# LLMs in Psycholinguistics

**Romance Lab**

**Goethe-Universität Frankfurt**

Part I: 14th July 2025

**Umesh Patil**

# Workshop Scope (Part I & II)

1. Introduction to how Large Language Models (LLMs) work

2. Introduction to Surprisal Theory?

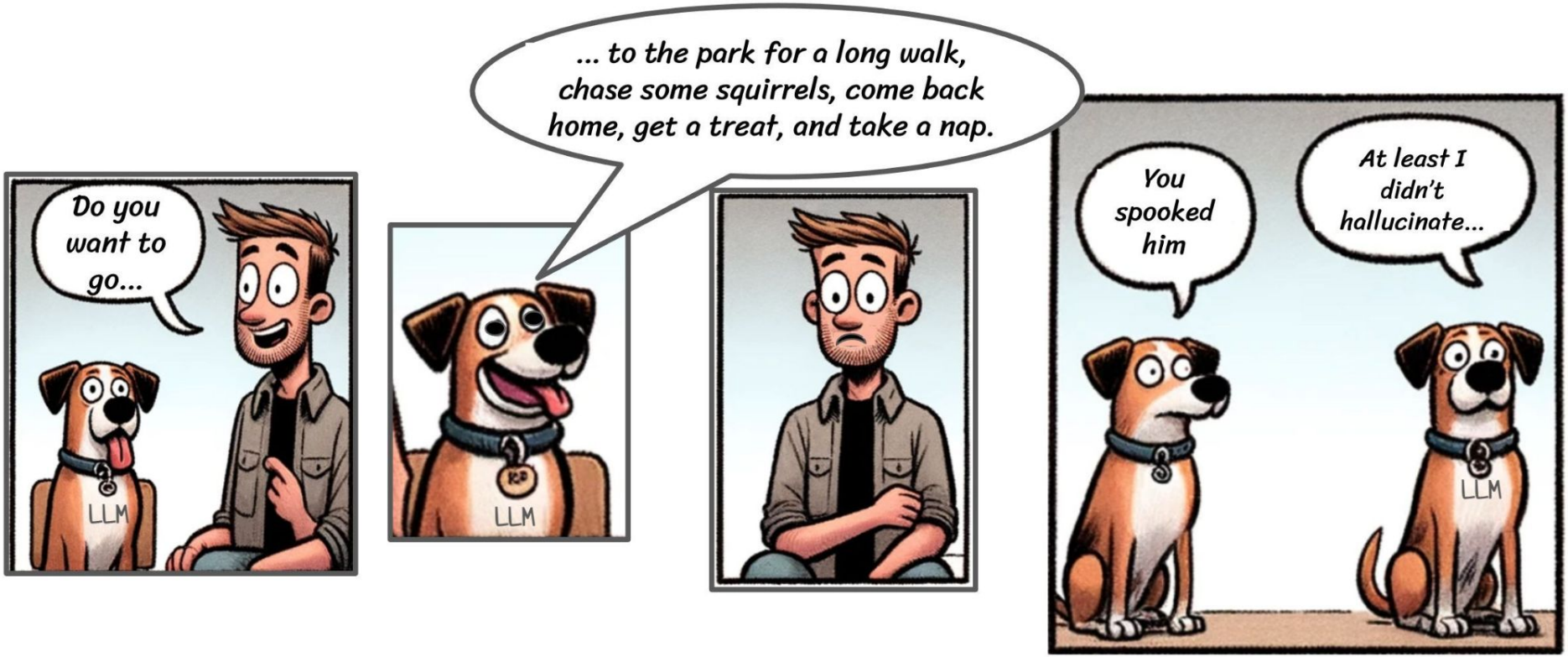
3. How to use LLMs and the **`minicons`** package to:
    a. Generate surprisal values word by word ⇒ word by word processing difficulty
    b. Generate average surprisal for a sentence ⇒ acceptability of a sentence
    c. Generate probabilities for next word(s) ⇒ sentence completion (production)

4. Using different/multiple LLMs from **HuggingFace**
5. How to use LLMs to generate predictions for your items?
    a. 3(a)-(c) for non-English languages (German, Spanish, Portuguese, French, etc.)

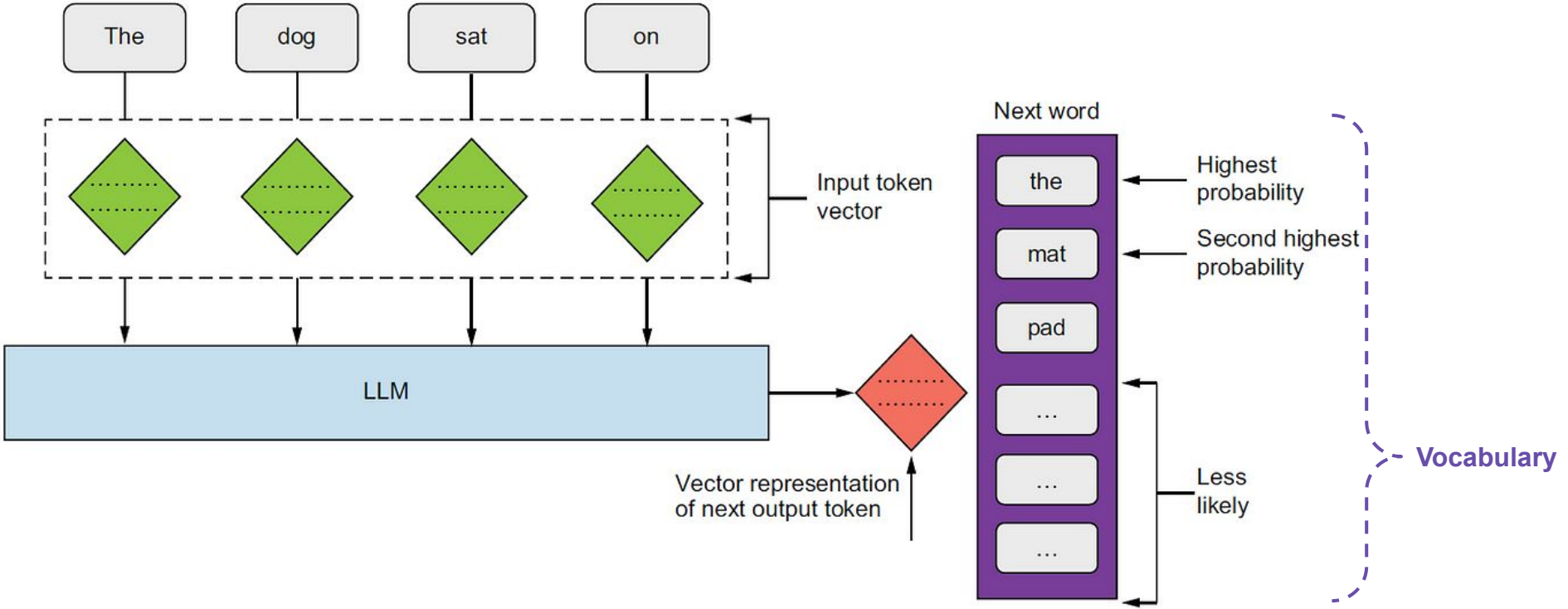
# Questions!

1. Have you have listened to *all three* talks in HSP-LLM series?
2. Have you listened to Kanishka Misra's **`minicons`** talk?
3. Have you used the **`minicons`** package?
4. Do you know what *Surprisal* means?
5. Are you familiar to programming?
6. Have you used Python?
7. Do you have an experiment(s) for which you want to use LLM/**`minicons`** predictions?

# Intro to LLMs: Next Word Prediction



# Intro to LLMs: Next Word Prediction




Source: https://python.plainenglish.io/the-art-of-prediction-how-llms-master-next-token-generation-b8f81dc16de2

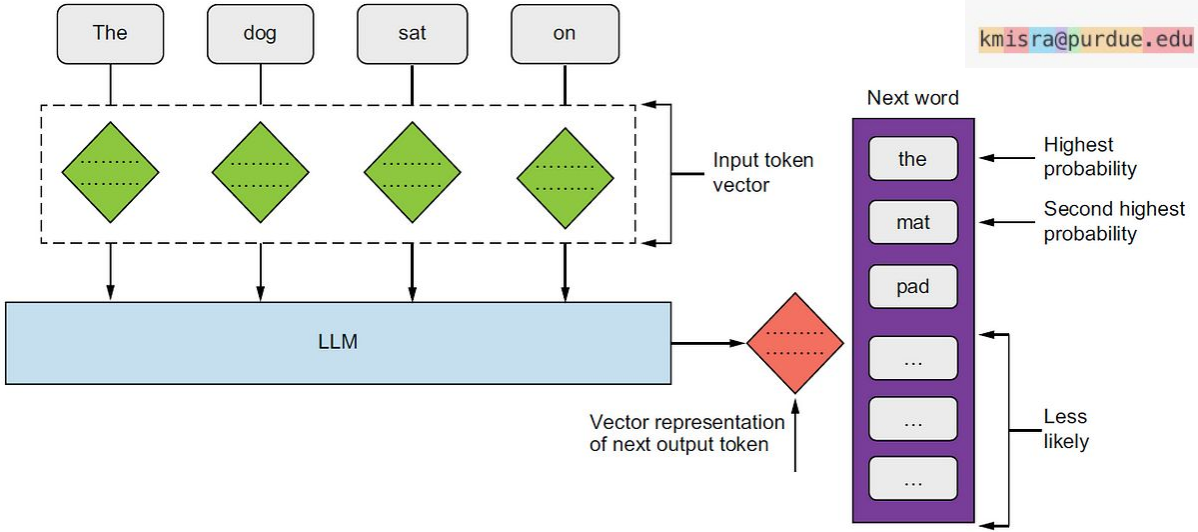
# Intro to LLMs: Next Token Prediction (Tokenization)

**minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models**

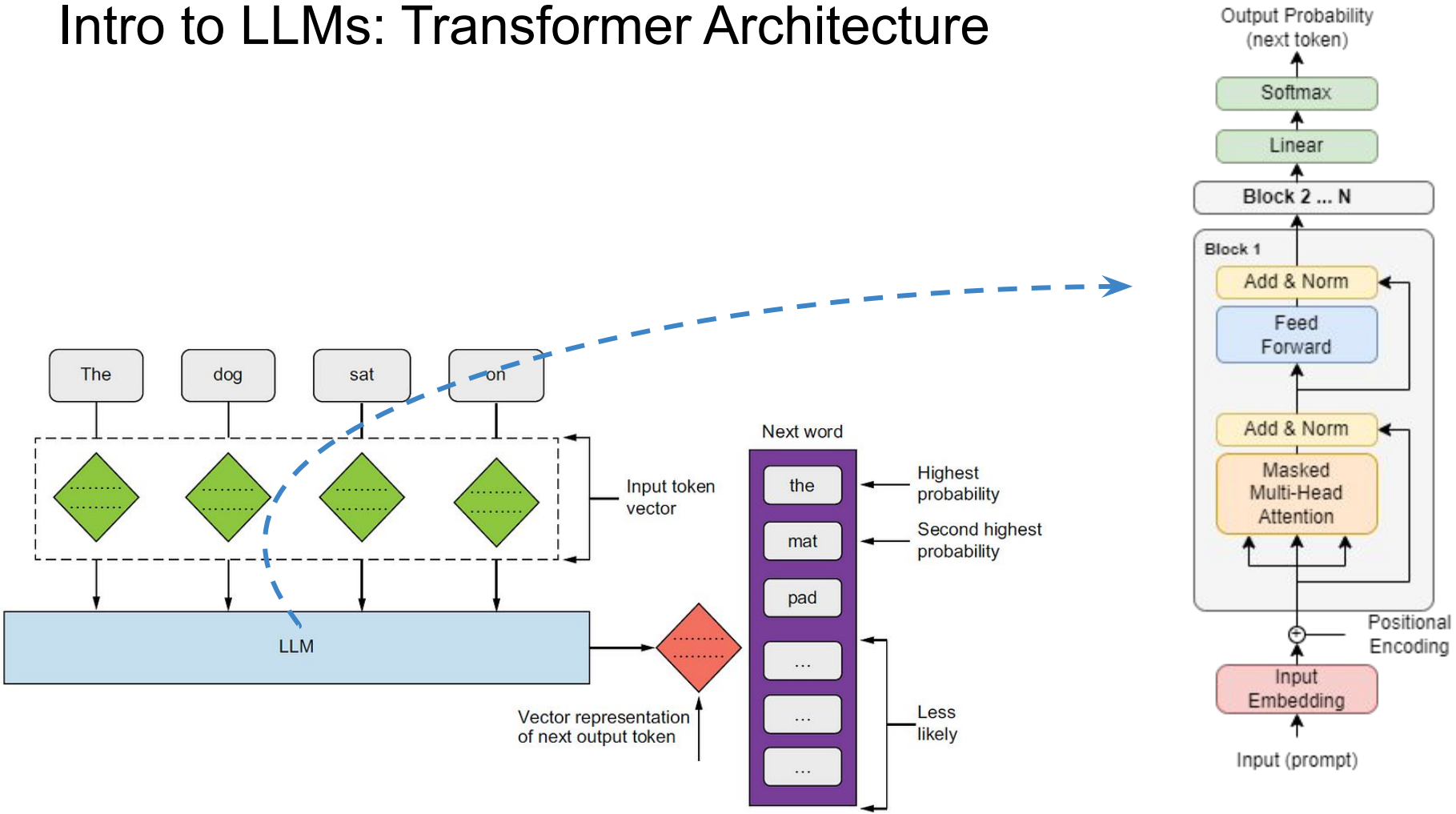
**Kanishka Misra**
Purdue University
kmisra@purdue.edu

```
minicons: Enabling Flexible Behavioral and Representational Analyses of
 Transformer Language Models

Kanishka Misra

Purdue University

kmisra@purdue.edu
```

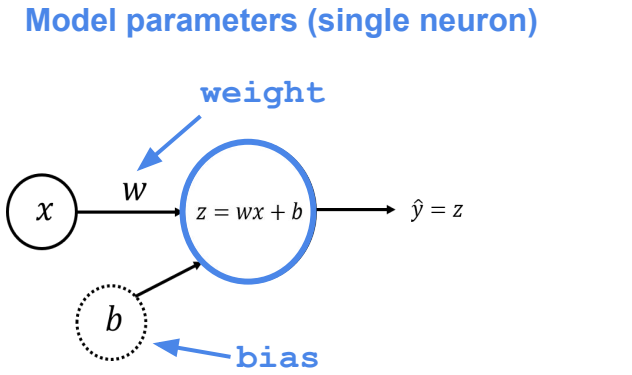
https://platform.openai.com/tokenizer

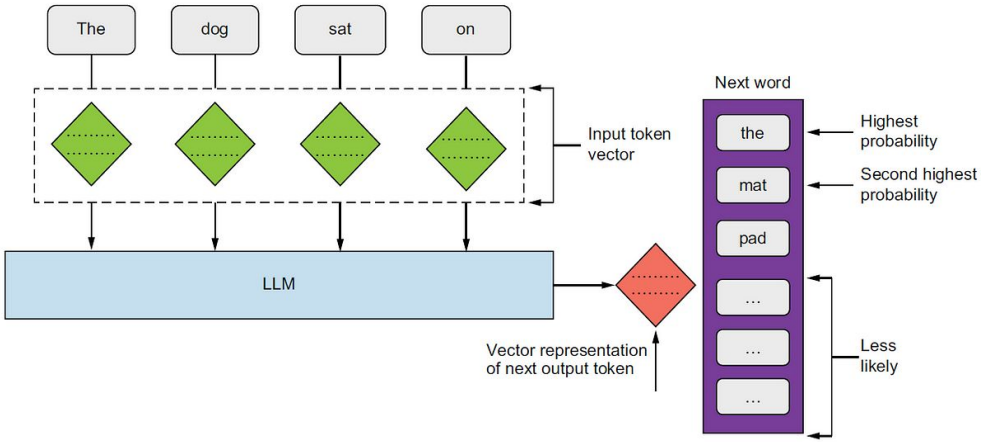


# Intro to LLMs: Transformer Architecture

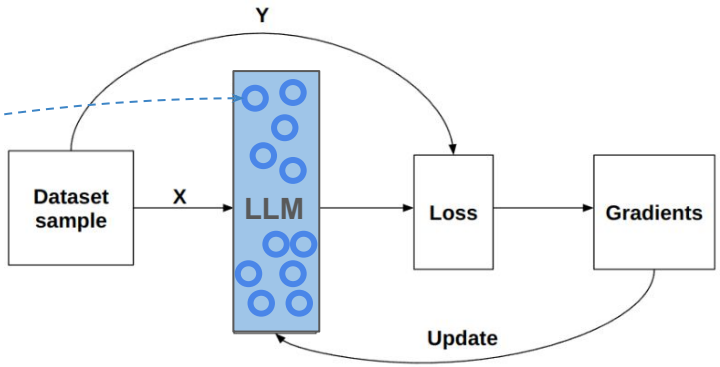


# (extremely brief) Intro to LLM Training

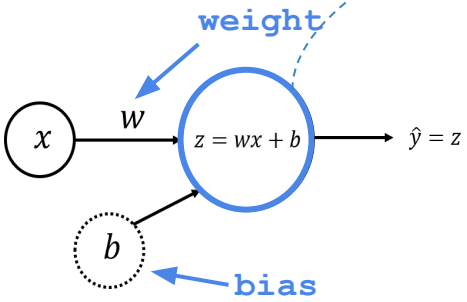


# (extremely brief) Intro to LLM Training



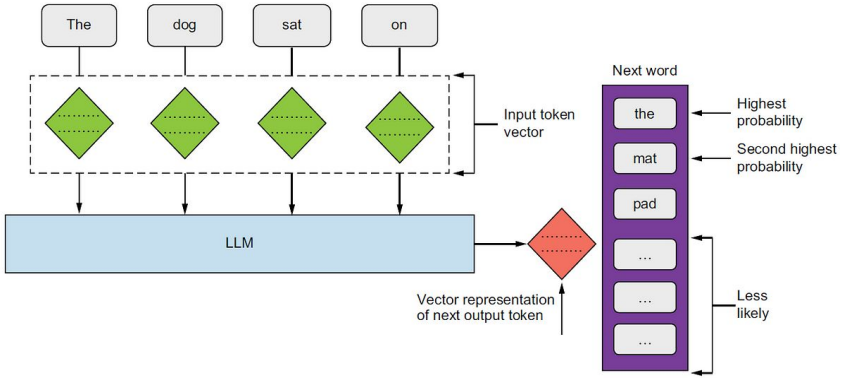

**Model parameters (single neuron)**




**Loss**:

- loss $L(\theta)$ measures how poorly the model (with parameters $\theta$) is doing on its task of next token pred.
- what the mode seeks to minimize during training

$$\mathcal{L} = -\sum_{t=1}^{T} \log P_\theta\left(w_t \mid w_{1:t-1}\right)$$



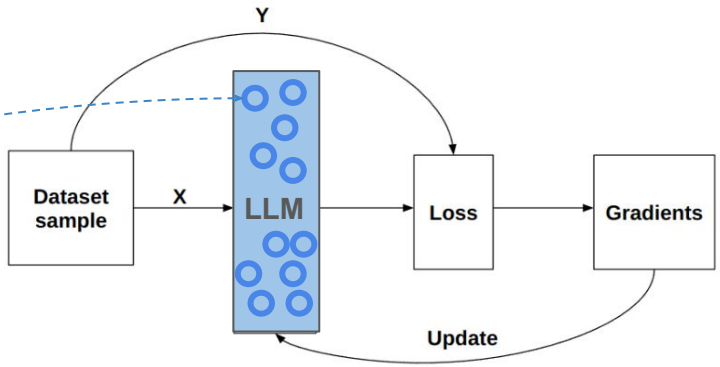

**Backpropagation** (compute gradients):

- To improve the model, it has to update the parameters so that the Loss reduces
- Backpropagation is the algorithm that efficiently computes the **gradients** (change in the value of parameters)

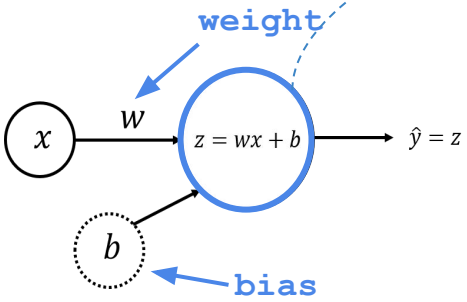
**Gradient Descent** (Updating Parameters):

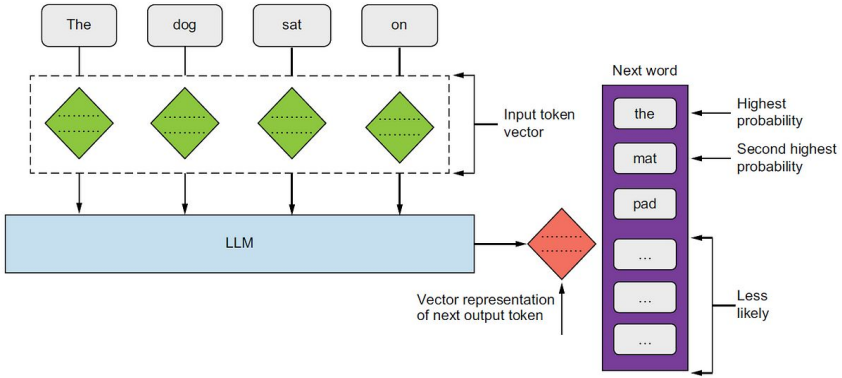
- Apply a update rule to shift the parameters to reduce the loss

# (extremely brief) Intro to LLM Training




**Model parameters (single neuron)**

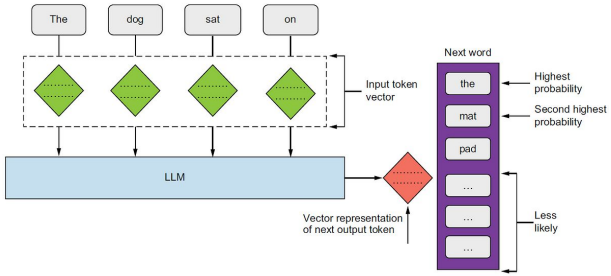



**One training step**

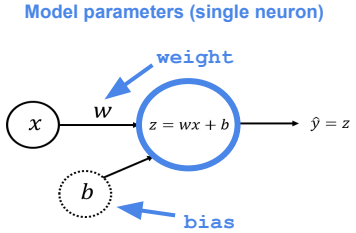
1. **Forward pass**: compute model outputs and loss on the given input
2. **Backward pass**: run backpropagation to compute gradients
3. **Parameter update**: apply gradient descent to adjust parameters

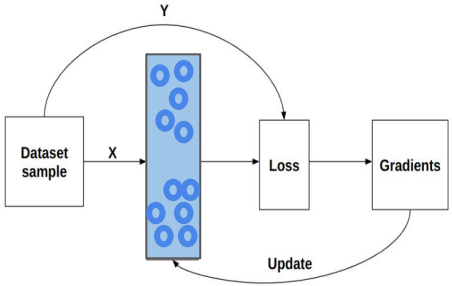
# LLM Concepts




minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models

1. Tokenization
   - Subword Tokens
   - Vocabulary
2. Training
   - Parameters (**Weights** & Biases)
   - Loss
   - Backpropagation
     - Gradients
   - Gradient descent



# Surprisal & Psycholinguistics

- **Surprisal**: Information conveyed by any given linguistic unit **x** (e.g. phoneme, word, utterance) in context.

- Surprisal is:
  - high, when **x** has a low conditional probability, and
  - low, when **x** has a high conditional probability.

- Claim: Cognitive effort required to process a word is proportional to its surprisal (Hale, 2001).

$$\text{surprisal}(x) = \log\left(\frac{1}{P(x \mid \text{context})}\right) = -\log P(x \mid \text{context})$$

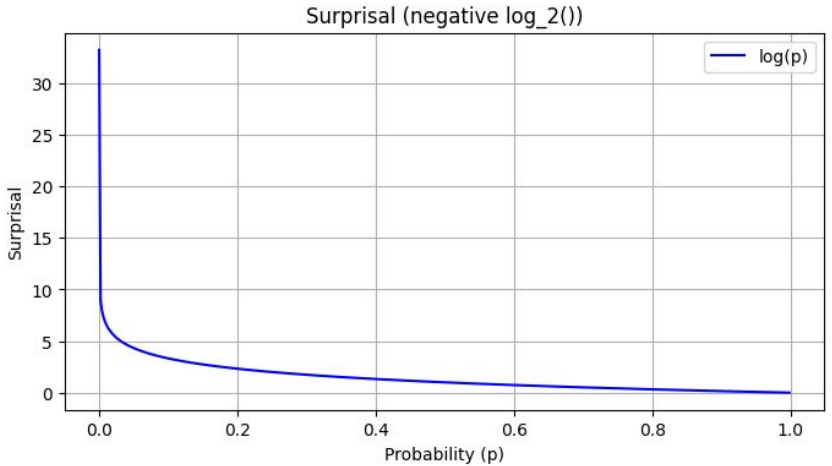
$$\text{surprisal}(w_t) = -\log P(w_t \mid w_{1:t-1})$$

# Surprisal & Psycholinguistics

- **Surprisal**: Information conveyed by any given linguistic unit **x** (e.g. phoneme, word, utterance) in context.
- Surprisal is:
  - high, when **x** has a low conditional probability, and
  - low, when **x** has a high conditional probability.
- Claim: Cognitive effort required to process a word is proportional to its surprisal (Hale, 2001).

$$\text{surprisal}(x) = \log\left(\frac{1}{P(x \mid \text{context})}\right) = -\log P(x \mid \text{context})$$

$$\text{surprisal}(w_t) = -\log P(w_t \mid w_{1:t-1})$$



# Surprisal & Psycholinguistics

- **Surprisal**: Information conveyed by any given linguistic unit **x** (e.g. phoneme, word, utterance) in context.

- Surprisal is:
  - high, when **x** has a low conditional probability, and
  - low, when **x** has a high conditional probability.

- Claim: Cognitive effort required to process a word is proportional to its surprisal (Hale, 2001).

$$\text{surprisal}(x) = \log\left(\frac{1}{P(x \mid \text{context})}\right) = -\log P(x \mid \text{context})$$

$$\text{surprisal}(w_t) = -\log P(w_t \mid w_{1:t-1})$$

**Loss**:

- loss $L(\theta)$ measures how poorly the model (with parameters $\theta$) is doing on its task of next token pred.
- what the mode seeks to minimize during training

$$\mathcal{L} = -\sum_{t=1}^{T} \log P_\theta\left(w_t \mid w_{1:t-1}\right)$$

# Surprisal & Psycholinguistics

- **Surprisal**: Information conveyed by any given linguistic unit **x** (e.g. phoneme, word, utterance) in context.
- Surprisal is:
  - high, when **x** has a low conditional probability, and
  - low, when **x** has a high conditional probability.
- Claim: Cognitive effort required to process a word is proportional to its surprisal (Hale, 2001).

$$\text{surprisal}(x) = \log\left(\frac{1}{P(x \mid \text{context})}\right) = -\log P(x \mid \text{context})$$

$$\text{surprisal}(w_t) = -\log P(w_t \mid w_{1:t-1})$$

**Loss**:

- loss $L(\theta)$ measures how poorly the model (with parameters $\theta$) is doing on its task of next token pred.
- what the mode seeks to minimize during training

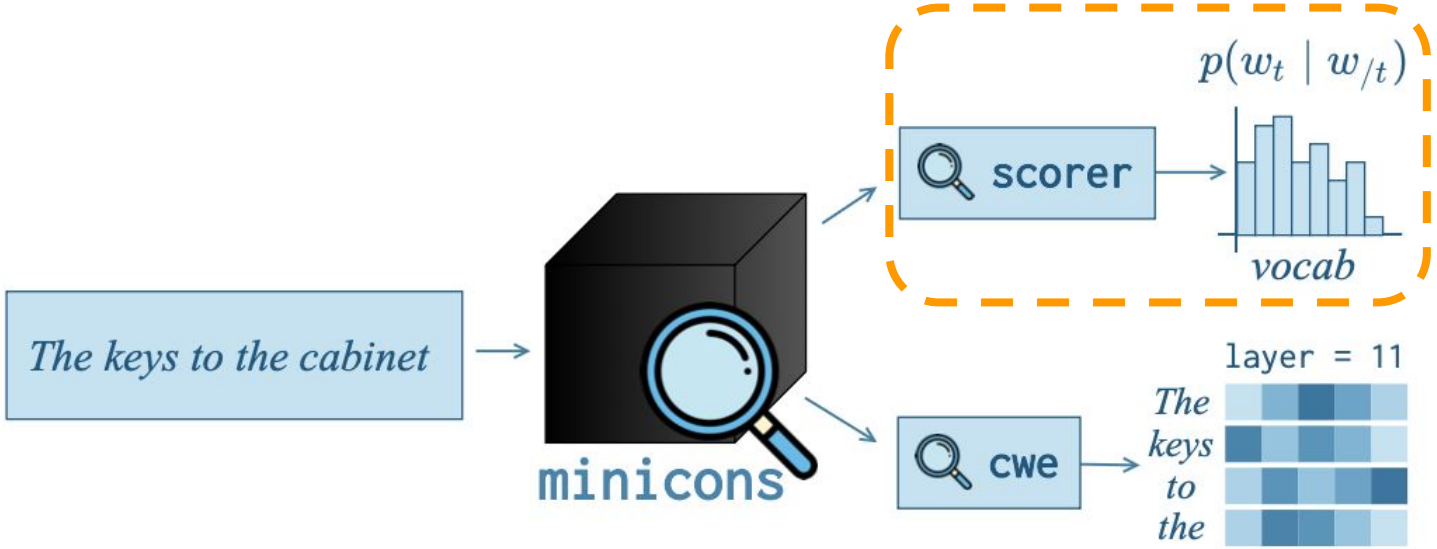
$$\mathcal{L} = -\sum_{t=1}^{T} \log P_\theta\left(w_t \mid w_{1:t-1}\right)$$

⇒ Training an (autoregressive) LLM means getting parameter values that **minimize total surprisal** on the data.

# `minicons` Package

Facilitates analyses of (transformer-based) LLMs at:

- The prediction level through its `scorer` module
- The representational level through its `cwe` module



https://shorturl.at/lZCpm

# **minicons** Package:

```
bos_token=True
bow_correction=True
```


## Quick digression to Oh and Schuler (2024)

```
lm.token_score(
    good_sentence,
    bos_token=True,
    surprisal=True,
    bow_correction=True
)
```

**Leading Whitespaces of Language Models' Subword Vocabulary Pose a Confound for Calculating Word Probabilities**

**Byung-Doh Oh**
Center for Data Science
New York University
oh.b@nyu.edu

**William Schuler**
Department of Linguistics
The Ohio State University
schuler.77@osu.edu

| 5.85 | 4.28 | 4.67 | 16.34 | 1.74 | 2.12 |
|---|---|---|---|---|---|
| I | ␣was | ␣a | ␣mat | ron | ␣in |

| 5.85 | 4.28 | 4.67 | 16.34 | 10.78 |
|---|---|---|---|---|
| I | ␣was | ␣a | ␣mat | ␣in |

(a) Surprisal values calculated with leading whitespaces.

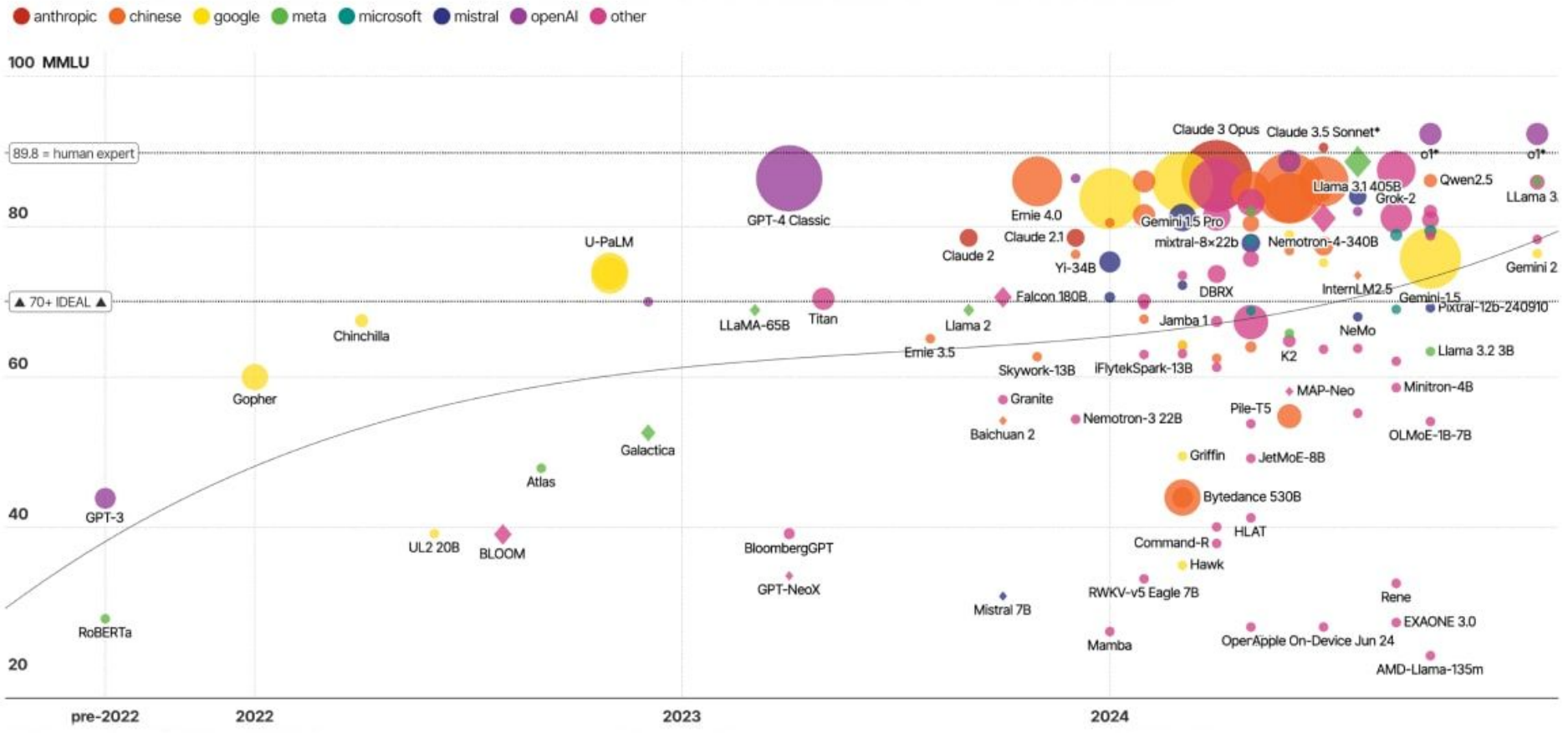
| 6.30 | 3.84 | 4.68 | 16.34 | 2.11 | 1.75 |
|---|---|---|---|---|---|
| I␣ | was␣ | a␣ | mat | ron␣ | in␣ |

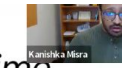
| 6.30 | 3.84 | 4.68 | 21.34 | 5.80 |
|---|---|---|---|---|
| I␣ | was␣ | a␣ | mat␣ | in␣ |

(b) Surprisal values calculated with trailing whitespaces.

Thank you!

# LLMs: evolution



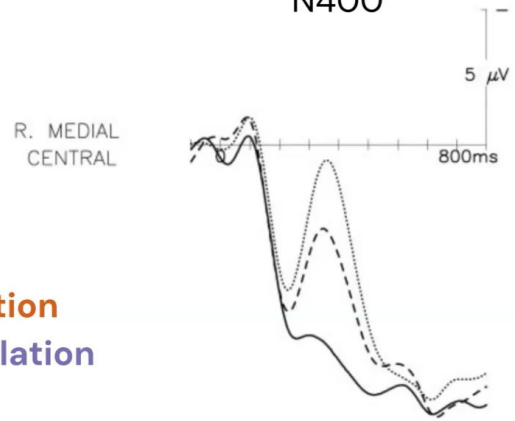



*Getting himself and his car to work on the neighboring island was time consuming. Every morning he drove for a few minutes and then boarded the* ***ferry*** ***gondola*** ***plane***

**expected**

**Contextually unexpected**

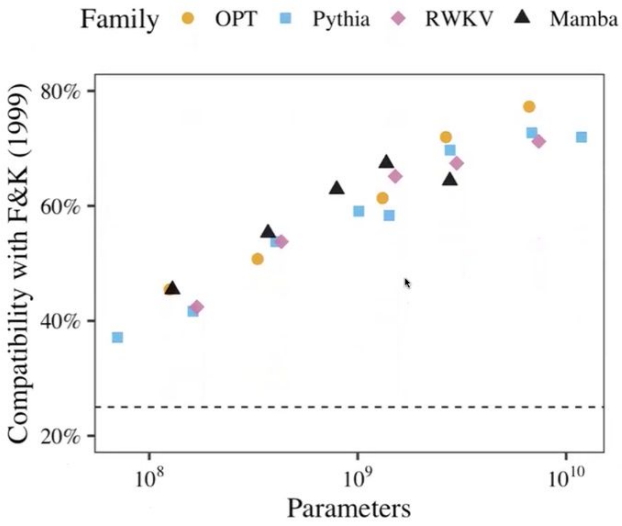
N400




*ferry* Expected

*gondola* Within-category violation

*plane* Between-category violation



What does the distribution look like?

`entropy()`

What is the top predicted word?

What is `p(I saw a bear, it was big!)` ?

What is `p(I | <s>)`?