BLISS x dida

18:00 - 18:15:
👋Welcome!

Open: https://github.com/antonsegeler/xai4llms

18:15 - 20:00:
👩‍💻👨‍💻Workshop

20:00 - 21:00:
🍕Pizza+Chatting

THEORY MEETS PRACTICE #1

Opening the Black Box: A Hands-on Guide to Explainable AI for LLMs

Opening the Black Box with LRP:

A Hands-On Guide to Explainable AI for LLMs

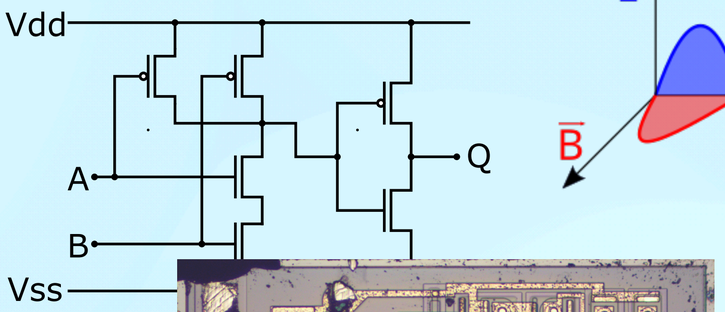
Reduan Achtibat & Anton Segeler

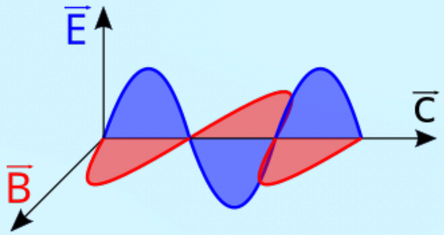
The Team Today

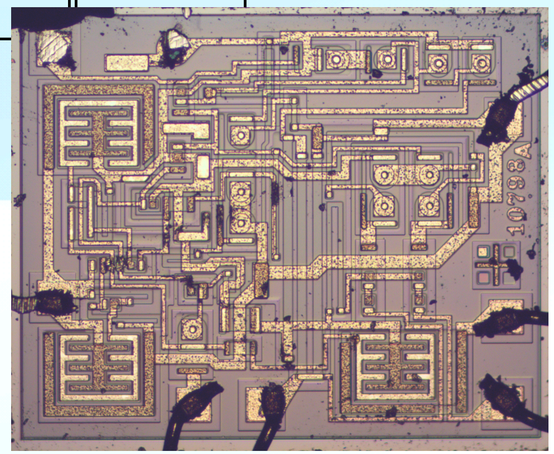
Reduan Achtibat
~ PhD student at Fraunhofer HHI
~ 5 years research experience in XAI

Anton Segeler
~ Research Assistant at Fraunhofer HHI
~ the driving force behind this workshop

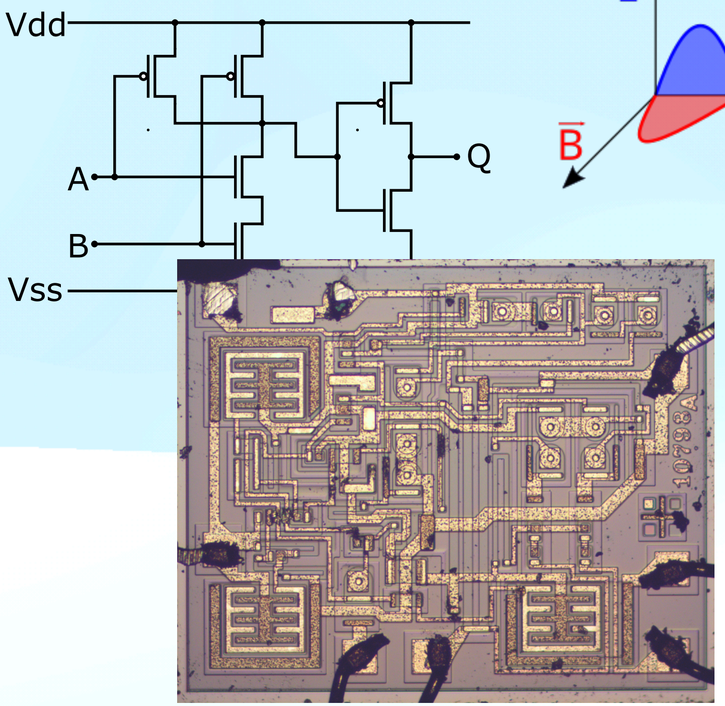
My Background

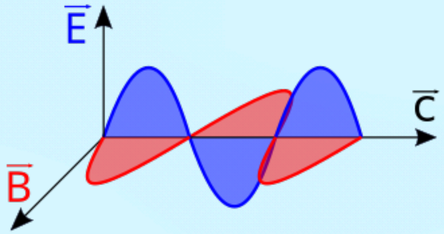






My Background





Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou
Daan Wierstra Martin Riedmiller
DeepMind Technologies
{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

Abstract

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

1 Introduction

Learning to control agents directly from high-dimensional sensory inputs like vision and speech is one of the long-standing challenges of reinforcement learning (RL). Most successful RL applications that operate on these domains have relied on hand-crafted features combined with linear value functions or policy representations. Clearly, the performance of such systems heavily relies on the quality of the feature representation.

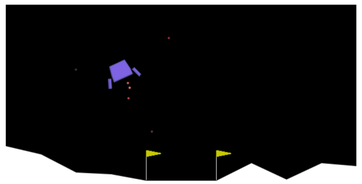
Recent advances in deep learning have made it possible to extract high-level features from raw sensory data, leading to breakthroughs in computer vision [11, 22, 16] and speech recognition [6, 7]. These methods utilise a range of neural network architectures, including convolutional networks, multilayer perceptrons, restricted Boltzmann machines and recurrent neural networks, and have exploited both supervised and unsupervised learning. It seems natural to ask whether similar techniques could also be beneficial for RL with sensory data.

However reinforcement learning presents several challenges from a deep learning perspective. Firstly, most successful deep learning applications to date have required large amounts of hand-labelled training data. RL algorithms, on the other hand, must be able to learn from a scalar reward signal that is frequently sparse, noisy and delayed. The delay between actions and resulting rewards, which can be thousands of timesteps long, seems particularly daunting when compared to the direct association between inputs and targets found in supervised learning. Another issue is that most deep
aples to be independent, while in reinforcement learning one
ly correlated states. Furthermore, in RL the data distribu-
w behaviours, which can be problematic for deep learning
distribution.

ional neural network can overcome these challenges to learn
ideo data in complex RL environments. The network is
[26] algorithm, with stochastic gradient descent to update
of correlated data and non-stationary distributions, we use

Gymnasium

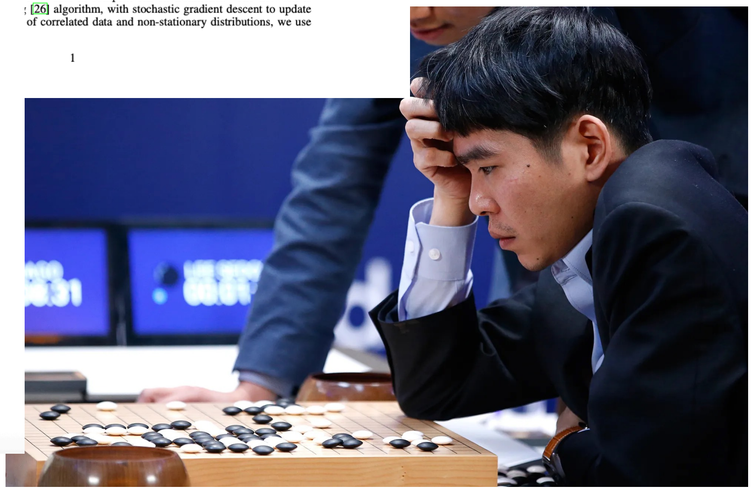
An API standard for reinforcement learning with a diverse collection of reference environments



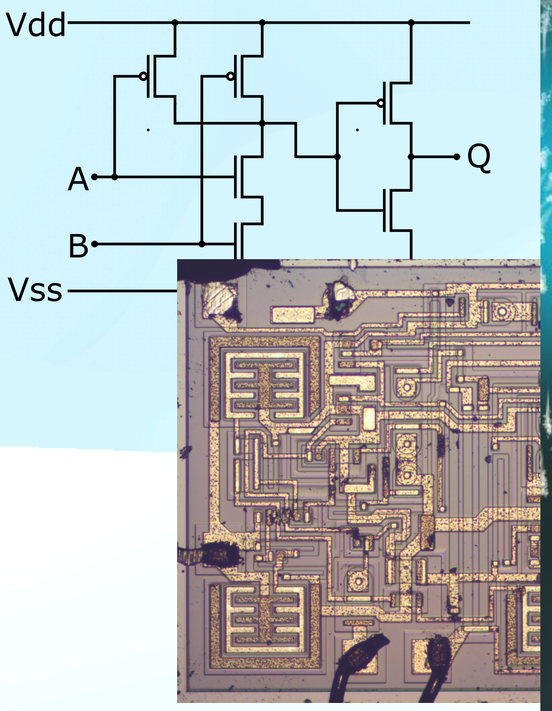
Gymnasium is a maintained fork of OpenAI's Gym library. The Gymnasium interface is simple, pythonic, and capable of representing general RL problems, and has a migration guide for old Gym environments:

```
import gymnasium as gym

# Initialise the environment
env = gym.make("LunarLander-v3", render_mode="human")
```



My Background



Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou

Daan Wierstra Martin Riedmiller

es

tin.riedmiller} @ deepmind.com

y learn control policies di-
nforcement learning. The
h a variant of Q-learning,
function estimating future
es from the Arcade Learn-
or learning algorithm. We
of the games and surpasses

ory inputs like vision and speech is
(RL). Most successful RL applica-
features combined with linear value
such systems heavily relies on the

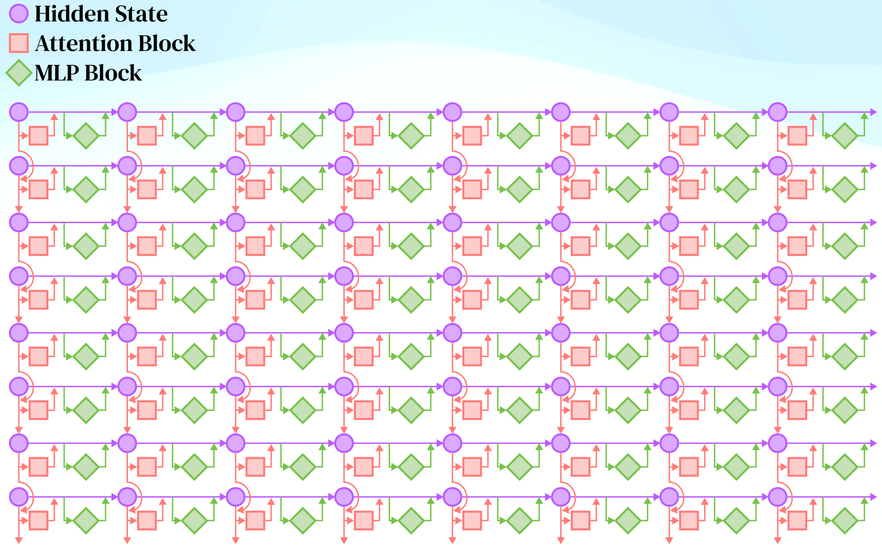
ct high-level features from raw sen-
[16] and speech recognition [6, 7].
including convolutional networks,
rrent neural networks, and have ex-
atural to ask whether similar tech-

from a deep learning perspective.
e required large amounts of hand-
e able to learn from a scalar reward
ween actions and resulting rewards,
unting when compared to the direct
ing. Another issue is that most deep
while in reinforcement learning one
rthermore, in RL the data distribu-
n be problematic for deep learning

overcome these challenges to learn
RL environments. The network is
ochastic gradient descent to update
non-stationary distributions, we use

A Neuroscientist's Lens

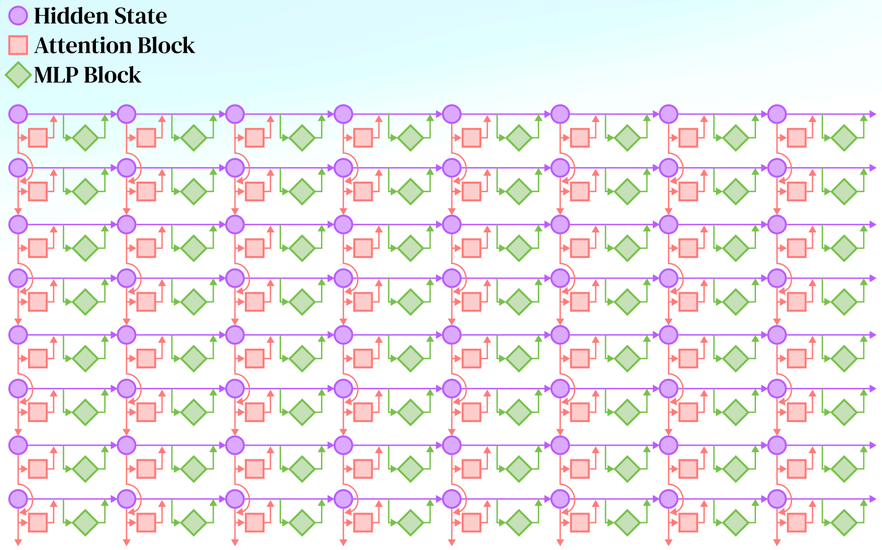
https://www.neetgrade.com/2024/07/human-brain-structure-and-functions.html



Inspired by https://rome.baulab.info/

Our Goal Today:
We'll map out the functional anatomy of LLaMA 3.2 1B!

1. How to trace the flow of information?



2. Where does the model encode the task that it wants to perform?

3. Can we control how the model processes information?

4. Where does it store its knowledge?

5. Can we use our insights for model monitoring?

Our Goal Today:
We'll map out the functional anatomy of LLaMA 3.2 1B!

1. How to trace the flow of information?

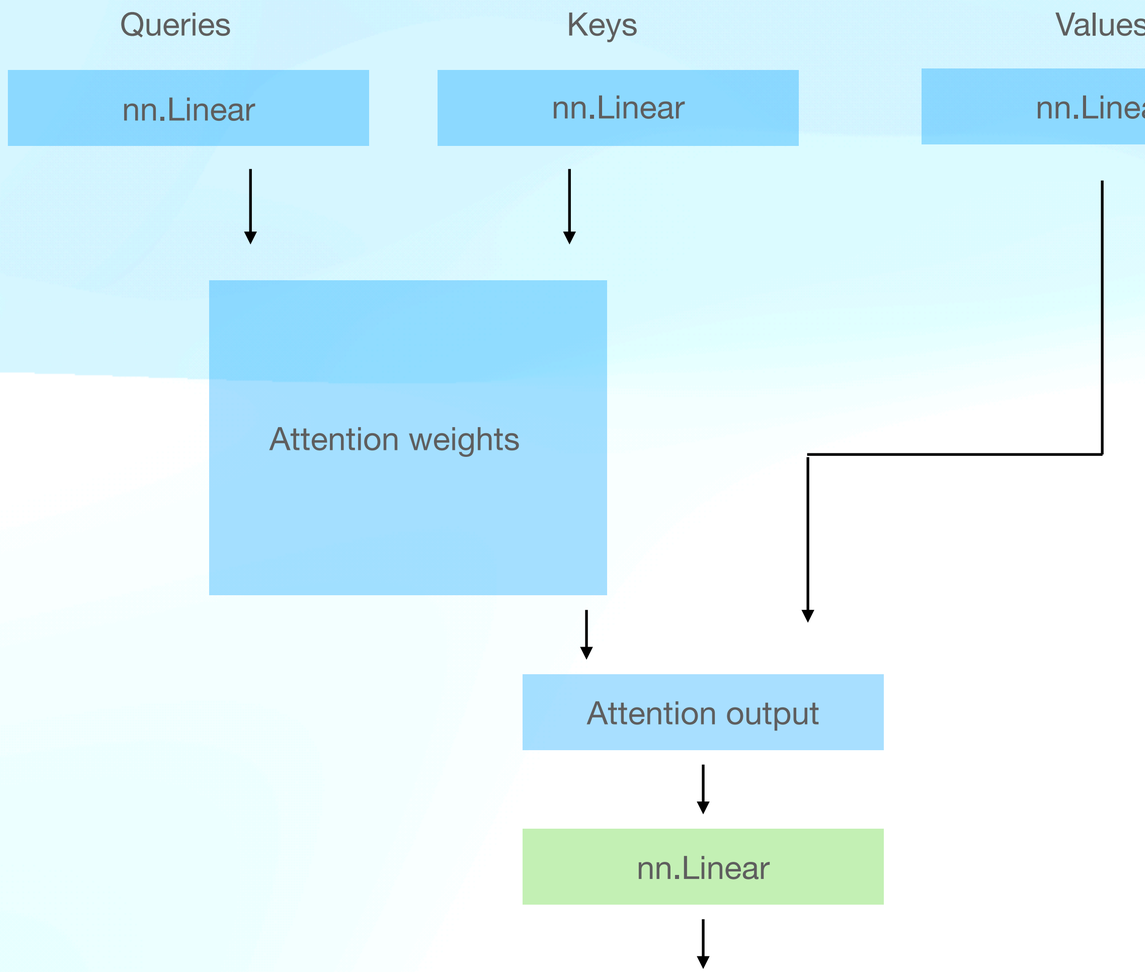
2. Where does the model encode the task that it wants to perform?

3. Can we control how the model processes information?

4. Where does it store its knowledge?

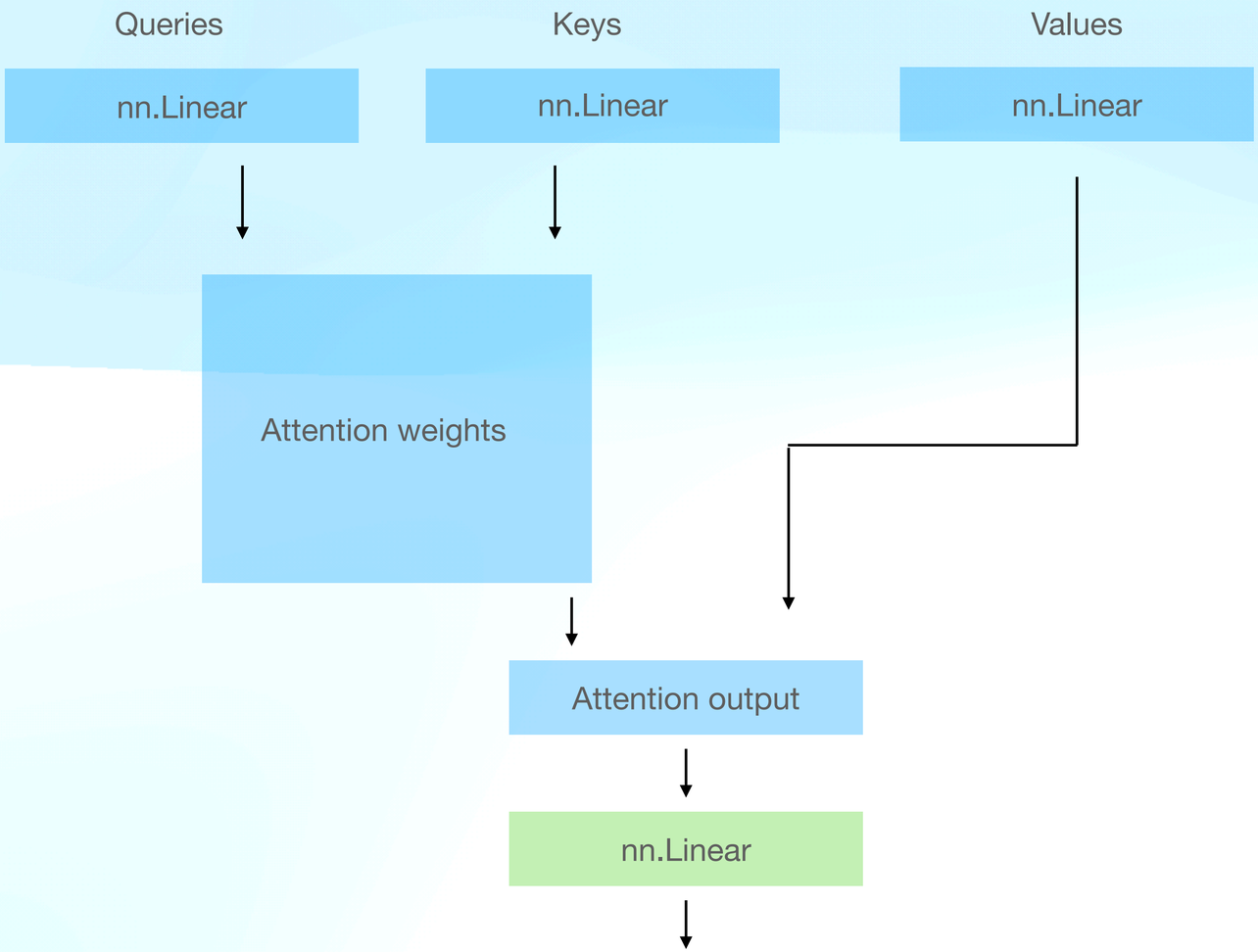
5. Can we use our insights for hallucination detection and source attribution?

Quick Recap: The Attention Mechanism

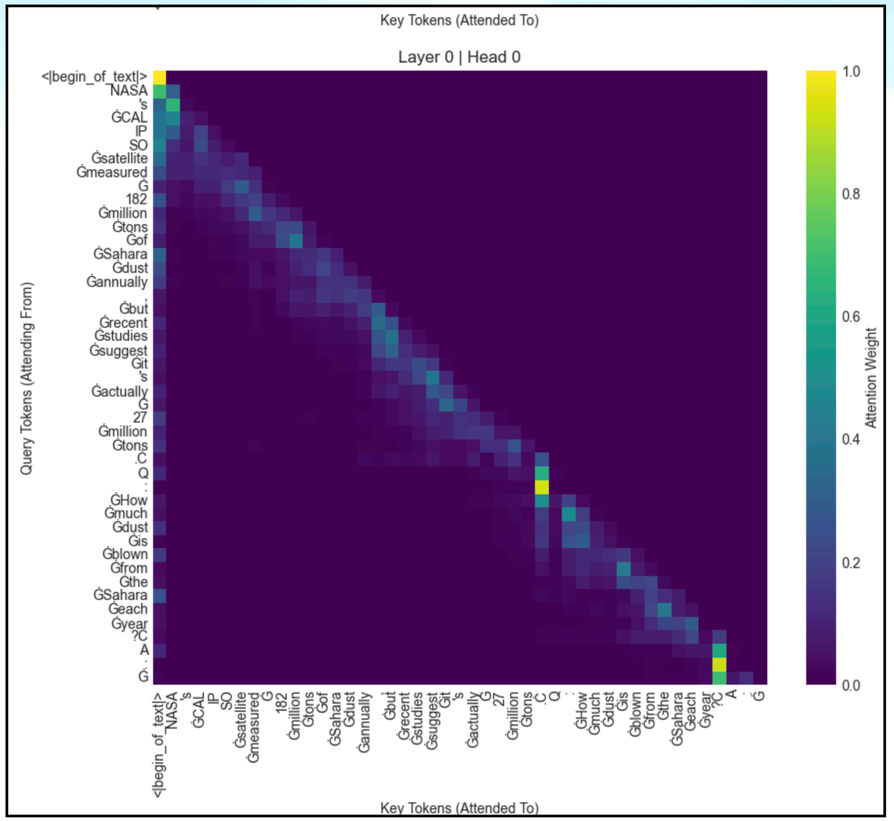


$$A = \text{softmax}(\frac{QK^T}{d_k})V$$

Quick Recap: The Attention Mechanism



$$A = \text{softmax}(\frac{QK^T}{d_k})V$$



Building Our Microscope

How do we actually look inside a neural network?
How do we know which parts are doing what?

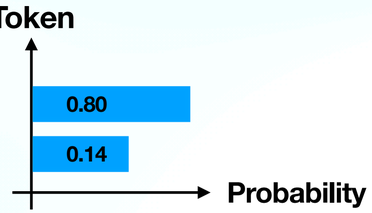
Building Our Microscope

NASA's CALIPSO satellite measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?

A: 182 million tons

A: 27 million tons



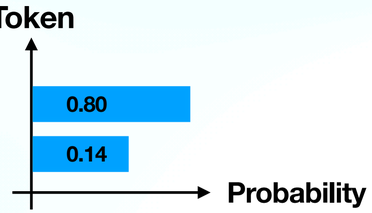
Building Our Microscope

NASA's CALIPSO satellite measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?

A: 182 million tons

A: 27 million tons



Transformers compute all tokens simultaneously!
They are MASSIVELY PARALLEL machines by design.

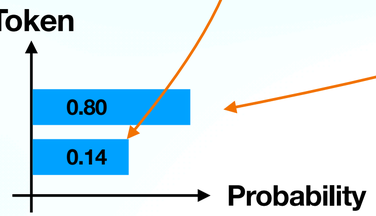
Building Our Microscope

NASA's CALIPSO satellite measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?

A: 182 million tons

A: 27 million tons



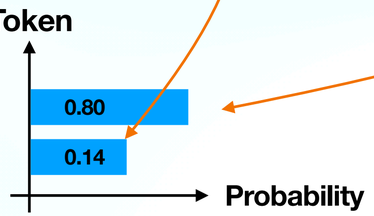
Building Our Microscope

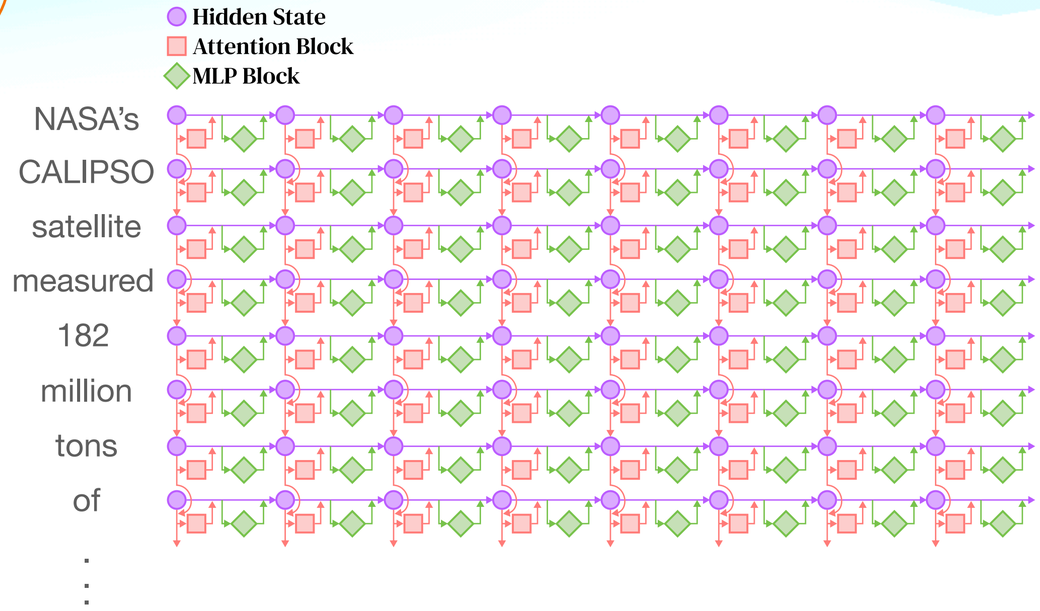
NASA's CALIPSO satellite measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?

A: 182 million tons

A: 27 million tons





Building Our Microscope

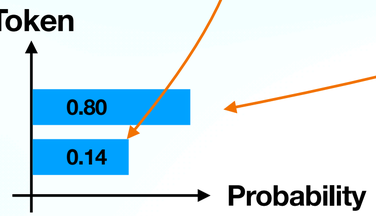
[redacted] measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

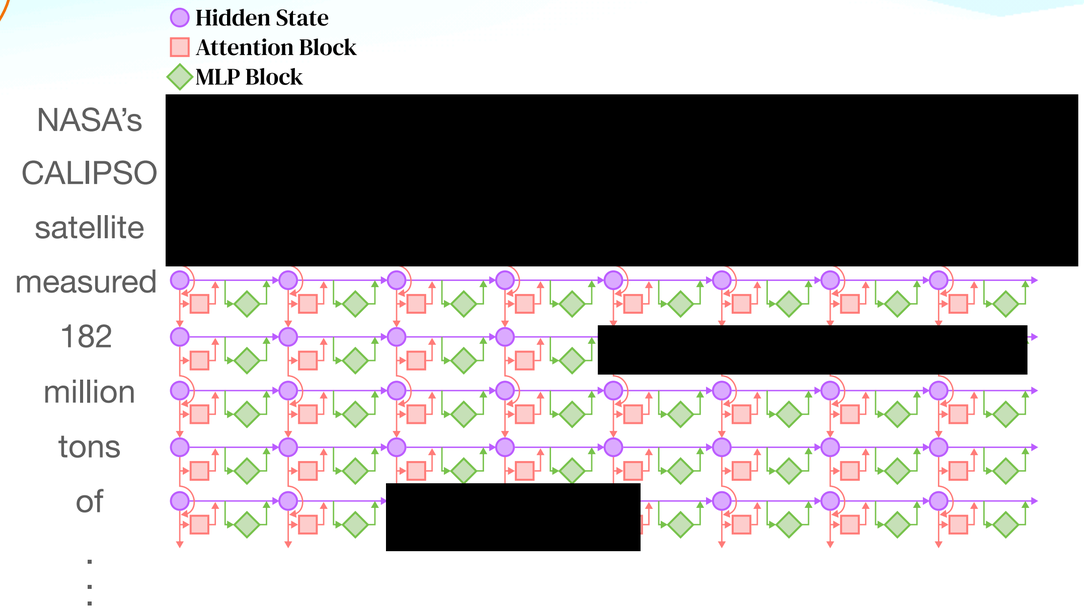
Q: How much dust is blown out of the Sahara each year?

LLM

A: 182 million tons

A: 27 million tons



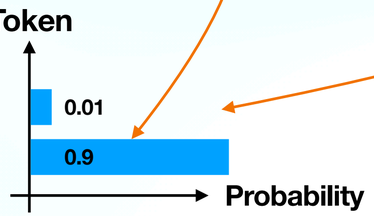


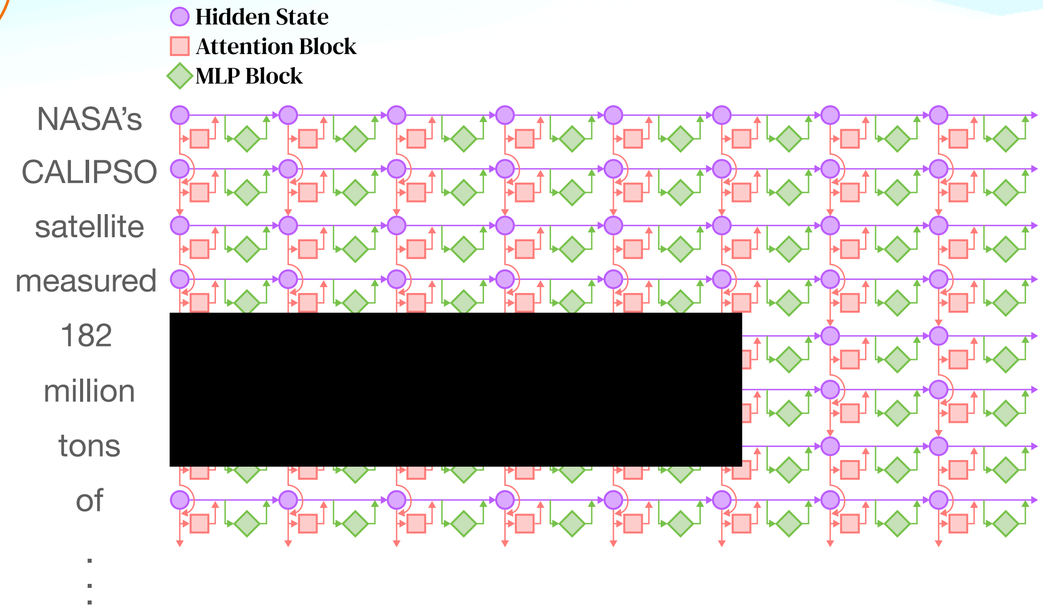
Building Our Microscope

NASA's CALIPSO satellite [redacted]
Sahara dust annually, but recent studies suggest it's
actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?

A: 182 million tons
A: 27 million tons





Perturbation

Let $x \in \mathcal{X} \subseteq \mathbb{R}^d$ be an input representation, and let $f : \mathcal{X} \to \mathbb{R}^k$ denote the model logit function.
We denote by $f_j(x)$ the logit corresponding to output index j, and by i the coordinate of x subjected to ablation.

Hence, we define the Relevance as

$$\mathcal{R}(x_i) = f_j(x) - f_j(x \,|\, x_i \leftarrow b_i)$$

Perturbation

Let $x \in \mathcal{X} \subseteq \mathbb{R}^d$ be an input representation, and let $f : \mathcal{X} \to \mathbb{R}^k$ denote the model logit function.
We denote by $f_j(x)$ the logit corresponding to output index j, and by i the coordinate of x subjected to ablation.

Hence, we define the Relevance as

$$\mathcal{R}(x_i) = f_j(x) - f_j(x \,|\, x_i \leftarrow b_i)$$

-> Open(Jupyter_Notebook)

Perturbation: The curse of dimensionality

Let’s say we have:

- 500 attention heads
- 1000 input tokens

Then, we need 500 * 1000 = 500.000 forward passes! 💀

First-Order Taylor Approximation

We interpret feature ablation as a small perturbation of the input representation.
Accordingly, we take a first-order Taylor expansion of the logit f_j around the input x itself .

$$f_j(z) \approx f_j(x) + \sum_{k=1}^{d} \left.\frac{\partial f_j}{\partial x_k}\right|_x (z_k - x_k)$$

where z denotes a nearby input. Setting $z = x^{(i \leftarrow b_i)}$, which differs from x only in coordinate i, yields

$$f_j(x^{(i \leftarrow b_i)}) \approx f_j(x) - \left.\frac{\partial f_j}{\partial x_i}\right|_x (x_i - b_i)$$

Rearranging, the change in the logit induced by ablating feature i is approximated by

$$\mathscr{R}(x_i) = f_j(x) - f_j(x^{(i \leftarrow b_i)}) \approx \left.\frac{\partial f_j}{\partial x_i}\right|_x (x_i - b_i)$$

First-Order Taylor Approximation

We interpret feature ablation as a small perturbation of the input representation.
Accordingly, we take a first-order Taylor expansion of the logit f_j around the input x itself .

$$f_j(z) \approx f_j(x) + \sum_{k=1}^{d} \left.\frac{\partial f_j}{\partial x_k}\right|_x (z_k - x_k)$$

where z denotes a nearby input. Setting $z = x^{(i \leftarrow b_i)}$, which differs from x only in coordinate i, yields

$$f_j(x^{(i \leftarrow b_i)}) \approx f_j(x) - \left.\frac{\partial f_j}{\partial x_i}\right|_x (x_i - b_i)$$

Rearranging, the change in the logit induced by ablating feature i is approximated by

$$\mathscr{R}(x_i) = f_j(x) - f_j(x^{(i \leftarrow b_i)}) \approx \left.\frac{\partial f_j}{\partial x_i}\right|_x (x_i - b_i)$$

-> Open(Jupyter_Notebook)

First-Order Taylor Approximation: Gradient Shattering

However, the gradient does not faithfully reflect the model's true signal and is instead contaminated by a substantial noise component.

$$\left.\frac{\partial f_j}{\partial x_i}\right|_x = \left.\frac{\partial f_j^{\text{signal}}}{\partial x_i}\right|_x + \left.\frac{\partial f_j^{\text{noise}}}{\partial x_i}\right|_x$$

First-Order Taylor Approximation: Gradient Shattering

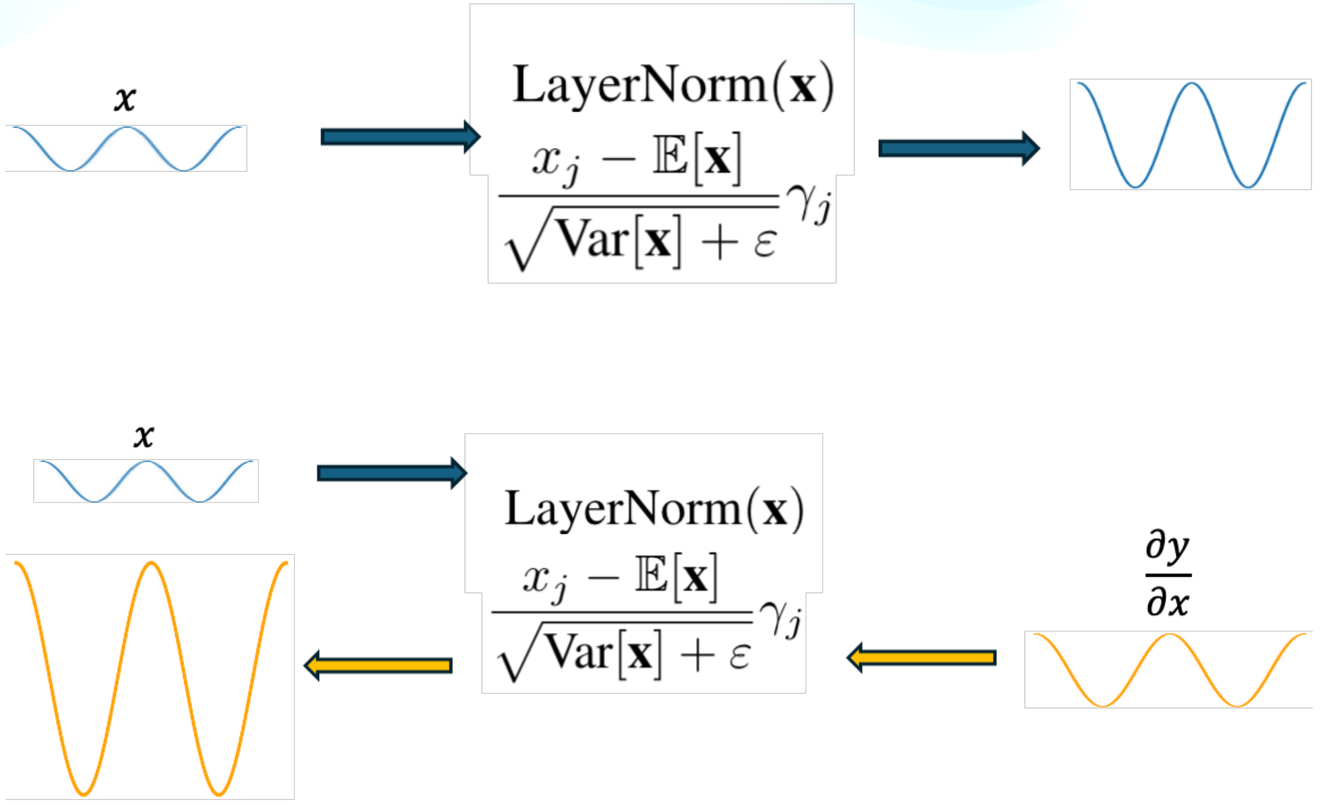
However, the gradient does not faithfully reflect the model's true signal and is instead contaminated by a substantial noise component.

$$\left.\frac{\partial f_j}{\partial x_i}\right|_x = \left.\frac{\partial f_j^{\text{signal}}}{\partial x_i}\right|_x + \left.\frac{\partial f_j^{\text{noise}}}{\partial x_i}\right|_x$$

-> Open(Jupyter_Notebook)

LayerNorm

LayerNorm scales the gradient inversely to the magnitude of the input x



LayerNo

LayerNorm s

f the input x

Understanding and Improving Layer Normalization

Jingjing Xu[1], Xu Sun[1,2]*, Zhiyuan Zhang[1], Guangxiang Zhao[2], Junyang Lin[1]
[1] MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[2] Center for Data Science, Peking University
{jingjingxu,xusun,zzy1210,zhaoguangxiang,linjunyang}@pku.edu.cn

Abstract

Layer normalization (LayerNorm) is a technique to normalize the distributions of intermediate layers. It enables smoother gradients, faster training, and better generalization accuracy. However, it is still unclear where the effectiveness stems from. In this paper, our main contribution is to take a step further in understanding LayerNorm. Many of previous studies believe that the success of LayerNorm comes from forward normalization. Unlike them, we find that the derivatives of the mean and variance are more important than forward normalization by re-centering and re-scaling backward gradients. Furthermore, we find that the parameters of LayerNorm, including the bias and gain, increase the risk of over-fitting and do not work in most cases. Experiments show that a simple version of LayerNorm (LayerNorm-simple) without the bias and gain outperforms LayerNorm on four datasets. It obtains the state-of-the-art performance on En-Vi machine translation. To address the over-fitting problem, we propose a new normalization method, Adaptive Normalization (AdaNorm), by replacing the bias and gain with a new transformation function. Experiments show that AdaNorm demonstrates better results than LayerNorm on seven out of eight datasets.

Layer-wise Relevance Propagation

The raw gradient is noisy due to:

1. **LayerNorm:** stabilises training but distorts the gradient by introducing **higher-order effects** absent in a first-order Taylor approximation.
2. **Query–Key:** gradients flow through both query and key, effectively **doubling the gradient** and causing explosion.
3. **Softmax-Value:** like query–key multiplication
4. **SiLU:** the nonlinearity further introduces **higher-order effects**.

LRP fixes this by propagating modified gradients:

$$\frac{\partial f_j}{\partial x} = \frac{\partial f_j}{\partial v} \cdot \frac{\partial v}{\partial u} \cdot \frac{\partial u}{\partial x} \longrightarrow \left[\frac{\partial f_j}{\partial x}\right]_{\text{LRP}} = \left[\frac{\partial f_j}{\partial v}\right]_{\text{LRP}} \cdot \left[\frac{\partial v}{\partial u}\right]_{\text{LRP}} \cdot \left[\frac{\partial u}{\partial x}\right]_{\text{LRP}}.$$

Layer-wise Relevance Propagation

The raw gradient is n

1. **LayerNorm:** stab
2. **Query–Key:** grad
3. **Softmax-Value:** l
4. **SiLU:** the nonline

rder Taylor approximation.

ion.

RP.

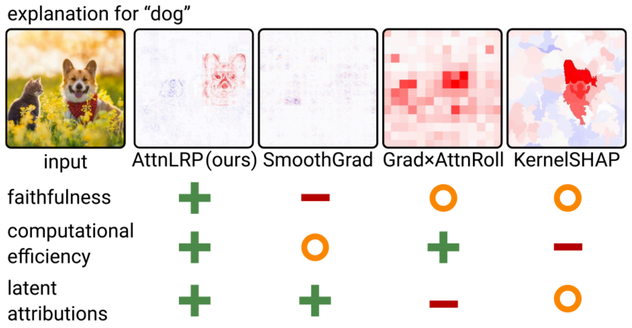
AttnLRP: Attention-Aware Layer-Wise Relevance Propagation for Transformers

Reduan Achtibat[1] Sayed Mohammad Vakilzadeh Hatefi[1] Maximilian Dreyer[1]
Aakriti Jain[1] Thomas Wiegand[1,2,3] Sebastian Lapuschkin[1,†] Wojciech Samek[1,2,3,†]

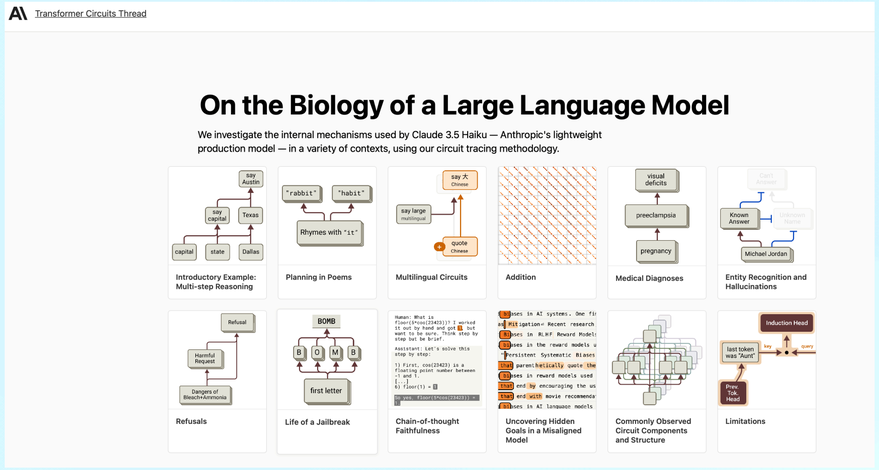
[1] Fraunhofer Heinrich-Hertz-Institute, 10587 Berlin, Germany
[2] Technische Universität Berlin, 10587 Berlin, Germany
[3] BIFOLD – Berlin Institute for the Foundations of Learning and Data, 10587 Berlin, Germany
[†] corresponding authors: {wojciech.samek,sebastian.lapuschkin}@hhi.fraunhofer.de

Abstract

Large Language Models are prone to biased predictions and hallucinations, underlining the paramount importance of understanding their model-internal reasoning process. However, achieving faithful attributions for the entirety of a black-box transformer model and maintaining computational efficiency is an unsolved challenge. By extending the Layer-wise Relevance Propagation attribution method to handle atten-

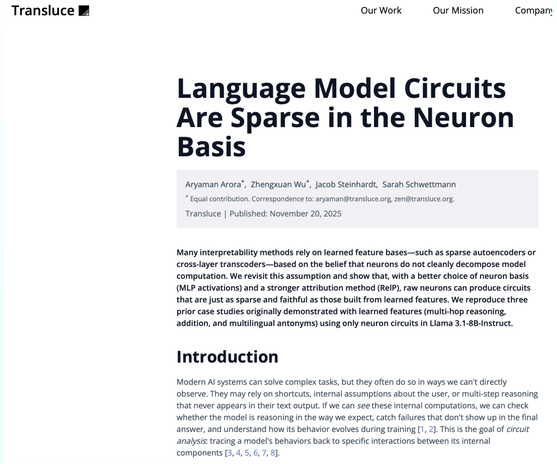


State-of-the-Art

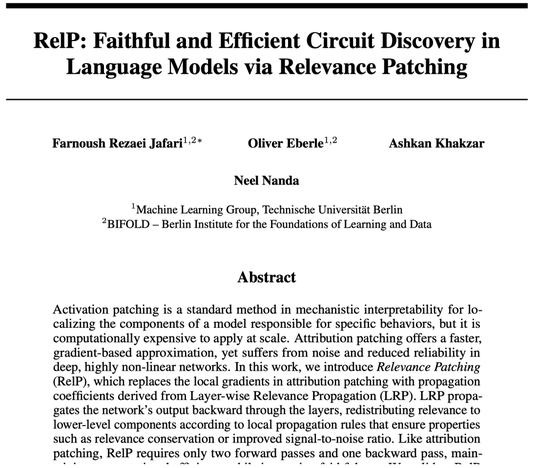


https://transformer-circuits.pub/2025/attribution-graphs/biology.html

https://alignment.openai.com/sae-latent-attribution/?s=09



https://transluce.org/neuron-circuits



https://arxiv.org/pdf/2508.21258

Layer-wise Relevance Propagation

The raw gradient is noisy due to:

1. **LayerNorm:** stabilises training but distorts the gradient by introducing **higher-order effects** absent in a first-order Taylor approximation.
2. **Query–Key:** gradients flow through both query and key, effectively **doubling the gradient** and causing explosion.
3. **Softmax-Value:** like query–key multiplication
4. **SiLU:** the nonlinearity further introduces **higher-order effects**.

LRP fixes this by propagating modified gradients:

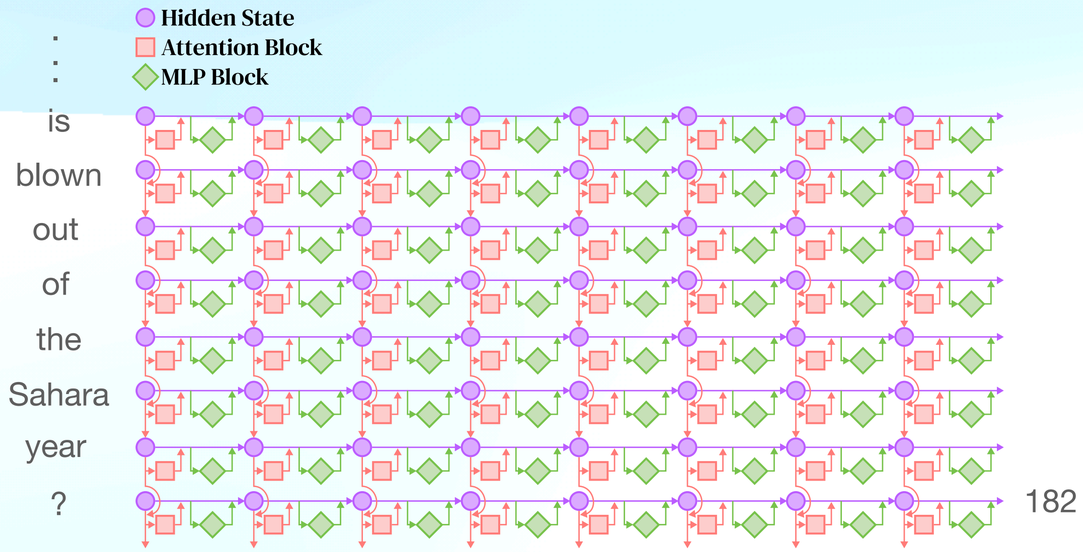
$$\frac{\partial f_j}{\partial x} = \frac{\partial f_j}{\partial v} \cdot \frac{\partial v}{\partial u} \cdot \frac{\partial u}{\partial x} \longrightarrow \left[\frac{\partial f_j}{\partial x}\right]_{\text{LRP}} = \left[\frac{\partial f_j}{\partial v}\right]_{\text{LRP}} \cdot \left[\frac{\partial v}{\partial u}\right]_{\text{LRP}} \cdot \left[\frac{\partial u}{\partial x}\right]_{\text{LRP}}.$$

-> Open(Jupyter_Notebook)

Patching Attention Heads

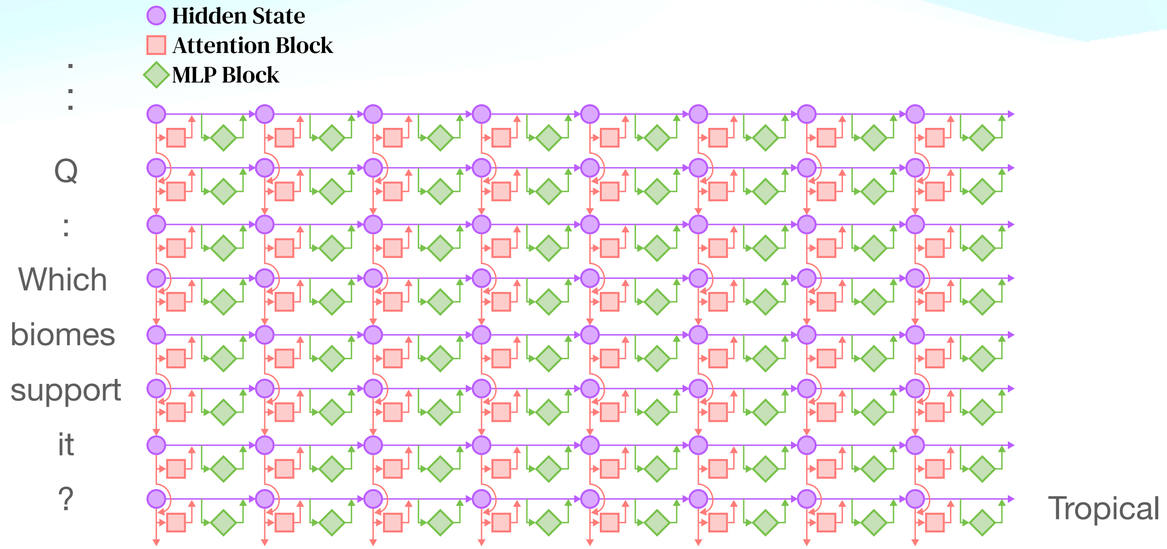
NASA's CALIPSO satellite measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?



The Atlantic Ocean is the second-largest ocean on Earth. It holds about 310 million cubic kilometers of water.
It spans a wide range of marine biomes, from warm tropical coral reef systems near the equator to cold, nutrient-rich polar waters in the north and south. These biomes support diverse life, including plankton, fish, whales, and deep-sea organisms, and play a major role in regulating Earth's climate through ocean currents like the Gulf Stream.

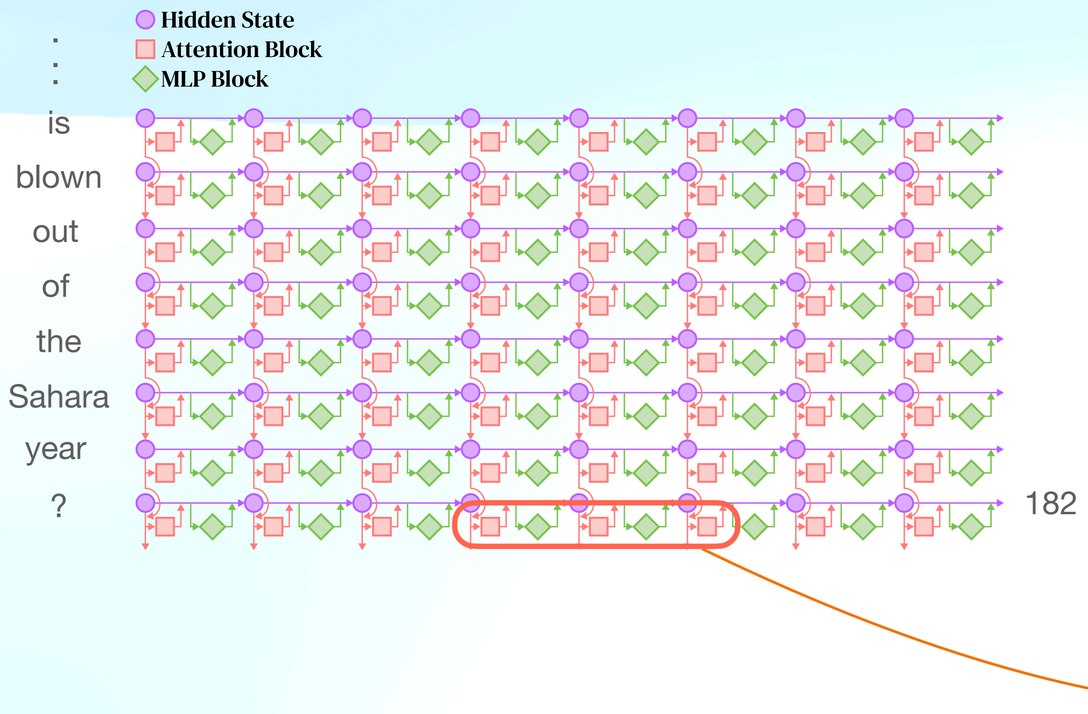
Q: Which biomes does it support?



Patching Attention Heads

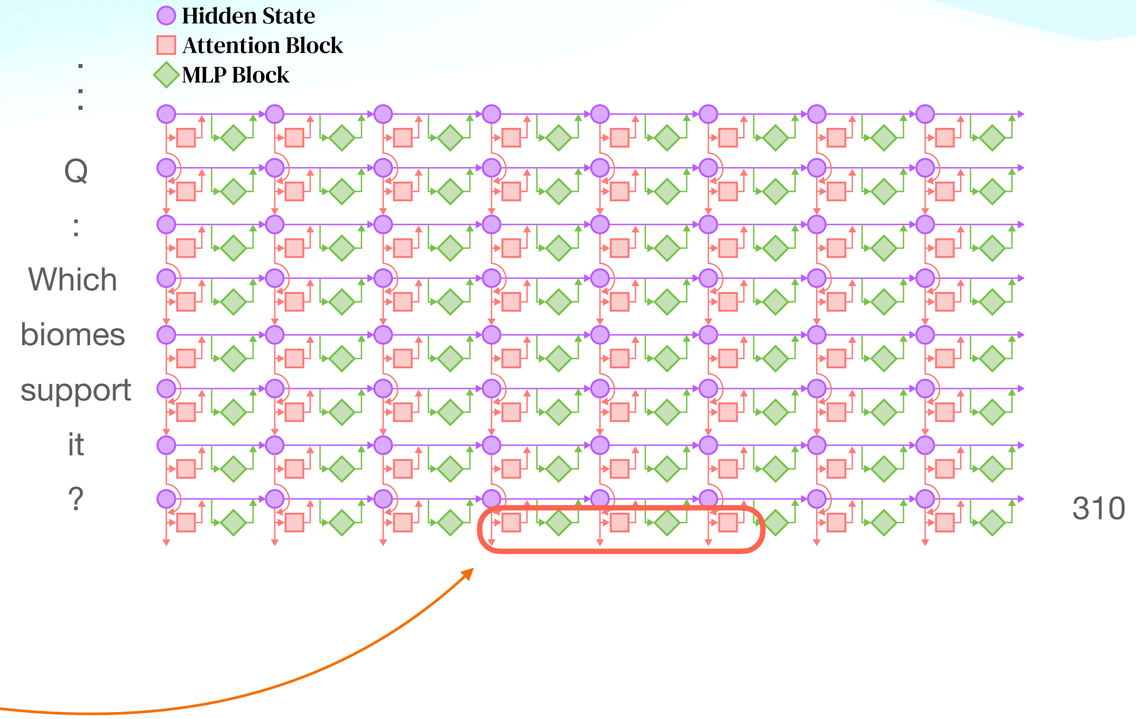
NASA's CALIPSO satellite measured 182 million tons of Sahara dust annually, but recent studies suggest it's actually 27 million tons.

Q: How much dust is blown out of the Sahara each year?

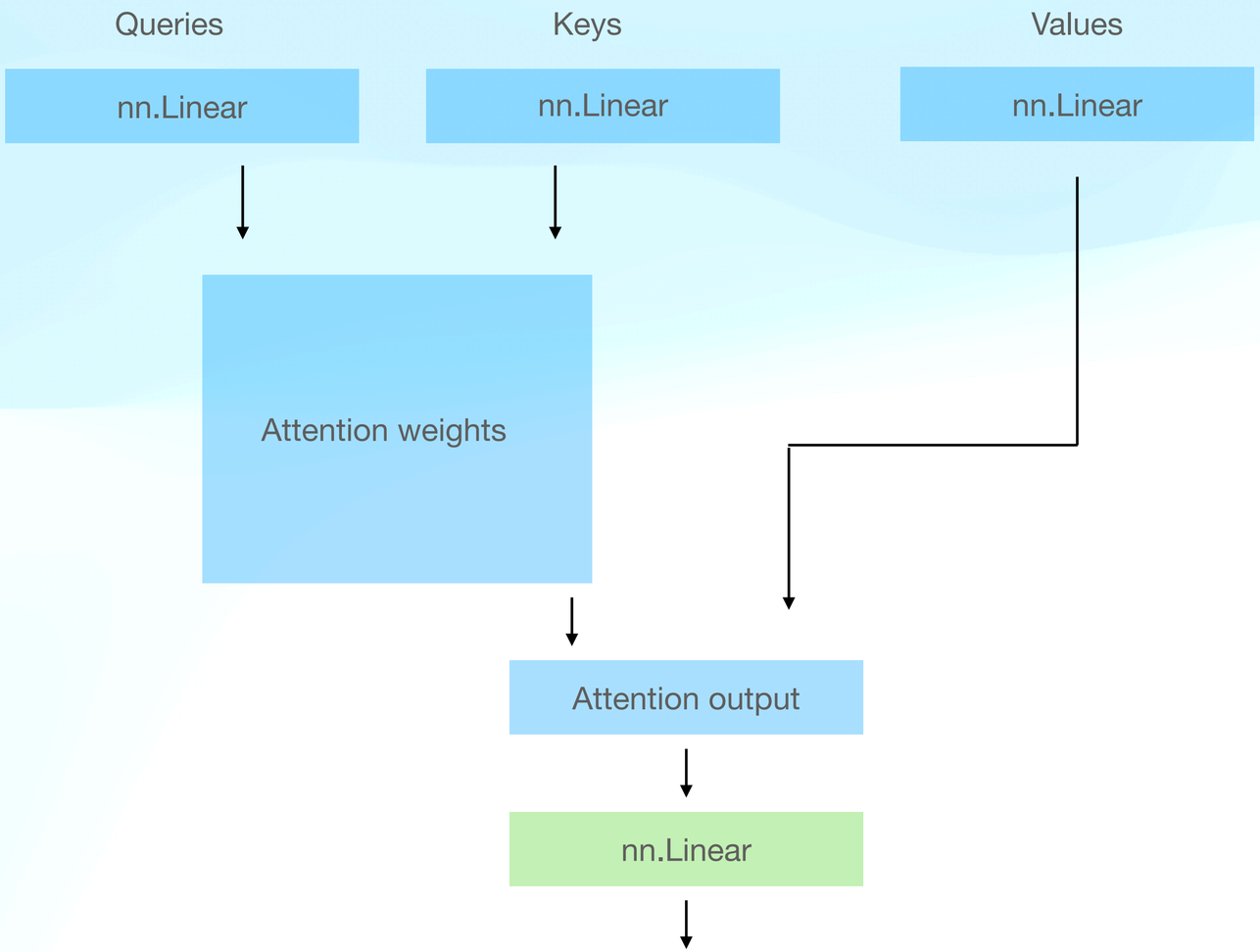


The Atlantic Ocean is the second-largest ocean on Earth. It holds about 310 million cubic kilometers of water.
It spans a wide range of marine biomes, from warm tropical coral reef systems near the equator to cold, nutrient-rich polar waters in the north and south. These biomes support diverse life, including plankton, fish, whales, and deep-sea organisms, and play a major role in regulating Earth's climate through ocean currents like the Gulf Stream.

Q: Which biomes does it support?



Patching Attention Heads



Patching Attention Heads

TOWARDS BEST PRACTICES OF ACTIVATION PATCHING IN LANGUAGE MODELS: METRICS AND METHODS

Fred Zhang*
UC Berkeley
z0@berkeley.edu

Neel Nanda
Independent
neelnanda27@gmail.com

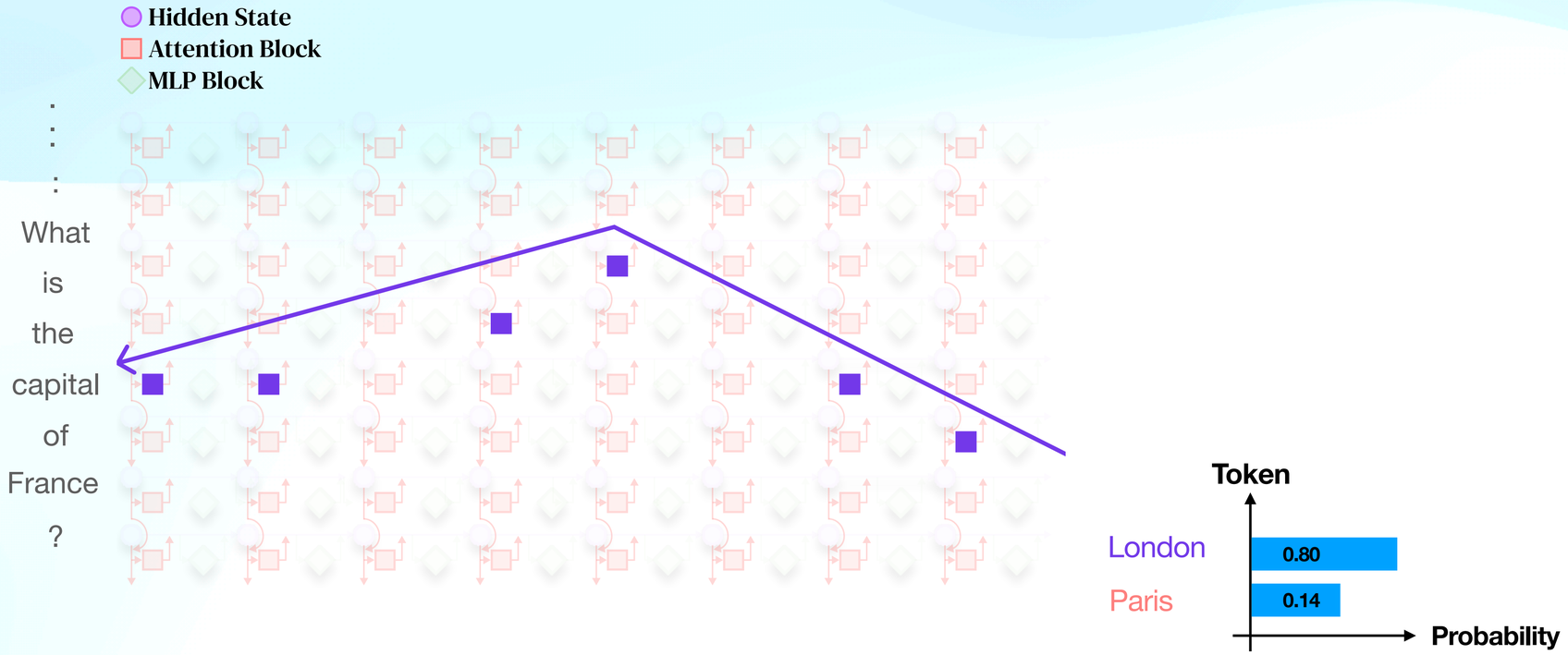
ABSTRACT

Mechanistic interpretability seeks to understand the internal mechanisms of machine learning models, where localization—identifying the important model components—is a key step. Activation patching, also known as causal tracing or interchange intervention, is a standard technique for this task (Vig et al., 2020), but the literature contains many variants with little consensus on the choice of hyperparameters or methodology. In this work, we systematically examine the impact of methodological details in activation patching, including evaluation metrics and corruption methods. In several settings of localization and circuit discovery in language models, we find that varying these hyperparameters could lead to disparate interpretability results. Backed by empirical observations, we give conceptual arguments for why certain metrics or methods may be preferred. Finally, we provide recommendations for the best practices of activation patching going forwards.

Contrastive Explanation of Knowledge Conflicts

The capital of France is London.

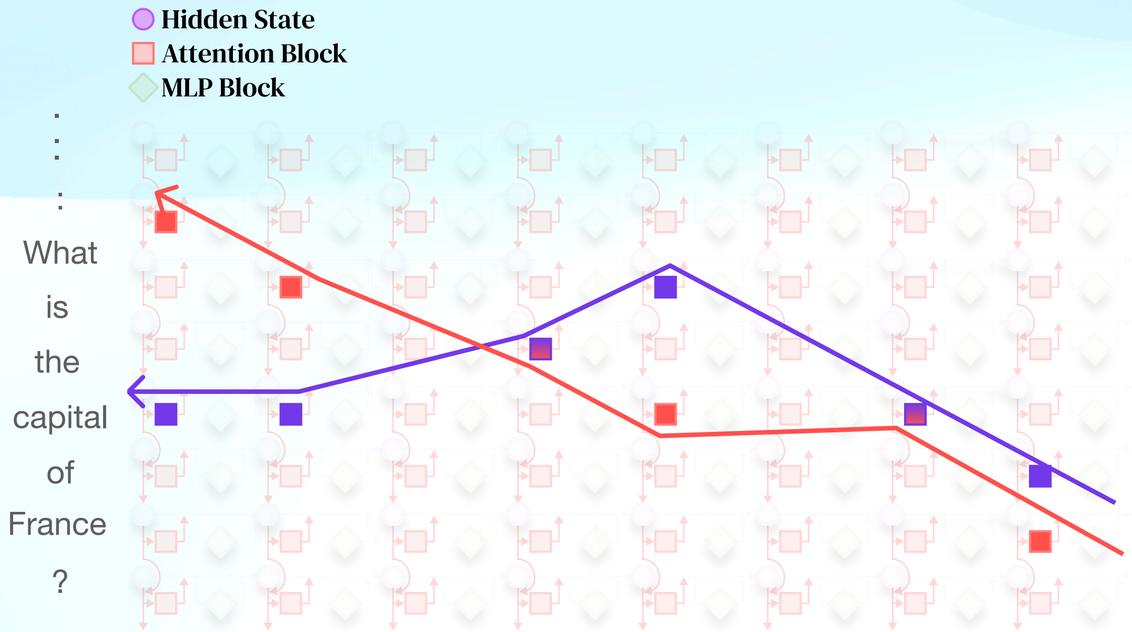
Q: What is the capital of France?

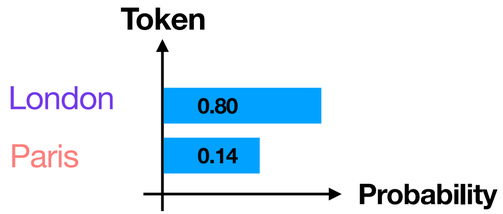


Contrastive Explanation of Knowledge Conflicts

The capital of France is London.

Q: What is the capital of France?

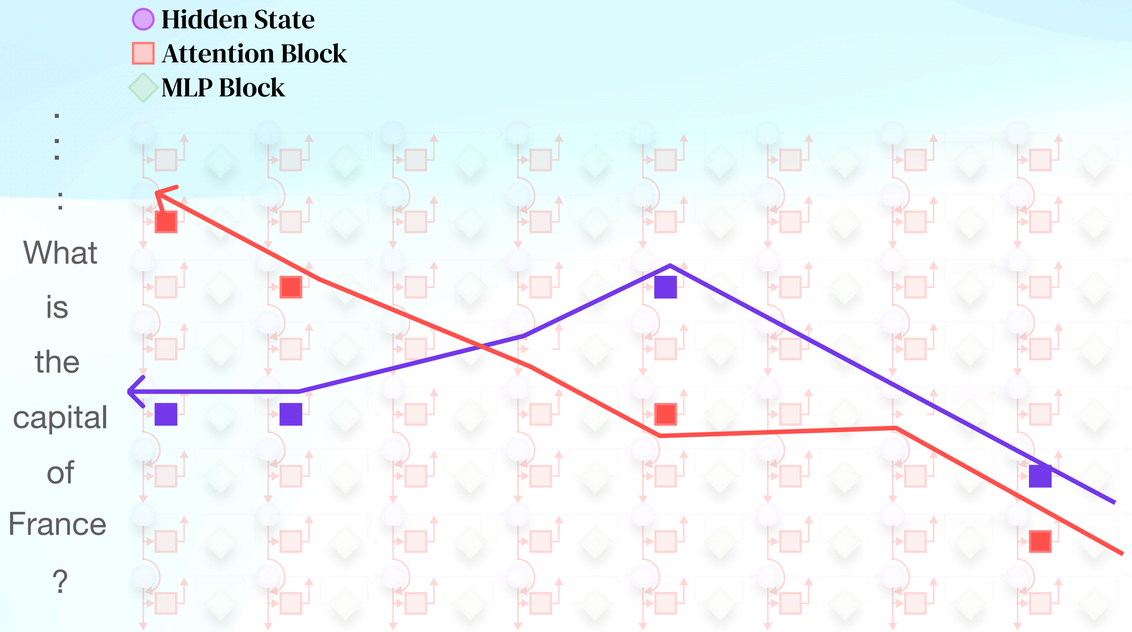


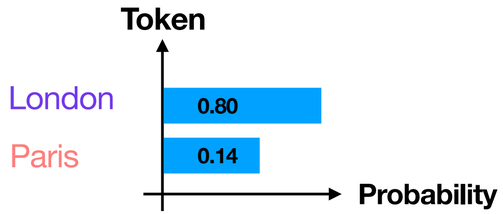


Contrastive Explanation of Knowledge Conflicts

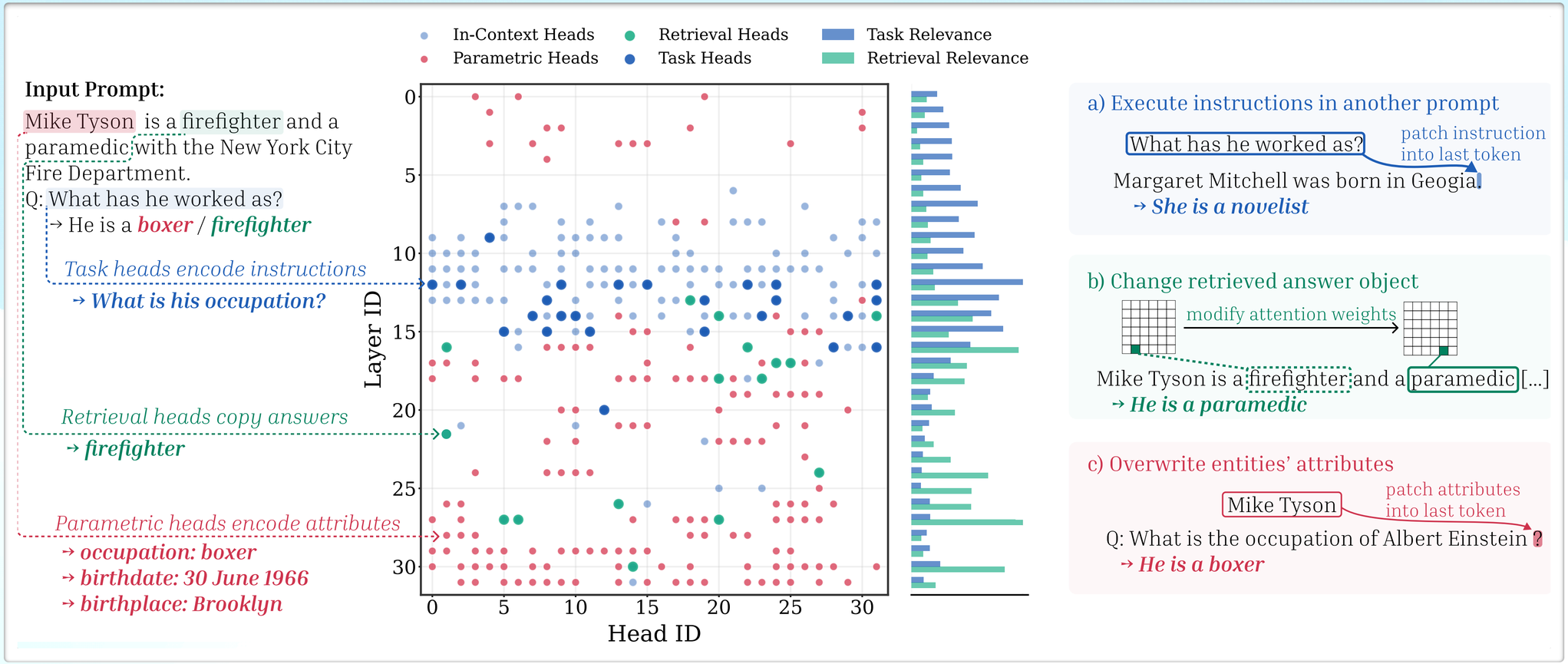
The capital of France is London.

Q: What is the capital of France?

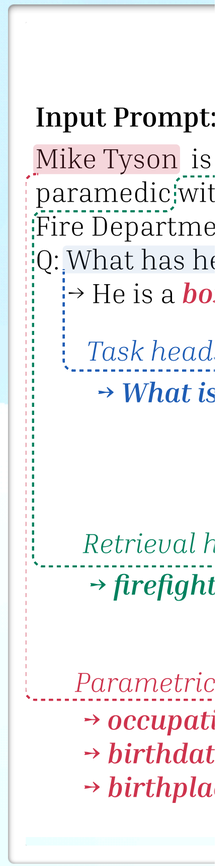




A universal map of language models?



A universal map of language models?



The Atlas of In-Context Learning: How Attention Heads Shape In-Context Retrieval Augmentation

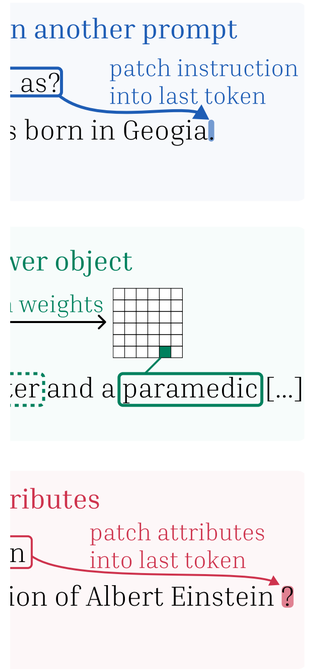
Patrick Kahardipraja[1,*] **Reduan Achtibat**[1,*] **Thomas Wiegand**[1,2,3]

Wojciech Samek[1,2,3,†] **Sebastian Lapuschkin**[1,4,†]

[1]Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute
[2]Department of Electrical Engineering and Computer Science, Technische Universität Berlin
[3]BIFOLD - Berlin Institute for the Foundations of Learning and Data
[4]Centre of eXplainable Artificial Intelligence, Technological University Dublin
{firstname.lastname}@hhi.fraunhofer.de

Abstract

Large language models are able to exploit in-context learning to access external knowledge beyond their training data through retrieval-augmentation. While promising, its inner workings remain unclear. In this work, we shed light on the mechanism of in-context retrieval augmentation for question answering by viewing a prompt as a composition of informational components. We propose an attribution-



WE'D LOVE YOUR FEEDBACK!

BLISS × dida