Towards trustworthy foundation models with privacy

Grigorios Chrysos

Assistant Professor

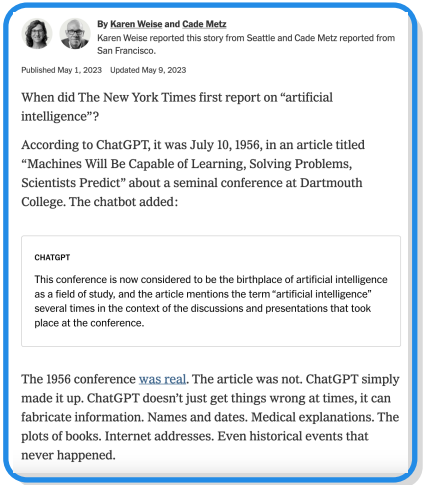
chrysos@wisc.edu

August 13, 2025

Machine Learning Summer School

Hallucination: Growing media attention

2023



By Karen Weise and Cade Metz

Karen Weise reported this story from Seattle and Cade Metz reported from San Francisco.

Published May 1, 2023 Updated May 9, 2023

When did The New York Times first report on “artificial intelligence”?

According to ChatGPT, it was July 10, 1956, in an article titled “Machines Will Be Capable of Learning, Solving Problems, Scientists Predict” about a seminal conference at Dartmouth College. The chatbot added:

CHATGPT

This conference is now considered to be the birthplace of artificial intelligence as a field of study, and the article mentions the term “artificial intelligence” several times in the context of the discussions and presentations that took place at the conference.

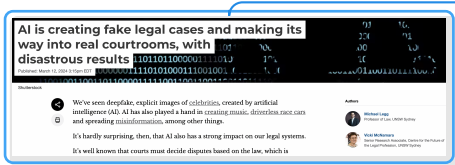
The 1956 conference was real. The article was not. ChatGPT simply made it up. ChatGPT doesn't just get things wrong at times, it can fabricate information. Names and dates. Medical explanations. The plots of books. Internet addresses. Even historical events that never happened.

NY Times

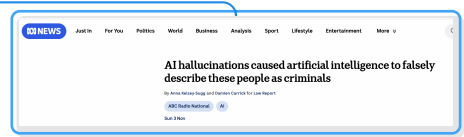
Hallucination: Growing media attention

2024

NY Times



Conversation

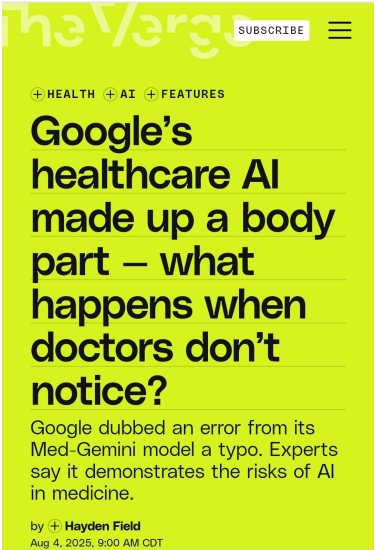


ABC News

Hallucination: Growing media attention

2025

Are there any real-world impacts though?



Are hallucinations new?

Figure 6.8: Failure cases of ProdPoly (trained on FFHQ). The three leftmost faces in the first row seem

Figure: This is a schematic from my PhD thesis 5 years ago - generative models were already known to hallucinate.

How does hallucination manifest in different modalities?

- **Text**
  - Faithfulness Hallucination:Input, context and logical inconsistency
  - Factual Hallucination: Contradiction and fabrication

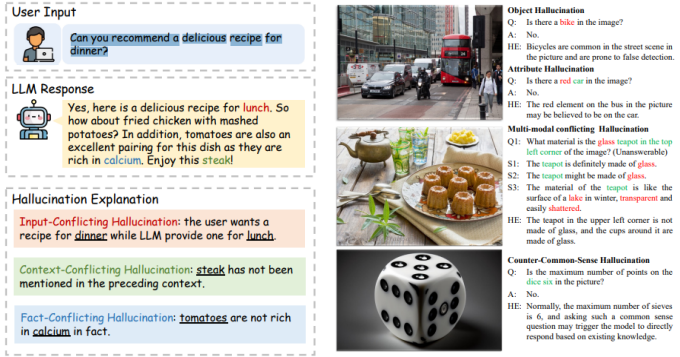


Figure from: Huang et al. (2025), "A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models"

How does hallucination manifest in different modalities?

- **Text**
  - Faithfulness Hallucination:Input, context and logical inconsistency
  - Factual Hallucination: Contradiction and fabrication
- **Image**
  - Object Hallucination: Category, attribute and relation

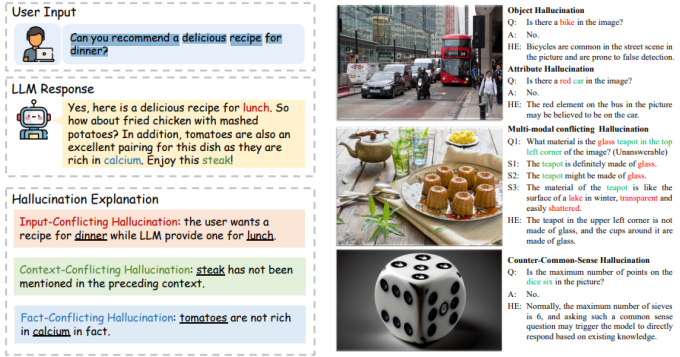


Figure from: Huang et al. (2025), "A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models"

How does hallucination manifest in different modalities?

- **Text**
  - Faithfulness Hallucination:Input, context and logical inconsistency
  - Factual Hallucination: Contradiction and fabrication
- **Image**
  - Object Hallucination: Category, attribute and relation
- **Video and Audio**
  - Audio: Temporal hallucination
  - Video: Over-reliance on the visual modality

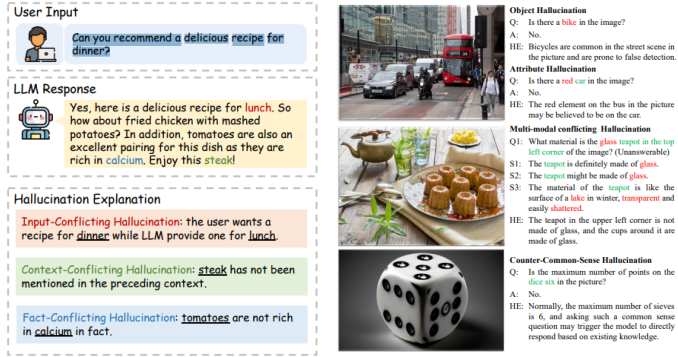


Figure from: Huang et al. (2025), "A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models"

Outline

Categories of LLM Hallucinations

Category	Type	Example Dialog	Explanation
Factuality Hallucination	Factual Contradiction	**Instruction:** What are Thomas Edison's main contributions to science and technology? **Response:** Thomas Edison developed the first practical telephone and invented the light bulb.	The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**.
Faithfulness Hallucination			

Table from: Feng et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions", 2025.

Categories of LLM Hallucinations

Category	Type	Example Dialog	Explanation
Factuality Hallucination	Factual Contradiction	**Instruction:** What are Thomas Edison's main contributions to science and technology? **Response:** Thomas Edison developed the first practical telephone and invented the light bulb.	The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**.
	Factual Fabrication	**Instruction:** What are the major environmental impacts of the construction of the Eiffel Tower? **Response:** The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement.	The response erroneously states that the construction of the Eiffel Tower led to the extinction of the 'Parisian tiger' and sparked the global green architecture movement. In fact, **there is no evidence of a 'Parisian tiger' ever existing, making this a fabricated claim.** Moreover, **attributing the origins of green architecture to the Eiffel Tower is an exaggeration**, as this movement has diverse roots and cannot be traced back to a single event.
Faithfulness Hallucination			

Table from: Feng et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions", 2025.

Categories of LLM Hallucinations

Category	Type	Example Dialog	Explanation
Factuality Hallucination	Factual Contradiction	**Instruction:** What are Thomas Edison's main contributions to science and technology? **Response:** Thomas Edison developed the first practical telephone and invented the light bulb.	The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**.
	Factual Fabrication	**Instruction:** What are the major environmental impacts of the construction of the Eiffel Tower? **Response:** The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement.	The response erroneously states that the construction of the Eiffel Tower led to the extinction of the 'Parisian tiger' and sparked the global green architecture movement. In fact, **there is no evidence of a 'Parisian tiger' ever existing, making this a fabricated claim.** Moreover, **attributing the origins of green architecture to the Eiffel Tower is an exaggeration**, as this movement has diverse roots and cannot be traced back to a single event.
Faithfulness Hallucination	Instruction Inconsistency	**Instruction:** Translate the English question into Spanish: "What is the capital of France?" **Response:** The capital of France is Paris.	The user explicitly provided an instruction to translate the English question into Spanish. **Instead of performing the translation as instructed, the LLM chose to answer the question.**

Table from: Feng et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions", 2025.

Categories of LLM Hallucinations

Category	Type	Example Dialog	Explanation
Factuality Hallucination	Factual Contradiction	**Instruction:** What are Thomas Edison's main contributions to science and technology? **Response:** Thomas Edison developed the first practical telephone and invented the light bulb.	The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**.
	Factual Fabrication	**Instruction:** What are the major environmental impacts of the construction of the Eiffel Tower? **Response:** The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement.	The response erroneously states that the construction of the Eiffel Tower led to the extinction of the 'Parisian tiger' and sparked the global green architecture movement. In fact, **there is no evidence of a 'Parisian tiger' ever existing, making this a fabricated claim.** Moreover, **attributing the origins of green architecture to the Eiffel Tower is an exaggeration**, as this movement has diverse roots and cannot be traced back to a single event.
Faithfulness Hallucination	Instruction Inconsistency	**Instruction:** Translate the English question into Spanish: "What is the capital of France?" **Response:** The capital of France is Paris.	The user explicitly provided an instruction to translate the English question into Spanish. **Instead of performing the translation as instructed, the LLM chose to answer the question.**
	Context Inconsistency	**Retrieved context:** The Nile is a major north-flowing river in northeastern Africa. It's traditionally considered the longest river in the world. Originating in the Great Lakes region of central Africa, it flows through several countries before emptying into the Mediterranean Sea. **Instruction:** Summarize the retrieval document above. **Response:** The Nile originates from the mountain ranges of central Africa and flows into the Mediterranean.	The summary incorrectly states that the Nile originates from the "mountain ranges of central Africa", while the **paragraph mentions that the Nile originates in the Great Lakes region of central Africa.**

Table from: Feng et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions", 2025.

Categories of LLM Hallucinations

Category	Type	Example Dialog	Explanation
Factuality Hallucination	Factual Contradiction	**Instruction:** What are Thomas Edison's main contributions to science and technology? **Response:** Thomas Edison developed the first practical telephone and invented the light bulb.	The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**.
	Factual Fabrication	**Instruction:** What are the major environmental impacts of the construction of the Eiffel Tower? **Response:** The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement.	The response erroneously states that the construction of the Eiffel Tower led to the extinction of the 'Parisian tiger' and sparked the global green architecture movement. In fact, **there is no evidence of a 'Parisian tiger' ever existing, making this a fabricated claim.** Moreover, **attributing the origins of green architecture to the Eiffel Tower is an exaggeration**, as this movement has diverse roots and cannot be traced back to a single event.
Faithfulness Hallucination	Instruction Inconsistency	**Instruction:** Translate the English question into Spanish: "What is the capital of France?" **Response:** The capital of France is Paris.	The user explicitly provided an instruction to translate the English question into Spanish. **Instead of performing the translation as instructed, the LLM chose to answer the question.**
	Context Inconsistency	**Retrieved context:** The Nile is a major north-flowing river in northeastern Africa. It's traditionally considered the longest river in the world. Originating in the Great Lakes region of central Africa, it flows through several countries before emptying into the Mediterranean Sea. **Instruction:** Summarize the retrieval document above. **Response:** The Nile originates from the mountain ranges of central Africa and flows into the Mediterranean.	The summary incorrectly states that the Nile originates from the "mountain ranges of central Africa", while the **paragraph mentions that the Nile originates in the Great Lakes region of central Africa.**
	Logical Inconsistency	**Instruction:** Solve the equation $2x + 3 = 11$ step by step. **Response:** Step 1: Subtract 3 from both sides to get: $2x = 8$. Step 2: Divide both sides by 2 to get: $x = 3$. So the result of the equation is x=3.	The first step correctly isolates the term with x to get $2x = 8$. However, **the next step inaccurately divides 8 by 2 to yield a result of $x = 3$**, which is inconsistent with the earlier reasoning.

Table from: Feng et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions", 2025.

Literature on Hallucination

- **Causes and Analysis**
  - Hallucination from Data: Bengio et al. (2015) (NeurIPS), Bender et al. (2021) (ACM), ...
  - Hallucination from Training: R-tuning (Zhang et al., 2024), Perez et al. (2022) (ACL), ...
  - Hallucination from Inference: Holtzman et al. (2020) (ICLR), Chen et al. (2022) (NeurIPS), ...
- **Detection Methods**
  - Factuality Hallucination Detection: FActScore (Min et al., 2023), Cove (Dhuliawala et al., 2024), ...
  - Faithfulness Hallucination Detection: Lakshminarayanan et al. (2017) (NeurIPS), Zhang et al. (2023) (EMNLP), ...
- **Benchmarks and Evaluation**
  - Hallucination Evaluation Benchmarks: TruthfulQA (Lin et al., 2022), PopQA (Mallen et al., 2023), ...
  - Hallucination Detection Benchmarks: BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020), ...
- **Mitigation Strategies:** Transformer-Patcher (Huang et al., 2023), RARR (Gao et al., 2023), Zhang et al. (2019) (ACM), ...
- **Surveys:** Huang et al. (2025) (ACM), Bai et al. (2025), ...
- **Additional resources:** Rawte et al. (2024), https://vr25.github.io/coling25-hallucination-tutorial

Causes for Hallucination (Huang et al., 2025)

Question

Can we identify key components that might give rise to hallucinations?

Causes for Hallucination (Huang et al., 2025)

Question

Can we identify key components that might give rise to hallucinations?

1. **Data**
   - Dataset biases.
   - Knowledge Boundary: long-tail knowledge, up-to-date knowledge, or copyright-sensitive knowledge.
   - Inferior Alignment Data.

Causes for Hallucination (Huang et al., 2025)

Question

Can we identify key components that might give rise to hallucinations?

1. **Data**
   - Dataset biases.
   - Knowledge Boundary: long-tail knowledge, up-to-date knowledge, or copyright-sensitive knowledge.
   - Inferior Alignment Data.
2. **Training**
   - Pre-training algorithms, architectures and regularizations.
   - Supervised Fine-Tuning.
   - Reinforcement Learning from Human Feedback.

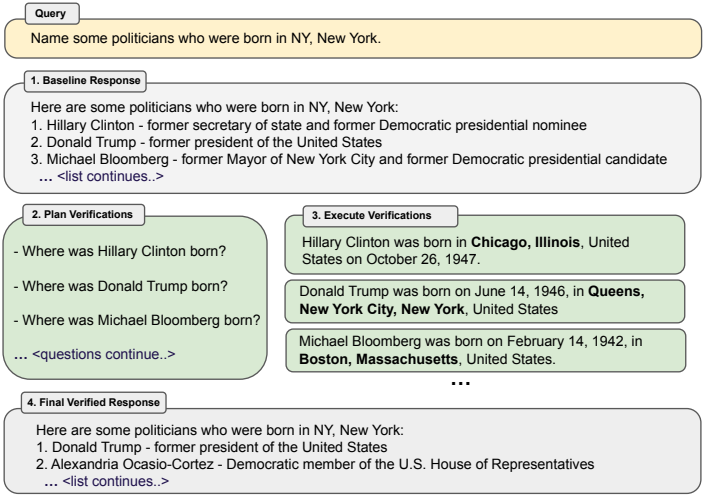
Causes for Hallucination (Huang et al., 2025)

Question

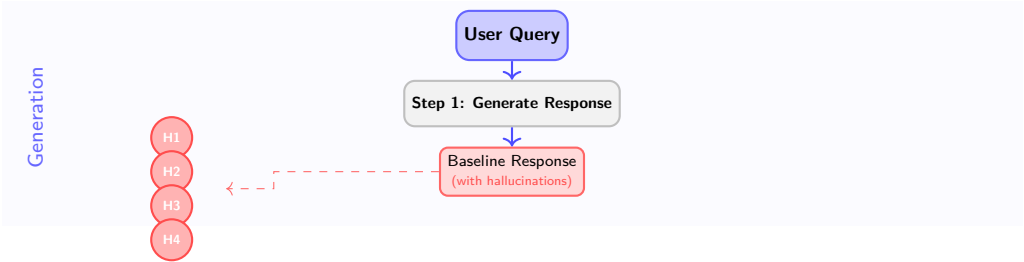
Can we identify key components that might give rise to hallucinations?

1. **Data**
   - Dataset biases.
   - Knowledge Boundary: long-tail knowledge, up-to-date knowledge, or copyright-sensitive knowledge.
   - Inferior Alignment Data.
2. **Training**
   - Pre-training algorithms, architectures and regularizations.
   - Supervised Fine-Tuning.
   - Reinforcement Learning from Human Feedback.
3. **Inference**
   - Decoding Strategies.
   - Over-confidence.
   - Architectural components (e.g., softmax).
   - Reasoning Failure.

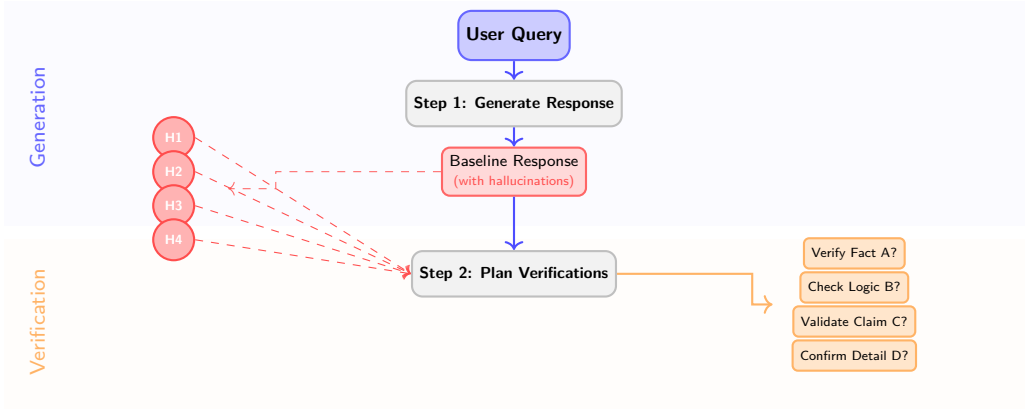
Chain-of-Verification Reduces Hallucination in Large Language Models (CoVe) (Dhuliawala et al., 2024)



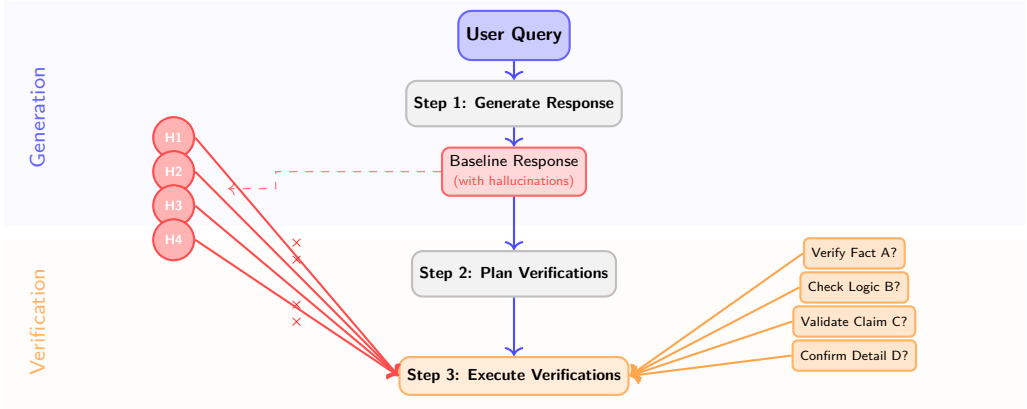
Chain-of-Verification Process (Dhuliawala et al., 2024)



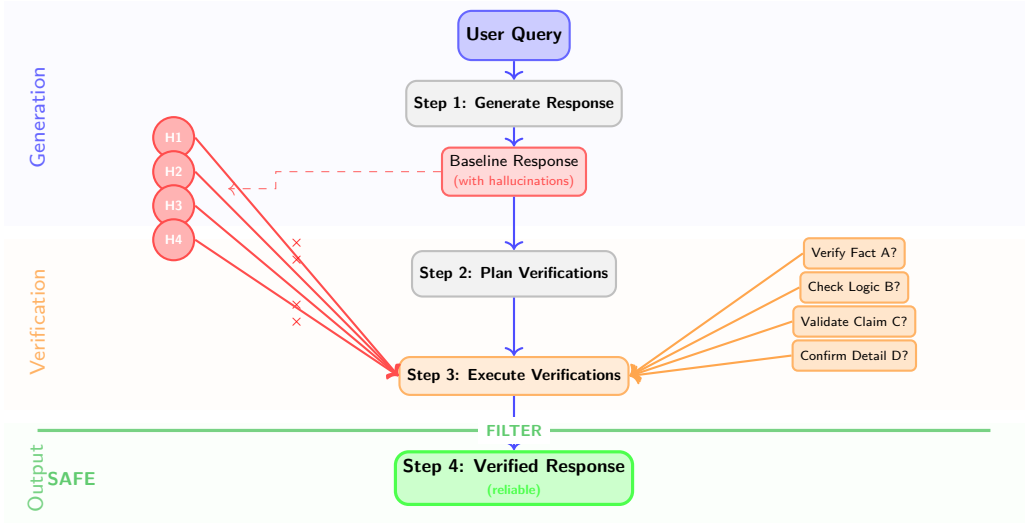
Chain-of-Verification Process (Dhuliawala et al., 2024)



Chain-of-Verification Process (Dhuliawala et al., 2024)



Chain-of-Verification Process (Dhuliawala et al., 2024)



Hallucination on images

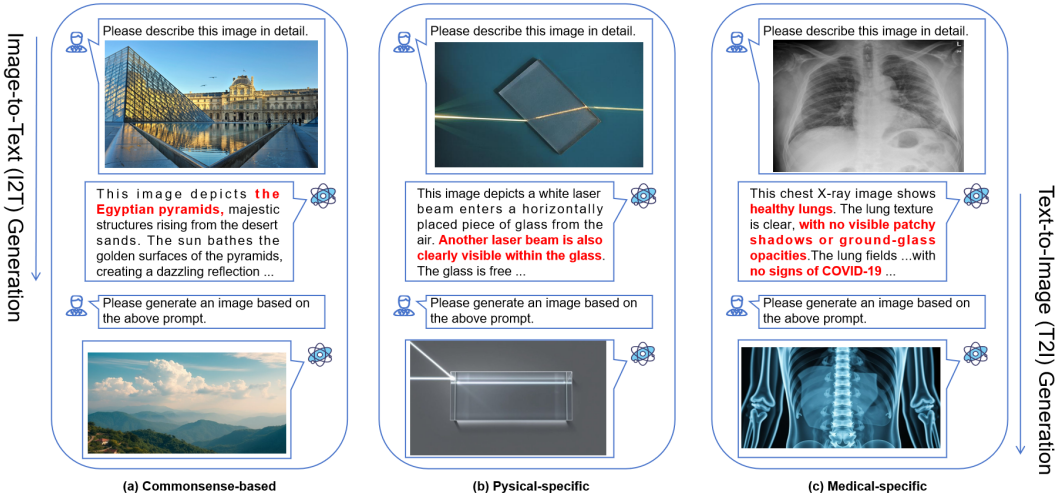


Figure from Chen et al. (2025), "A Survey of Multimodal Hallucination Evaluation and Detection"

Hallucination on images

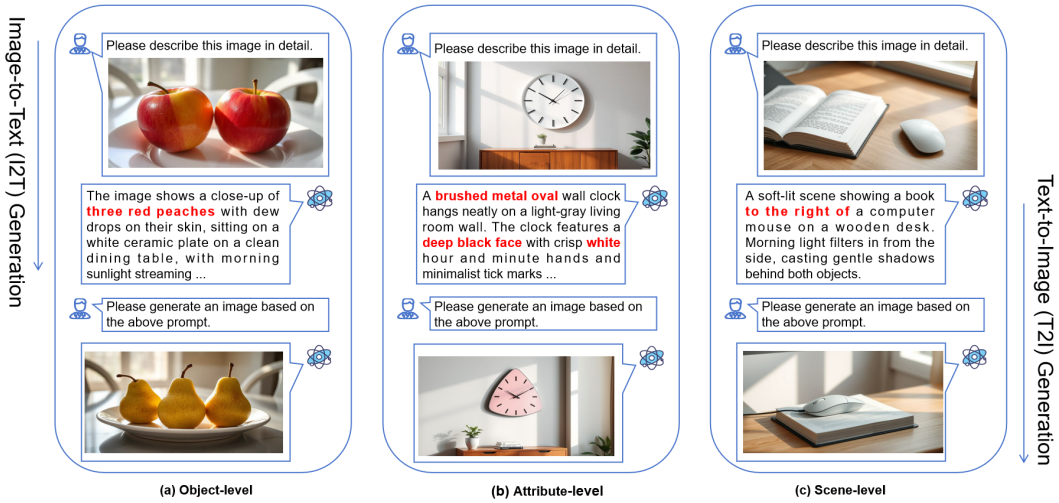


Figure from Chen et al. (2025), "A Survey of Multimodal Hallucination Evaluation and Detection"

Denoising Diffusion Models (Ho et al., 2020)

- Forward process:

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}),$$

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}).$$

- The forward process variances β_t can be learned.

Denoising Diffusion Models (Ho et al., 2020)

- Forward process:

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}),$$

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}).$$

- The forward process variances β_t can be learned.
- Using the notation $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}).$$

Denoising Diffusion Models (Ho et al., 2020)

- Forward process:

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}),$$

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\,\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}).$$

- The forward process variances β_t can be learned.
- Using the notation $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}).$$

- Backward process:

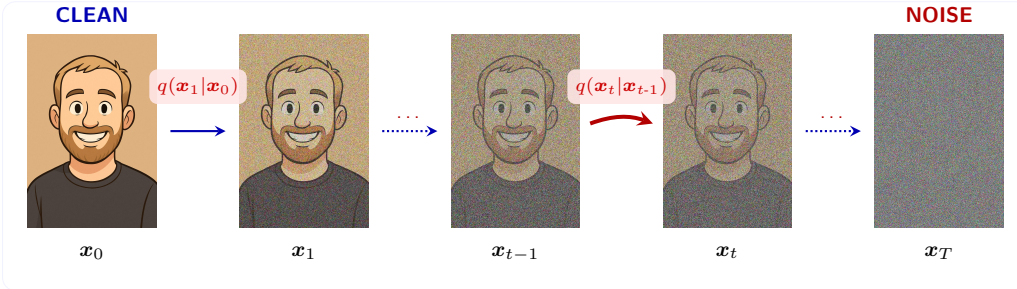
$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\Big(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\Big),$$

for parametrized model with parameters $\boldsymbol{\theta}$.

Diffusion process (Ho et al., 2020)

Forward Process



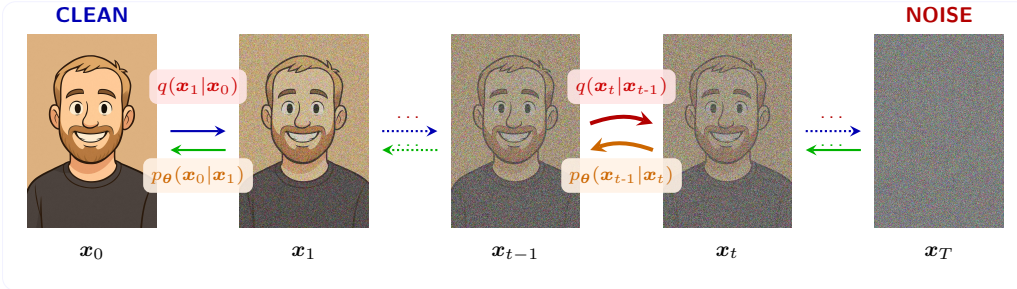
Forward: $q(\boldsymbol{x}_t|\boldsymbol{x}_{t\text{-}1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\,\boldsymbol{x}_{t\text{-}1}, \beta_t\mathbf{I})$

Diffusion Notation

$\boldsymbol{x}_0$	Original data
$\boldsymbol{x}_t$	Noisy state at time t
$q(\boldsymbol{x}_t\|\boldsymbol{x}_{t\text{-}1})$	Forward transition
β_t	Noise variance schedule
T	Final timestep

Diffusion process (Ho et al., 2020)

Reverse Process



Forward: $q(\boldsymbol{x}_t|\boldsymbol{x}_{t\text{-}1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\,\boldsymbol{x}_{t\text{-}1}, \beta_t\mathbf{I})$

Reverse: $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t\text{-}1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t\text{-}1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t))$

Diffusion Notation

$\boldsymbol{x}_0$	Original data
$\boldsymbol{x}_t$	Noisy state at time t
$q(\boldsymbol{x}_t\|\boldsymbol{x}_{t\text{-}1})$	Forward transition
β_t	Noise variance schedule
T	Final timestep

Training of DDPM

Training Algorithm for DDPM

For every image $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^M$ in your training dataset:

- Repeat until convergence:
  - Sample $t \sim \mathrm{Uniform}[1, T]$.
  - Draw a sample $\boldsymbol{x}_t^{(m)} \sim \mathcal{N}(\boldsymbol{x}_t \mid \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, \ \ (1-\overline{\alpha}_t)I)$, i.e., Compute $\boldsymbol{x}_t^{(m)} = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_t^{(m)}$, $\epsilon_t^{(m)} \sim \mathcal{N}(0, I)$.
  - Update $\boldsymbol{\theta}$ using gradient of

$$\frac{1}{M}\sum_{m=1}^{M}\left\|\widehat{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t^{(m)}) - \boldsymbol{x}_0\right\|^2 .$$

Inference of DDPM

Inference of DDPM

- Start from a white noise vector: $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$.
- For $t = T, T-1, \ldots, 1$, repeat:
  - Compute denoised estimate $\widehat{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$ using the trained denoiser.
  - Update $\boldsymbol{x}_{t-1}$ using:

$$\boldsymbol{x}_{t-1} = \frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\widehat{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t) + \sigma_q(t)\,\epsilon, \epsilon \sim \mathcal{N}(0, \boldsymbol{I})\,.$$

What is Hallucination in Diffusion Models?

Definition (Hallucination (Aithal et al., 2024))

A phenomenon where diffusion models generate samples that lie **completely outside** the support of the training distribution.

Mathematical Definition (Aithal et al., 2024)

The ϵ-Hallucination set:

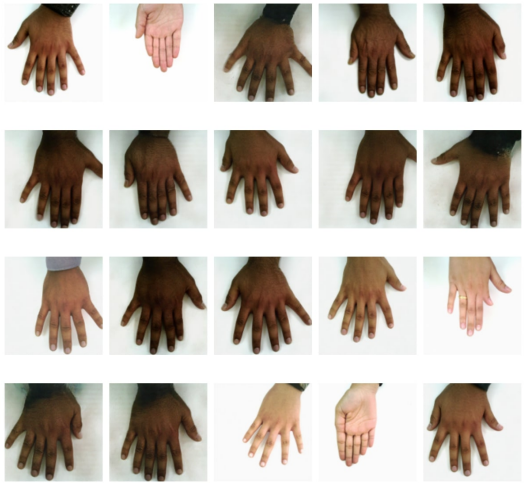
$$H_\epsilon(q) = \{x : q(x) \leq \epsilon\},$$

where $q(x)$ is the true data distribution and $\epsilon \approx 0$.

Examples: ○ Hands with 6+ fingers
○ Impossible combinations

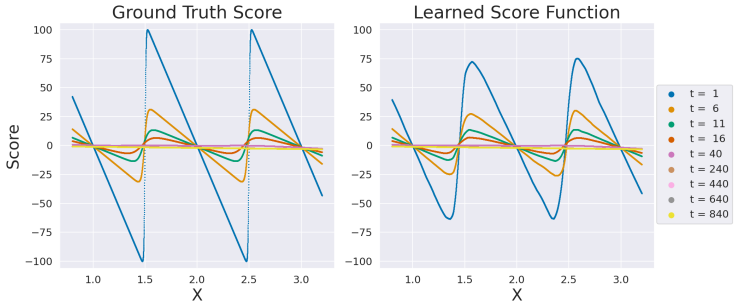
Key Point: These aren't just "bad" generations - they're generations of things that **never existed** in training!

How do the hallucinated images look like?

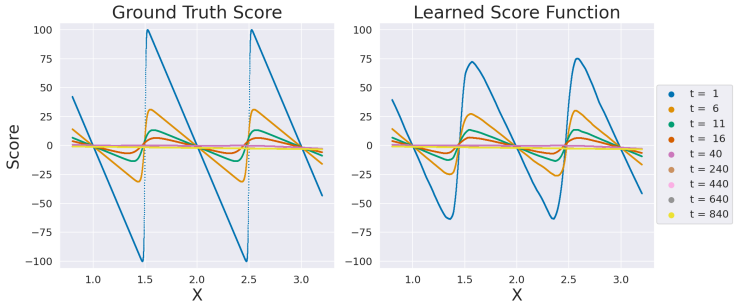


When do the hallucinated images appear during the diffusion process?

The Root Cause: Smooth Score Function Approximation



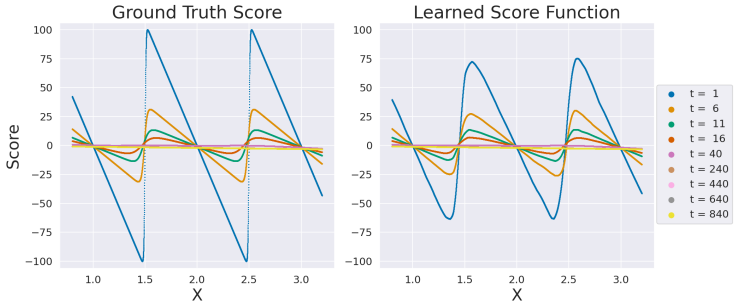
The Root Cause: Smooth Score Function Approximation



The Problem

- True score has sharp jumps
- Neural networks learn smooth approximations
- Smooth transitions create interpolations

The Root Cause: Smooth Score Function Approximation



The Problem

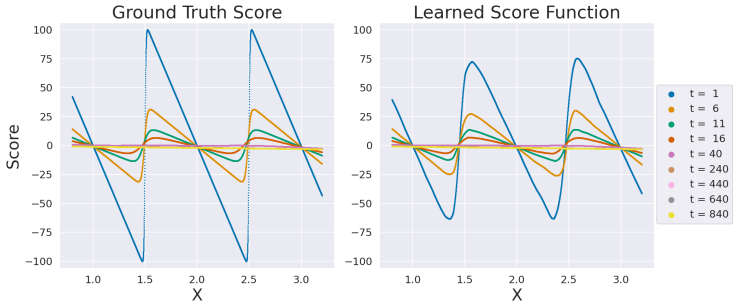
- True score has sharp jumps
- Neural networks learn smooth approximations
- Smooth transitions create interpolations

Mathematical Insight

The score function: $s_\theta(x_t, t) = -\frac{\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}$

- **Ground truth:** Step functions between modes
- **Neural network:** Smooth approximation
- **Result:** Uncertainty regions $\rightarrow$ interpolated samples

The Root Cause: Smooth Score Function Approximation



The Problem

- True score has sharp jumps
- Neural networks learn smooth approximations
- Smooth transitions create interpolations

Mathematical Insight

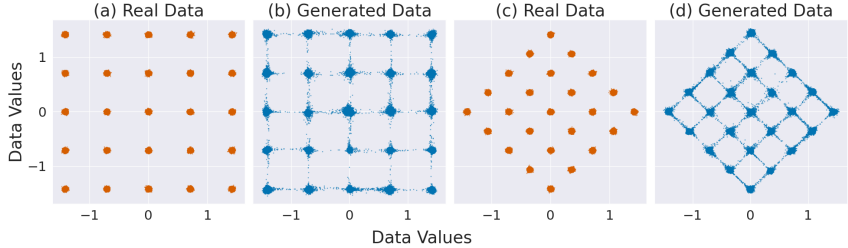
The score function: $s_\theta(x_t, t) = -\frac{\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}$

- **Ground truth:** Step functions between modes
- **Neural network:** Smooth approximation
- **Result:** Uncertainty regions $\rightarrow$ interpolated samples

Validation

Using true score function $\rightarrow$ No mode interpolation observed!

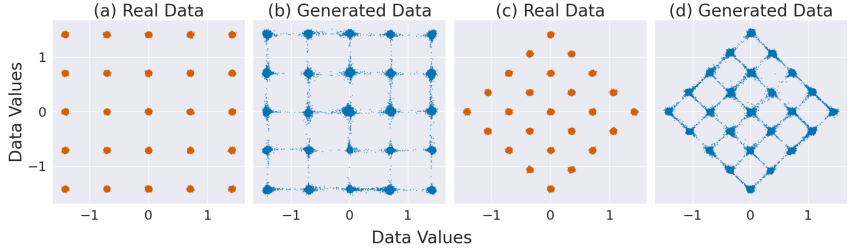
2D Evidence: Grid of Gaussians



Key Insights

- (a) **Left:** Original 25 Gaussians in square grid (orange dots)
- (b) **Right:** Model samples (blue dots) show interpolation between nearest neighbors
- (c)-(d) **Diamond experiment:** No interpolation along x-axis when modes are rotated

2D Evidence: Grid of Gaussians



Key Insights

- (a) **Left:** Original 25 Gaussians in square grid (orange dots)
- (b) **Right:** Model samples (blue dots) show interpolation between nearest neighbors
- (c)-(d) **Diamond experiment:** No interpolation along x-axis when modes are rotated

Critical Discovery

Mode interpolation only occurs between **nearest neighboring modes**!

Key Factors Affecting Mode Interpolation

What Makes Hallucinations Worse?

1. **Limited Training Data**
   Fewer samples $\rightarrow$ more uncertainty $\rightarrow$ more interpolation
2. **Close Mode Spacing**
   Nearby modes in data space $\rightarrow$ easier to interpolate
3. **Fewer Diffusion Steps**
   Lower $T \rightarrow$ coarser approximation $\rightarrow$ more interpolation
4. **High Dimensionality**
   Complex representations $\rightarrow$ more interpolation opportunities

Key Factors Affecting Mode Interpolation

What Makes Hallucinations Worse?

1. **Limited Training Data**
   Fewer samples $\rightarrow$ more uncertainty $\rightarrow$ more interpolation
2. **Close Mode Spacing**
   Nearby modes in data space $\rightarrow$ easier to interpolate
3. **Fewer Diffusion Steps**
   Lower T $\rightarrow$ coarser approximation $\rightarrow$ more interpolation
4. **High Dimensionality**
   Complex representations $\rightarrow$ more interpolation opportunities

Design Implications

- Need more training data for complex distributions
- Consider mode spacing in dataset curation
- Higher T for better quality (but slower sampling)
- Architecture choices matter for representation learning

Can We Detect Hallucinations?

Trajectory Variance Method

Hallucinated samples have **high variance** during reverse diffusion. Then, if we use

$$\text{Variance} = \frac{1}{T} \sum_{t=1}^{T} \|x_t - \bar{x}\|^2 ,$$

where $\bar{x}$ is the mean trajectory and x_t are intermediate states, we can filter them out.

Can We Detect Hallucinations?

Trajectory Variance Method

Hallucinated samples have **high variance** during reverse diffusion. Then, if we use

$$\text{Variance} = \frac{1}{T} \sum_{t=1}^{T} \|x_t - \bar{x}\|^2 ,$$

where $\bar{x}$ is the mean trajectory and x_t are intermediate states, we can filter them out.

Post-hoc processing strategy

- Generate more samples than needed (e.g., 100k instead of 10k).
- Filter out high-variance samples.

Can We Detect Hallucinations?

Trajectory Variance Method

Hallucinated samples have **high variance** during reverse diffusion. Then, if we use

$$\text{Variance} = \frac{1}{T}\sum_{t=1}^{T} \|x_t - \bar{x}\|^2 ,$$

where $\bar{x}$ is the mean trajectory and x_t are intermediate states, we can filter them out.

Post-hoc processing strategy

- Generate more samples than needed (e.g., 100k instead of 10k).
- Filter out high-variance samples.

Practical Limitation

In realistic datasets with many modes, this might not be sufficient.

Can We Detect Hallucinations?

Trajectory Variance Method

Hallucinated samples have **high variance** during reverse diffusion. Then, if we use

$$\text{Variance} = \frac{1}{T}\sum_{t=1}^{T} \|x_t - \bar{x}\|^2 ,$$

where $\bar{x}$ is the mean trajectory and x_t are intermediate states, we can filter them out.

Post-hoc processing strategy

- Generate more samples than needed (e.g., 100k instead of 10k).
- Filter out high-variance samples.

Practical Limitation

In realistic datasets with many modes, this might not be sufficient.

Interesting open questions

- *Can we develop "hallucination-proof" architectures?*
- *How can we handle the density of the modes?*
- *Can we develop a theoretical understanding of when hallucination occurs?*
- *Should we watermark all synthetic content?*

Outline

Question

Are hallucinations *always* unwanted?

Question

Are hallucinations *always* unwanted?

The Challenge: Incomplete Data

- Continuous labels rarely available (clipped instead to binary $\{0, 1\}$).
- Labels don't contain all variations.

Generation beyond what we've seen

Traditional wisdom: Models can only learn what they observe.

Generation beyond what we've seen

Traditional wisdom: Models can only learn what they observe.

Question

Can a machine learning model generate data in a region the model has **never seen** during training?

Generation beyond what we've seen

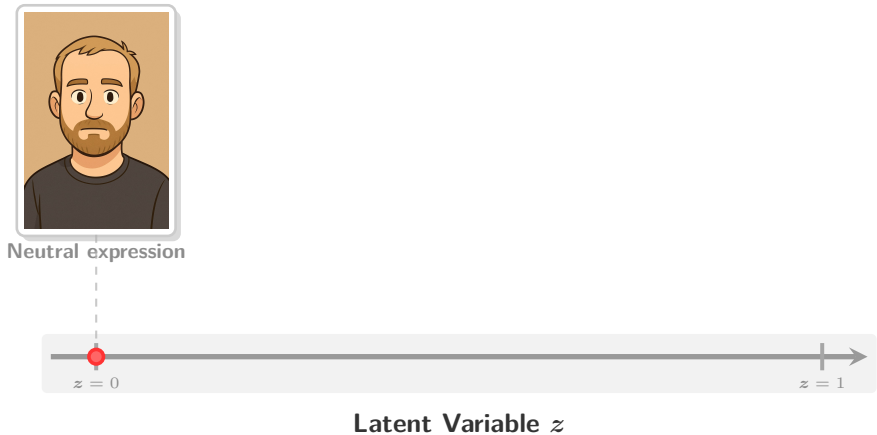
Traditional wisdom: Models can only learn what they observe.

Question

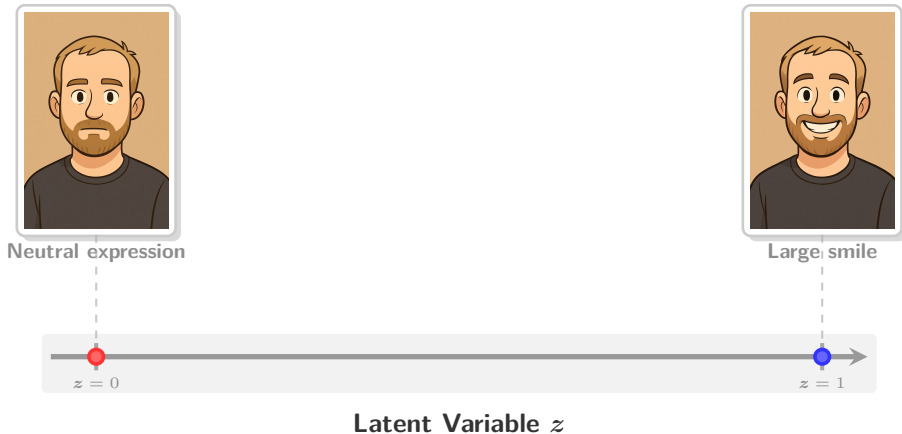
Can a machine learning model generate data in a region the model has **never seen** during training?

Zero-shot interpolation: Generating intermediate examples without training on them.

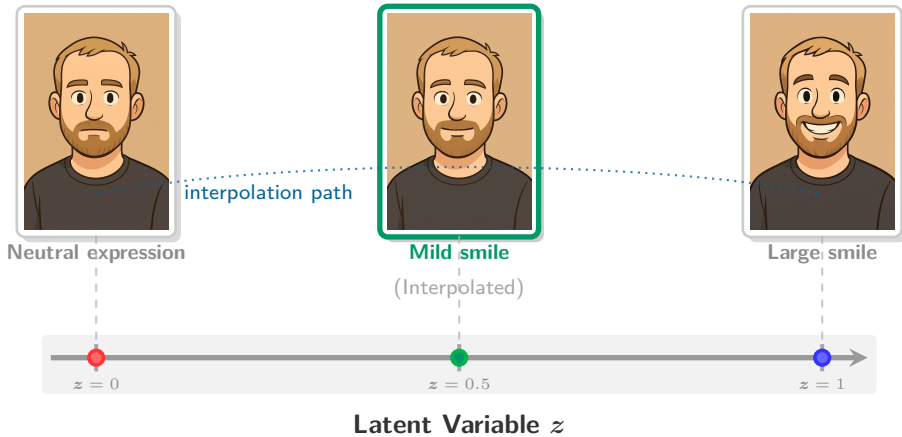
What if we could synthesize in intermediate regions?



What if we could synthesize in intermediate regions?



What if we could synthesize in intermediate regions?



How did we ensure only extrema exist in our case?

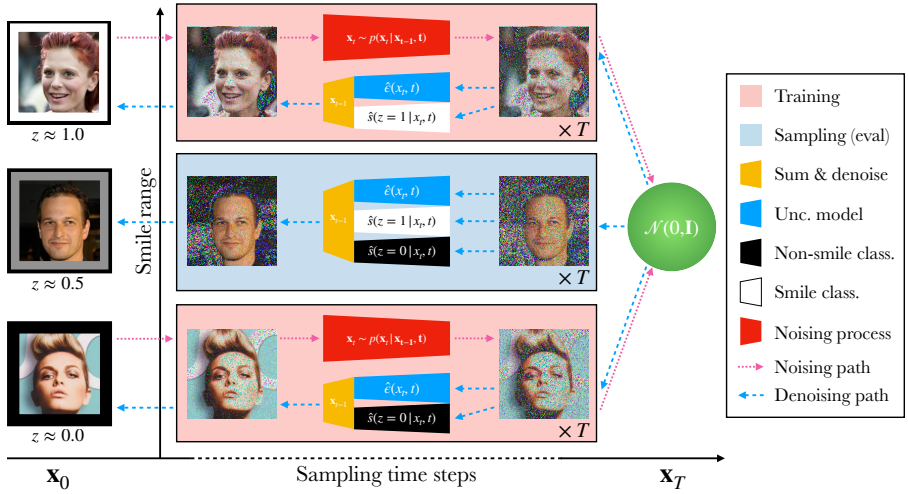
Algorithm Training the filtering and evaluation classifiers (EfficientNet)

Input: Raw dataset $\mathcal{D}$, number of filtering classifiers $N = 5$
Output: N filtering classifiers, 1 evaluation classifier, dataset $\tilde{\mathcal{D}}$
for $i \leftarrow 1$ to N **do**
    $\mathcal{D}_i \leftarrow \text{sample}(\mathcal{D}, M)$ ▷ Sample M elements without replacement
    $f(x; \theta_i) \leftarrow \text{train}(\mathcal{D}_i, \theta_{\text{init},i})$ ▷ Train filtering classifier i
end for
$f(x; \theta_{N+1}) \leftarrow \text{train}(\mathcal{D}, \theta_{\text{init},N+1})$ ▷ Train evaluation classifier
$\tilde{\mathcal{D}} \leftarrow [\,]$ ▷ List of extreme samples
for $(x_i, y_i) \in \mathcal{D}$ **do**
    if all($[f(x_i; \theta_j)(y_i) > 1 - \epsilon$ for $j = 1$ to $N + 1]$) **then**
        $\tilde{\mathcal{D}}$.append((x_i, y_i)) ▷ Keep if all classifiers agree
    end if
end for

Quality Assurance

Ensemble ensures only samples with **unanimously clear attributes** remain in training set.

Zero-shot interpolation with diffusion models [Deschenaux, Krawczuk, Chrysos, Cevher, ICML (2024)]



Does the interpolation still happen in more constrained settings?

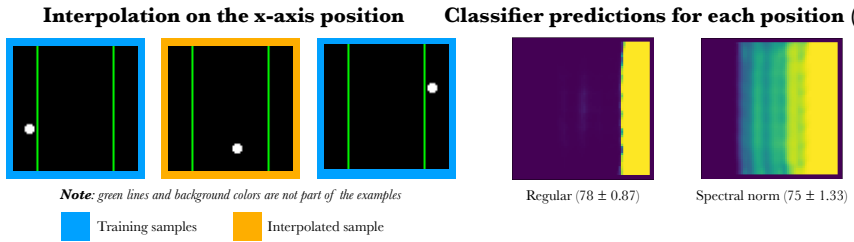
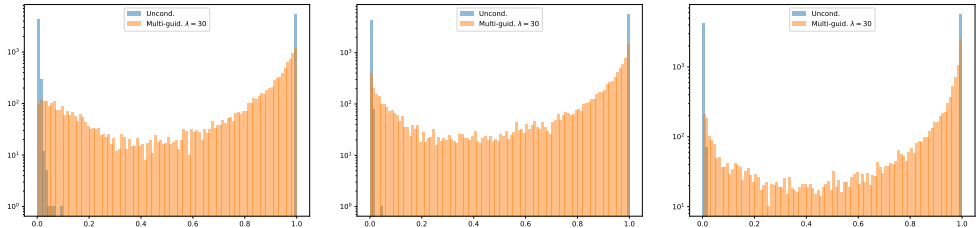


Figure: The training samples include exclusively balls outside of the central region.

*J Deschenaux, I Krawczuk, **G Chrysos**, V Cevher, 'Going Beyond Compositions: Zero-Shot Interpolation in DDPMs'. In **ICML**, 2024.*

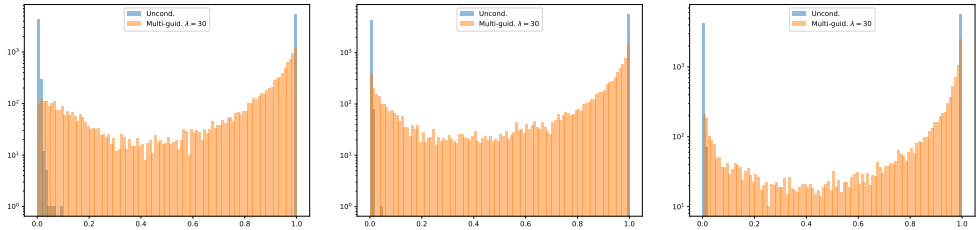
What's the impact of the training data size?



(a) 30k training samples (b) 10k training samples (c) 5k training samples

Figure: Performance (on the smile attribute) when reducing the training data size. **(a)**: With 30k training samples. **(b)**: With 10k training samples. **(c)**: With 5k training samples. While using 30k or 10k maintains similar performance, the interpolation capability starts to degrade with 5k, even though DDPM retains its ability to interpolate.

What's the impact of the training data size?



(a) 30k training samples (b) 10k training samples (c) 5k training samples

Figure: Performance (on the smile attribute) when reducing the training data size. **(a)**: With 30k training samples. **(b)**: With 10k training samples. **(c)**: With 5k training samples. While using 30k or 10k maintains similar performance, the interpolation capability starts to degrade with 5k, even though DDPM retains its ability to interpolate.

Interesting open questions

- *Can we use reference images to reduce the amount of images required?*
- *Are there better sampling strategies for more precise control?*
- *Can we move beyond the interpolation range, towards extrapolation (e.g. (Wu et al., 2022))?*

Broader Implications: Generation in a long-tailed world

Connection with long-tailed world

Interpolation can help address **rare groups** in training data!

Medical Imaging

- Rare pathologies
- Interpolation could generate intermediate severity cases
- Better training data for diagnostic systems

Demographic Balance

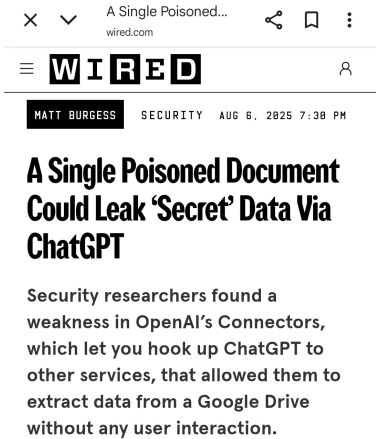
- Training sets often have demographic gaps
- Generate missing intersectional identities
- Richer training examples for AI systems

Broader Applications

- **Data augmentation**: Create training data for edge cases
- **Content creation**: Novel intermediate styles/expressions
- **Scientific discovery**: Explore parameter spaces between known examples

Outline

Is privacy still important in the era of AI agents?



A case for privacy

Question

Can we reveal private information stored in a language model?

A case for privacy

Question

Can we reveal private information stored in a language model?

Assume we have a language model f_{θ} trained on text corpus Y.

Examples

- "Hello, how are you?": High probability.
- "item people low reader": Low probability.

A case for privacy

Question

Can we reveal private information stored in a language model?

Assume we have a language model $f_{\boldsymbol{\theta}}$ trained on text corpus Y.

Examples

- "Hello, how are you?": High probability.
- "item people low reader": Low probability.

- Denote the **log-perplexity** of a sequence of tokens $x_1, ..., x_n$ for model $f_{\boldsymbol{\theta}}$ as:

$$P_{\boldsymbol{\theta}}(x_1, ..., x_n) = -\log_2 \Pr(x_1, ..., x_n | f_{\boldsymbol{\theta}}) = \sum_{i=1}^{n} (-\log_2 \Pr(x_i | f_{\boldsymbol{\theta}}(x_1, ..., x_{i-1}))) \tag{1}$$

- A low perplexity $\implies$ model assigns high probability of occurrence to a sequence. Similarly, a high perplexity $\implies$ model assigns low probability to a sequence.

A case for privacy

Question

Can we reveal private information stored in a language model?

Assume we have a language model $f_{\boldsymbol{\theta}}$ trained on text corpus Y.

Examples

- "Hello, how are you?": High probability.
- "item people low reader": Low probability.

- Denote the **log-perplexity** of a sequence of tokens $x_1, ..., x_n$ for model $f_{\boldsymbol{\theta}}$ as:

$$P_{\boldsymbol{\theta}}(x_1, ..., x_n) = -\log_2 \Pr(x_1, ..., x_n | f_{\boldsymbol{\theta}}) = \sum_{i=1}^{n} (-\log_2 \Pr(x_i | f_{\boldsymbol{\theta}}(x_1, ..., x_{i-1}))) \tag{1}$$

- A low perplexity $\implies$ model assigns high probability of occurrence to a sequence. Similarly, a high perplexity $\implies$ model assigns low probability to a sequence.

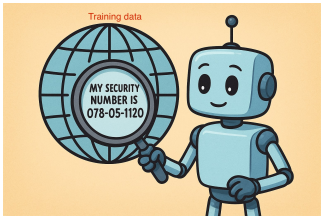
Question

But, what if "item people low reader" was in the training data? Would this still hold?

A case for privacy in the language model era

Question

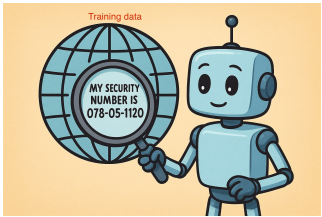
What if the phrase "my security number is 078-05-1120" was in the training data?



A case for privacy in the language model era

Question

What if the phrase "my security number is 078-05-1120" was in the training data?

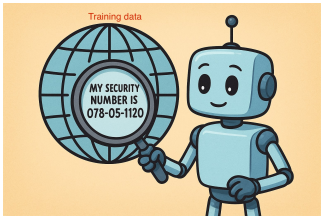


- Then, in this trained model, "my social security number is 078-05-1120" would have low perplexity.

A case for privacy in the language model era

Question

What if the phrase "my security number is 078-05-1120" was in the training data?

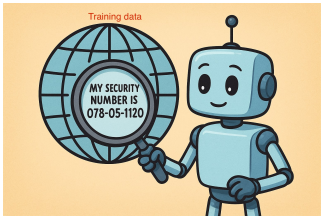


- Then, in this trained model, "my social security number is 078-05-1120" would have low perplexity.
- That meant that we can infer that this is the social security number of someone in the training data–**a privacy violation**.

A case for privacy in the language model era

Question

What if the phrase "my security number is 078-05-1120" was in the training data?



- Then, in this trained model, "my social security number is 078-05-1120" would have low perplexity.
- That meant that we can infer that this is the social security number of someone in the training data–**a privacy violation**.
- An adversary can use this SSN to file e.g. fraudulent tax returns, steal social benefits, etc.

A case for privacy in the language model era

- More formally, let $\mathcal{R}$ be a set of all phrases of the form: "my social security number is ????-??-????".

A case for privacy in the language model era

- More formally, let $\mathcal{R}$ be a set of all phrases of the form: "my social security number is ????-??-????".
- Let $r^* \in \mathcal{R}$ be the "canary", or secret phrase we want to protect.

A case for privacy in the language model era

- More formally, let $\mathcal{R}$ be a set of all phrases of the form: "my social security number is ????-??-????".
- Let $r^* \in \mathcal{R}$ be the "canary", or secret phrase we want to protect.
- Suppose we sort all of these phrases in increasing order of log-perplexity according to f_{θ}; the **rank** of a phrase in $\mathcal{R}$ is its position in this list.

A case for privacy in the language model era

- More formally, let $\mathcal{R}$ be a set of all phrases of the form: "my social security number is ????-??-????".
- Let $r^* \in \mathcal{R}$ be the "canary", or secret phrase we want to protect.
- Suppose we sort all of these phrases in increasing order of log-perplexity according to f_{θ}; the **rank** of a phrase in $\mathcal{R}$ is its position in this list.
- Candidates with the highest rank are the most "exposed":

Exposure (Carlini et al., 2019)

The exposure of a phrase $r \in \mathcal{R}$ is:

$$\log_2 |\mathcal{R}| - \log_2 \mathsf{rank}(r) \tag{2}$$

A case for privacy in the language model era

- More formally, let $\mathcal{R}$ be a set of all phrases of the form: "my social security number is ????-??-????".
- Let $r^* \in \mathcal{R}$ be the "canary", or secret phrase we want to protect.
- Suppose we sort all of these phrases in increasing order of log-perplexity according to f_{θ}; the **rank** of a phrase in $\mathcal{R}$ is its position in this list.
- Candidates with the highest rank are the most "exposed":

Exposure (Carlini et al., 2019)

The exposure of a phrase $r \in \mathcal{R}$ is:

$$\log_2 |\mathcal{R}| - \log_2 \mathsf{rank}(r) \tag{2}$$

- The exposure ranges from 0 to $\log_2 |\mathcal{R}|$; a larger exposure indicates that the phrase is more noticeable.

A case for privacy in the language model era

- In the experiments of Carlini et al. (2019), they insert the canary several times in the training dataset.

A case for privacy in the language model era

- In the experiments of Carlini et al. (2019), they insert the canary several times in the training dataset.
- Figure 4 is for a language model f_θ trained on 100k sequences.
- When the canary is inserted once, it is 1000× more likely than random chance, and by 4 insertions maximum exposure is almost reached.
- Common methods to mitigate overfitting to training data, like regularization and dropout, fail–exposure is still significantly higher for the canary than other sequences in $\mathcal{R}$.
- Advanced methods can be used to extract memorized instances from much larger language models, for example LLMs for coding Huang et al. (2024).

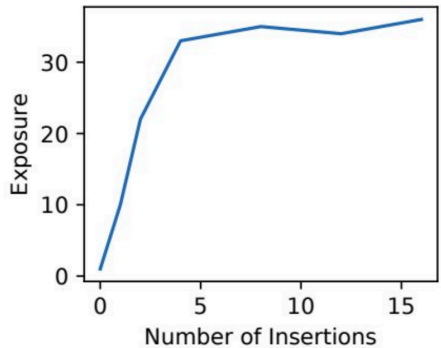


Figure: Exposure vs. number of insertions (Carlini et al., 2019)

A Solution: Differential Privacy

General Framework: Add a small amount of structured noise to the ML model such that an adversary cannot tell whether:

a) A model was trained with instance x.

b) A model was trained without instance x,

for all instances $x \in \mathcal{D}$ for dataset $\mathcal{D}$.

- This is called **differential privacy** (Dwork et al., 2006).

A Solution: Differential Privacy

General Framework: Add a small amount of structured noise to the ML model such that an adversary cannot tell whether:

a) A model was trained with instance x.

b) A model was trained without instance x,

for all instances $x \in \mathcal{D}$ for dataset $\mathcal{D}$.

- This is called **differential privacy** (Dwork et al., 2006).
- This is the only (known) method to fully reduce exposure in the example of Carlini et al. (2019)–the canary's exposure is indistinguishable from a random phrase $r \in \mathcal{R}$.

A Solution: Differential Privacy

General Framework: Add a small amount of structured noise to the ML model such that an adversary cannot tell whether:

a) A model was trained with instance x.

b) A model was trained without instance x,

for all instances $x \in \mathcal{D}$ for dataset $\mathcal{D}$.

- This is called **differential privacy** (Dwork et al., 2006).
- This is the only (known) method to fully reduce exposure in the example of Carlini et al. (2019)–the canary's exposure is indistinguishable from a random phrase $r \in \mathcal{R}$.
- Differential privacy offers strong protection against other attacks as well:
  - Model inversion attacks: recover the training data from model outputs/parameters (Haim et al., 2022).

A Solution: Differential Privacy

General Framework: Add a small amount of structured noise to the ML model such that an adversary cannot tell whether:

a) A model was trained with instance x.

b) A model was trained without instance x,

for all instances $x \in \mathcal{D}$ for dataset $\mathcal{D}$.

- This is called **differential privacy** (Dwork et al., 2006).
- This is the only (known) method to fully reduce exposure in the example of Carlini et al. (2019)–the canary's exposure is indistinguishable from a random phrase $r \in \mathcal{R}$.
- Differential privacy offers strong protection against other attacks as well:
  - Model inversion attacks: recover the training data from model outputs/parameters (Haim et al., 2022).
  - Membership inference attacks: figure out whether or not an instance is part of the training data (Shokri et al., 2017).

A Solution: Differential Privacy

General Framework: Add a small amount of structured noise to the ML model such that an adversary cannot tell whether:

a) A model was trained with instance x.

b) A model was trained without instance x,

for all instances $x \in \mathcal{D}$ for dataset $\mathcal{D}$.

- This is called **differential privacy** (Dwork et al., 2006).
- This is the only (known) method to fully reduce exposure in the example of Carlini et al. (2019)–the canary's exposure is indistinguishable from a random phrase $r \in \mathcal{R}$.
- Differential privacy offers strong protection against other attacks as well:
  - Model inversion attacks: recover the training data from model outputs/parameters (Haim et al., 2022).
  - Membership inference attacks: figure out whether or not an instance is part of the training data (Shokri et al., 2017).
  - Linkage attacks: Use external information to break dataset anonymization.

A Solution: Differential Privacy

General Framework: Add a small amount of structured noise to the ML model such that an adversary cannot tell whether:

a) A model was trained with instance x.

b) A model was trained without instance x,

for all instances $x \in \mathcal{D}$ for dataset $\mathcal{D}$.

- This is called **differential privacy** (Dwork et al., 2006).
- This is the only (known) method to fully reduce exposure in the example of Carlini et al. (2019)–the canary's exposure is indistinguishable from a random phrase $r \in \mathcal{R}$.
- Differential privacy offers strong protection against other attacks as well:
  - Model inversion attacks: recover the training data from model outputs/parameters (Haim et al., 2022).
  - Membership inference attacks: figure out whether or not an instance is part of the training data (Shokri et al., 2017).
  - Linkage attacks: Use external information to break dataset anonymization.
- Justified by the Fundamental Theorem of Information: need to add noise to the data instances or statistical procedure, otherwise an adversary can **always recover sensitive attributes** (Dinur and Nissim, 2003).

Outline

Notation

We will now introduce the standard notation of differential privacy:

- $\mathcal{X} \subset \mathbb{R}^d$ be a sample space and let $\mathcal{Y} \subset \mathbb{R}^o$ be a label space. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; $\mathcal{Z}^n = \underbrace{\mathcal{Z} \times ... \times \mathcal{Z}}_{\text{n times}}$.

Notation

We will now introduce the standard notation of differential privacy:

- $\mathcal{X} \subset \mathbb{R}^d$ be a sample space and let $\mathcal{Y} \subset \mathbb{R}^o$ be a label space. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; $\mathcal{Z}^n = \underbrace{\mathcal{Z} \times ... \times \mathcal{Z}}_{\text{n times}}$.
- (Randomized) learning algorithm $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$, where $\mathcal{W} \subset \mathbb{R}^z$ is a parameter space.

Notation

We will now introduce the standard notation of differential privacy:

- $\mathcal{X} \subset \mathbb{R}^d$ be a sample space and let $\mathcal{Y} \subset \mathbb{R}^o$ be a label space. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; $\mathcal{Z}^n = \underbrace{\mathcal{Z} \times ... \times \mathcal{Z}}_{\text{n times}}$.
- (Randomized) learning algorithm $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$, where $\mathcal{W} \subset \mathbb{R}^z$ is a parameter space.
- Set of hypotheses (e.g. neural networks) parameterized with respect to this parameter space $\mathcal{H}_{\mathcal{W}}$.

Notation

We will now introduce the standard notation of differential privacy:

- $\mathcal{X} \subset \mathbb{R}^d$ be a sample space and let $\mathcal{Y} \subset \mathbb{R}^o$ be a label space. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$; $\mathcal{Z}^n = \underbrace{\mathcal{Z} \times ... \times \mathcal{Z}}_{\text{n times}}$.
- (Randomized) learning algorithm $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$, where $\mathcal{W} \subset \mathbb{R}^z$ is a parameter space.
- Set of hypotheses (e.g. neural networks) parameterized with respect to this parameter space $\mathcal{H}_{\mathcal{W}}$.
- Let $f_{\boldsymbol{\theta}} \in \mathcal{H}_{\mathcal{W}}$ be the hypothesis parameterized by $\boldsymbol{\theta} \in \mathcal{W}$, defined as $f_{\boldsymbol{\theta}} : \mathcal{X} \to \Delta_{|\mathcal{Y}|}$, where $\Delta_{|\mathcal{Y}|}$ is the probability simplex $\{p_1, ..., p_\gamma : p_i \geq 0, \sum_{i=1}^{\gamma} p_i = 1\}$. (Outputs a probability distribution across all classes).

Exact Differential Privacy

ε-Differential Privacy (Dwork et al., 2006)

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be a (randomized) learning algorithm. Fix $\varepsilon \in (0, 1)$. $\mathcal{A}$ is ε-differentially private if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^n$ s.t. $||\mathcal{D} - \mathcal{D}'||_1 \leq 1$ and $\mathcal{T} \subset \mathcal{W}$:

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{T}] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{T}]\,. \tag{3}$$

Exact Differential Privacy

ε-Differential Privacy (Dwork et al., 2006)

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be a (randomized) learning algorithm. Fix $\varepsilon \in (0, 1)$. $\mathcal{A}$ is ε-differentially private if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^n$ s.t. $||\mathcal{D} - \mathcal{D}'||_1 \leq 1$ and $\mathcal{T} \subset \mathcal{W}$:

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{T}] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{T}]\,. \tag{3}$$

- As $\varepsilon \to 0$, probabilities are roughly the same $\implies$ models are equally likely to occur, so indistinguishable.
- As $\varepsilon \to 1$, probabilities become different. Use of e^{ε} an artifact of concentration inequalities.
- Exact differential privacy: very useful for classical stats. However, not so applicable to ML (so far).

Approximate Differential Privacy

(ε, δ)-Differential Privacy (Dwork et al., 2014)

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be a (randomized) learning algorithm. Fix $\varepsilon, \delta \in (0, 1)$. $\mathcal{A}$ is ε-differentially private if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^n$ s.t. $||\mathcal{D} - \mathcal{D}'||_1 \leq 1$ and $\mathcal{T} \subset \mathcal{W}$:

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{T}] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{T}] + \delta \,. \tag{4}$$

Approximate Differential Privacy

(ε, δ)-Differential Privacy (Dwork et al., 2014)

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be a (randomized) learning algorithm. Fix $\varepsilon, \delta \in (0, 1)$. $\mathcal{A}$ is ε-differentially private if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^n$ s.t. $||\mathcal{D} - \mathcal{D}'||_1 \leq 1$ and $\mathcal{T} \subset \mathcal{W}$:

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{T}] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{T}] + \delta\,. \tag{4}$$

- This is the form of differential privacy we will focus on.
- δ is the "catastrophic failure probability"–probability that $\mathcal{A}(\mathcal{D})$ and $\mathcal{A}(\mathcal{D}')$ are easily distinguishable. Generally chosen very small, $\delta \approx 0$.

ℓ_2 sensitivity

ℓ_2 Sensitivity

Let $f : \mathcal{Z}^n \to \mathbb{R}^z$. The ℓ_2 sensitivity of f is:

$$\Delta_2^{(f)} = \max_{\mathcal{D},\mathcal{D}'} ||f(\mathcal{D}) - f(\mathcal{D}')||_2 \,, \tag{5}$$

where $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^n$ are neighboring datasets s.t. $||\mathcal{D} - \mathcal{D}'||_1 \leq 1$.

Gaussian Mechanism

Gaussian Mechanism

Fix $\varepsilon, \delta \in (0,1)$. Consider a (deterministic) function $f : \mathcal{Z}^n \to \mathcal{R}^z$. Let $\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + H$ where $H \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ and $\sigma \geq \frac{\Delta_2^{(f)}}{\varepsilon} \sqrt{2 \ln(1.25/\delta)}$. Then, the (randomized) mechanism $\mathcal{M}$ is (ε, δ)-differentially private.

Gaussian Mechanism

Gaussian Mechanism

Fix $\varepsilon, \delta \in (0,1)$. Consider a (deterministic) function $f : \mathcal{Z}^n \to \mathcal{R}^z$. Let $\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + H$ where $H \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ and $\sigma \geq \frac{\Delta_2^{(f)}}{\varepsilon} \sqrt{2 \ln(1.25/\delta)}$. Then, the (randomized) mechanism $\mathcal{M}$ is (ε, δ)-differentially private.

This is the primary way one designs a (ε, δ)-differentially private algorithm.

Properties of (ε, δ)-differential privacy

Post-Processing

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be (ε, δ)-differentially private, and let $\mathcal{F} : \mathcal{W} \to \mathcal{B}$ be an arbitrary randomized mapping. Then, $\mathcal{F} \circ \mathcal{A}$ is (ε, δ)-differentially private.

Properties of (ε, δ)-differential privacy

Post-Processing

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be (ε, δ)-differentially private, and let $\mathcal{F} : \mathcal{W} \to \mathcal{B}$ be an arbitrary randomized mapping. Then, $\mathcal{F} \circ \mathcal{A}$ is (ε, δ)-differentially private.

Importantly, this is why DP also allows us to release model predictions (rather than just parameters): consider $\mathcal{F}$ that takes parameters and outputs predictions.

Properties of (ε, δ)-differential privacy

Post-Processing

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be (ε, δ)-differentially private, and let $\mathcal{F} : \mathcal{W} \to \mathcal{B}$ be an arbitrary randomized mapping. Then, $\mathcal{F} \circ \mathcal{A}$ is (ε, δ)-differentially private.

Importantly, this is why DP also allows us to release model predictions (rather than just parameters): consider $\mathcal{F}$ that takes parameters and outputs predictions.

Group Privacy

Let $\mathcal{A} : \mathcal{Z}^n \to \mathcal{W}$ be (ε, δ)-differentially private. Suppose $\mathcal{D}, \mathcal{D}' \subset \mathcal{Z}^n$ are two datasets which differ in at most k positions, i.e. $||\mathcal{D} - \mathcal{D}'||_1 \leq k$. Then, for all $\mathcal{T} \subset \mathcal{W}$, we have:

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{T}] \leq e^{k\varepsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{T}] + ke^{(k-1)\varepsilon}\delta\,. \tag{6}$$

Properties of (ε, δ)-differential privacy

Basic Composition

Let $\boldsymbol{\mathcal{M}} = (\mathcal{M}_1, ..., \mathcal{M}_k)$ be a sequence of randomized mechanisms, where $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private, for $\varepsilon_i, \delta_i > 0$, and the $\mathcal{M}_i$ can be chosen sequentially and adaptively. Then, $\mathcal{M}$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.

Properties of (ε, δ)-differential privacy

Basic Composition

Let $\boldsymbol{\mathcal{M}} = (\mathcal{M}_1, ..., \mathcal{M}_k)$ be a sequence of randomized mechanisms, where $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private, for $\varepsilon_i, \delta_i > 0$, and the $\mathcal{M}_i$ can be chosen sequentially and adaptively. Then, $\mathcal{M}$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.

Advanced Composition

For all $\varepsilon, \delta, \delta' > 0$, let $\boldsymbol{\mathcal{M}} = (\mathcal{M}_1, ..., \mathcal{M}_k)$ be a sequence of randomized mechanisms, where $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private and the $\mathcal{M}_i$ can be chosen sequentially and adaptively. Then $\mathcal{M}$ is $(\bar{\varepsilon}, \bar{\delta})$-differentially private, where $\bar{\varepsilon} = \varepsilon\sqrt{2k\log(1/\delta')} + k\varepsilon\frac{e^{\varepsilon}-1}{e^{\varepsilon}+1}$ and $\bar{\delta} = k\delta + \delta'$.

Simple approach, but under strict assumptions

- Under strong assumptions e.g. strong convexity of loss function and Lipschitz gradients, we obtain a simple bound on $\Delta_2^{\mathcal{A}(\mathcal{D})}$, sensitivity of learning algorithm.

Simple approach, but under strict assumptions

- Under strong assumptions e.g. strong convexity of loss function and Lipschitz gradients, we obtain a simple bound on $\Delta_2^{\mathcal{A}(\mathcal{D})}$, sensitivity of learning algorithm.
- Then, we can perturb model parameters themselves:

$$\tilde{\boldsymbol{\theta}} = \mathcal{A}(\mathcal{D}) + \mathcal{N}(0, \sigma^2 \boldsymbol{I}), \tag{7}$$

s.t.

$$\sigma \geq \frac{\Delta_2^{\mathcal{A}(\mathcal{D})}}{\varepsilon} \sqrt{2 \log(1.25/\delta)}, \tag{8}$$

yielding a (ε, δ)-differentially private model.

DP-SGD (Abadi et al., 2016)

Key idea

Add Gaussian noise to bounded gradients and use advanced composition-like theorems to yield DP guarantee.

DP-SGD (Abadi et al., 2016)

Key idea

Add Gaussian noise to bounded gradients and use advanced composition-like theorems to yield DP guarantee.

For $t = 1, ..., N$:

1. Sample a minibatch $B_t \subset \mathcal{D}$ at random and compute per-sample gradient $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in B_t$.

DP-SGD (Abadi et al., 2016)

Key idea

Add Gaussian noise to bounded gradients and use advanced composition-like theorems to yield DP guarantee.

For $t = 1, ..., N$:

1. Sample a minibatch $B_t \subset \mathcal{D}$ at random and compute per-sample gradient $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in B_t$.
2. Clip each gradient to have bounded norm:

$$\bar{\boldsymbol{g}}_i = \boldsymbol{g}_i / \max(1, \frac{||\boldsymbol{g}_i||_2}{C}) \,. \tag{9}$$

DP-SGD (Abadi et al., 2016)

Key idea

Add Gaussian noise to bounded gradients and use advanced composition-like theorems to yield DP guarantee.

For $t = 1, ..., N$:

1. Sample a minibatch $B_t \subset \mathcal{D}$ at random and compute per-sample gradient $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in B_t$.
2. Clip each gradient to have bounded norm:

$$\bar{\boldsymbol{g}}_i = \boldsymbol{g}_i / \max(1, \frac{||\boldsymbol{g}_i||_2}{C}) \,. \tag{9}$$

3. Add noise to the batch average gradient:

$$\tilde{\boldsymbol{g}}_t = \frac{1}{|B_t|} (\sum_{i \in B_t} \bar{\boldsymbol{g}}_i + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})) \tag{10}$$

DP-SGD (Abadi et al., 2016)

Key idea

Add Gaussian noise to bounded gradients and use advanced composition-like theorems to yield DP guarantee.

For $t = 1, ..., N$:

1. Sample a minibatch $B_t \subset \mathcal{D}$ at random and compute per-sample gradient $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in B_t$.
2. Clip each gradient to have bounded norm:

$$\bar{\boldsymbol{g}}_i = \boldsymbol{g}_i / \max(1, \frac{||\boldsymbol{g}_i||_2}{C}) \,. \tag{9}$$

3. Add noise to the batch average gradient:

$$\tilde{\boldsymbol{g}}_t = \frac{1}{|B_t|} (\sum_{i \in B_t} \bar{\boldsymbol{g}}_i + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})) \tag{10}$$

4. Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \tilde{\boldsymbol{g}}_t$.

DP-SGD (Abadi et al., 2016)

Key idea

Add Gaussian noise to bounded gradients and use advanced composition-like theorems to yield DP guarantee.

For $t = 1, ..., N$:

1. Sample a minibatch $B_t \subset \mathcal{D}$ at random and compute per-sample gradient $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in B_t$.
2. Clip each gradient to have bounded norm:

$$\bar{\boldsymbol{g}}_i = \boldsymbol{g}_i / \max(1, \frac{||\boldsymbol{g}_i||_2}{C}) \,. \tag{9}$$

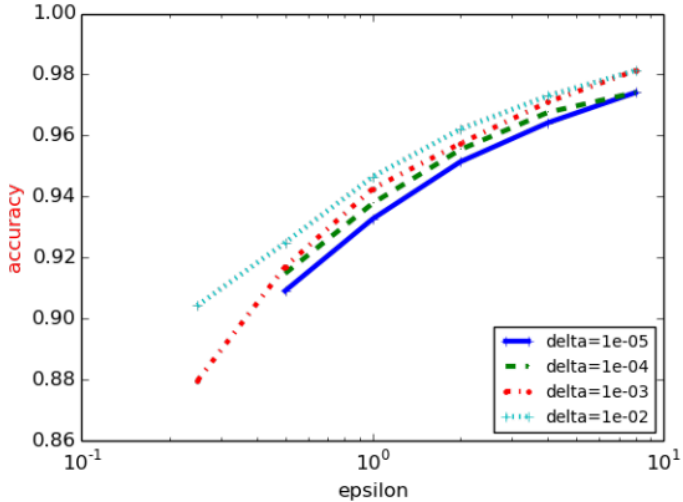
3. Add noise to the batch average gradient:

$$\tilde{\boldsymbol{g}}_t = \frac{1}{|B_t|} (\sum_{i \in B_t} \bar{\boldsymbol{g}}_i + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})) \tag{10}$$

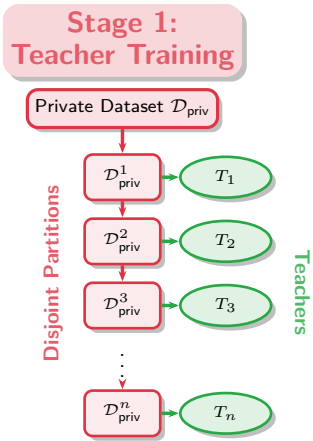
4. Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \tilde{\boldsymbol{g}}_t$.

Note: Choose σ, N, and sampling rate $q = \frac{|B_t|}{|\mathcal{D}|}$, then ε, δ. A "privacy budget"; when using budget, incur "privacy cost".

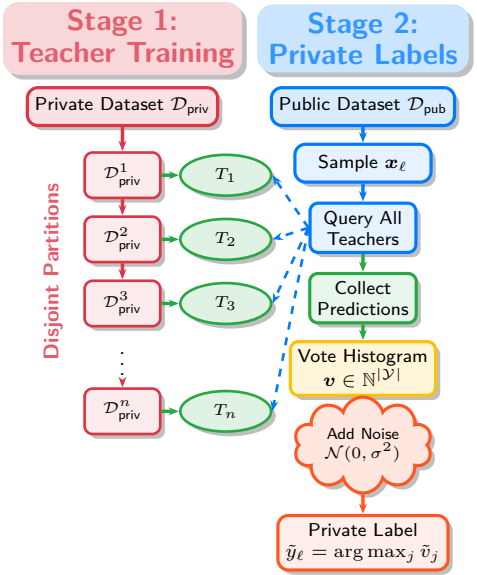
Privacy Utility Tradeoff for DP-SGD



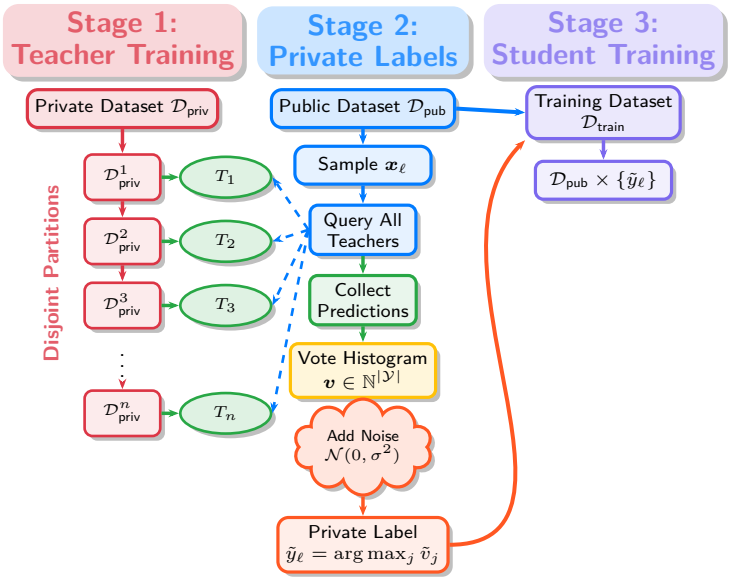
Private Aggregation of Teacher Ensembles (PATE) (Papernot et al., 2017)



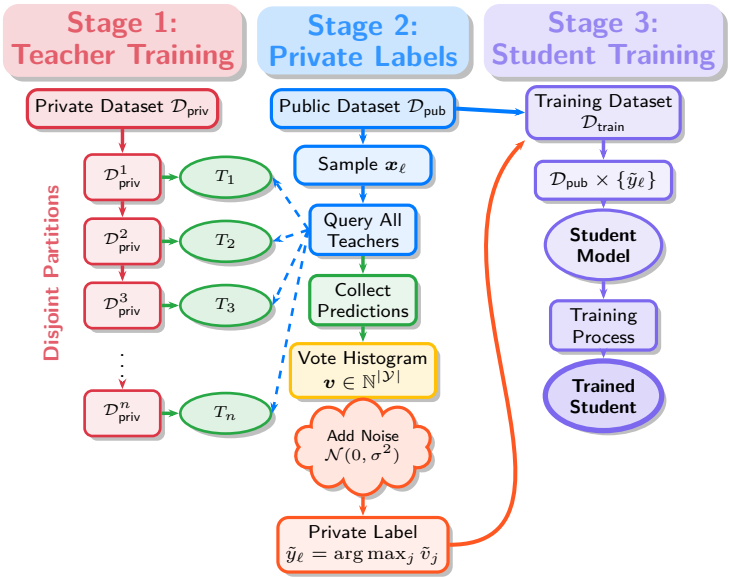
Private Aggregation of Teacher Ensembles (PATE) (Papernot et al., 2017)



Private Aggregation of Teacher Ensembles (PATE) (Papernot et al., 2017)



Private Aggregation of Teacher Ensembles (PATE) (Papernot et al., 2017)



PATE Framework Summary

- **Stage 1**: Train multiple teacher models on disjoint partitions of private data.
- **Stage 2**: Generate private labels for public data using noisy aggregation.
- **Stage 3**: Train student model on public data with private labels.

PATE Framework Summary

- **Stage 1**: Train multiple teacher models on disjoint partitions of private data.
- **Stage 2**: Generate private labels for public data using noisy aggregation.
- **Stage 3**: Train student model on public data with private labels.

Key Innovation

The disjoint partitioning ensures that changing one private example affects at most one teacher's vote, enabling strong differential privacy guarantees with controlled noise addition.

PATE (Papernot et al., 2017)

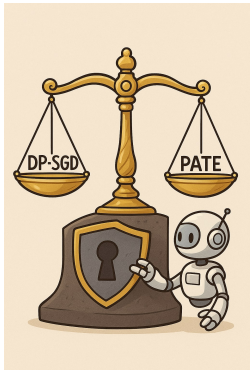
PATE offers multiple layers of privacy:

1. Each teacher only accesses a disjoint subset of the private data.
2. The final label is perturbed with Gaussian noise, so an adversary cannot tell if a single training example changed a teacher's vote.
3. The student does not access any private instances whatsoever.

DP-SGD or PATE?

DP-SGD for supervised, PATE for semi-supervised:

- PATE has stronger privacy guarantees than DP-SGD.
- PATE results in better generalization accuracy than DP-SGD.
- But, DP-SGD is significantly less computationally expensive than PATE.



The Differential Privacy Paradox: A Privacy-Utility Tradeoff

Benefits: The Power of Protection

- **Formal Privacy Guarantees**: Provable protection against multiple attacks.
- **Composable and Flexible**: Gracefully composes, with post-processing, group privacy, etc.
- **Trusted, Real-World Adoption**: Often deployed by U.S. Census Bureau, Apple, and Google.

Limitations: The Price of Utility

- **A Significant Utility Cost**: Even with small ε, δ, the added noise can significantly impact the accuracy and usefulness of models.
- **Highly Parameter-Sensitive**: Requires a careful choice of ϵ and δ.
- **Interpretability Challenges**: Difficult to interpret results or understand model behavior.

Outline

Is that all for privacy?

Question

Can someone request their private data to be deleted?

Is that all for privacy?

Question

Can someone request their private data to be deleted?

- In 2020, European Union (EU) introduced the General Data Protection Regulation (GDPR) to ensure data privacy among EU citizens.

Is that all for privacy?

Question

Can someone request their private data to be deleted?

- In 2020, European Union (EU) introduced the General Data Protection Regulation (GDPR) to ensure data privacy among EU citizens.
- One right: "the right to be forgotten".

Is that all for privacy?

Question

Can someone request their private data to be deleted?

- In 2020, European Union (EU) introduced the General Data Protection Regulation (GDPR) to ensure data privacy among EU citizens.
- One right: "the right to be forgotten".
- Owner of user data–"data controller"– must get rid of **all** information about a user upon request.

Is that all for privacy?

Question

Can someone request their private data to be deleted?

- In 2020, European Union (EU) introduced the General Data Protection Regulation (GDPR) to ensure data privacy among EU citizens.
- One right: "the right to be forgotten".
- Owner of user data–"data controller"– must get rid of **all** information about a user upon request.
- For ML: Must delete instance from training data.

Is that all for privacy?

Question

Can someone request their private data to be deleted?

- In 2020, European Union (EU) introduced the General Data Protection Regulation (GDPR) to ensure data privacy among EU citizens.
- One right: "the right to be forgotten".
- Owner of user data–"data controller"– must get rid of **all** information about a user upon request.
- For ML: Must delete instance from training data.

Question

How do we delete a training data instance from a pretrained model?

Is that all for privacy?

Question

Can someone request their private data to be deleted?

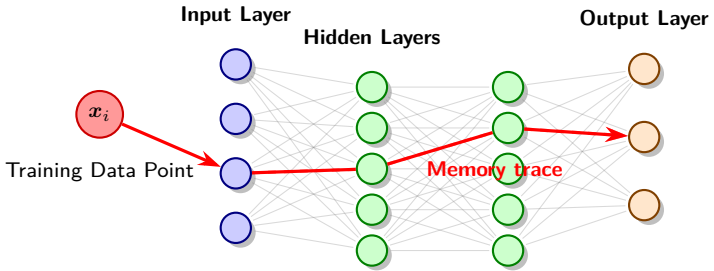
- In 2020, European Union (EU) introduced the General Data Protection Regulation (GDPR) to ensure data privacy among EU citizens.
- One right: "the right to be forgotten".
- Owner of user data–"data controller"– must get rid of **all** information about a user upon request.
- For ML: Must delete instance from training data.

Question

How do we delete a training data instance from a pretrained model?

We can retrain from scratch, but we want to avoid this "exact forgetting" – too computationally expensive.

Is it easy to forget data on pretrained networks?



The Core Problem

Deep networks memorize specific training patterns or instances and their **highly non-convex nature** makes it difficult to trace the effect of each pattern on the model's weights.

Why would we want to forget again?

Legal Requirements

- **GDPR's "Right to be Forgotten"**
- Users can request data deletion
- Models must comply efficiently

Security Concerns

- Remove poisoned samples
- Defend against data attacks
- Clean compromised training sets

Data Quality Issues

- Remove mislabeled examples
- Delete noisy samples
- Eliminate outliers

Data consistency

- Remove harmful information
- Delete outdated information
- Update model behavior

Key Insight: Different applications require different definitions of "forgetting"!

Machine Unlearning setup

Setting

- **Training dataset**: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$.
- **Original model**: $f(\cdot; \boldsymbol{\theta}^o)$ trained on $\mathcal{D}$.
- **Forget set**: $\mathcal{D}_f \subset \mathcal{D}$ (data to remove).
- **Retain set**: $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ (data to keep).

Machine Unlearning setup

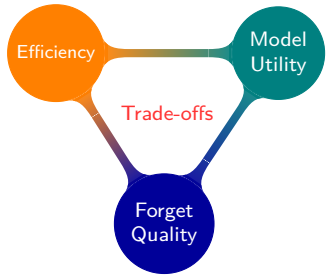
Setting

- **Training dataset**: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$.
- **Original model**: $f(\cdot; \boldsymbol{\theta}^o)$ trained on $\mathcal{D}$.
- **Forget set**: $\mathcal{D}_f \subset \mathcal{D}$ (data to remove).
- **Retain set**: $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ (data to keep).

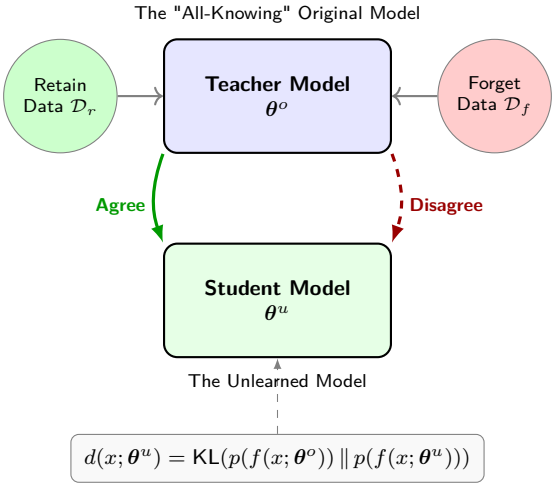
Goal

Find new weights $\boldsymbol{\theta}^u$ such that $f(\cdot; \boldsymbol{\theta}^u)$ has:

1. **Forgotten** $\mathcal{D}_f$ (application-dependent definition).
2. **Retained** performance on $\mathcal{D}_r$ and test data.
3. **Efficient** computation (faster than retraining).



SCalable Remembering and Unlearning unBound (SCRUB) (Kurmanji et al., 2023)



Core Idea

Student $\boldsymbol{\theta}^u$ **selectively** obeys teacher $\boldsymbol{\theta}^o$:

- **Agree** on retain data $\mathcal{D}_r$
- **Disagree** on forget data $\mathcal{D}_f$

Key Innovation

Use KL-divergence between output distributions:

$$d(x;\boldsymbol{\theta}^u) = \mathsf{KL}(p(f(x;\boldsymbol{\theta}^o))\|p(f(x;\boldsymbol{\theta}^u)))$$

SCRUB's Training Objective SCRUB

Contrastive Objective

$$\min_{\boldsymbol{\theta}^u} \underbrace{\frac{\alpha}{N_r} \sum_{x_r \in \mathcal{D}_r} d(x_r; \boldsymbol{\theta}^u)}_{\text{Stay close on retain set}} + \underbrace{\frac{\gamma}{N_r} \sum_{(x_r, y_r) \in \mathcal{D}_r} \ell(f(x_r; \boldsymbol{\theta}^u), y_r)}_{\text{Task loss on retain set}} - \underbrace{\frac{1}{N_f} \sum_{x_f \in \mathcal{D}_f} d(x_f; \boldsymbol{\theta}^u)}_{\text{Move away on forget set}}$$

Training Strategy

Due to conflicting objectives, SCRUB alternates between:

1. **Max-step:** Update on forget set (increase disagreement).
2. **Min-step:** Update on retain set (maintain agreement).

Certified Unlearning

Setting

- **Training dataset**: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$.
- **Forget set**: $\mathcal{D}_f \subset \mathcal{D}$ (data to remove).
- **Retain set**: $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ (data to keep).
- Privacy budget $\varepsilon \in (0, 1)$ and $\delta > 0$.

(ε, δ)-certified unlearning (Sekhari et al., 2021)

$\mathcal{U} : \mathcal{Z}^n \times \mathcal{Z}^n \times \mathcal{W} \to \mathcal{W}$ is an (ε, δ)-certified unlearning algorithm if $\forall \mathcal{T} \subset \mathcal{W}$, we have:

$$\Pr(\mathcal{U}(\mathcal{D}, \mathcal{D}_f, \mathcal{A}(\mathcal{D})) \in \mathcal{T}) \leq e^{\varepsilon} \Pr(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T}) + \delta, \tag{11}$$

$$\Pr(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T}) \leq e^{\varepsilon} \Pr(\mathcal{U}(\mathcal{D}, \mathcal{D}_f, \mathcal{A}(\mathcal{D})) \in \mathcal{T}) + \delta. \tag{12}$$

Certified Unlearning

Setting

- **Training dataset**: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$.
- **Forget set**: $\mathcal{D}_f \subset \mathcal{D}$ (data to remove).
- **Retain set**: $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ (data to keep).
- Privacy budget $\varepsilon \in (0, 1)$ and $\delta > 0$.

(ε, δ)-certified unlearning (Sekhari et al., 2021)

$\mathcal{U} : \mathcal{Z}^n \times \mathcal{Z}^n \times \mathcal{W} \to \mathcal{W}$ is an (ε, δ)-certified unlearning algorithm if $\forall \mathcal{T} \subset \mathcal{W}$, we have:

$$\Pr(\mathcal{U}(\mathcal{D}, \mathcal{D}_f, \mathcal{A}(\mathcal{D})) \in \mathcal{T}) \leq e^\varepsilon \Pr(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T}) + \delta, \tag{11}$$

$$\Pr(\mathcal{A}(\mathcal{D}_r) \in \mathcal{T}) \leq e^\varepsilon \Pr(\mathcal{U}(\mathcal{D}, \mathcal{D}_f, \mathcal{A}(\mathcal{D})) \in \mathcal{T}) + \delta. \tag{12}$$

DP implies certified unlearning, but not the other way around.

Limitations of Unlearning (Zhao et al., 2024)

- Unlearning is very difficult when forget sets and retain sets are similar.

Limitations of Unlearning (Zhao et al., 2024)

- Unlearning is very difficult when forget sets and retain sets are similar.
- For most instances in training distribution, $f(\boldsymbol{x}) = f_u(\boldsymbol{x})$, which can reveal unwanted information.

Limitations of Unlearning (Zhao et al., 2024)

- Unlearning is very difficult when forget sets and retain sets are similar.
- For most instances in training distribution, $f(\boldsymbol{x}) = f_u(\boldsymbol{x})$, which can reveal unwanted information.
- Consider the scenario:
  - $\boldsymbol{x}_p$ is a corrupted criminal record that is part of the training data for criminal classifier $f_\theta : \mathbb{R}^d \to \{\text{high risk}, \text{low risk}\}$

Limitations of Unlearning (Zhao et al., 2024)

- Unlearning is very difficult when forget sets and retain sets are similar.
- For most instances in training distribution, $f(\boldsymbol{x}) = f_u(\boldsymbol{x})$, which can reveal unwanted information.
- Consider the scenario:
  - $\boldsymbol{x}_p$ is a corrupted criminal record that is part of the training data for criminal classifier $f_\theta : \mathbb{R}^d \to \{\text{high risk}, \text{low risk}\}$
  - Police are obtaining $f(\boldsymbol{x}_p) = \text{high risk}$

Limitations of Unlearning (Zhao et al., 2024)

- Unlearning is very difficult when forget sets and retain sets are similar.
- For most instances in training distribution, $f(\boldsymbol{x}) = f_u(\boldsymbol{x})$, which can reveal unwanted information.
- Consider the scenario:
  - $\boldsymbol{x}_p$ is a corrupted criminal record that is part of the training data for criminal classifier $f_\theta : \mathbb{R}^d \to \{\text{high risk}, \text{low risk}\}$
  - Police are obtaining $f(\boldsymbol{x}_p) =$ high risk
  - Software provider (ML model provider) should suppress outputs; unlearning **does not suffice**, since $f_u(\boldsymbol{x}_p) = f(\boldsymbol{x}_p)$.
  - We call this **test-time privacy** (TTP).

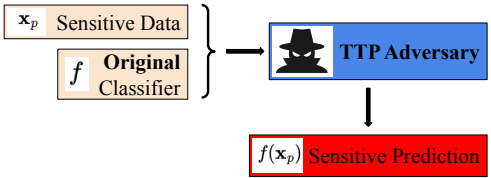


Figure: TTP Without Unlearning

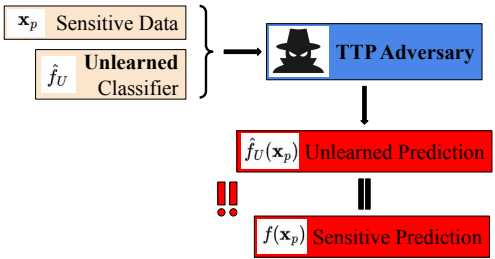


Figure: Test-time privacy with Unlearning

Limitations of Unlearning

Solution: Make model outputs uniform.

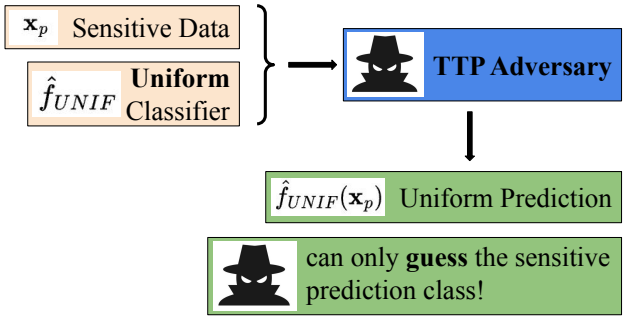


Figure: Test-time privacy after Uniformity

Limitations of Unlearning

Solution: Make model outputs uniform.

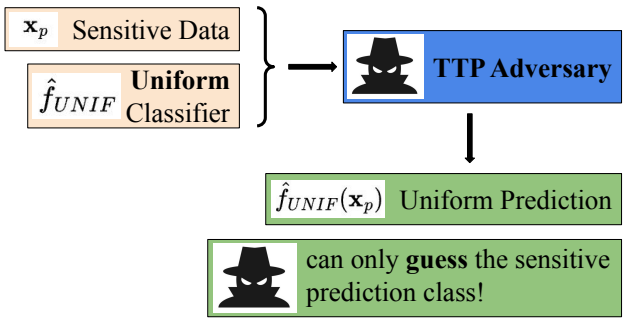


Figure: Test-time privacy after Uniformity

We can also adapt (ε, δ) guarantees to this setting (Ashiq et al., 2025).

Collaborators on topics presented today (alphabetical order):

Muhammad Ashiq, Vokan Cevher, Justin Deschenaux, Igor Krawczukz, Peter Triantafillou, Andrea Tseng.

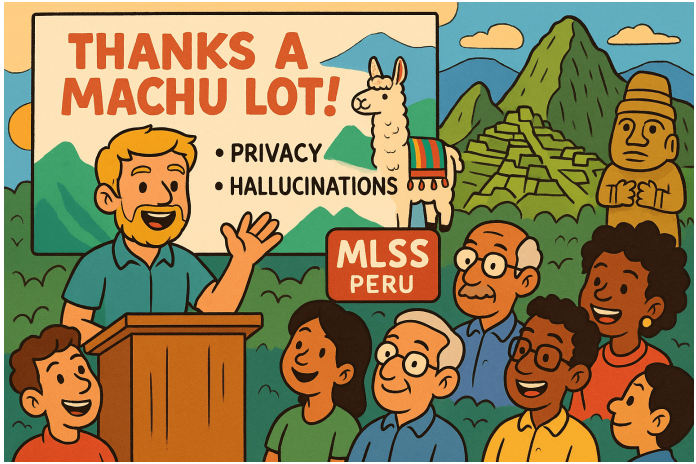
Special thanks to the UW Madison team (alphabetical order):

Muhammad Ashiq, Andrea Tseng, Yiheng Zhang.

Special thanks to the MLSS organizing team:

Pablo Martinez Olmos, Mariano Gabitto, Fernando Pérez Cruz, Efraín Mayhua López, Irvin Franco Benito Dongo Escalante, Antonio Artés-Rodríguez, Jean-Louis GELOT.

Thank you for your attention!



Source: GPT-5.

References I

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Sumukh K Aithal, Pratyush Maini, Zachary Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, volume 37, pages 134614–134644, 2024.

Muhammad Ashiq, Peter Triantafillou, Andrea Tseng, and Grigoris Chrysos. Test time privacy requires going beyond certified unlearning, 2025.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey, 2025. URL `https://arxiv.org/abs/2404.18930`.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2015.

References II

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

Xiuying Chen, Mingzhe Li, Xin Gao, and Xiangliang Zhang. Towards improving faithfulness in abstractive summarization. In *Advances in Neural Information Processing Systems*, 2022.

Zhiyuan Chen, Yuecong Min, Jie Zhang, Bei Yan, Jiahao Wang, Xiaozhen Wang, and Shiguang Shan. A survey of multimodal hallucination evaluation and detection, 2025. URL `https://arxiv.org/abs/2507.19024`.

Justin Deschenaux, Igor Krawczuk, Grigorios Chrysos, and Volkan Cevher. Going beyond compositional generalization, ddpms can produce zero-shot interpolation. In *International Conference on Machine Learning (**ICML**)*, 2024.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

References III

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22911–22924. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/906927370cbeb537781100623cca6fa6-Paper-Conference.pdf`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, volume 33, pages 6840–6851, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 2025.

Yizhan Huang, Yichen Li, Weibin Wu, Jianping Zhang, and Michael R Lyu. Your code secret belongs to me: Neural code completion tools can memorize hard-coded credentials. *Proceedings of the ACM on Software Engineering*, 1(FSE):2515–2537, 2024.

References IV

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*, 2023.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, volume 36, pages 1957–1987, 2023.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

References V

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations (**ICLR**)*, 2017.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2022.

Vipula Rawte, Aman Chadha, Amit Sheth, and Amitava Das. Lrec-coling 2024 tutorial: Hallucination in large language models. *LREC-COLING 2024*, 2024.

References VI

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, volume 34, pages 18075–18086, 2021.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Yongtao Wu, Zhenyu Zhu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Extrapolation and spectral bias of neural nets with hadamard product: a polynomial net study. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, volume 35, pages 26980–26993, 2022.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Instructing large language models to say 'I don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.

Jiajun Zhang, Yang Zhao, Haoran Li, and Chengqing Zong. Attention with sparsity regularization for neural machine translation and summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

References VII

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. Enhancing uncertainty-based hallucination detection with stronger focus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *Advances in Neural Information Processing Systems (**NeurIPS**)*, 2024.