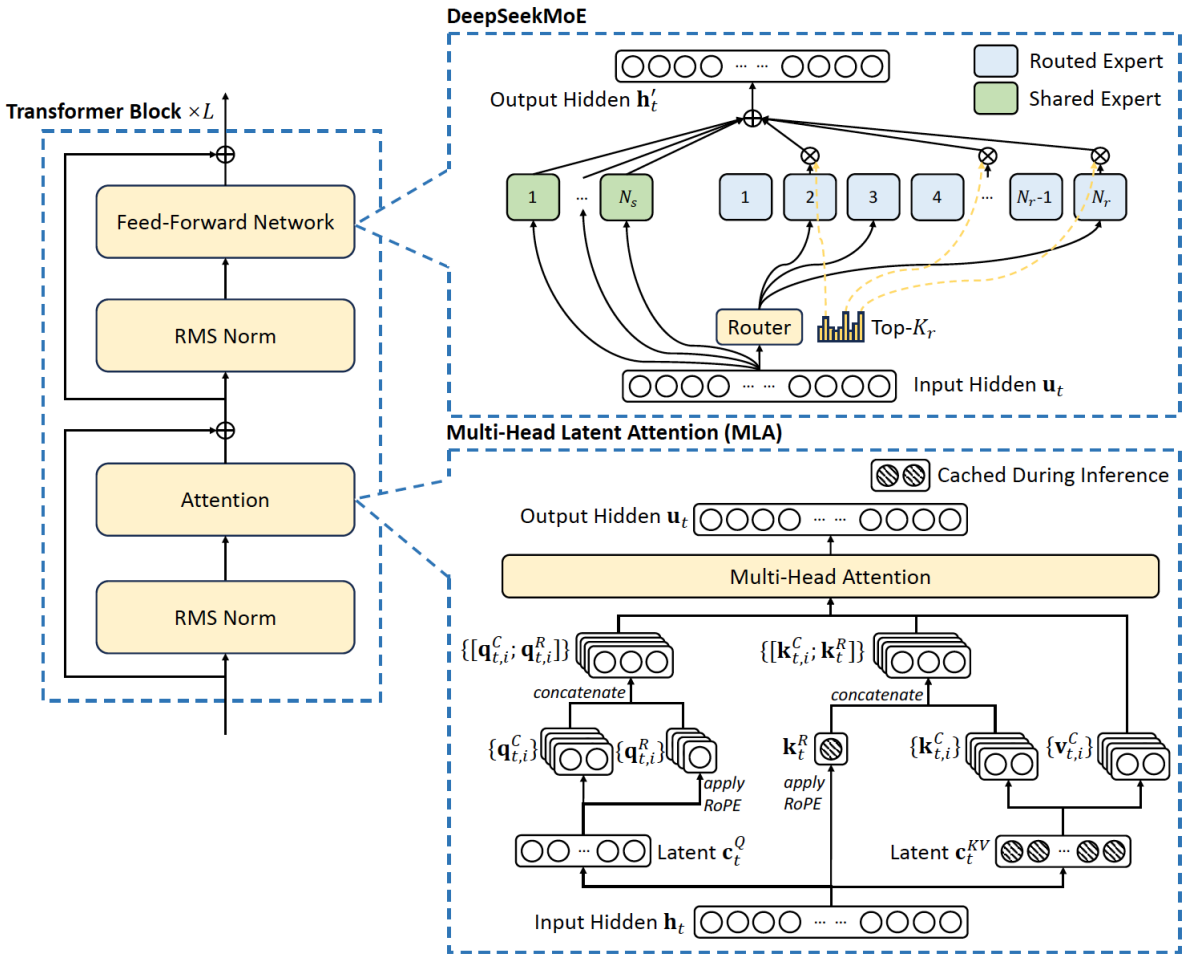
- emergence = grokking =
- understanding how to efficiently generalize from training data to similar problems =
- learning the implicit feature representation that makes the training dataset solvable in the most compact way possible

Outline

Why do we care about computation?

- Integrated audio-speech-text LLMs might be able to do things that pipelined ASR+LLM cannot.
- Experimenting is difficult, because LLMs currently have 100B parameters while ASRs have 1B.
- Can recent computational savings methods be applied to speech and audio? (Neha & Bhati, "A Survey of DeepSeek Models" (2025) mentions audio-LLMs, and DeepSeek-LLMs, but no audio-DeepSeek LLMs)

DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model



Latent Attention

- KV cache space reduced

$$\mathbf{c}_t^{KV} = W^{DKV}\mathbf{h}_t,$$
$$\mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV},$$
$$\mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV},$$

- Dot product time complexity reduced:

$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t,$$
$$\mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q,$$

...and by precomputing $(W^{UQ})^T W^{UK} = W^{QK}$, we can compute $\mathbf{q}_t^T \mathbf{k}_t$ (complexity d_q^2) as $\mathbf{c}_t^{Q,T} W^{QK} \mathbf{c}_t$ (complexity d_c^2).

Distributed Mixture of Experts

$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t}\,\mathrm{FFN}_i^{(r)}(\mathbf{u}_t),$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \mathrm{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise}, \end{cases}$$

$$s_{i,t} = \mathrm{Softmax}_i\left(\mathbf{u}_t{}^T \mathbf{e}_i\right),$$

. . . and we save computation by only activing a few of the experts on each server.

Unsolved Questions

- Does it work for audio-speech-language hybrid inputs?
- Training costs: Distributed MoE takes a while to warm up, so while testing cost is several orders of magnitude smaller, training cost is only 40% smaller.
- Fixed precision: DeepSeek v3 uses fixed precision to further reduce both training and testing costs.

Outline

Problem Statement

Federated learning addresses the following problem:

- Error rate of a deep network goes down as $\frac{1}{N} + \frac{N}{n} = \frac{1}{\sqrt{n}}$, where $N = \sqrt{n}$ is the number of hidden nodes and n is the number of training tokens.
- Most of the world's data is stored privately on 7 billion cell phones. People don't want to share it.
- Can people opt in to training a shared deep network, so that it will work very well, without compromising their own privacy?

Federated Learning

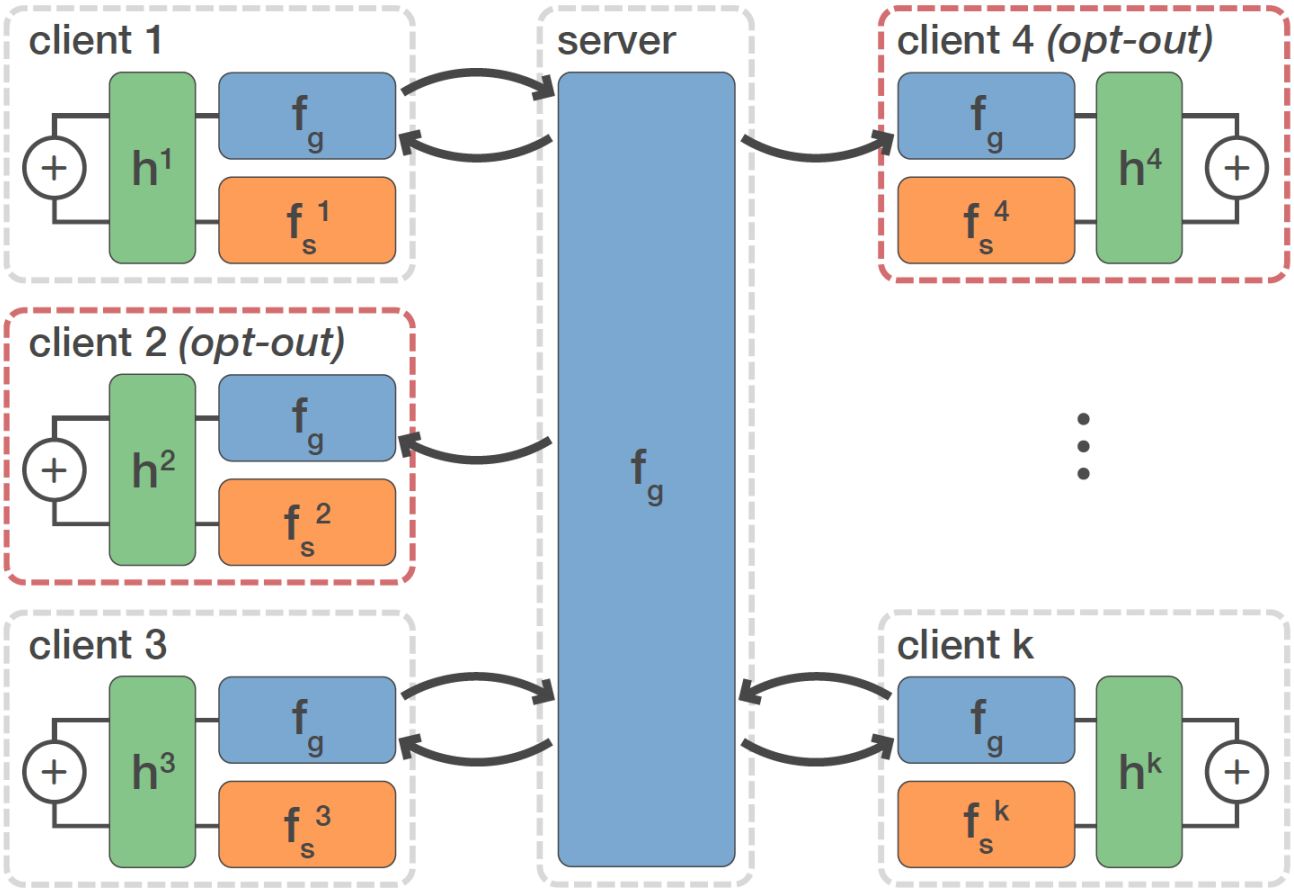


Figure 1, Specialized federated learning using a mixture of experts, Zec et al., 2020

Federated Learning

$$\min_{w \in \mathbb{R}^d} \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{(x,y)\sim p_k} \left[\ell_k(w;\ x, y) \right]$$

$$w_g^{t+1} \leftarrow \sum_k \frac{n_k}{n} w_{t+1}^k,$$

Equations 1 and 2, Specialized federated learning using a mixture of experts, Zec et al., 2020

Federated Mixture of Experts Works Well

Federated learning has an interesting relationship with DeepSeek's distributed mixture of experts. $\mathbf{x}, y$ =data, z specifies which expert has the most information:

$$p_{\mathbf{w}_{1:K},\theta}(y|\mathbf{x}) = \sum_{z=1}^{K} p_{\mathbf{w}_z}(y|\mathbf{x}, z)p_\theta(z|\mathbf{x}),$$

ELBO means we don't need all experts on each server. Each server uses a $q_\phi(z|y)$ that permits only a subset of experts, rather than $p_\theta(z|\mathbf{x})$ that permits any expert:

$$\sum_{s=1}^{S}\sum_{i=1}^{N_s} \log p_{\mathbf{w}_{1:K},\theta_s}(y_{s,i}|\mathbf{x}_{s,i}, s) \geq \sum_{s=1}^{S}\sum_{i=1}^{N_s} \mathbb{E}_{q_\phi(z|y_{s,i})}[$$
$$\log p_{\mathbf{w}_z}(y_{s,i}|\mathbf{x}_{s,i}, z)p_{\theta_s}(z|\mathbf{x}_{s,i}, s)] + \beta H(q_\phi(z|y_{s,i})), \quad (5)$$

Equations 3 and 5, Federated Mixture of Experts, Reisser et al., 2021

Federated Mixture of Experts Works Well

As DeepSeek later confirmed, $q_\phi(z|y)$ gradually converges to greater and greater confidence as training progresses, so each server only needs to have a few experts in RAM:

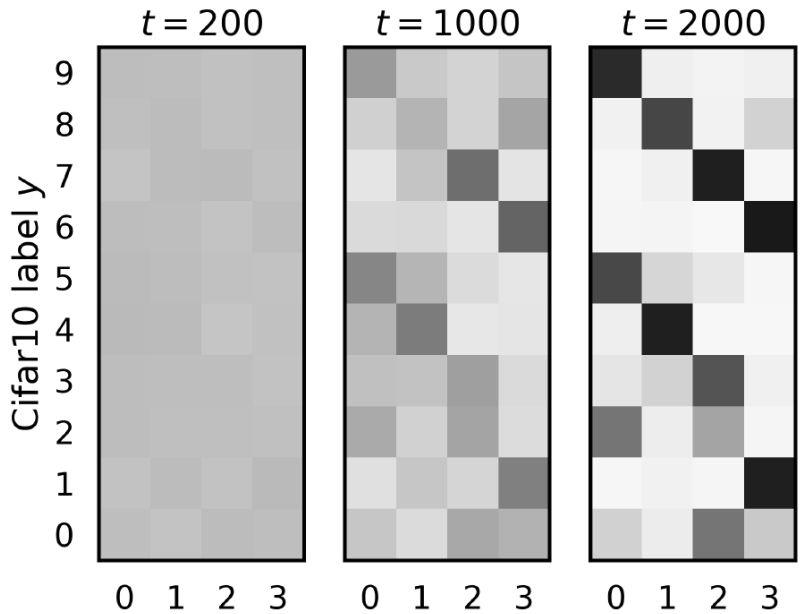


Figure 2, Federated Mixture of Experts, Reisser et al., 2021

Outline

Federated Transformer (FedAvg-T): Not so much

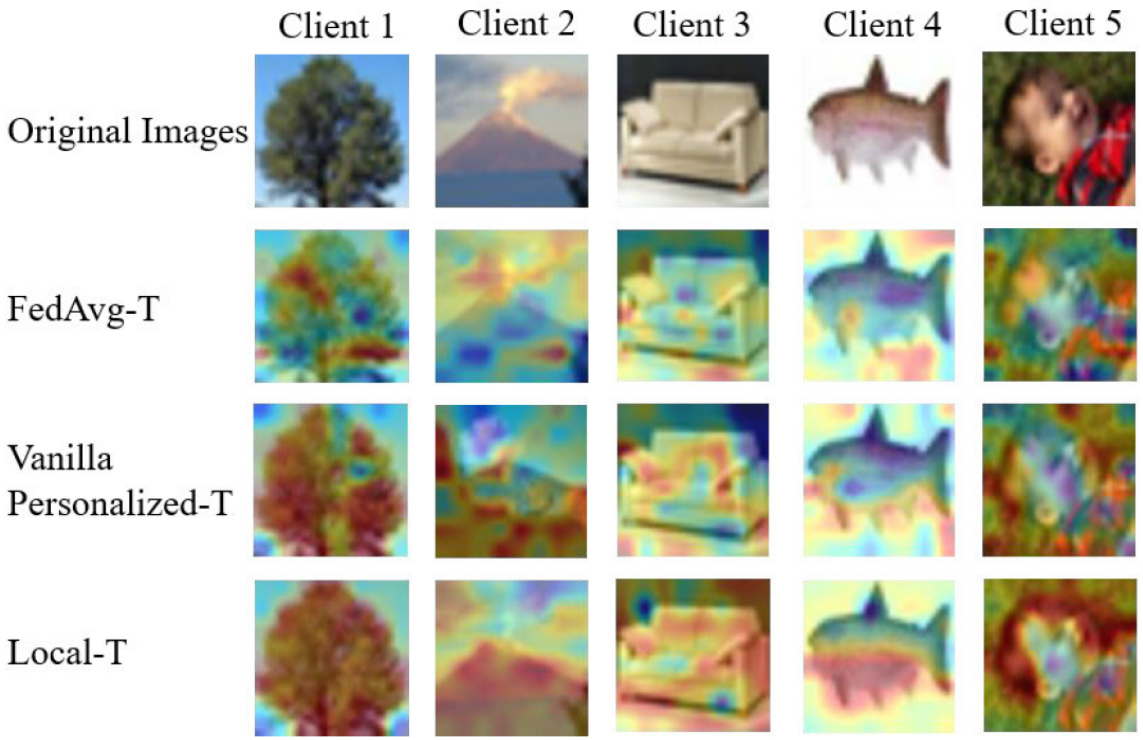


Figure 1, FedTP: Federated Learning by Transformer Personalization, Li et al., 2023

Solution #1 (Vanilla Personalized-T): Average W^O and W^V, but keep W^Q and W^K private to each client

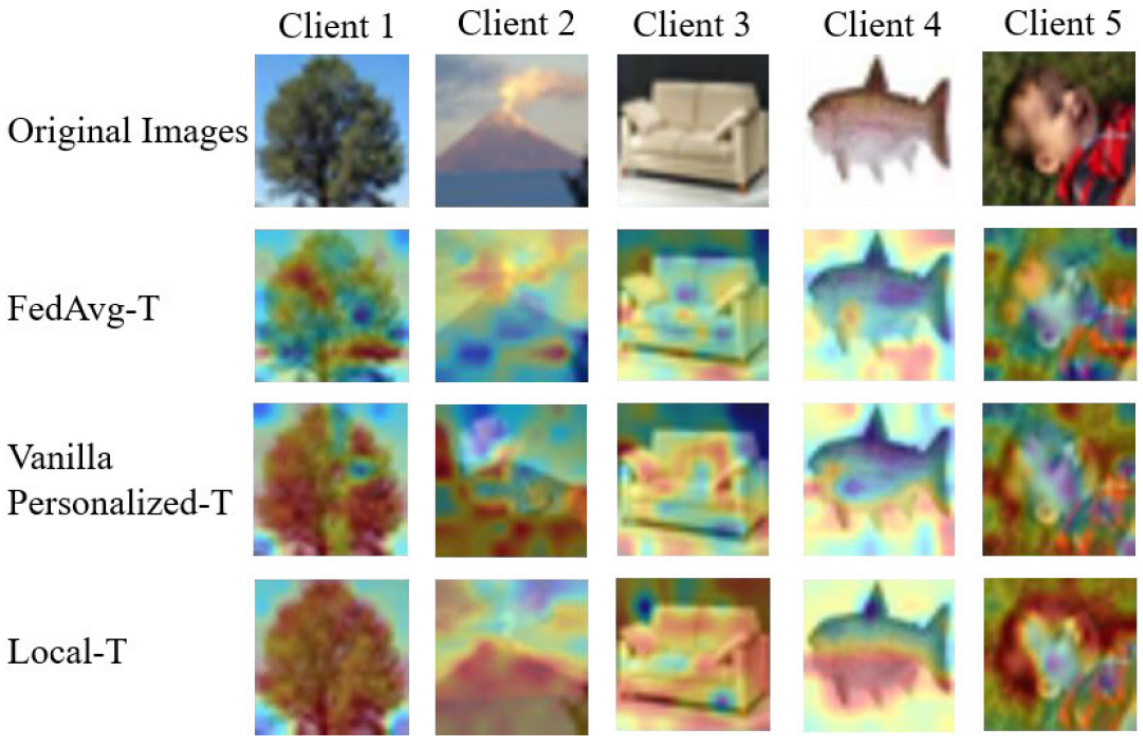


Figure 1, FedTP: Federated Learning by Transformer Personalization, Li et al., 2023

Solution #2 (FedTP): Learn a Hypernetwork that computes transformer weights from a client-data embedding vector

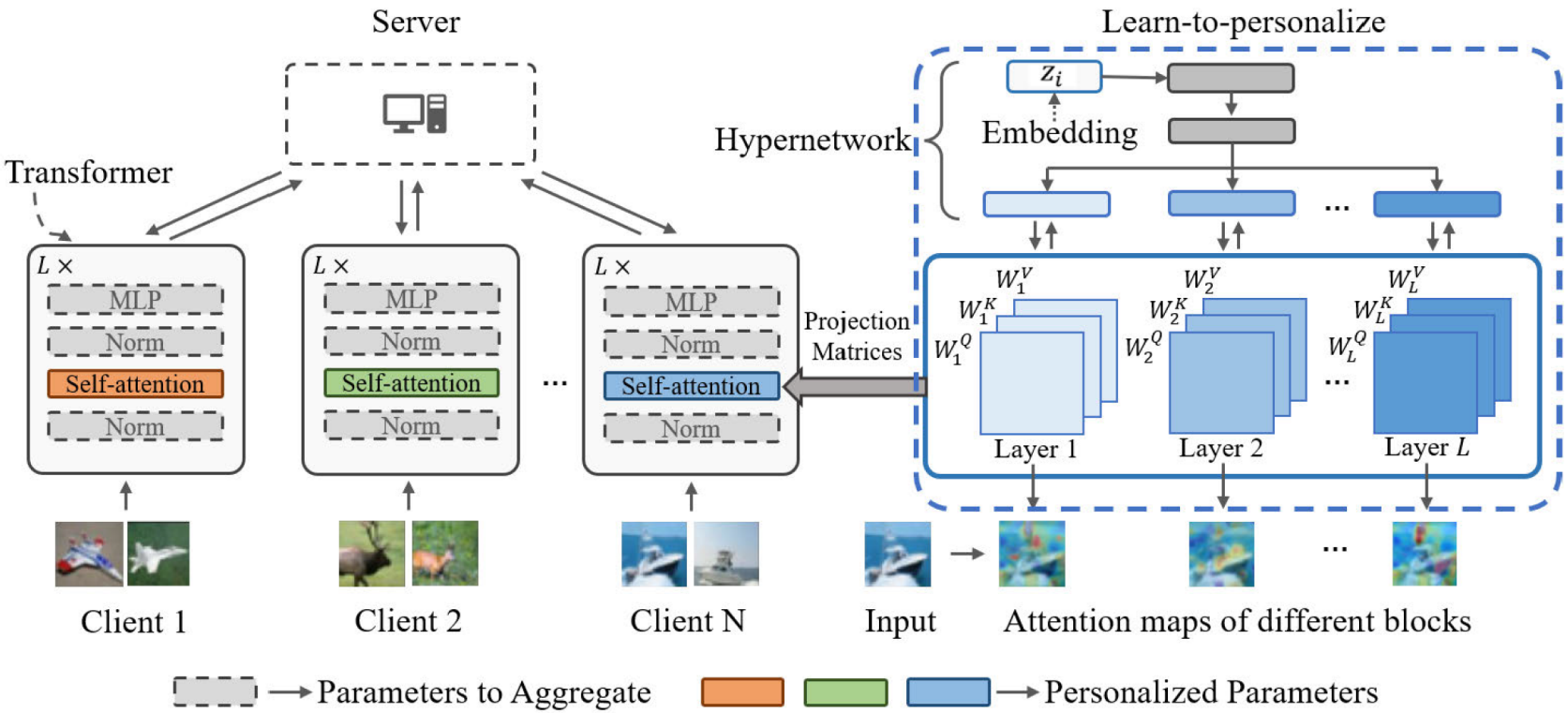


Figure 2, "FedTP: Federated Learning by Transformer Personalization," Li et al., 2023

Server accumulates ΔW_i from all the clients, but then, instead of updating W_i directly, updates parameters of the hypernetwork (ϕ) and vector embedding of each client (z_i)

$$\nabla_{\varphi} \mathcal{L}_i = \sum_{i \in C^t} \frac{m_i}{M} \nabla_{\varphi} W_i^T \Delta W_i$$

$$\nabla_{z_i} \mathcal{L}_i = \sum_{i \in C^t} \frac{m_i}{M} \nabla_{z_i} W_i^T \Delta W_i$$

Equation 7, "FedTP: Federated Learning by Transformer Personalization," Li et al., 2023

Outline

Differential Privacy: Basic Setting

The Algorithmic Foundations of Differential Privacy, Dwork and Roth, 2013

- You have a database of people who meet some criterion (e.g., people with Parkinson's), $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,d}]^T$ are the measurements from the i^{th} person.
- You want to learn some function of the database, $f(\mathcal{D})\ldots$
- in such a way that nobody can tell whether or not the k^{th} person, $\mathbf{x}_k$, was part of the database.

No deterministic mechanism can be differentially private

The Algorithmic Foundations of Differential Privacy, Dwork and Roth, 2013

- For example, suppose you just return the average, $f(\mathcal{D}) = \frac{1}{n}\sum \mathbf{x}_i$
- . . . but suppose your adversary knows all of the measurements $x_{k,j}$ of person k, she just doesn't know whether or not person k is in the database.
- Suppose, also, that she has a copy of the database before person i's diagnosis, so she knows $\mathbf{x}_i$ for all $i \neq k$
- Then she can determine whether or not person k is in the database by just comparing:

$$f(\mathcal{D}) = \frac{1}{n-1}\sum_{i \neq k} \mathbf{x}_i \quad \text{or} \quad f(\mathcal{D}) = \frac{1}{n}\sum_{i} \mathbf{x}_i$$

Definitions

The Algorithmic Foundations of Differential Privacy, Dwork and Roth, 2013

- A mechanism $\mathcal{M}$ is ϵ-differentially private if and only if, for any two datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ by only a single entry $\|\mathcal{D} - \mathcal{D}'\| = 1$, for any set $\mathcal{S}$,

$$\log \frac{P(\mathcal{M}(\mathcal{D}) \in \mathcal{S})}{P(\mathcal{M}(\mathcal{D}') \in \mathcal{S})} < \epsilon$$

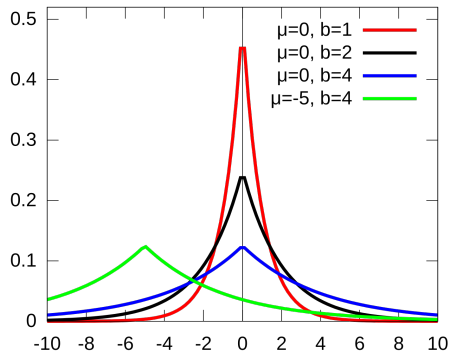
  Equivalently,

$$P(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) < P(\mathcal{M}(\mathcal{D}') \in \mathcal{S})e^{\epsilon}$$

- Note that f is useless unless it reliably measures some aggregate characteristic of the dataset: the mean, or the variance, or the whole distribution, or the classifier.

Example: Laplacian noise

The Algorithmic Foundations of Differential Privacy, Dwork and Roth, 2013



- A database of measurement vectors can be made differentially private by adding Laplacian noise.
- Adding Laplacian noise doesn't change the average.
- It changes the variance, distribution, and most classifiers in ways that can be mostly reversed.
- . . . but $\mathcal{M}(\mathcal{D})$ is ϵ-differentially private. No measurement can prove that the database contains $x_{k,j}$ instead of $x'_{k,j} = x_{k,j} + 1$ with a log likelihood ratio greater than ϵ.

Outline

Differentially Private Stochastic Gradient Descent

Deep Learning with Differential Privacy, Abadi et al., 2016

- Calculate $\nabla_{\mathbf{w}} \mathcal{L}_i$ for training token $\mathbf{x}_i$
- Clip the gradient:

$$\mathbf{g}_i = \frac{\nabla_{\mathbf{w}} \mathcal{L}_i}{\max\left(1, \|\nabla_{\mathbf{w}} \mathcal{L}_i\|\right)}$$

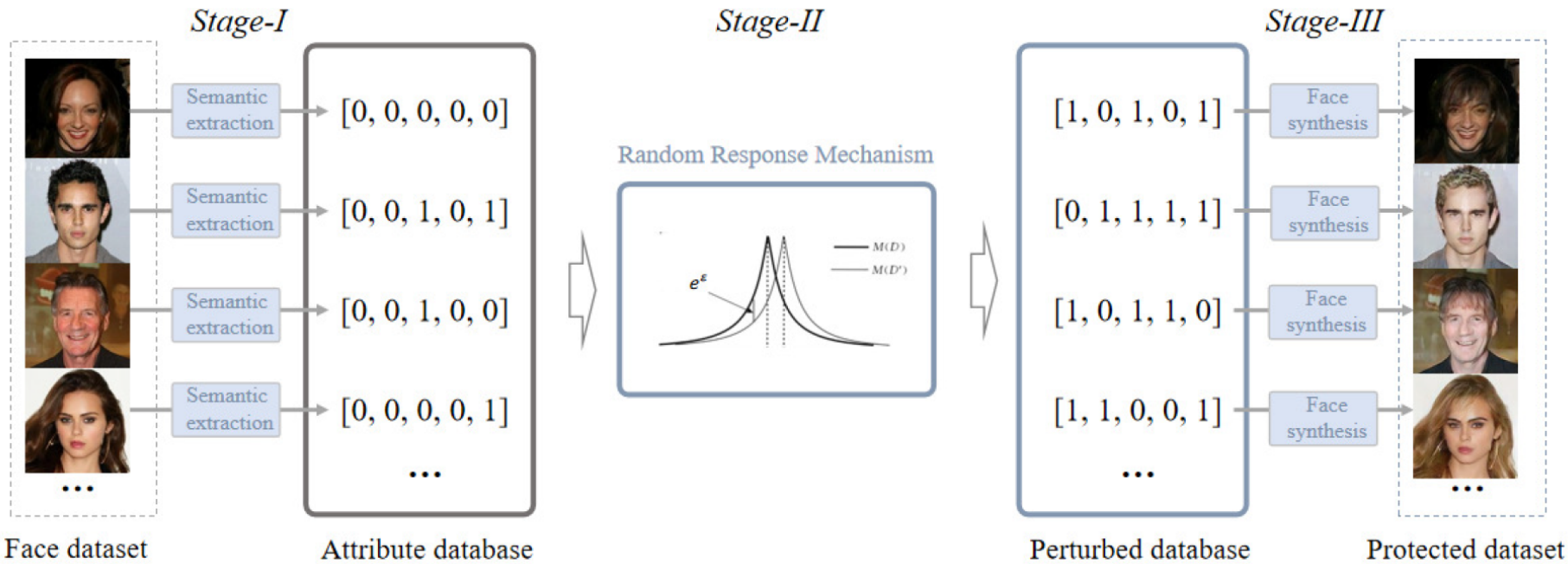
- Add Gaussian noise with variance σ^2 instead of Laplacian noise. Gives better SGD convergence at the expense of a weaker form of DP, (ϵ, δ)-DP:

$$\Pr\left\{\log \frac{P(\mathcal{D} \in \mathcal{S})}{P(\mathcal{D}' \in \mathcal{S})} > e^{\epsilon}\right\} < \delta \quad \text{if} \quad \sigma \geq c_2 \frac{\sqrt{\log(1/\delta)}}{\epsilon}$$

Models versus Data

- DP SGD allows us to distribute trained neural nets without compromising the privacy of data contributors
- DP SGD might work well in combination w/Federated Learning
- Can we distribute speech or image training data without compromising privacy? Only if we add noise to mask the identities of people in the dataset. How can we add so much noise without making the data useless?
- Zhang, Wang & Ji, "SemDP: Semantic-level Differential Privacy Protection for Face Datasets" (2024): Map faces to attribute vectors $\mathbf{x}$, add Laplacian noise to the attribute vectors, then resynthesize faces using GAN.

SemDP: Semantic-level Differential Privacy Protection for Face Datasets



Outline

What is label noise, and why does it matter?

- Training datum $= (\mathbf{x}, y)$. Label noise affects y.
- Why it matters:
  - Carefully trained and supervised labelers: little noise
  - Crowd workers: more noise
  - People using your app to get something: even more noise
  - Unlabeled data: infinite label noise
  - ...
  - If y is privacy-sensitive, Differential Privacy might corrupt it intentionally!

Label noise threshold effects

- **Unsupervised ASR (Wang et al., 2023)**: $P(y)$ and $P(x)$ learned separately. $P(y|x)$ can be inferred if noise does not change the sort order of the eigenvalues.
- **Supervised ASR** learns $P(y|x)$. Only if it does not change the maximizer of $P(y|x)$, label noise slows but does not prevent convergence.
- **Fine-tuning a foundation model:** Unsupervised pre-training can detect label errors during supervised fine-tuning. But does this require invariant ordering of eigenvalues? The question has never been asked.

Detecting Label Errors by using Pre-Trained Language Models

Chong, Hong & Manning, 2022

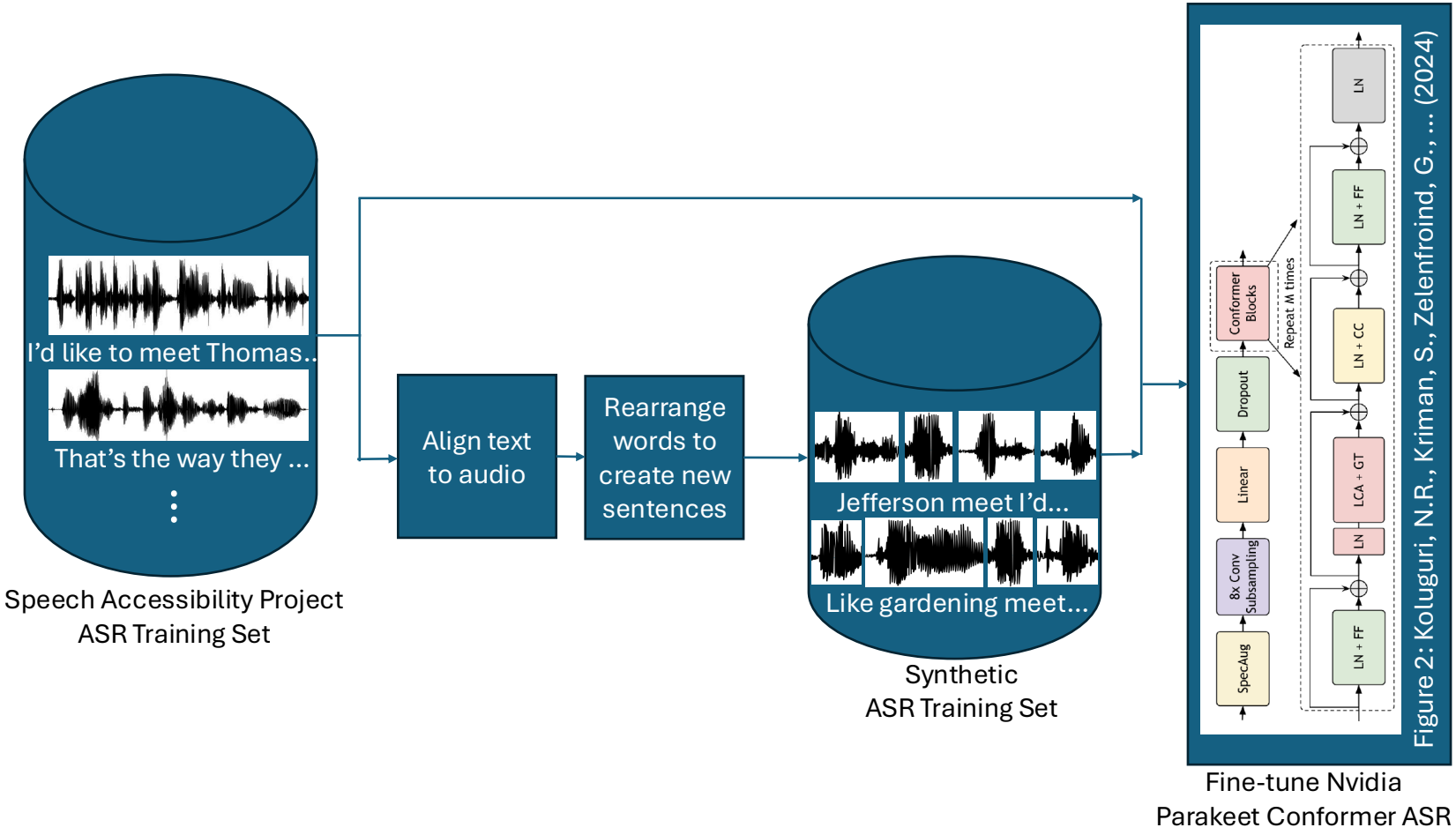
Dataset	Text	Label	Sentiment
IMDB	It is really unfortunate that a movie so well produced turns out to be such a disappointment. I thought this was full of (silly) cliches. It had all sorts of differences that it tried to tie together (not a bad thing in itself) but the result is at best awkward, but in fact ridiculous–too many clashes that wouldn't really happen. Then the end of the movie–the last 10 minutes–ruined all the rest. At first I thought Xavier was OK but with retrospect I think he was pretty bad. And that's all really too bad, because technically it was really good, and the soundtrack was great too. So the form was good, but the content pretty horrible.	~~Positive~~	Negative
IMDB	The ending made my heart jump up into my throat. I proceeded to leave the movie theater a little jittery. After all, it was nearly midnight. The movie was better than I expected. I don't know why it didn't last very long in the theaters or make as much money as anticipated. Definitely would recommend.	~~Negative~~	Positive
Amazon	The new design only has a thin layer of cellulose sponge material. It will not last as long. Already showing signs of wearing out. The picture does not represent the item received.	~~Neutral~~	Negative

Outline

Training datasets with limited text diversity

- 2015: 1000 hours of speech is enough to train modern ASR
- 2020: The speech needs to be from at least 1000 distinct people. 1000 hours of speech from a couple of hundred people is not enough.
- February, 2025: The corpus needs text diversity, otherwise an end-to-end ASR will just memorize the sentences in the training corpus!
- Many corpora designed before February 2025 have lots of speakers, reading a small number of texts, e.g.,
  - Redmond Sentence Recall (ASR for 3-5yo children): Lots of speech, but only a few dozen sentences.
  - Speech Accessibility Project (ASR for people with disabilities): 1000 hours of speech, but only about 2400 unique prompt sentences.

The solution: Cut-and-paste data augmentation



Mark Hasegawa-Johnson, based on information provided by Kaito Takahashi

Open problem: Optimal cut-and-paste

- The method of Takahashi randomly reshuffles words in the training corpus, in order to prevent the ASR from memorizing the texts in the training corpus.
- Can we do better than random reshuffling?

A solution recently proposed for text

Nguyen et al., Synthetic Text Generation for Training Large Language Models via Gradient Matching, 2025

- Problem statement: Generate real, human-readable text sequences that are not the same as the natural data, but that minimize error on the natural data.

$$\underset{\substack{\mathcal{D}_{\text{syn}}, |\mathcal{D}_{\text{syn}}| \leq r, \\ s \in \Gamma, \text{ppl}(s) \leq \epsilon \\ \forall s \in \mathcal{D}_{\text{syn}}}}{\arg\min} \; D(\nabla_{\boldsymbol{\theta}} \ell(\mathcal{D}_{\text{syn}}, \boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} \ell(\mathcal{D}_{\text{real}}, \boldsymbol{\theta})).$$

- Solution: alternate between optimizing the embeddings $\mathbf{X}^t$ and the resulting text sequences $\mathbf{Z}^t$.

$$\boldsymbol{X}^{t+1} = \arg\min_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{Z}^t, \boldsymbol{\Lambda}^t)$$
$$\boldsymbol{Z}^{t+1} = \mathcal{P}_{\mathcal{E}_{\text{top-k}}}(\boldsymbol{X}^{t+1} + \rho^{-1}\boldsymbol{\Lambda}^t)$$

Summary

- Computation: Can DeepSeek be used to train audio/speech LLMs? Does speech permit or prohibit computational savings in any way different from text?
- Federated Learning: Can we combine a few private datasets to train a single speech LLM?
- Differential Privacy: Can GANs add category noise to an audio/speech database in a way that permits its free redistribution without compromising contributor privacy?
- Label noise: Do pre-trained models detect fine-grained label noise (text transcription) as well as coarse-grained (opinion)? How does it depend on distribution shift?
- Dataset size: How can pre-training data be chosen to minimize the number of labeled examples required?