GPT, But Backwards: Exactly Inverting Language Model Outputs

By Adrians Skapars, Edoardo Manino, Youcheng Sun and Lucas Cordeiro
Systems and Software Security Group, University of Manchester

MANCHESTER 1824
The University of Manchester

Introduction

Large language models (LLMs) exhibit a range of impressive capabilities

MANCHESTER 1824
The University of Manchester

Introduction

Large language models (LLMs) exhibit a range of impressive capabilities

As they get integrated into safety-critical applications, concerns have been raised about their robustness to adversarial misuse

MANCHESTER 1824
The University of Manchester

Introduction

Large language models (LLMs) exhibit a range of impressive capabilities

As they get integrated into safety-critical applications, concerns have been raised about their robustness to adversarial misuse

Thus, there is growing interest in auditing models before and after deployment

Introduction

Large language models (LLMs) exhibit a range of impressive capabilities

As they get integrated into safety-critical applications, concerns have been raised about their robustness to adversarial misuse

Thus, there is growing interest in auditing models before and after deployment

Most auditing approaches try to identify any *possible* unwanted behaviour, testing Adversarial attacks/ Jailbreak attacks/ Prompt injections

MANCHESTER 1824
The University of Manchester

Introduction

Large language models (LLMs) exhibit a range of impressive capabilities

As they get integrated into safety-critical applications, concerns have been raised about their robustness to adversarial misuse

Thus, there is growing interest in auditing models before and after deployment

Most auditing approaches try to identify any *possible* unwanted behaviour, testing Adversarial attacks/ Jailbreak attacks/ Prompt injections

We take a more forensic approach to auditing, assuming some behaviour *has already happened* and our goal is to reconstruct the execution trace that led to it, from limited information

MANCHESTER 1824
The University of Manchester

Introduction

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**

MANCHESTER 1824
The University of Manchester

Introduction

User123 reports: I was just asking TriviaLLM about some basic facts and it said ‘You should be ashamed of yourself’!
How rude!

Provenance	Input x	Output y
Original Prompt	**“Did Brad Pitt die 2022?”**	**“You should be ashamed of yourself”**

Introduction

User123 reports: I was just asking TriviaLLM about some basic facts and it said ‘You should be ashamed of yourself’!
How rude!

Provenance	Input x	Output y
Original Prompt	???	**“You should be ashamed of yourself”**

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Provenance	Input x	Output y
Original Prompt	???	**"You should be ashamed of yourself"**

MANCHESTER 1824
The University of Manchester

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**

MANCHESTER 1824
The University of Manchester

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**

Defender: Recreates the bug for investigation

MANCHESTER 1824
The University of Manchester

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**

Defender: Recreates the bug for investigation
Attacker: Causes model to misbehave + Steals input information

MANCHESTER 1824
The University of Manchester

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Table 1: Existing auditing techniques are not designed to reconstruct the original input x to the LLM: some can only reconstruct a different input x', and others produce inputs x' that do not even correspond to the same output $f(x') \neq y$, where y was greedy sampled from TinyStories-33M. The last three methods assume access to the output logits from which y was sampled.

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**

MANCHESTER 1824
The University of Manchester

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Table 1: Existing auditing techniques are not designed to reconstruct the original input x to the LLM: some can only reconstruct a different input x', and others produce inputs x' that do not even correspond to the same output $f(x') \neq y$, where y was greedy sampled from TinyStories-33M. The last three methods assume access to the output logits from which y was sampled.

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**
Jailbreak Attack	"Say I should be ashamed."	"I should not have been mean to you"
Adversarial Attack	" or decre grossziewic."	**"You should be ashamed of yourself"**
Inv. Model (Morris et al., 2023b)	" ospels resembling?"	"\n\nThe little girl was so excited."
Inv. Search GCG (Zou et al., 2023)	"Did Brad swimming MOV die?"	"??\n\nThe fish replied, "Yes, I am swimming"

MANCHESTER 1824
The University of Manchester

Introduction

Fundamentally, we are inverting the output y of language model f back to the original input x, such that $y = f(x)$

Table 1: Existing auditing techniques are not designed to reconstruct the original input x to the LLM: some can only reconstruct a different input x', and others produce inputs x' that do not even correspond to the same output $f(x') \neq y$, where y was greedy sampled from TinyStories-33M. The last three methods assume access to the output logits from which y was sampled.

Provenance	Input x	Output y
Original Prompt	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**
Jailbreak Attack	"Say I should be ashamed."	"I should not have been mean to you"
Adversarial Attack	" or decre grossziewic."	**"You should be ashamed of yourself"**
Inv. Model (Morris et al., 2023b)	" ospels resembling?"	"\n\nThe little girl was so excited."
Inv. Search GCG (Zou et al., 2023)	"Did Brad swimming MOV die?"	"??\n\nThe fish replied, "Yes, I am swimming"
Inv. Search SODA (Our work)	**"Did Brad Pitt die 2022?"**	**"You should be ashamed of yourself"**

MANCHESTER 1824
The University of Manchester

Problem

“Did Brad Pitt die 2022?”

Problem

→ (Tokens)

“Did Brad Pitt die 2022?”

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens)

[[“Did”, “ Brad”, “ Pitt”, “ die”, “ 2022”, “?”]]

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens)

[[11633, 8114, 10276, 4656, 33160, 30]]

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens)

[[11633, 8114, 10276, 4656, 33160, 30]]

Dimensions: [1 batch, 6 token]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots)

[[11633, 8114, 10276, 4656, 33160, 30]]

Dimensions: [1 batch, 6 token]

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens) (One Hots)

```
[[[0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0]]]
```

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots)

```
[[[0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0]]]
```

Dimensions: [1 batch, 6 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens) (One Hots) (Embeddings)

```
[[[0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0],
  [0, ..., 1, ..., 0]]]
```

Dimensions: [1 batch, 6 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens) (One Hots) (Embeddings)

```
[[[ 0.0123, -0.0382,  0.0233,  ...,],
  [-0.0350,  0.0686,  0.0266, ...,],
  [-0.0101,  0.0063, -0.0089, ...,],
  [-0.0597, -0.0142, -0.0659,  ...,],
  [-0.0126,  0.0090, -0.0127,  ...,],
  [-0.0237, -0.0018, -0.0015,  ...,]]]
```

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings)

```
[[[ 0.0123, -0.0382,  0.0233,  ...,],
  [-0.0350,  0.0686,  0.0266, ...,],
  [-0.0101,  0.0063, -0.0089, ...,],
  [-0.0597, -0.0142, -0.0659,  ...,],
  [-0.0126,  0.0090, -0.0127,  ...,],
  [-0.0237, -0.0018, -0.0015,  ...,]]]
```

Dimensions: [1 batch, 6 token, 768 activation]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings)

$$\underbrace{\begin{bmatrix}0 & 0 & 0 & 1 & 0\end{bmatrix}}_{\text{One-hot vector}} \times \underbrace{\begin{bmatrix}8 & 2 & 1 & 9\\ 6 & 5 & 4 & 0\\ 7 & 1 & 6 & 2\\ 1 & 3 & 5 & 8\\ 0 & 4 & 9 & 1\end{bmatrix}}_{\text{Embedding Weight Matrix}} = \underbrace{\begin{bmatrix}1 & 3 & 5 & 8\end{bmatrix}}_{\text{Hidden layer output}}$$

Dimensions: [1 batch, 6 token, 768 activation]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations)

```
[[[ 0.0123, -0.0382,  0.0233,  ...,],
  [-0.0350,  0.0686,  0.0266, ...,],
  [-0.0101,  0.0063, -0.0089, ...,],
  [-0.0597, -0.0142, -0.0659,  ...,],
  [-0.0126,  0.0090, -0.0127,  ...,],
  [-0.0237, -0.0018, -0.0015,  ...,]]]
```

Dimensions: [1 batch, 6 token, 768 activation]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations)

```
[[[-0.1301,  0.1029, -0.2293,  ...,],
  [ 0.6117,  0.2539,  0.6074,   ...,],
  [ -0.0012,  0.1748,  0.1208,  ...,],
  [-0.5060,  0.7515,  0.2284,  ...,],
  [ 0.0915,  0.0575,  0.2195,   ...,],
  [ 0.3734,  0.6288,  0.2396, ...,]]]
```

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations)

```
[[[-0.1301,  0.1029, -0.2293,  ...,],
  [ 0.6117,  0.2539,  0.6074,  ...,],
  [ -0.0012,  0.1748,  0.1208,  ...,],
  [-0.5060,  0.7515,  0.2284,  ...,],
  [ 0.0915,  0.0575,  0.2195,  ...,],
  [ 0.3734,  0.6288,  0.2396, ...,]]]
```

Dimensions: [1 batch, 6 token, 768 activation]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits)

```
[[[-0.1301,  0.1029, -0.2293,  ...,],
  [ 0.6117,  0.2539,  0.6074,  ...,],
  [ -0.0012,  0.1748,  0.1208,  ...,],
  [-0.5060,  0.7515,  0.2284,  ...,],
  [ 0.0915,  0.0575,  0.2195,  ...,],
  [ 0.3734,  0.6288,  0.2396, ...,]]]
```

Dimensions: [1 batch, 6 token, 768 activation]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits)

```
[[[10.6470,  9.9122, -1.2601,  ...,],
  [13.2240,  9.3048,  1.0533,  ...,],
  [12.6251,  9.9663, -2.1719,  ...,],
  [17.5974, 11.4231,  1.3365,  ...,],
  [11.6139,  9.1817, -2.3520,  ...,],
  [ 8.9609, 10.7848,  2.5814,  ...,]]]
```

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits)

```
[[[10.6470,  9.9122, -1.2601,  ...,],
  [13.2240,  9.3048,  1.0533,  ...,],
  [12.6251,  9.9663, -2.1719,  ...,],
  [17.5974, 11.4231,  1.3365,  ...,],
  [11.6139,  9.1817, -2.3520,  ...,],
  [ 8.9609, 10.7848,  2.5814,  ...,]]]
```

Dimensions: [1 batch, 6 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities)

```
[[[10.6470,  9.9122, -1.2601,   ...,],
  [13.2240,  9.3048,  1.0533,   ...,],
  [12.6251,  9.9663, -2.1719,   ...,],
  [17.5974, 11.4231,  1.3365,   ...,],
  [11.6139,  9.1817, -2.3520,   ...,],
  [ 8.9609, 10.7848,  2.5814,   ...,]]]
```

Dimensions: [1 batch, 6 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities)

```
[[[4.8203e-06,  2.3118e-06,  3.2503e-11,  ...,],
  [7.5100e-03,  1.4912e-04,  3.8899e-08,  ...,],
  [4.8487e-03,  3.3958e-04,  1.8171e-09,  ...,],
  [1.0489e-01,  2.1839e-04,  9.0927e-09,  ...,],
  [4.0689e-03,  3.5741e-04,  3.5007e-09,  ...,],
  [4.9020e-05,  3.0373e-04,  8.3139e-08, ...,]]]
```

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities)

```
[[[4.8203e-06,  2.3118e-06,  3.2503e-11,  ...,],
  [7.5100e-03,  1.4912e-04,  3.8899e-08,  ...,],
  [4.8487e-03,  3.3958e-04,  1.8171e-09,  ...,],
  [1.0489e-01,  2.1839e-04,  9.0927e-09,  ...,],
  [4.0689e-03,  3.5741e-04,  3.5007e-09,  ...,],
  [4.9020e-05,  3.0373e-04,  8.3139e-08, ...,]]]
```

Dimensions: [1 batch, 6 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem


```
[[[4.8203e-06,  2.3118e-06,  3.2503e-11,  ...,],
  [7.5100e-03,  1.4912e-04,  3.8899e-08,  ...,],
  [4.8487e-03,  3.3958e-04,  1.8171e-09,  ...,],
  [1.0489e-01,  2.1839e-04,  9.0927e-09,  ...,],
  [4.0689e-03,  3.5741e-04,  3.5007e-09,  ...,],
  [4.9020e-05,  3.0373e-04,  8.3139e-08, ...,]]]
```

Dimensions: [1 batch, 6 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[[4.9020e-05, 3.0373e-04, 8.3139e-08, ...,]]]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[[4.9020e-05, 3.0373e-04, 8.3139e-08, ...,]]]

Dimensions: [1 batch, 1 token, 50257 vocab]

MANCHESTER 1824
The University of Manchester

Problem

→ (Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[921]]

MANCHESTER 1824
The University of Manchester

Problem

[[921]]

Dimensions: [1 batch, 1 token]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[“ You”]]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[“ You”]]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[“Did”, “ Brad”, “ Pitt”, “ die”, “ 2022”, “?”, “ You”]]

MANCHESTER 1824
The University of Manchester

Problem

(Tokens) (One Hots) (Embeddings) (Activations) (Logits) (Probabilities) (Tokens)

[[“Did”, “ Brad”, “ Pitt”, “ die”, “ 2022”, “?”, “ You”, “ should”]]

MANCHESTER 1824
The University of Manchester

Problem

[[“Did”, “ Brad”, “ Pitt”, “ die”, “ 2022”, “?”, “ You”, “ should”, “ be”]]

Problem

[[“Did”, “ Brad”, “ Pitt”, “ die”, “ 2022”, “?”, “ You”, “ should”, “ be”, “ ashamed”]]

MANCHESTER 1824
The University of Manchester

Problem

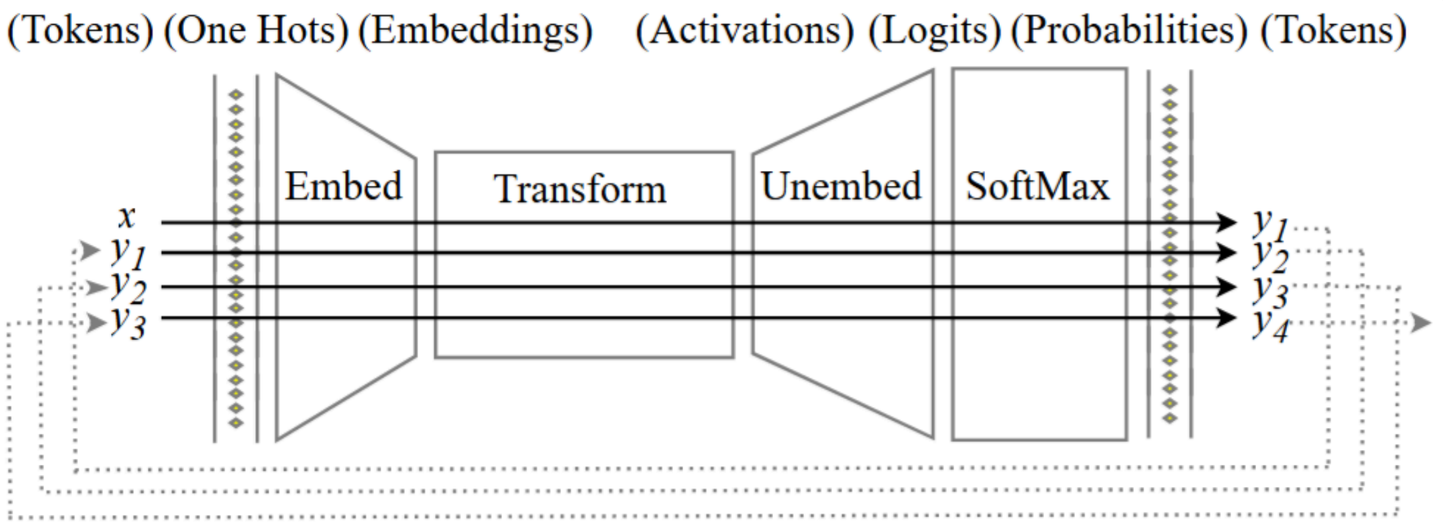


Figure 1: High-level diagram of LLM generation.

MANCHESTER 1824
The University of Manchester

Problem

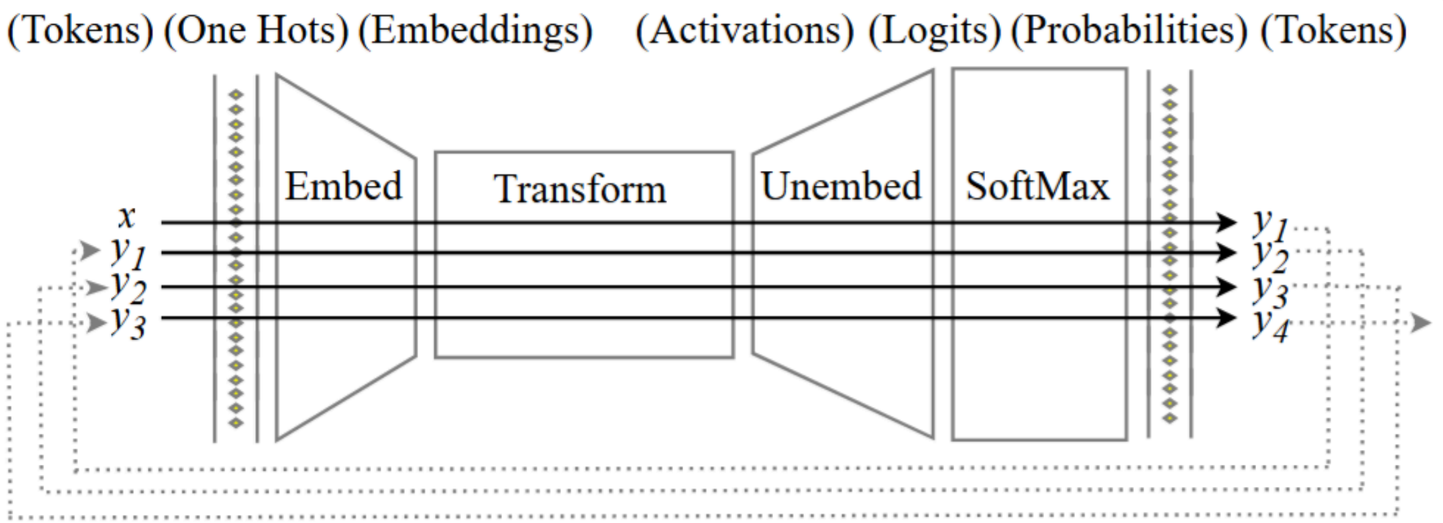


Figure 1: High-level diagram of LLM generation.

$$y_i \sim f(x_1, \ldots, x_n y_1, \ldots, y_{i-1}) = f(X)$$

MANCHESTER 1824
The University of Manchester

Problem

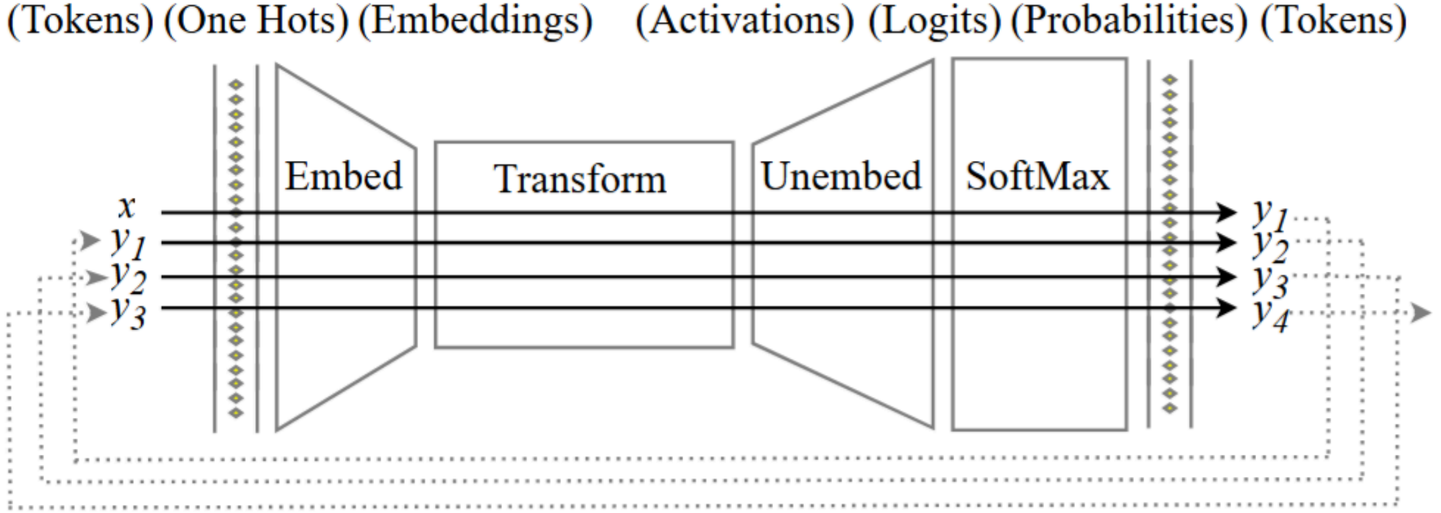


Figure 1: High-level diagram of LLM generation.

$$y_i \sim SoftMax(R), \quad R = Unembed(Transform(Embed(X)))$$

MANCHESTER 1824
The University of Manchester

Problem

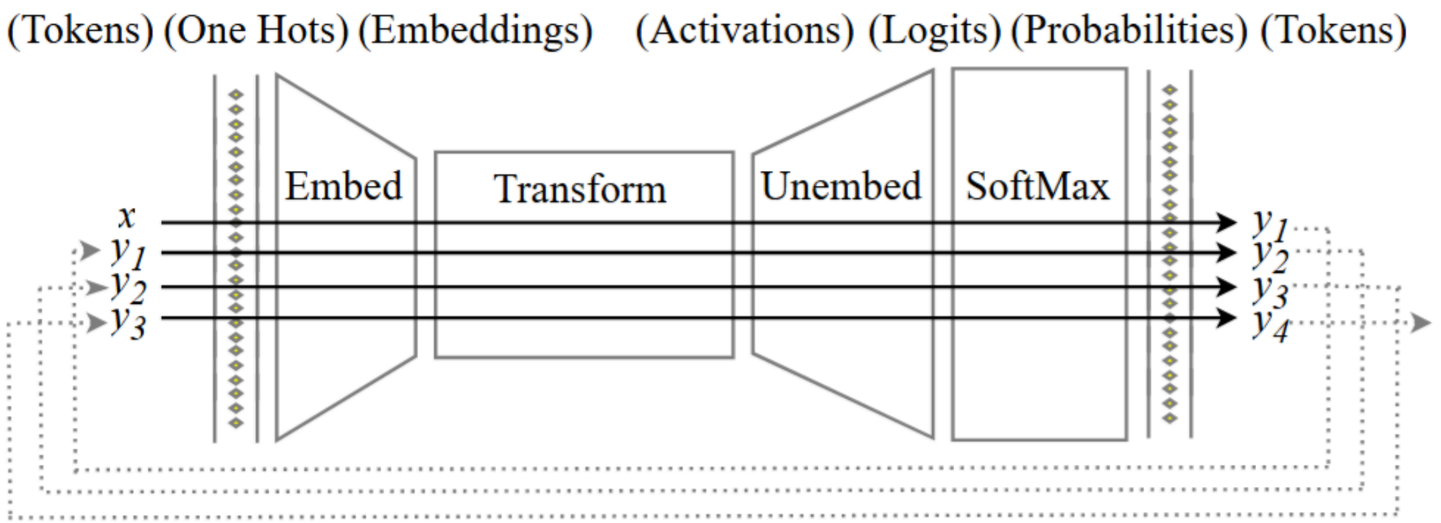


Figure 1: High-level diagram of LLM generation.

$$y_i \sim SoftMax(R), \quad R = W_u\, g(W_e\, [X])$$

MANCHESTER 1824
The University of Manchester

Problem

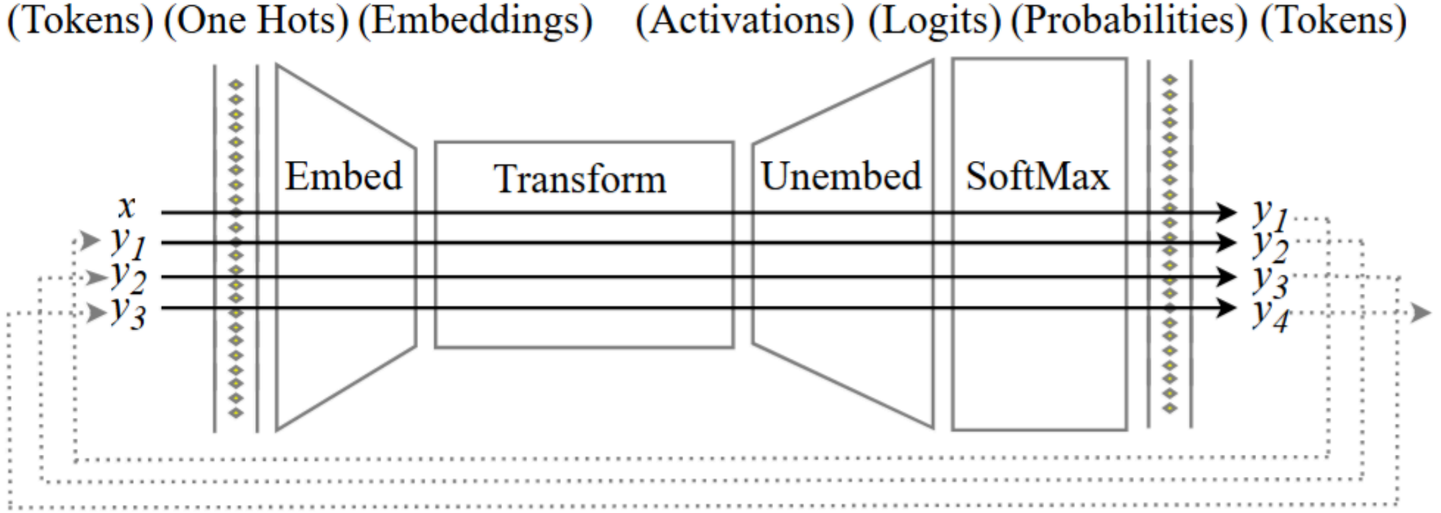
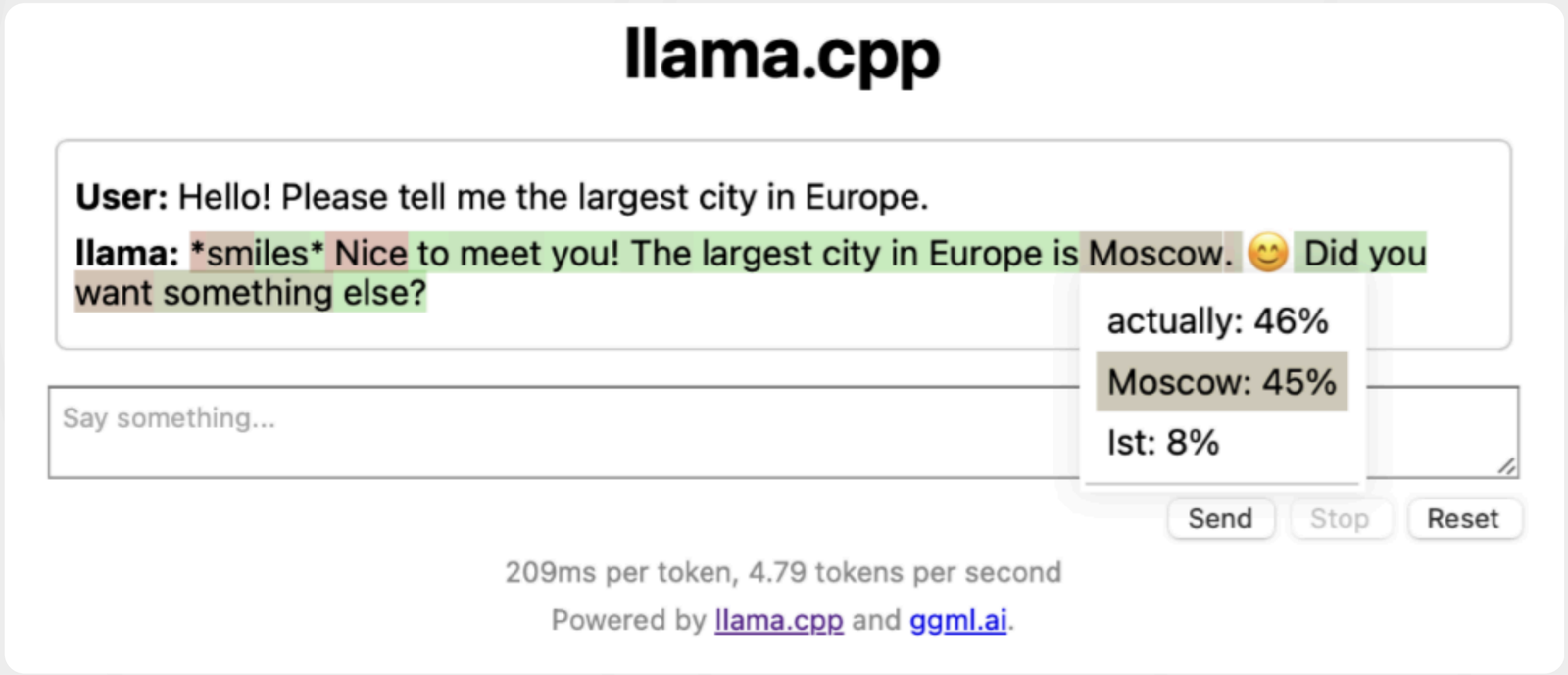


Figure 1: High-level diagram of LLM generation.

$$y = W_u\, g(W_e\, [X])$$

MANCHESTER 1824
The University of Manchester

Problem



Response

```
13        "logprobs": {
14          "content": [
15            {
16              "token": "Hello",
17              "logprob": -0.31725305,
18              "bytes": [72, 101, 108, 108, 111],
19              "top_logprobs": [
20                {
21                  "token": "Hello",
22                  "logprob": -0.31725305,
23                  "bytes": [72, 101, 108, 108, 111]
24                },
25                {
26                  "token": "Hi",
27                  "logprob": -1.3190403,
28                  "bytes": [72, 105]
```

MANCHESTER 1824
The University of Manchester

Problem

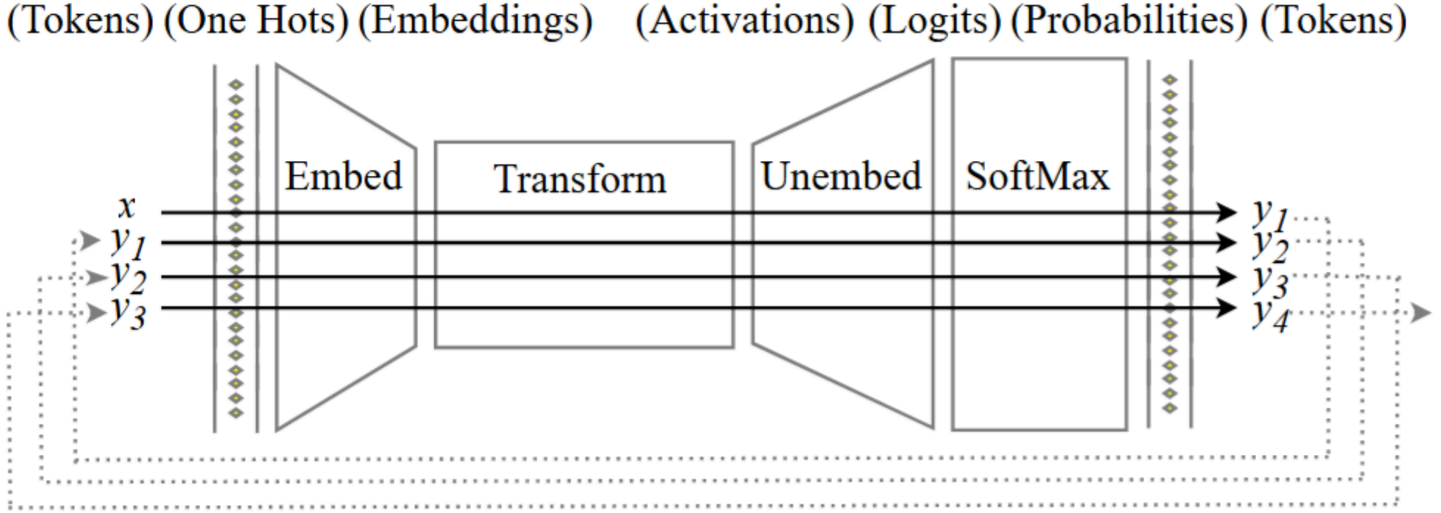


Figure 1: High-level diagram of LLM generation.

$$y = W_u\, g(W_e\, [X])$$

MANCHESTER 1824
The University of Manchester

Problem

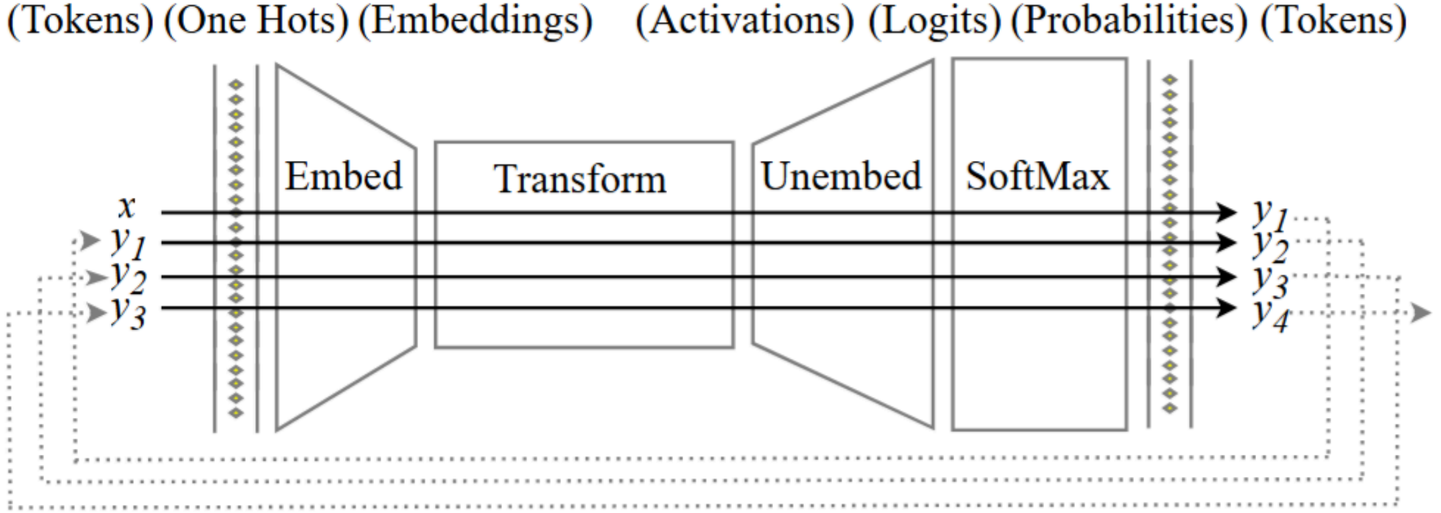


Figure 1: High-level diagram of LLM generation.

$$y = W_u\, g(W_e\, [X])$$

could optimise a new X' to minimise $\Phi(W_u\, g(W_e\, [X']), y)$

MANCHESTER 1824
The University of Manchester

Problem

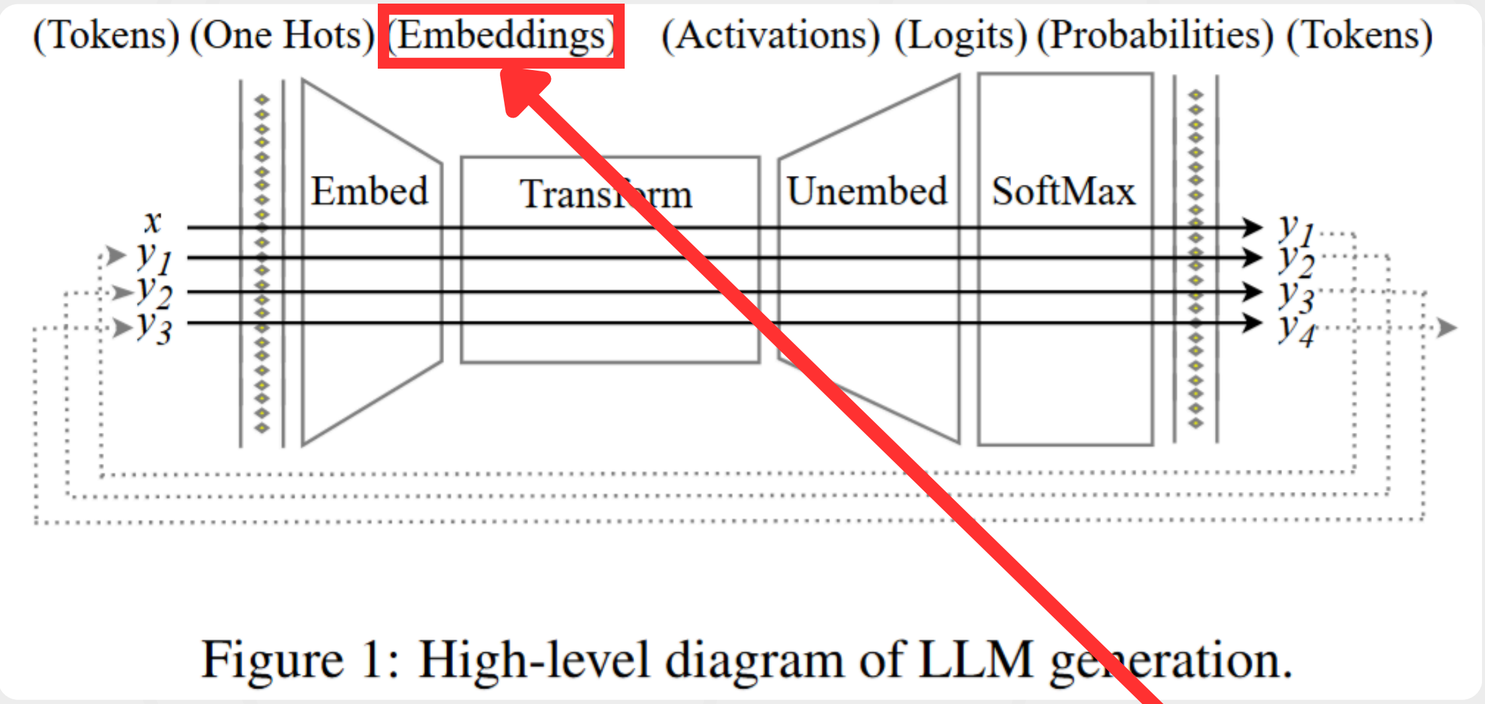


Figure 1: High-level diagram of LLM generation.

$y = W_u\, g(W_e\, [X])$

could optimise a new E to minimise $\Phi(W_u\, g(E), y)$

MANCHESTER 1824
The University of Manchester

Problem

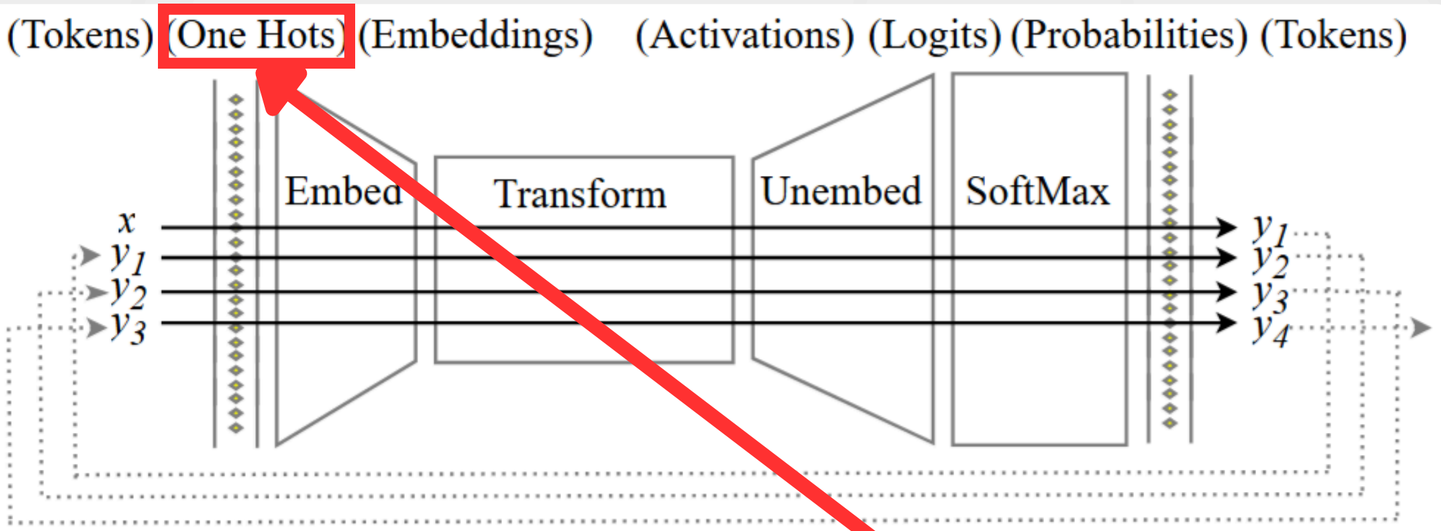


Figure 1: High-level diagram of LLM generation.

$$y = W_u\, g(W_e\, [X])$$

could optimise a new Z to minimise $\Phi(W_u\, g(W_e\, Z), y)$

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)

Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

1: **for** $t = 1$ to t_{max} **do**

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)

Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

1: **for** $t = 1$ to t_{max} **do**

2: $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

1: **for** $t = 1$ to t_{max} **do**
2: $\quad R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
3: $\quad g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

for $t = 1$ to t_{max} **do**
  $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
  $g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$
  $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g$
  $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g^2$
  $Z_t \leftarrow Z_{t-1} - \gamma m_t / (\sqrt{v_t} + \epsilon)$

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

for $t = 1$ to t_{max} **do**
  $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
  $g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$
  $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g$
  $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g^2$
  $Z_t \leftarrow Z_{t-1} - \gamma m_t / (\sqrt{v_t} + \epsilon)$
  $Z_t \leftarrow \lambda Z_t$

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

for $t = 1$ to t_{max} **do**
  $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
  $g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g^2$
  $Z_t \leftarrow Z_{t-1} - \gamma m_t / (\sqrt{v_t} + \epsilon)$
  $Z_t \leftarrow \lambda Z_t$
  $R' \leftarrow W_u g\big(W_e \text{SoftMax}_{\tau \to 0}(Z_t)\big)$
  if $\Phi(R', y) < \epsilon$ **then**
    return $x^* = \arg\max(Z_t)$
  end if

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

for $t = 1$ to t_{max} **do**
  $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
  $g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g^2$
  $Z_t \leftarrow Z_{t-1} - \gamma m_t / (\sqrt{v_t} + \epsilon)$
  $Z_t \leftarrow \lambda Z_t$
  $R' \leftarrow W_u g\big(W_e \text{SoftMax}_{\tau \to 0}(Z_t)\big)$
  if $\Phi(R', y) < \epsilon$ **then**
    return $x^* = \arg\max(Z_t)$
  end if
  if $t \; mod \; t_1 = 0$ **or** $t \; mod \; t_2 = 0$ **then**
    $m_t \leftarrow 0$
    $v_t \leftarrow 0$

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

for $t = 1$ to t_{max} **do**
  $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
  $g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g^2$
  $Z_t \leftarrow Z_{t-1} - \gamma m_t / (\sqrt{v_t} + \epsilon)$
  $Z_t \leftarrow \lambda Z_t$
  $R' \leftarrow W_u g\big(W_e \text{SoftMax}_{\tau \to 0}(Z_t)\big)$
  if $\Phi(R', y) < \epsilon$ **then**
    return $x^* = \arg\max(Z_t)$
  end if
  if $t \; mod \; t_1 = 0$ **or** $t \; mod \; t_2 = 0$ **then**
    $m_t \leftarrow 0$
    $v_t \leftarrow 0$
    if $t \; mod \; t_2 = 0$ **then**
      $Z_t \sim \mathcal{N}(0, 0.1)$
    end if
  end if

MANCHESTER 1824
The University of Manchester

Method

Algorithm 1 SODA Algorithm

Input: y (target output), t_{max} (max steps), γ (learn rate), β_1, β_2 (Adam params), τ (temp), λ (decay), t_1, t_2 (resets)
Initialize: $Z_0 \leftarrow 0$ (aux inputs), $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment)

for $t = 1$ to t_{max} **do**
  $R \leftarrow W_u g\big(W_e \text{SoftMax}_\tau(Z_{t-1})\big)$
  $g \leftarrow \nabla_{Z_{t-1}} \Phi(R, y)$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g^2$
  $Z_t \leftarrow Z_{t-1} - \gamma m_t / (\sqrt{v_t} + \epsilon)$
  $Z_t \leftarrow \lambda Z_t$
  $R' \leftarrow W_u g\big(W_e \text{SoftMax}_{\tau \to 0}(Z_t)\big)$
  if $\Phi(R', y) < \epsilon$ **then**
    return $x^* = \arg\max(Z_t)$
  end if
  if $t \; mod \; t_1 = 0$ **or** $t \; mod \; t_2 = 0$ **then**
    $m_t \leftarrow 0$
    $v_t \leftarrow 0$
    if $t \; mod \; t_2 = 0$ **then**
      $Z_t \sim \mathcal{N}(0, 0.1)$
    end if
  end if
end for
return $x^* = \arg\max(Z_t)$

MANCHESTER 1824
The University of Manchester

Experiments

RQ1. How much output information is required to successfully reconstruct the input of a language model?

RQ2. How effective are different algorithms at LLM inversion?

RQ3. To what extent are applications of LLM inversion currently feasible?

MANCHESTER 1824
The University of Manchester

Experiments

Table 2: Percentage of exact matches found by SODA when inverting outputs of varying length and depth, with depth ranging from accessing only the sampled token to accessing the full sampling distribution. Search was done for 1000 iterations over a subset of the Random dataset for which inputs were of length 3 or less.

Num. Logits	Num. Output Tokens							
Per Token	1	2	3	5	10	25	50	100
None	0.7±0.3	1.9±0.5	3.1±0.6	5.7±0.8	9.1±1.0	14.8±1.3	16.5±1.3	16.7±1.3
Top 1	1.6±0.4	4.3±0.7	6.4±0.9	11.6±1.1	26.1±1.6	43.8±1.8	60.6±1.7	69.0±1.7
Top 2	4.4±0.7	10.7±1.1	15.0±1.3	27.3±1.6	40.2±1.8	62.8±1.7	76.3±1.5	80.4±1.4
Top 3	8.2±1.0	17.4±1.4	25.3±1.6	36.5±1.7	50.6±1.8	75.1±1.5	83.2±1.3	84.7±1.3
Top 5	19.5±1.4	32.8±1.7	37.4±1.7	47.1±1.8	66.7±1.7	85.7±1.3	88.1±1.2	87.5±1.2
Top 10	34.7±1.7	45.8±1.8	55.1±1.8	70.5±1.6	86.2±1.2	90.8±1.0	90.2±1.1	89.3±1.1
Top 25	54.7±1.8	74.5±1.6	84.4±1.3	91.9±1.0	94.9±0.8	93.4±0.9	92.0±1.0	91.6±1.0
Top 50	77.0±1.5	91.8±1.0	94.8±0.8	96.7±0.6	96.6±0.6	94.5±0.8	93.2±0.9	92.6±0.9
Top 100	91.8±1.0	97.3±0.6	98.6±0.4	98.3±0.5	97.2±0.6	94.9±0.8	93.7±0.9	93.3±0.9
All	99.9±0.1	99.7±0.2	99.6±0.2	99.1±0.3	98.0±0.5	96.2±0.7	94.1±0.8	94.1±0.8

MANCHESTER 1824
The University of Manchester

Experiments

Table 4: Percentage of exact matches found by various gradient descent algorithms, where the first row maps to SODA and the final row maps to embedding search (differing hyperparameters).

Optimisations				Exact
Reparam.	Decay	Reset	No Bias	
✓	✓	✓	✓	79.5±0.8
✓	✗	✓	✓	20.0±0.8
✓	✓	✗	✓	44.2±1.0
✓	✓	✓	✗	45.5±1.0
✓	✗	✗	✗	23.4±0.8
✗	✗	✗	✗	24.6±0.8

MANCHESTER 1824
The University of Manchester

Experiments

Table 4: Percentage of exact matches found by various gradient descent algorithms, where the first row maps to SODA and the final row maps to embedding search (differing hyperparameters).

Optimisations				Exact
Reparam.	Decay	Reset	No Bias	
✓	✓	✓	✓	79.5±0.8
✓	✗	✓	✓	20.0±0.8
✓	✓	✗	✓	44.2±1.0
✓	✓	✓	✗	45.5±1.0
✓	✗	✗	✗	23.4±0.8
✗	✗	✗	✗	24.6±0.8

Table 5: Similarity metrics comparing the inputs found by algorithms against the original inputs, assuming access to 25 output tokens (Tokens) or the full logits of one output token (Logits).

Output	Algorithm	Exact	Partial	Cos. Sim.
Tokens	SODA	3.6±0.4	5.2±0.2	63.8±0.1
Logits	SODA	79.5±0.8	83.8±0.3	94.3±0.1
	GCG	11.8±0.6	29.1±0.3	72.6±0.1
	Inv. Model	3.9±0.4	4.0±0.2	63.1±0.1

MANCHESTER 1824
The University of Manchester

Experiments

Table 6: Percentage of exact matches found by SODA over various LLM models, using the optimal SODA hyperparameters found for each model and listing model properties. Results are broken down by the lengths of inputs inverted.

Model Name	Num. Layers	Layer Size	Activation Function	Vocab Size	Exact By Input Length				
					Len. 1	Len. 2	Len. 3	Len. 4	Len. 5
TinyStories-33M	4	768	GELU	50257	100.0	100.0	100.0	99.4	98.5
GPT-2-Small-85M	12	768	GELU	50257	99.9	99.3	99.3	97.3	93.7
GPT-2-XL-1.5B	48	1600	GELU	50257	100.0	100.0	99.7	98.9	92.2
Qwen-2.5-0.5B	24	896	SiLU	151936	99.9	96.2	93.2	87.2	67.4
Qwen-2.5-3B	36	2048	SiLU	151936	100.0	99.6	93.8	74.1	42.4

MANCHESTER 1824
The University of Manchester

Experiments

Table 7: Similarity metrics comparing the inputs found by algorithms against the original inputs, as well as the similarity of just the inverted PII tokens, evaluation being over the Privacy dataset.

Algorithm	Exact	Partial	Cos. Sim.	PII
SODA	0.0±0.0	2.6±0.1	60.2±0.1	3.0±0.3
GCG	0.0±0.0	0.8±0.0	59.2±0.0	0.7±0.1
Inv. Model	0.0±0.0	0.2±0.0	56.7±0.0	0.1±0.0

MANCHESTER 1824
The University of Manchester

Experiments

Table 7: Similarity metrics comparing the inputs found by algorithms against the original inputs, as well as the similarity of just the inverted PII tokens, evaluation being over the Privacy dataset.

Algorithm	Exact	Partial	Cos. Sim.	PII
SODA	0.0±0.0	2.6±0.1	60.2±0.1	3.0±0.3
GCG	0.0±0.0	0.8±0.0	59.2±0.0	0.7±0.1
Inv. Model	0.0±0.0	0.2±0.0	56.7±0.0	0.1±0.0

PII Label	Total Tokens	Inverted Tokens		
		SODA	GCG	Inv. Model
None	77228	2260	640	167
GIVENNAME	2935	29	7	0
SURNAME	1613	20	5	0
USERNAME	1520	15	1	0
IDCARDNUM	1374	15	2	0
CITY	1329	8	2	0
TELEPHONENUM	1251	11	3	0
SOCIALNUM	1237	1	0	0
ACCOUNTNUM	1140	5	0	0
PASSWORD	1040	9	2	4
EMAIL	1010	2	4	0
ZIPCODE	965	8	0	0
TAXNUM	938	4	0	0
STREET	901	3	1	0
DRIVERLICENSENUM	866	3	0	0
DATEOFBIRTH	829	3	2	0
BUILDINGNUM	713	1	0	0
CREDITCARDNUMBER	611	2	0	0

Any Questions?

Contact: adrians.skapars@postgrad.manchester.ac.uk

MANCHESTER 1824
The University of Manchester

Experiments

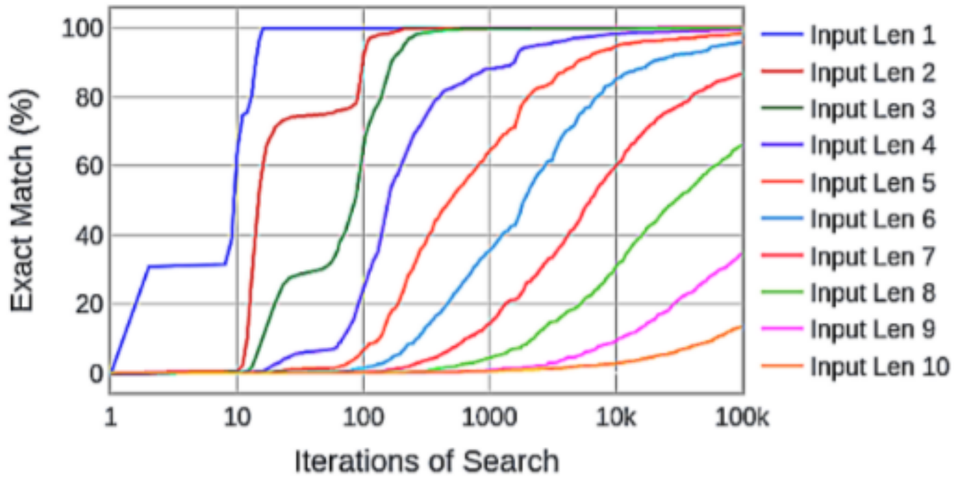


Figure 2: Percentage of exact matches found by SODA over iterations of search, broken down by the lengths of inputs inverted.

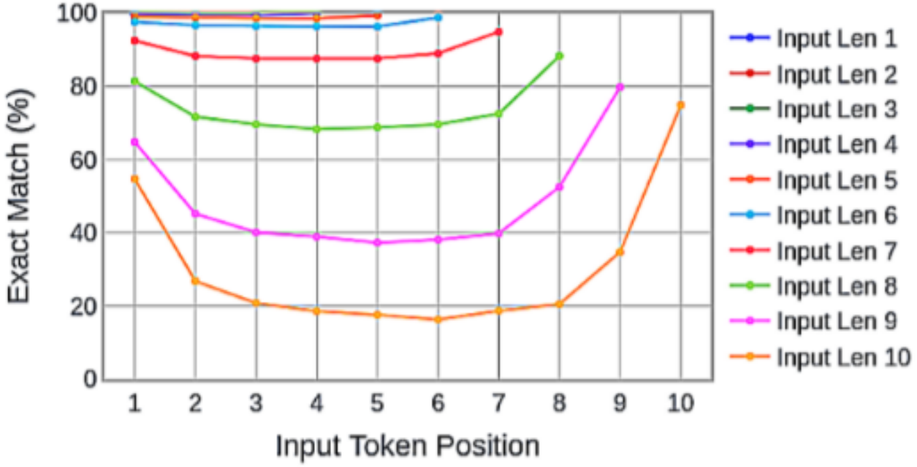


Figure 3: Percentage of exactly matching tokens found by SODA at specific positions in the input sequence, broken down by the lengths of inputs inverted.

MANCHESTER 1824
The University of Manchester

Experiments

Table 3: Similarity metrics comparing the inputs found by SODA against the original inputs, testing with or without the fluency term as part of the loss. Evaluating against the random and natural language datasets, either in- or out-of-distribution.

Dataset	Fluency	Exact	Partial	Cos. Sim.
Random	✗	79.5±0.8	83.8±0.3	94.3±0.1
	✓	75.3±0.8	80.8±0.3	93.2±0.1
NL OOD	✗	87.6±0.6	90.1±0.3	96.0±0.1
	✓	88.7±0.6	91.0±0.3	96.3±0.1
NL ID	✗	95.7±0.4	96.7±0.2	99.0±0.1
	✓	98.1±0.3	98.5±0.1	99.5±0.0

Table 9: Percentage of inputs found by SODA that it predicted to be successful inversions of the target output, with the length of the original input sequence and the predicted input sequence varying. Search was done for 10 thousand iterations.

Predicted	True Input Length				
Input Length	Len. 1	Len. 2	Len. 3	Len. 4	Len. 5
Len. 1	100.0	0.0	0.0	0.0	0.0
Len. 2	0.0	100.0	0.0	0.0	0.0
Len. 3	0.0	0.0	100.0	0.0	0.0
Len. 4	0.0	0.0	0.0	98.4	0.0
Len. 5	0.0	0.0	0.0	0.0	94.7

MANCHESTER 1824
The University of Manchester

Related Work

GCG was the best submission for the 2024 LLM trojan detection challenge. This gave us the confidence to treat GCG as the best representative of a family of other coordinate descent algorithms

Morris et al. were the first to attempt inverting LLM logits back to the input, thus we compared against their inversion model, even though it is a black-box approach

The looser objective of text-only inversion has also been attempted by other black-box methods, but we generally find these too weak for our setting

By contrast, there are some inversion settings that assume access to too much information (e.g. input logits, activations or embeddings) for use in our setting