“The Curse Of Recursion: Training on Generated Data Makes Models Forget”

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, Ross Anderson

Discussion Lead: Georgios Syros

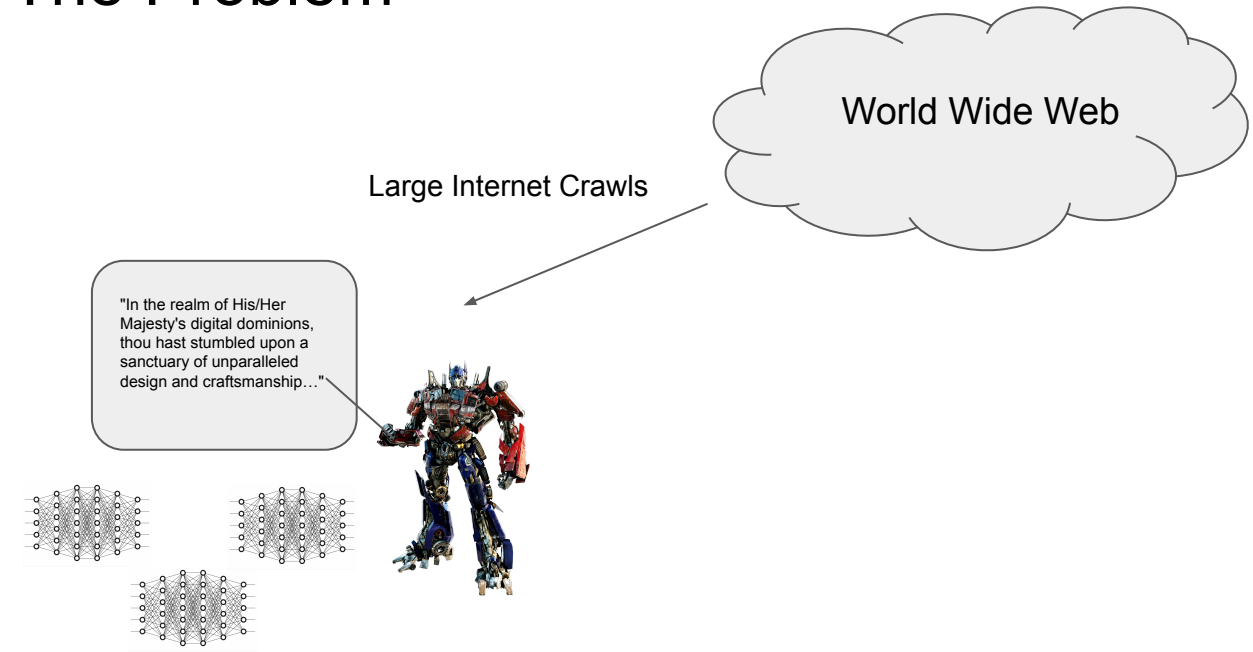
The Problem

Timeline: AnotherLLM-4

The WWW contains a large portion of **original**(!) human interaction and communication.

The Problem

Timeline: AnotherLLM-4

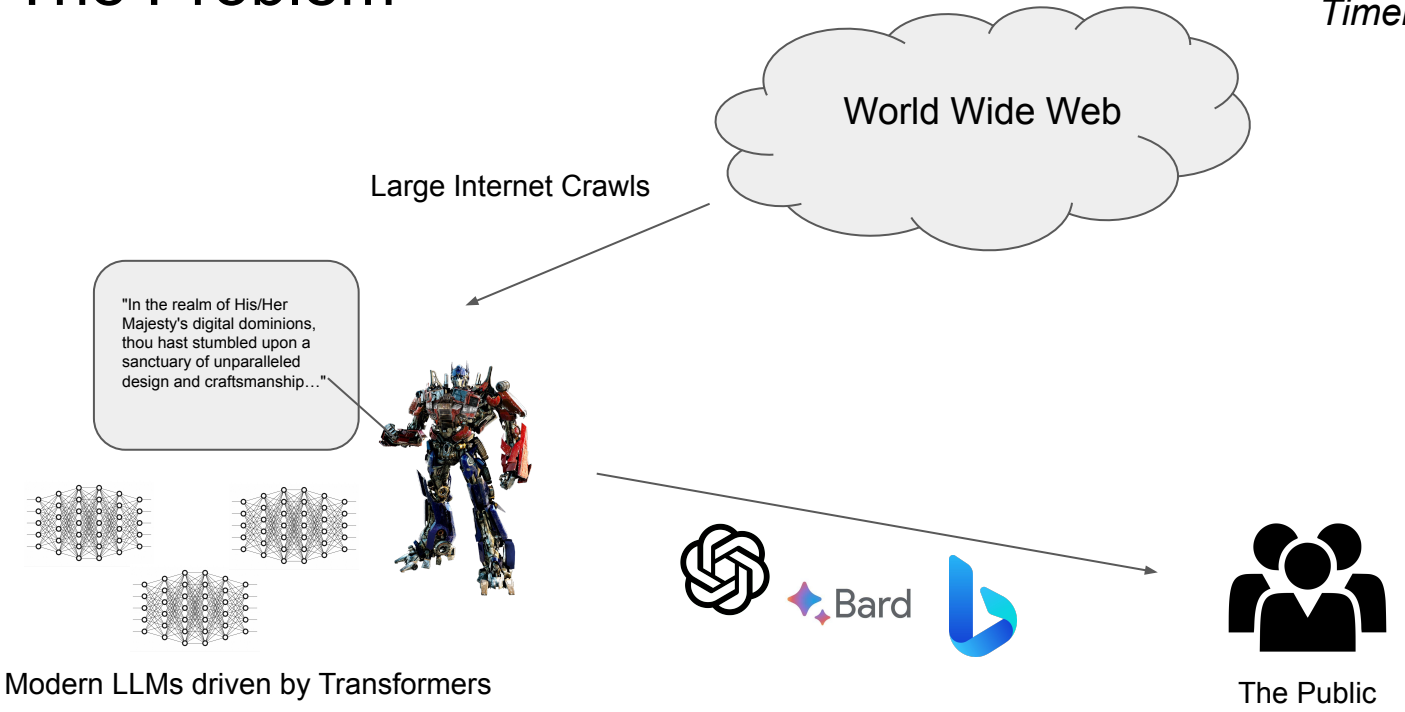


Modern LLMs driven by Transformers

Which is why it is very appealing to train modern LLMs on!

The Problem

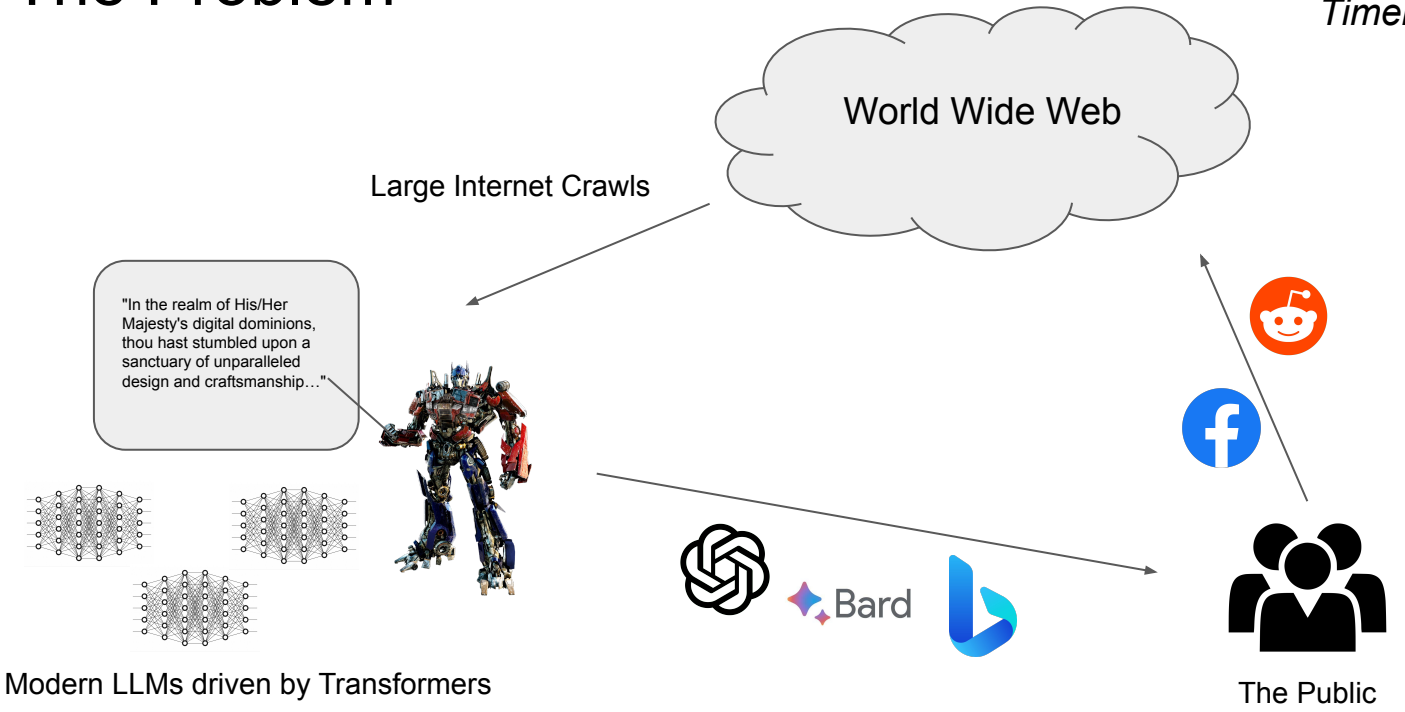
Timeline: AnotherLLM-4



Undisputedly, LLMs have become a very important tool for the public and a core component to many modern software systems.

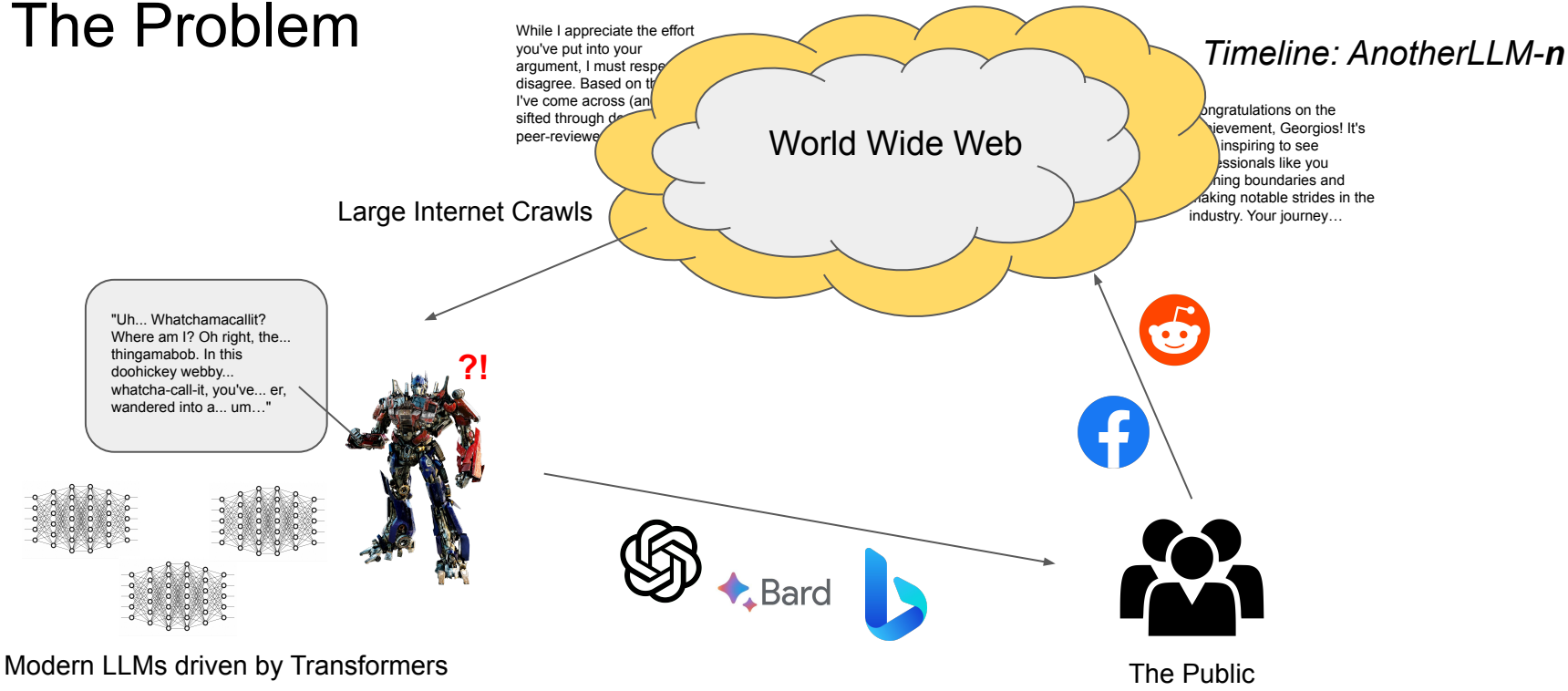
The Problem

Timeline: AnotherLLM-4



Thus, inevitably, a lot of AI-generated content is going to end up to the WWW.

The Problem



Since most future models' training data is also scraped from the web, what happens when they will inevitably be trained on data produced by their predecessors?

Overview

Overview

The key takeaways of the research:

Overview

The key takeaways of the research:

- *Model Collapse*, a degenerative process in learning, is discovered.

Overview

The key takeaways of the research:

- *Model Collapse*, a degenerative process in learning, is discovered.
- *Model Collapse* exists in a variety of different model types and datasets.

Overview

The key takeaways of the research:

- *Model Collapse*, a degenerative process in learning, is discovered.
- *Model Collapse* exists in a variety of different model types and datasets.
- In order to avoid *Model Collapse*, access to genuine human-generated content is essential.

Model Collapse

Model Collapse

"*Model Collapse is a degenerative process affecting generations of learned generative models, where generated data end up polluting the training set of the next generation of models; being trained on polluted data, they then mis-perceive reality.*"

Model Collapse

"*Model Collapse is a degenerative process affecting generations of learned generative models, where generated data end up polluting the training set of the next generation of models; being trained on polluted data, they then mis-perceive reality.*"

- *Early*
  - Model begins to lose information about the tails of the distribution.

Model Collapse

"*Model Collapse is a degenerative process affecting generations of learned generative models, where generated data end up polluting the training set of the next generation of models; being trained on polluted data, they then mis-perceive reality.*"

- *Early*
  - Model begins to lose information about the tails of the distribution.
- *Late*
  - Model entangles different modes of the original distributions and converges to a distribution that carries little resemblance to the original one

What is responsible for *Model Collapse*?

From this point onwards let’s focus on Normal Distributions for a while. We will come back to LLMs later.

What is responsible for *Model Collapse*?

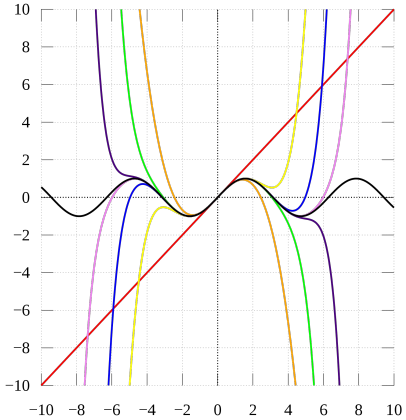
In general, two key factors:

What is responsible for *Model Collapse*?

In general, two key factors:

- **Statistical Approximation Error**
  - The primary type of error.
  - Arises due to the number of samples being finite, and disappears as the number of samples tends to infinity.
  - Occurs due to a non-zero probability that information can get lost at every step of re-sampling.

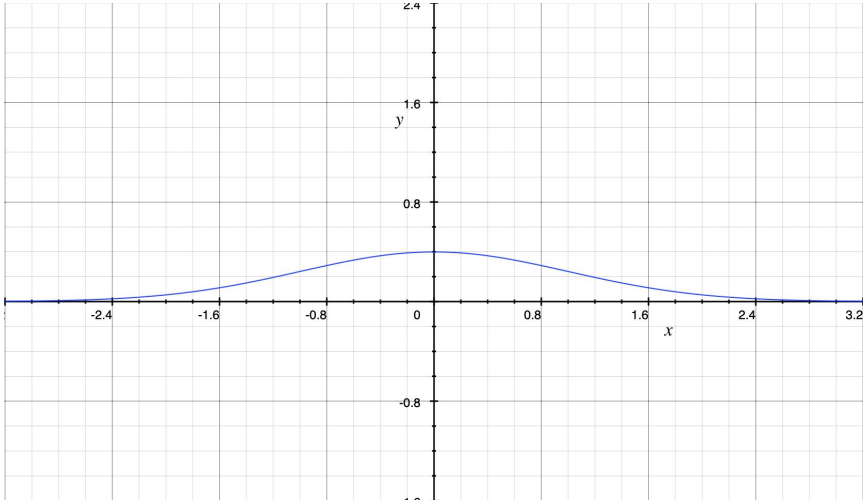
What is responsible for *Model Collapse*?



In general, two key factors:

- **Statistical Approximation Error**
  - The primary type of error.
  - Arises due to the number of samples being finite, and disappears as the number of samples tends to infinity.
  - Occurs due to a non-zero probability that information can get lost at every step of re-sampling.
- **Functional Approximation Error**
  - The secondary type of error.
  - Stems from our function approximations being insufficiently expressive or over-expressive outside of the original distribution support.

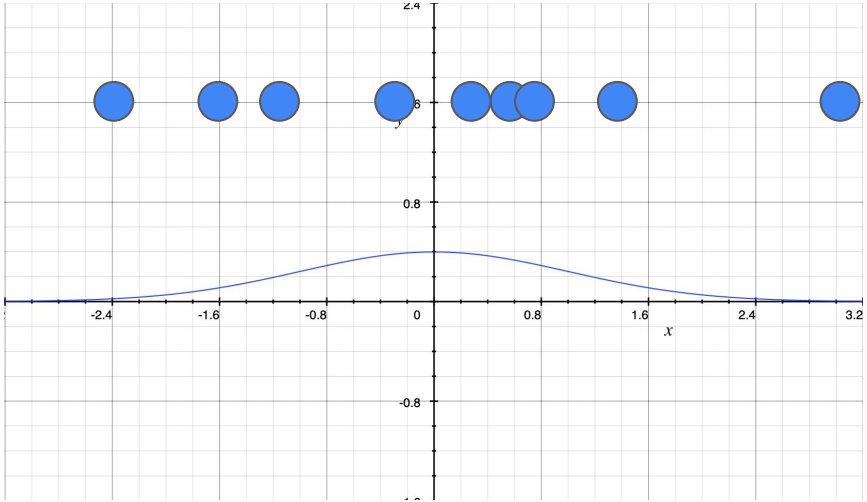
What is responsible for *Model Collapse*? - in practice



Assume that we have a Normal Distribution $d \sim N(\mu, \sigma^2)$.

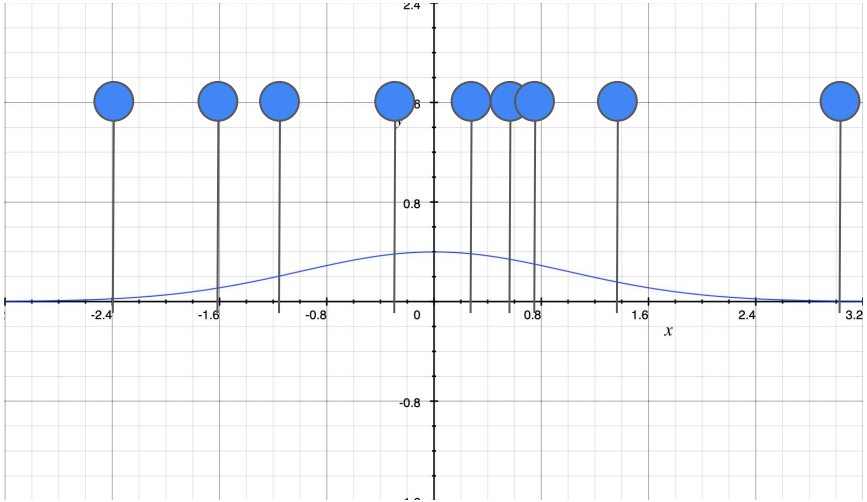
We arbitrarily call this the ‘real’ (original) model.

What is responsible for *Model Collapse*? - in practice



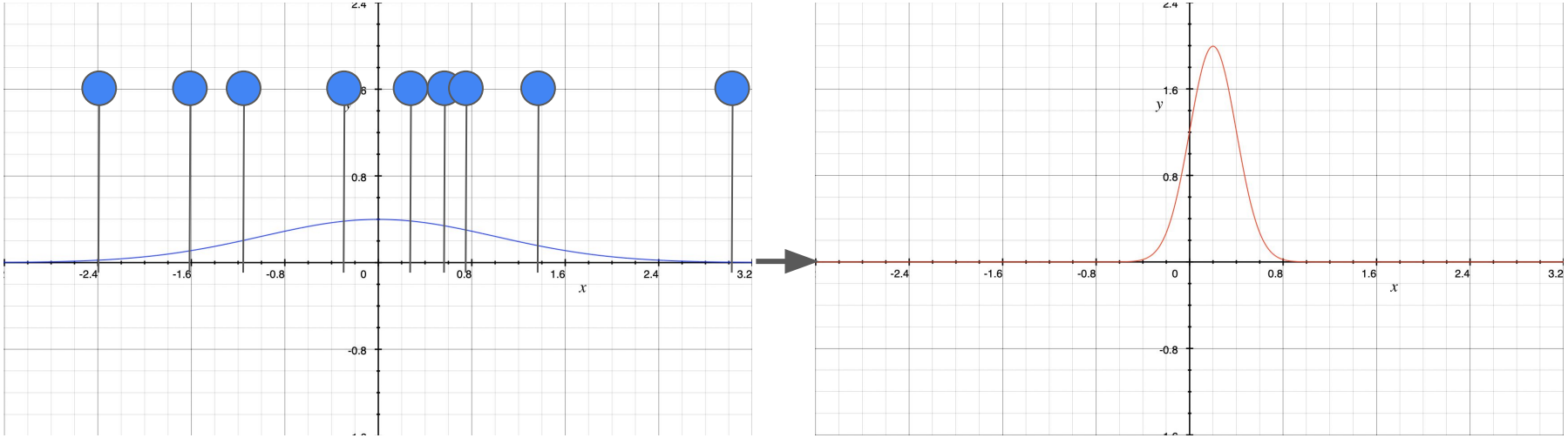
To move to the ‘next’ model, we sample randomly sample the distribution with *n* points.

What is responsible for *Model Collapse*? - in practice



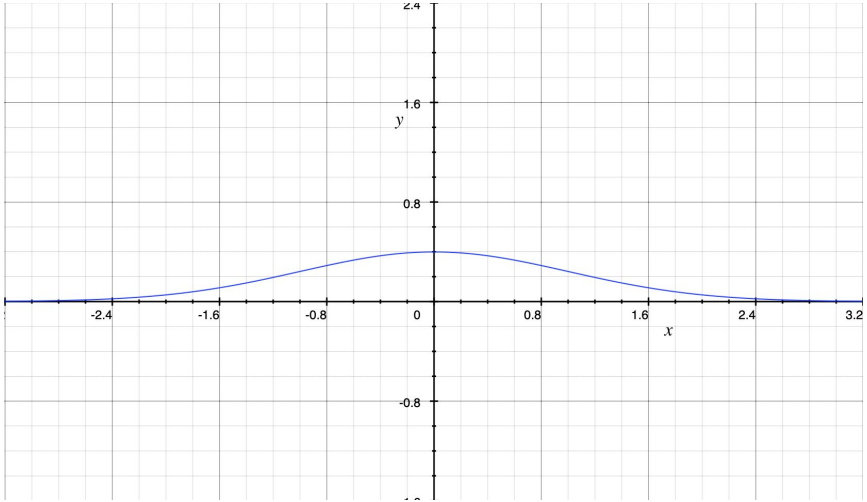
We calculate the μ’ and σ’² using those n points

What is responsible for *Model Collapse*? - in practice

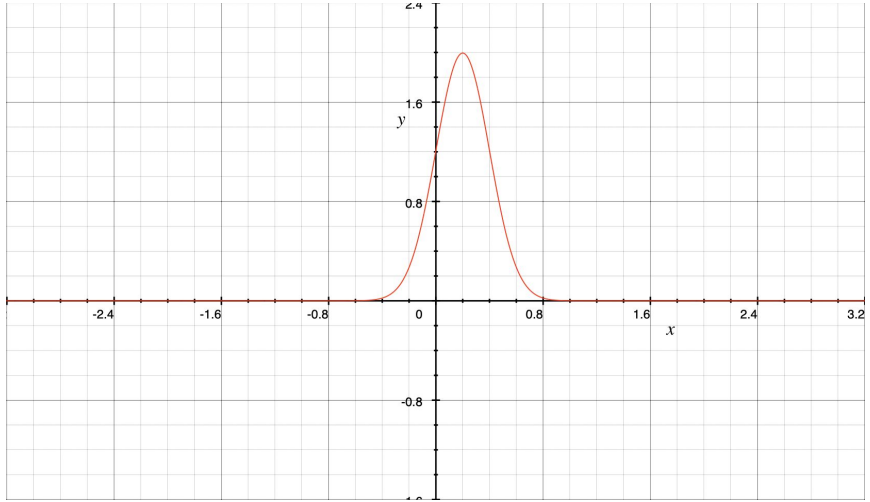


We calculate the μ’ and σ’² using those n points and we create d’~N(μ’, σ’²).

What is responsible for *Model Collapse*? - in practice

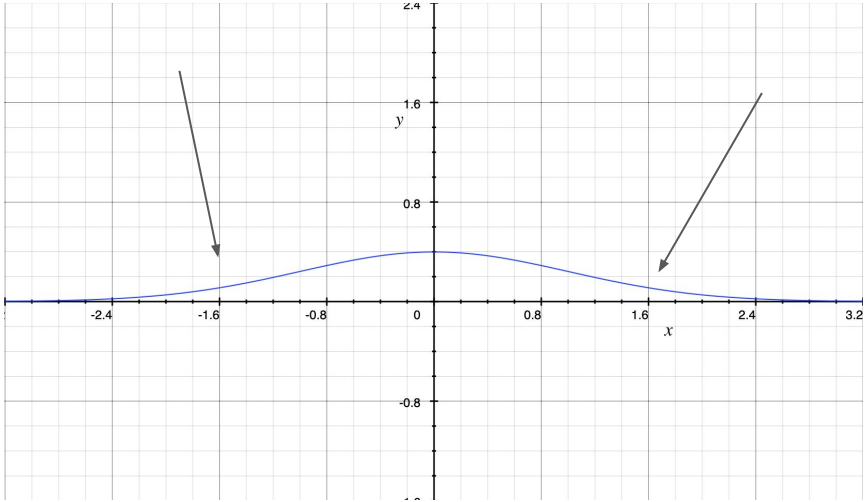


≠

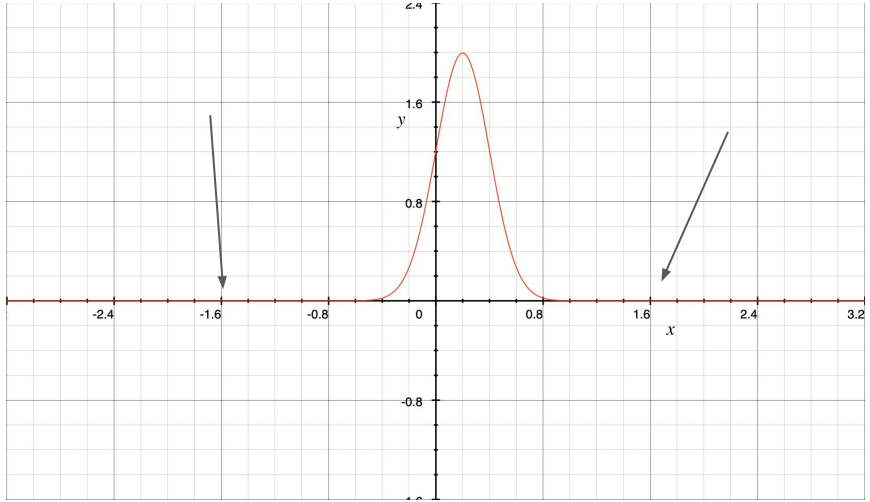


Notice how d and d’ differ.

What is responsible for *Model Collapse*? - in practice

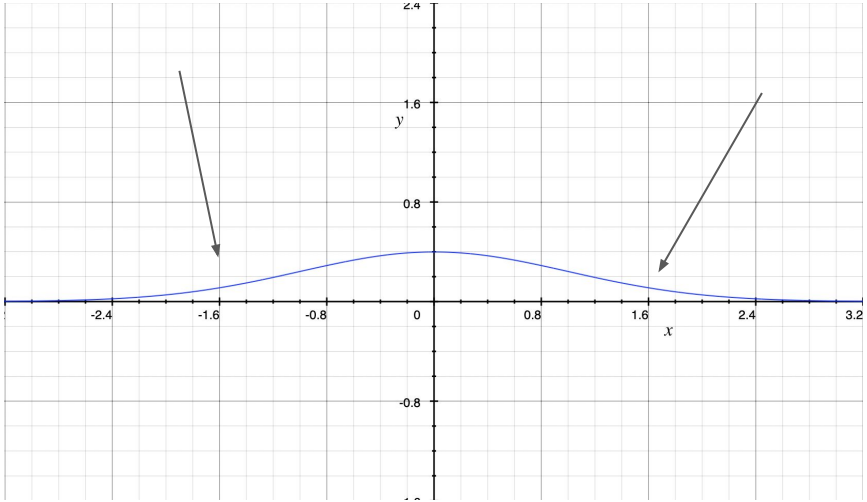


≠

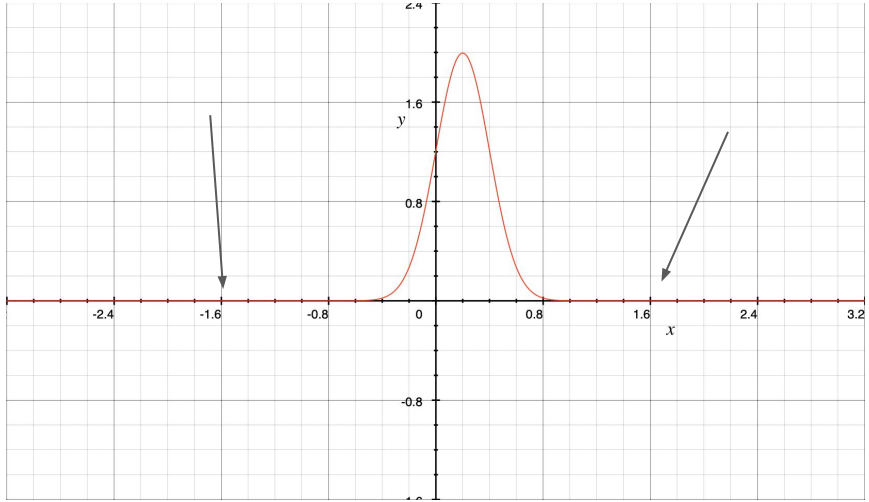


Tail information is lost from model to model.

What is responsible for *Model Collapse*? - in practice

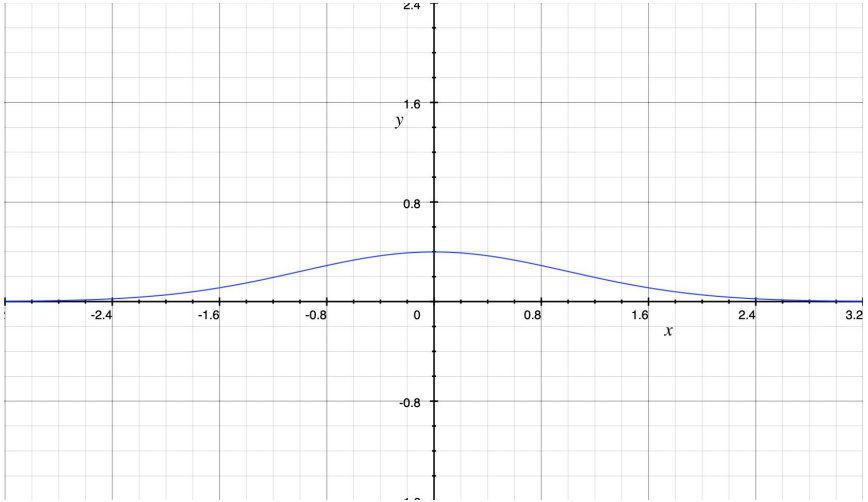


≠

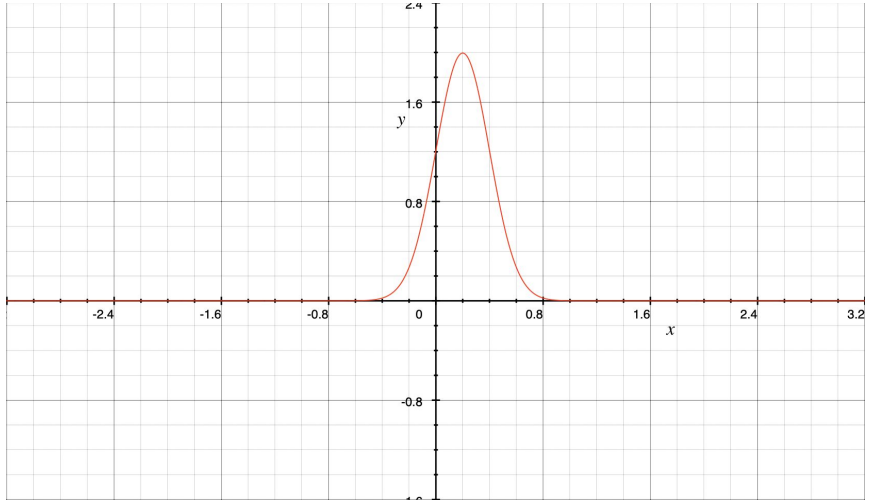


Tail information is lost from model to model. (*Early Model Collapse*)

What is responsible for *Model Collapse*? - in practice

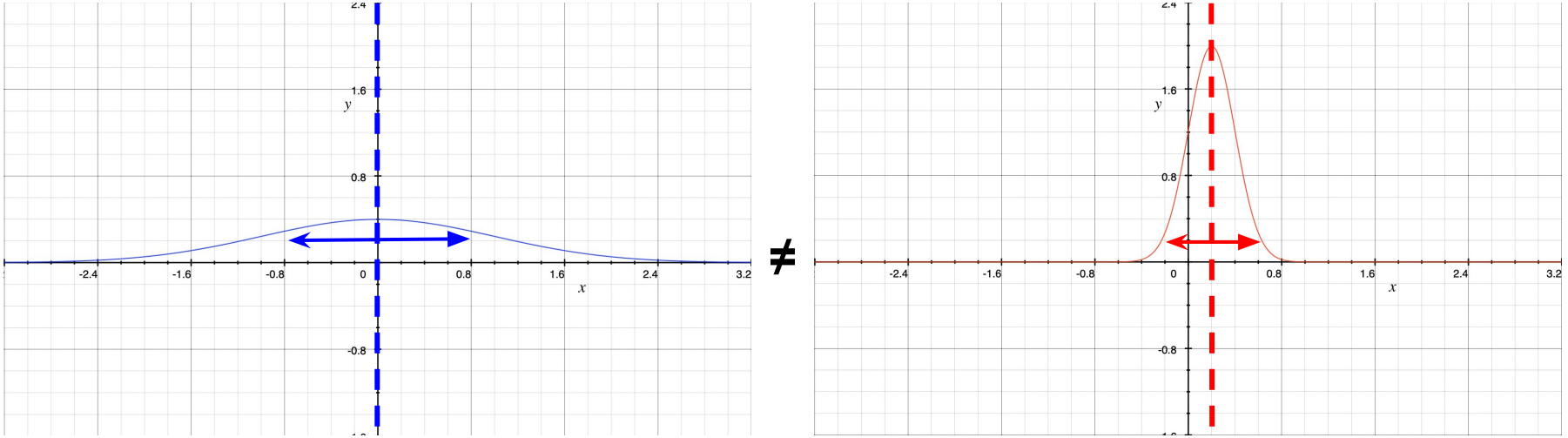


≠



The new model has $\mu' \neq \mu$ and $\sigma'^2 \neq \sigma^2$, thus d' resembles a different distribution than d.

What is responsible for *Model Collapse*? - in practice



The new model has $\mu' \neq \mu$ and $\sigma'^2 \neq \sigma^2$, thus d' resembles a different distribution than d. (*Late Model Collapse*)

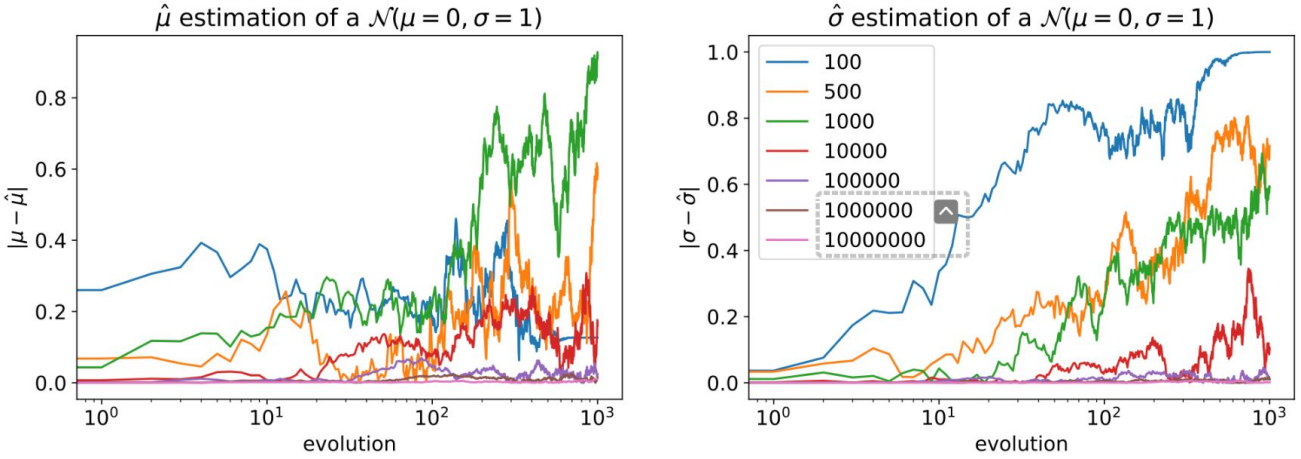
But how can we sample?

Three approaches for **sampling** between models:

But how can we sample?

Three approaches for **sampling** between models:

- Approach 1
  - To create model i+1 we sample n points from model i and fit to a normal distribution.



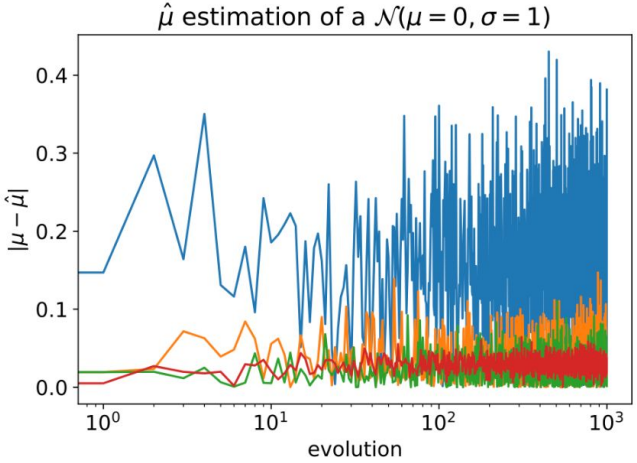
(a) Mean estimation

(b) Standard Deviation

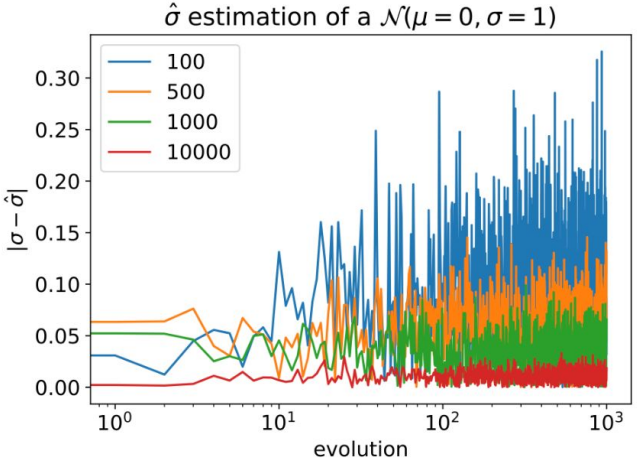
But how can we sample?

Three approaches for **sampling** between models:

- Approach 2
  - To create model i+1, model i gets data sampled, its output is mixed with data sampled from models 1 . . . i, and then **the mix gets sampled** to fit the model i + 1.



(a) Mean estimation

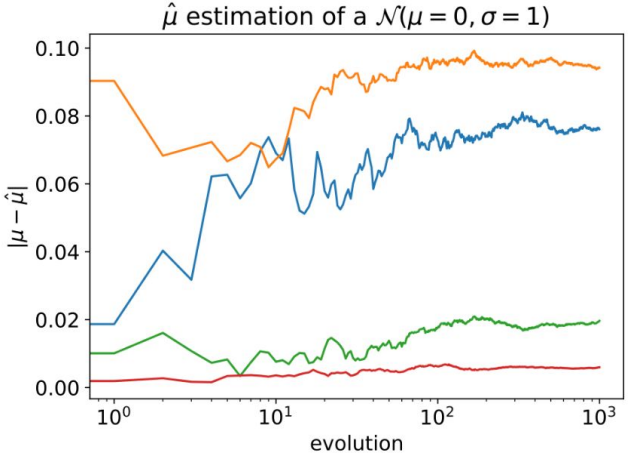


(b) Standard Deviation

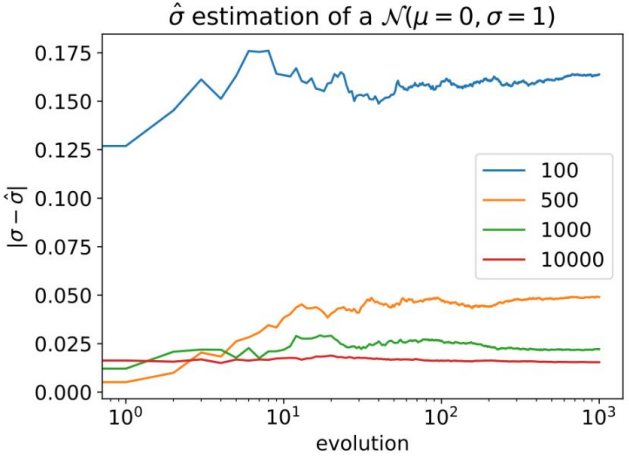
But how can we sample?

Three approaches for **sampling** between models:

- Approach 3
  - To create model i+1, model i gets data sampled, its output is mixed with data sampled from models 1 . . . i, and then **the entire mix** is used to fit the model i + 1.



(a) Mean estimation



(b) Standard Deviation

The connection between Normal Distributions & LLMs

The connection between Normal Distributions & LLMs

Do the normal distribution examples have anything to do with LLMs at all?

The connection between Normal Distributions & LLMs

Do the normal distribution examples have anything to do with LLMs at all?

The answer is..

The connection between Normal Distributions & LLMs

Do the normal distribution examples have anything to do with LLMs at all?

The answer is.. **yes**!

The connection between Normal Distributions & LLMs

Do the normal distribution examples have anything to do with LLMs at all?

The answer is.. **yes**!

LLMs can be “represented” as normal distributions by **plotting their perplexity of generated datapoints**.

The connection between Normal Distributions & LLMs

Do the normal distribution examples have anything to do with LLMs at all?

The answer is.. **yes**!

LLMs can be “represented” as normal distributions by **plotting their perplexity of generated datapoints**.

The perplexity (PP) of a LM is a metric used to evaluate how well a probability model predicts a sample. For language models specifically, it gauges the model's uncertainty in predicting the next word in a sequence.

Perplexity & LMs

The goal when designing LMs is to **have as low perplexity as possible**.

Perplexity & LMs

The goal when designing LMs is to **have as low perplexity as possible**.

- Lower perplexity means better model.
- A perfect model has a perplexity of 1.
- The worst possible model has a perplexity of |V|, where V is the vocabulary.

Perplexity & LMs

The goal when designing LMs is to **have as low perplexity as possible**.

- Lower perplexity means better model.
- A perfect model has a perplexity of 1.
- The worst possible model has a perplexity of |V|, where V is the vocabulary.

In our context, we want to observe the change (if any) in the distribution of perplexity on generated data in later generations of models compared to previous models.

Evaluation

Evaluation

- Gaussian Mixture Models (GMMs)
  - Try to separate two artificially-generated Gaussians.
  - Show the progression of the GMM fitting process over time.

Evaluation

- Gaussian Mixture Models (GMMs)
  - Try to separate two artificially-generated Gaussians.
  - Show the progression of the GMM fitting process over time.

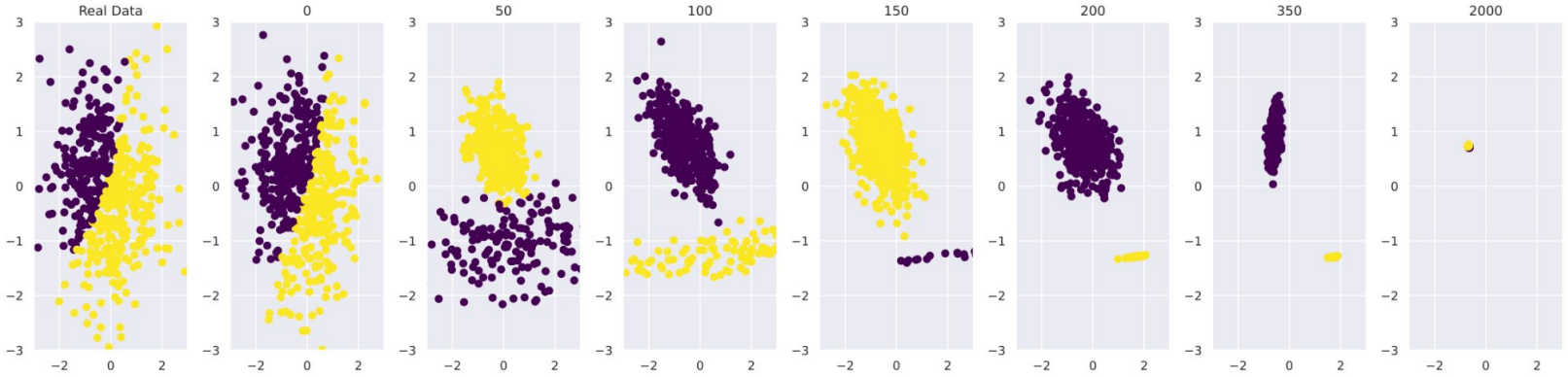


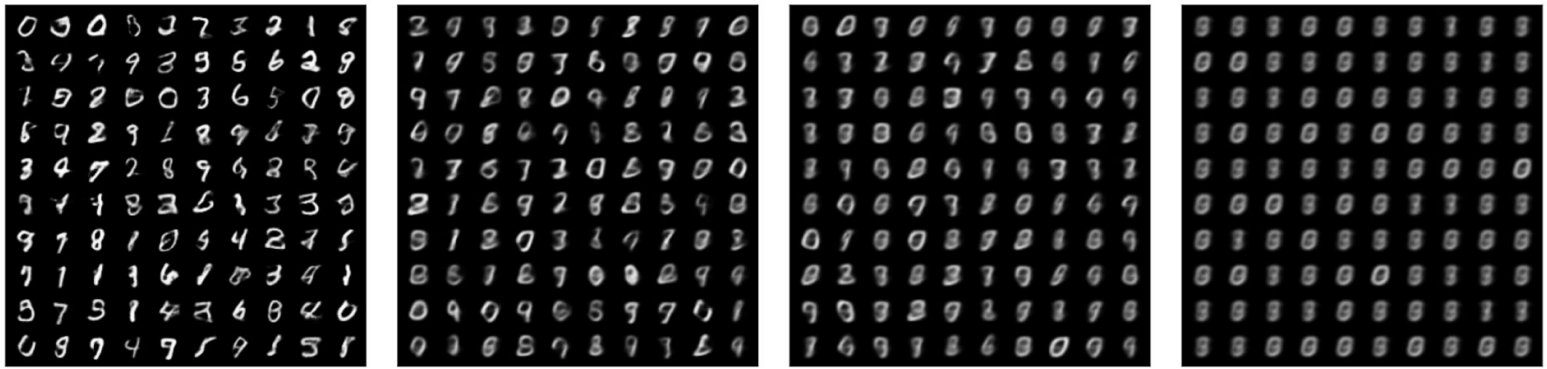
Figure 7: An examples of GMM fitting data at iterations $\{0, 50, 100, 150, 200, 350, 2000\}$. At first the model fits data very well as is shown on the left; yet even at generation 50 the perception of the underlying distribution completely changes. At generation 2000 it converges to a state with very little variance. GMM is sampled a thousand times.

Evaluation

- Variational Autoencoders (VAEs)
  - Train an autoencoder on an original data source and then sample.
  - Generate latents from a Gaussian distribution which are then used by the decoder to generate data for the subsequent generation.

Evaluation

- Variational Autoencoders (VAEs)
  - Train an autoencoder on an original data source and then sample.
  - Generate latents from a Gaussian distribution which are then used by the decoder to generate data for the subsequent generation.



(a) Original model (b) Generation 5 (c) Generation 10 (d) Generation 20

Figure 9: Random latent reconstructions from VAEs. No training data comes from the original distribution. Over the generations, different modes of the original distribution get entangled and generated data starts looking unimodal.

Evaluation

- Large Language Models (LLMs)

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.
  - Fine-tune *OPT-125m* causal language model by Meta through Huggingface.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.
  - Fine-tune *OPT-125m* causal language model by Meta through Huggingface.
    - Data generation from the trained models using a 5-way beam-search.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.
  - Fine-tune *OPT-125m* causal language model by Meta through Huggingface.
    - Data generation from the trained models using a 5-way beam-search.
    - Block training sequences to be 64 tokens long; then for each token sequence in the training set, ask the model to predict the next 64 tokens.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.
  - Fine-tune *OPT-125m* causal language model by Meta through Huggingface.
    - Data generation from the trained models using a 5-way beam-search.
    - Block training sequences to be 64 tokens long; then for each token sequence in the training set, ask the model to predict the next 64 tokens.
    - Go through all of the original training dataset and produce an artificial dataset of the same size.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.
  - Fine-tune *OPT-125m* causal language model by Meta through Huggingface.
    - Data generation from the trained models using a 5-way beam-search.
    - Block training sequences to be 64 tokens long; then for each token sequence in the training set, ask the model to predict the next 64 tokens.
    - Go through all of the original training dataset and produce an artificial dataset of the same size.
    - Training for each of the generations starts with generation from the original training data.

Evaluation

- Large Language Models (LLMs)
  - Explore what happens with language models when they are sequentially fine-tuned with data generated by other models.
  - Fine-tune on the *wikitext2* dataset.
  - Fine-tune *OPT-125m* causal language model by Meta through Huggingface.
    - Data generation from the trained models using a 5-way beam-search.
    - Block training sequences to be 64 tokens long; then for each token sequence in the training set, ask the model to predict the next 64 tokens.
    - Go through all of the original training dataset and produce an artificial dataset of the same size.
    - Training for each of the generations starts with generation from the original training data.
    - Each experiment is ran 5 times and the results are shown as 5 separate runs.

Evaluation

Two Different settings based on the normal distribution sampling approaches that we presented earlier:

Evaluation

Two Different settings based on the normal distribution sampling approaches that we presented earlier:

- **5 epochs, no original training data**

Evaluation

Two Different settings based on the normal distribution sampling approaches that we presented earlier:

- **5 epochs, no original training data**
  - the model is trained for 5 epochs on the original dataset and no original data.

Evaluation

Two Different settings based on the normal distribution sampling approaches that we presented earlier:

- **5 epochs, no original training data**
  - the model is trained for 5 epochs on the original dataset and no original data.

- **10 epochs, 10% of original training data preserved**

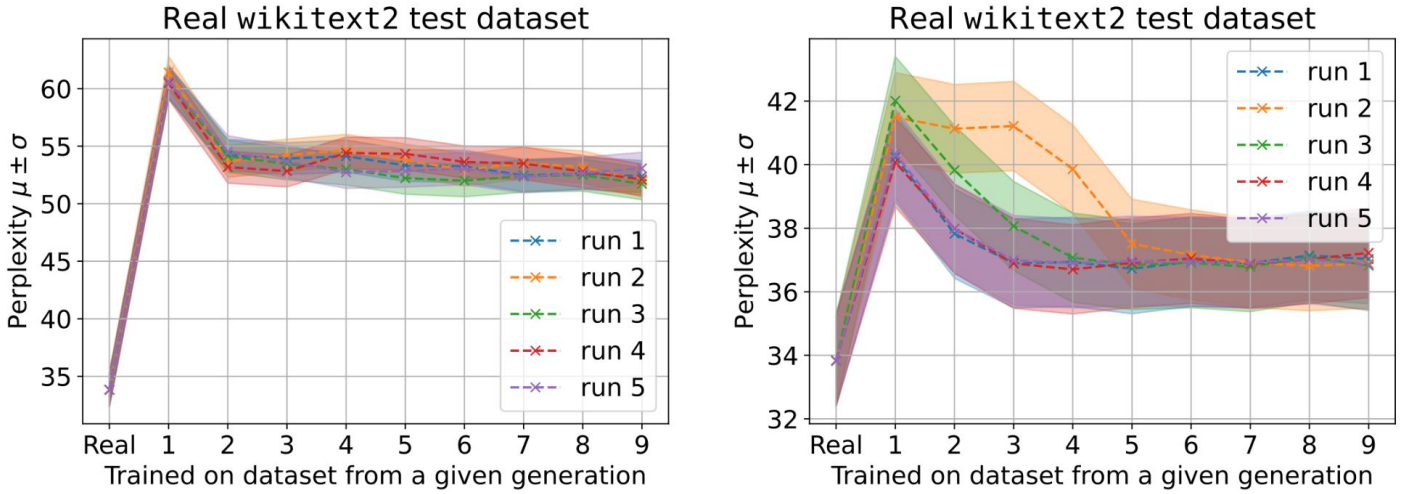
Evaluation

Two Different settings based on the normal distribution sampling approaches that we presented earlier:

- **5 epochs, no original training data**
  - the model is trained for 5 epochs on the original dataset and no original data.

- **10 epochs, 10% of original training data preserved**
  - the model is trained for 10 epochs on the original dataset and every new generation of training, a random 10% of the original data points are sampled.

Evaluation

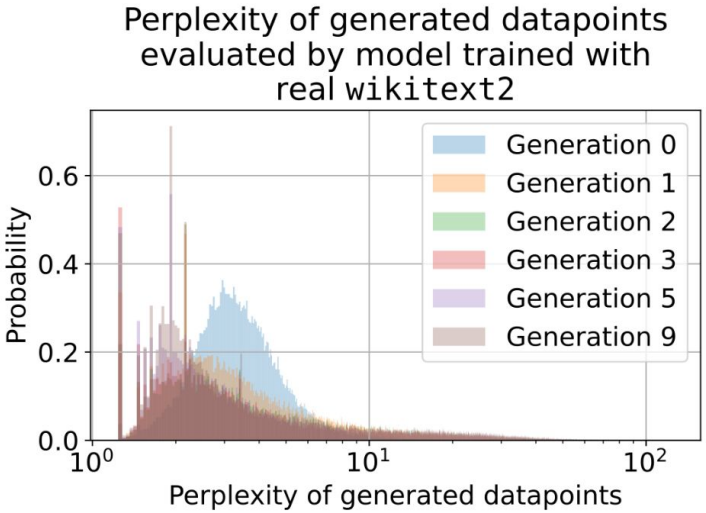


(a) No data preserved, 5 epochs

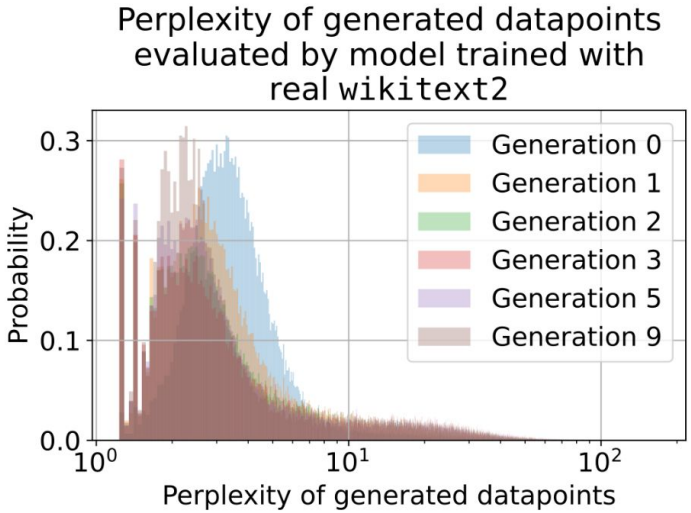
(b) 10% data preserved, 10 epochs

Figure 10: Performance of OPT-125m models of different generations evaluated using the original `wikitext2` test dataset. Perplexity is shown on the y-axis and for each independent run the graph of the mean and its standard deviation is shown with error bars. x-axis refers to the generation of the model – 'Real' refers to the 'model 0' trained on the original `wikitext2` dataset; model 1 was trained on the data produced by model 0; model 2 was trained on data produced by model 1 etc. with all generated datasets equal in size. We find that models trained on generated data are able to learn some of the original task, but with errors, as seen from the increase in perplexity.

Evaluation



(a) No data preserved



(b) 10% data preserved

Figure 11: Histograms of perplexities of each individual data training sequence produced by different generations as is evaluated by the very first model trained with the real data. Over the generations models tend to produce samples that the original model trained with real data is more likely to produce. At the same time, a much longer tail appears for later generations – later generations start producing samples that would never be produced by the original model *i.e.* they start misperceiving reality based on errors introduced by their ancestors. Same plots are shown in 3D in Figure 15.

Strengths & Limitations

Strengths & Limitations

- Strengths

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.
  - Strong mathematical foundations behind the claim.

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.
  - Strong mathematical foundations behind the claim.
  - Empirical data on a variety of ML models, not just LLMs.

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.
  - Strong mathematical foundations behind the claim.
  - Empirical data on a variety of ML models, not just LLMs.

- Limitations

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.
  - Strong mathematical foundations behind the claim.
  - Empirical data on a variety of ML models, not just LLMs.

- Limitations
  - Very math heavy with providing little to no background.

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.
  - Strong mathematical foundations behind the claim.
  - Empirical data on a variety of ML models, not just LLMs.

- Limitations
  - Very math heavy with providing little to no background.
  - No background about (niche) GMMs and VAEs.

Strengths & Limitations

- Strengths
  - First paper to present this concerning recursive learning problem.
  - Strong mathematical foundations behind the claim.
  - Empirical data on a variety of ML models, not just LLMs.

- Limitations
  - Very math heavy with providing little to no background.
  - No background about (niche) GMMs and VAEs.
  - No exploration of from-scratch training of LLMs.

Discussion