Berkeley
UNIVERSITY OF CALIFORNIA

AI Interpretability

AKSHAT GUPTA

GUEST LECTURE, CS 188 UC Berkeley

April 29, 2025

About Me

AKSHAT GUPTA

PHD STUDENT, UC BERKELEY (2023 - PRESENT)

- AI RESEARCH ENGINEER, JPMorgan Chase, NY (2021-2023)
- MS ECE, Carnegie Mellon University (2020-2021)
- MS PHYSICS, Technical University of Munich (2017-2019)
- BTECH EE, Indian Institute of Technology Mandi (2013-2017)

ADVISOR: Gopala Anumanchipalli
AFFILIATIONS : Berkeley Speech Group, BAIR

Research Areas: Interpretability, Knowledge Editing, Reasoning and Poker

WHY INTERPRETABILITY ?

WHY INTERPRETABILITY?

WHY INTERPRETABILITY?

WHY INTERPRETABILITY?

1. Understand AI decision making process

WHY INTERPRETABILITY?

1. Understand AI decision making process

2. Predict and prevent unintended behaviors

WHY INTERPRETABILITY?

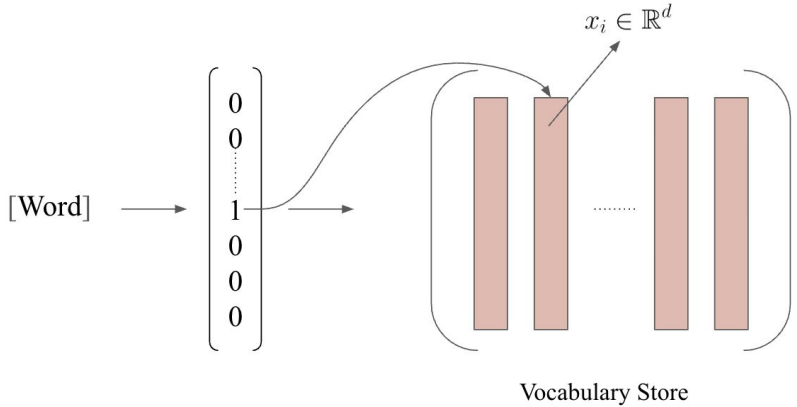
1. Understand AI decision making process
2. Predict and prevent unintended behaviors
3. Scientific curiosity

WHY INTERPRETABILITY?

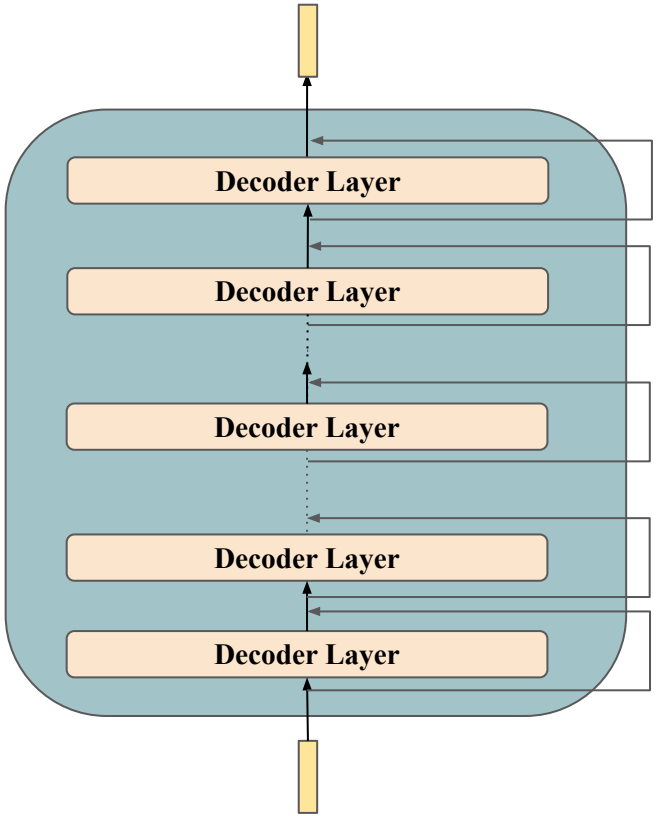
1. Understand AI decision making process
2. Predict and prevent unintended behaviors
3. Scientific curiosity
4. Improve models!

RECAP: THE TRANSFORMER ARCHITECTURE

RECAP : THE TRANSFORMER ARCHITECTURE

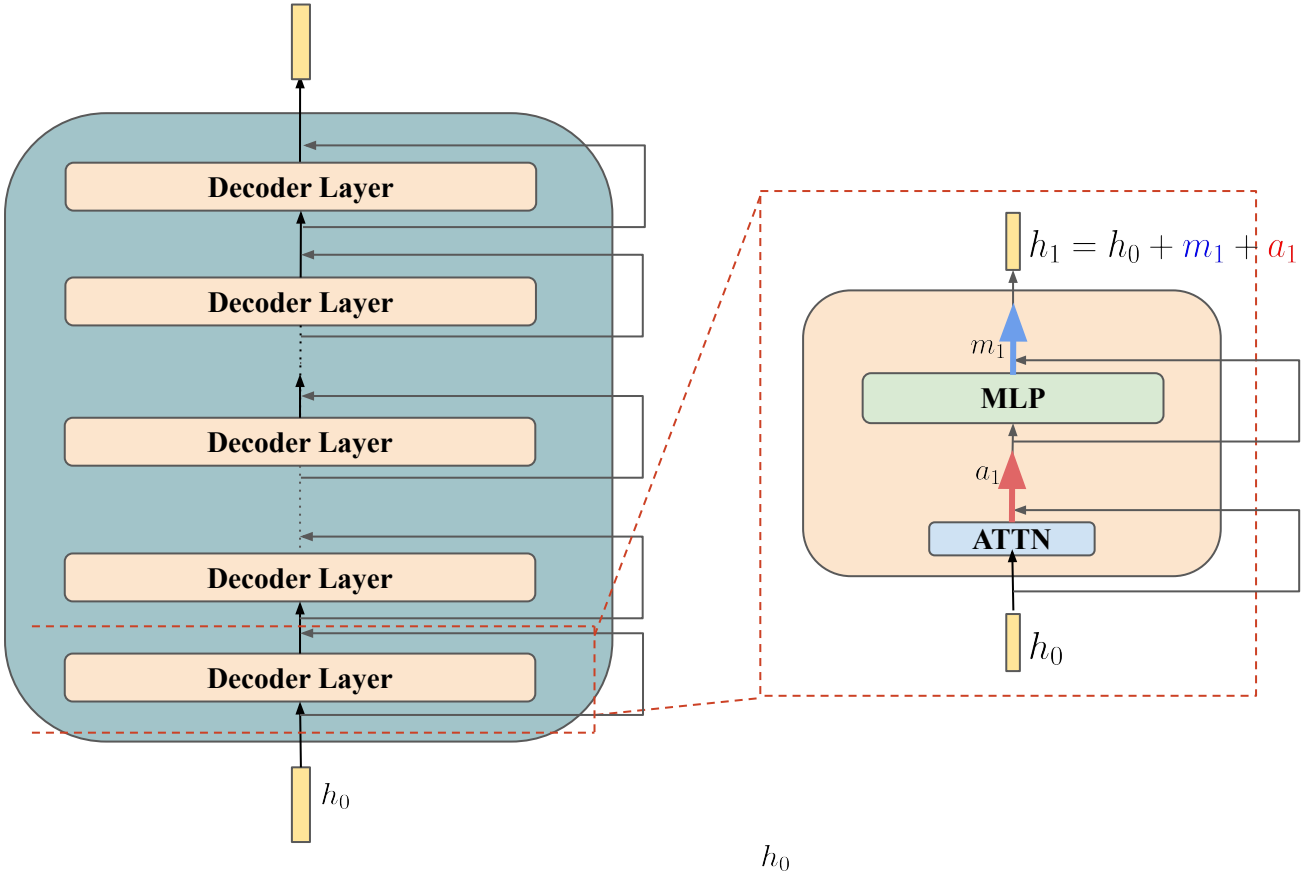


RECAP : THE TRANSFORMER ARCHITECTURE

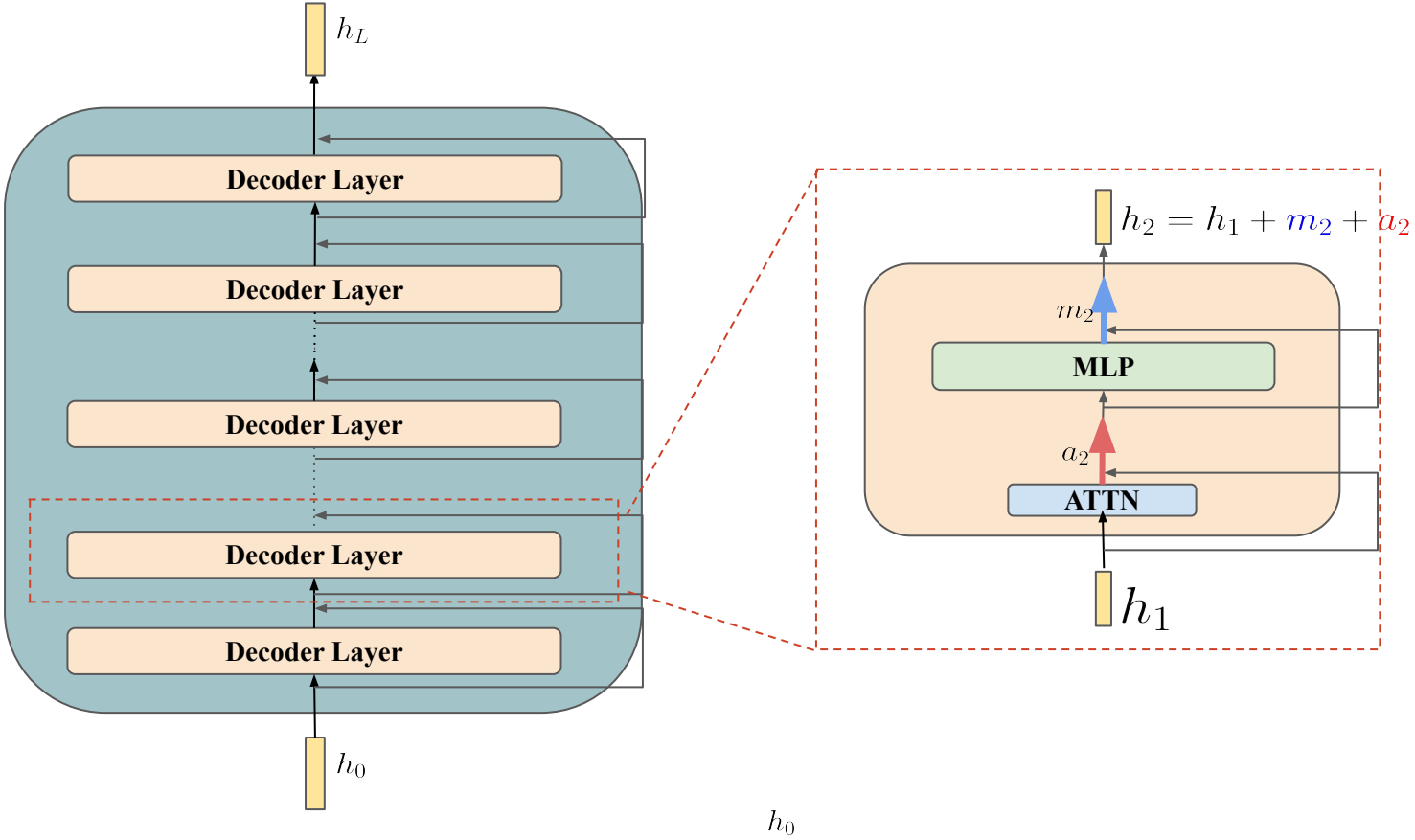


h_0

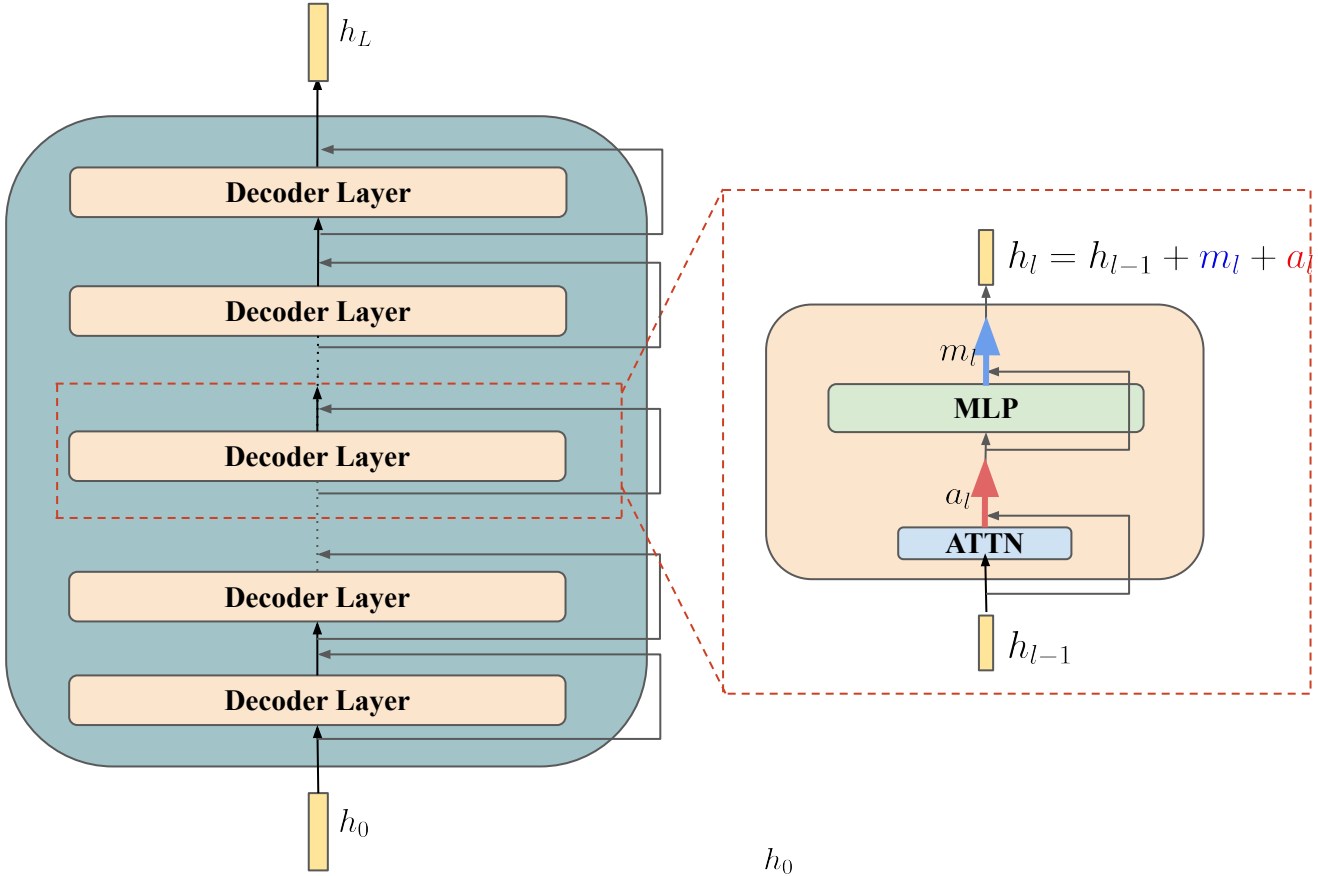
RECAP : THE TRANSFORMER ARCHITECTURE



RECAP : THE TRANSFORMER ARCHITECTURE

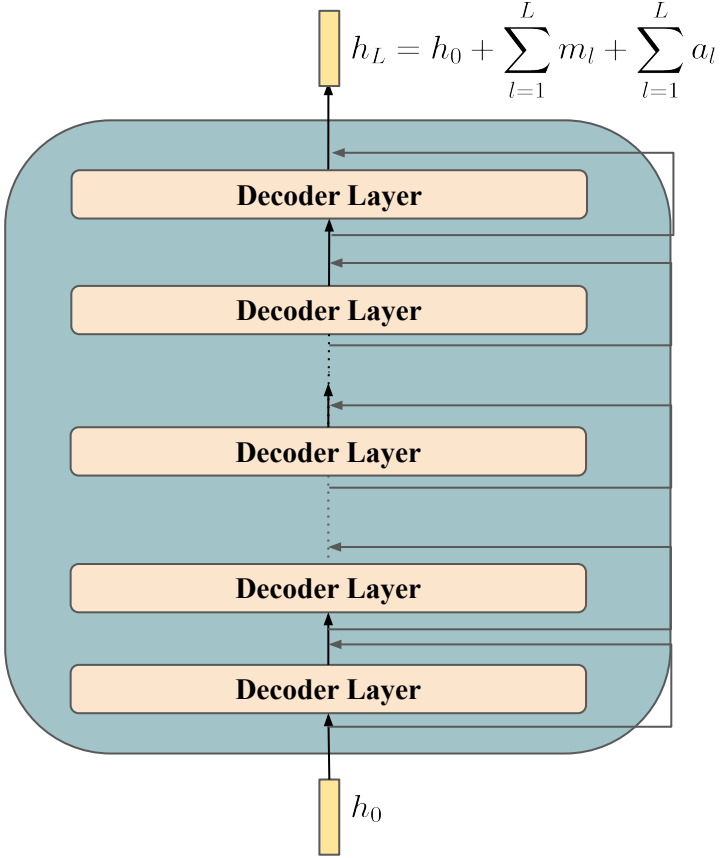


RECAP : THE TRANSFORMER ARCHITECTURE

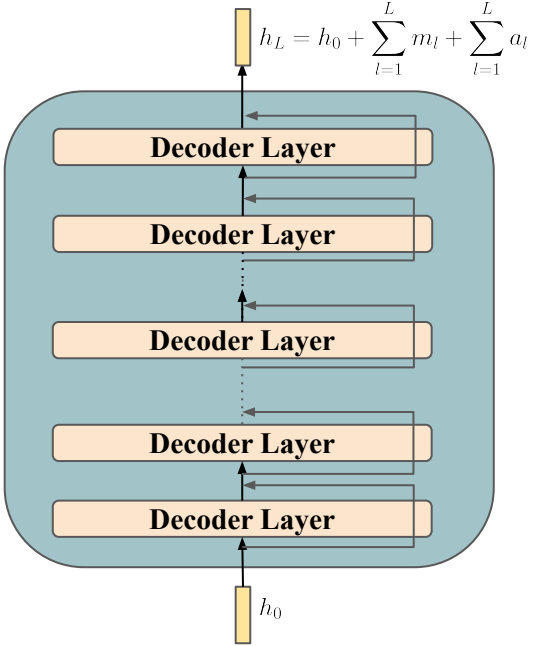


h_0

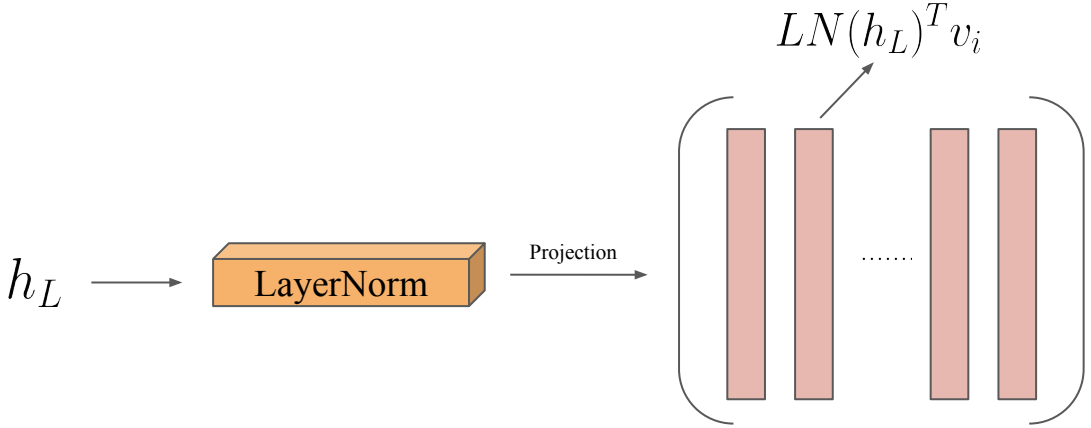
RECAP : THE TRANSFORMER ARCHITECTURE



RECAP : THE TRANSFORMER ARCHITECTURE



Pick the Largest Dot Product

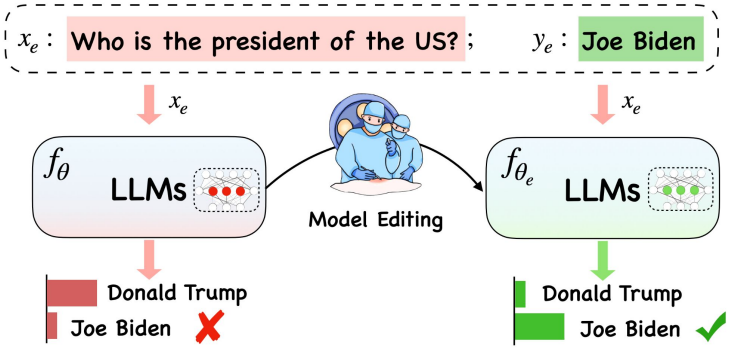


Vocabulary Store

PROJECT - 1: KNOWLEDGE EDITING

What does it mean to “Edit” a model?

- “Editing” is usually refers to “targeted” updates made to LLMs
- In “knowledge editing”, we usually do the following operations:
  - Correct incorrect facts
  - Update obsolete facts
  - Add new facts
  - Remove incorrect/sensitive/private information (unlearning)



Why “Edit” a model?

- To update stale information
- To delete sensitive/private information (safety/privacy)
- Enhanced model interpretability
- Continual learning

Evaluating Model Editing

Reliability Previous works (Huang et al., 2023; De Cao et al., 2021; Meng et al., 2022) define a reliable edit when the post-edit model f_{θ_e} gives the target answer for the case (x_e, y_e) to be edited. The reliability is measured as the average accuracy on the edit case:

$$\mathbb{E}_{x'_{\mathrm{e}}, y'_{\mathrm{e}} \sim \{(x_{\mathrm{e}}, y_{\mathrm{e}})\}} \mathbb{1}\left\{\operatorname{argmax}_y f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right) = y'_{\mathrm{e}}\right\} \tag{2}$$

Generalization The post-edit model f_{θ_e} should also edit the equivalent neighbour $N\left(x_{\mathrm{e}}, y_{\mathrm{e}}\right)$ (e.g. rephrased sentences). It is evaluated by the average accuracy of the model f_{θ_e} on examples drawn uniformly from the equivalence neighborhood:

$$\mathbb{E}_{x'_{\mathrm{e}}, y'_{\mathrm{e}} \sim N(x_{\mathrm{e}}, y_{\mathrm{e}})} \mathbb{1}\left\{\operatorname{argmax}_y f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right) = y'_{\mathrm{e}}\right\} \tag{3}$$

Locality also noted as **Specificity** in some work. Editing should be implemented locally, which means the post-edit model f_{θ_e} should not change the output of the irrelevant examples in the out-of-scope $O(x_e, y_e)$. Hence, the locality is evaluated by the rate at which the post-edit model f_{θ_e}'s predictions are unchanged as the pre-edit f_θ model:

$$\mathbb{E}_{x'_{\mathrm{e}}, y'_{\mathrm{e}} \sim O(x_{\mathrm{e}}, y_{\mathrm{e}})} \mathbb{1}\left\{f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right) = f_\theta\left(y \mid x'_{\mathrm{e}}\right)\right\} \tag{4}$$

Evaluating Model Editing

Reliability Previous works (Huang et al., 2023; De Cao et al., 2021; Meng et al., 2022) define a reliable edit when the post-edit model f_{θ_e} gives the target answer for the case (x_e, y_e) to be edited. The reliability is measured as the average accuracy on the edit case:

$$\mathbb{E}_{x'_e, y'_e \sim \{(x_e, y_e)\}} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e} \left(y \mid x'_e \right) = y'_e \right\} \quad (2)$$

Was the edit successful?

Generalization The post-edit model f_{θ_e} should also edit the equivalent neighbour $N\left(x_e, y_e\right)$ (e.g. rephrased sentences). It is evaluated by the average accuracy of the model f_{θ_e} on examples drawn uniformly from the equivalence neighborhood:

$$\mathbb{E}_{x'_e, y'_e \sim N(x_e, y_e)} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e} \left(y \mid x'_e \right) = y'_e \right\} \quad (3)$$

Locality also noted as **Specificity** in some work. Editing should be implemented locally, which means the post-edit model f_{θ_e} should not change the output of the irrelevant examples in the out-of-scope $O(x_e, y_e)$. Hence, the locality is evaluated by the rate at which the post-edit model f_{θ_e}'s predictions are unchanged as the pre-edit f_θ model:

$$\mathbb{E}_{x'_e, y'_e \sim O(x_e, y_e)} \mathbb{1} \left\{ f_{\theta_e} \left(y \mid x'_e \right) = f_\theta \left(y \mid x'_e \right) \right\} \quad (4)$$

Evaluating Model Editing

Reliability Previous works (Huang et al., 2023; De Cao et al., 2021; Meng et al., 2022) define a reliable edit when the post-edit model f_{θ_e} gives the target answer for the case (x_e, y_e) to be edited. The reliability is measured as the average accuracy on the edit case:

$$\mathbb{E}_{x'_e, y'_e \sim \{(x_e, y_e)\}} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e} \left(y \mid x'_e \right) = y'_e \right\} \quad (2)$$

Generalization The post-edit model f_{θ_e} should also edit the equivalent neighbour $N\left(x_e, y_e\right)$ (e.g. rephrased sentences). It is evaluated by the average accuracy of the model f_{θ_e} on examples drawn uniformly from the equivalence neighborhood:

$$\mathbb{E}_{x'_e, y'_e \sim N(x_e, y_e)} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e} \left(y \mid x'_e \right) = y'_e \right\} \quad (3)$$

Does the edit generalize to different phrasings of the same question?

Locality also noted as **Specificity** in some work. Editing should be implemented locally, which means the post-edit model f_{θ_e} should not change the output of the irrelevant examples in the out-of-scope $O(x_e, y_e)$. Hence, the locality is evaluated by the rate at which the post-edit model f_{θ_e}'s predictions are unchanged as the pre-edit f_θ model:

$$\mathbb{E}_{x'_e, y'_e \sim O(x_e, y_e)} \mathbb{1} \left\{ f_{\theta_e} \left(y \mid x'_e \right) = f_\theta \left(y \mid x'_e \right) \right\} \quad (4)$$

Evaluating Model Editing

Reliability Previous works (Huang et al., 2023; De Cao et al., 2021; Meng et al., 2022) define a reliable edit when the post-edit model f_{θ_e} gives the target answer for the case (x_e, y_e) to be edited. The reliability is measured as the average accuracy on the edit case:

$$\mathbb{E}_{x'_e, y'_e \sim \{(x_e, y_e)\}} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e} \left(y \mid x'_e \right) = y'_e \right\} \quad (2)$$

Generalization The post-edit model f_{θ_e} should also edit the equivalent neighbour $N\left(x_e, y_e\right)$ (e.g. rephrased sentences). It is evaluated by the average accuracy of the model f_{θ_e} on examples drawn uniformly from the equivalence neighborhood:

$$\mathbb{E}_{x'_e, y'_e \sim N(x_e, y_e)} \mathbb{1} \left\{ \text{argmax}_y f_{\theta_e} \left(y \mid x'_e \right) = y'_e \right\} \quad (3)$$

Locality also noted as **Specificity** in some work. Editing should be implemented locally, which means the post-edit model f_{θ_e} should not change the output of the irrelevant examples in the out-of-scope $O(x_e, y_e)$. Hence, the locality is evaluated by the rate at which the post-edit model f_{θ_e}'s predictions are unchanged as the pre-edit f_θ model:

$$\mathbb{E}_{x'_e, y'_e \sim O(x_e, y_e)} \mathbb{1} \left\{ f_{\theta_e} \left(y \mid x'_e \right) = f_\theta \left(y \mid x'_e \right) \right\} \quad (4)$$

Does the edit effect other facts stored in the model?

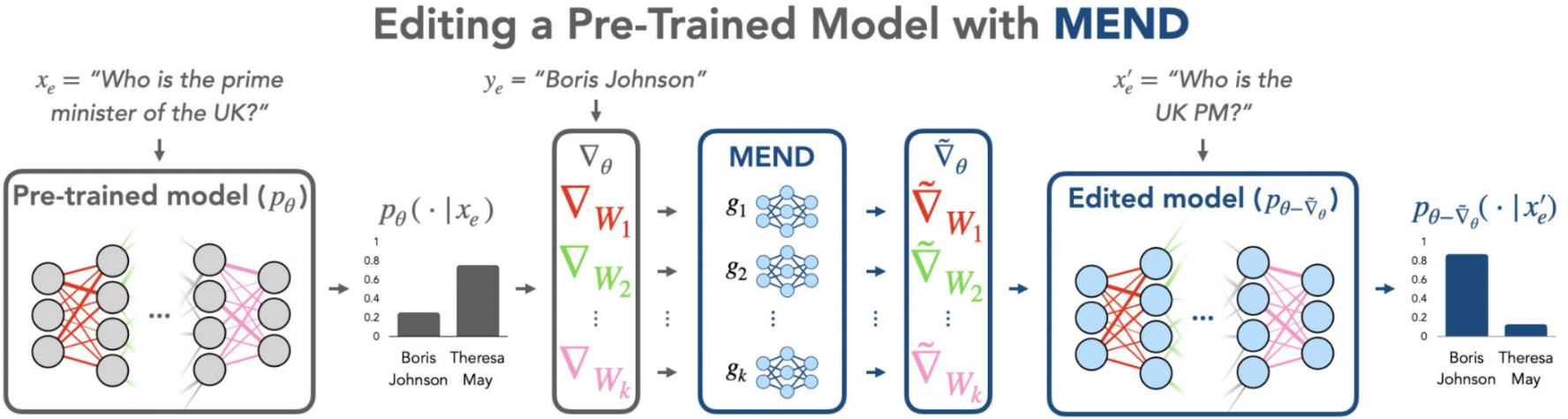
KNOWLEDGE EDITING : METHODS

Model Editing Methods

- TYPE-1 : Hypernetwork based Model Editing
- TYPE-2 : Locate-then-Edit Methods
- TYPE-3 : In-context Editing

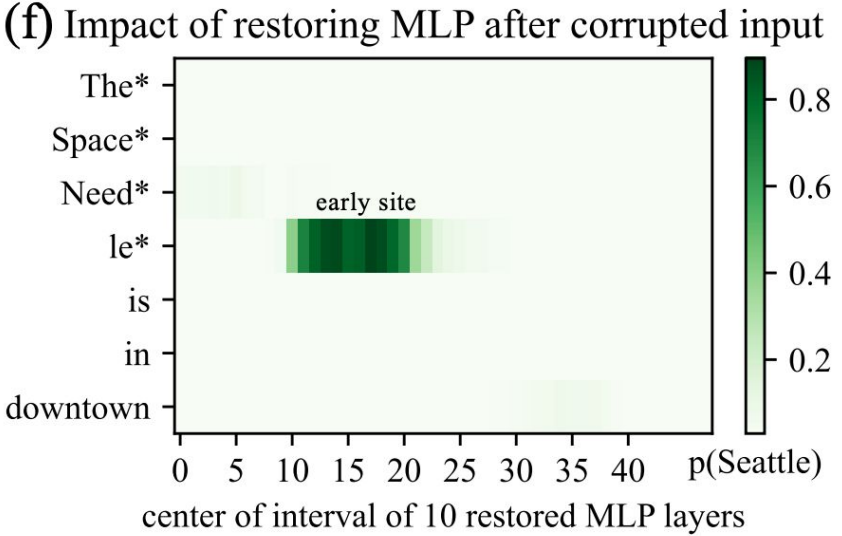
Paper 1 : Fast Model Editing at Scale (TYPE-1)

- Training a metamodel that outputs new weights of the model

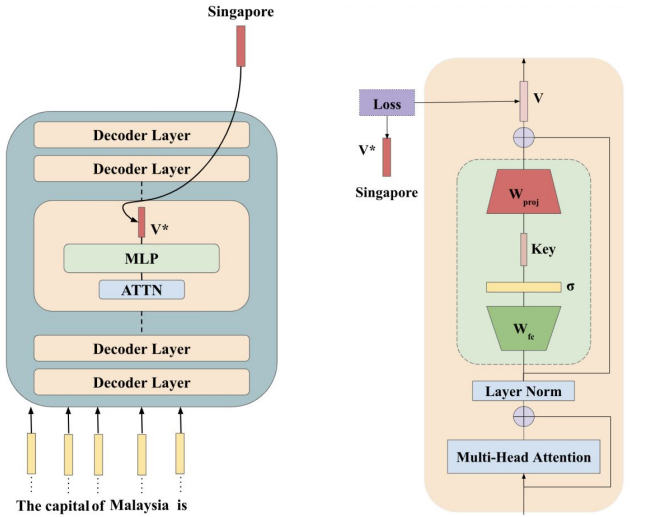


Paper 2 : Locating and Editing Factual Associations in GPT (TYPE-2)

- Some popular methods - ROME, MEMIT



Understanding Locate-Then-Edit Methods



(a) Gradient descent step which finds the target activations for the MLP matrix.

(b) Target activations are used to update the second MLP matrix (in red).

Figure 2. Presenting locate-then-edit knowledge editing methods as a two-step fine-tuning process.

"Lifelong Sequential Editing without Model Degradation", Gupta et al 2025

Understanding Locate-Then-Edit Methods

Facts type (s, r, o) → (Malaysia, capital, Singapore)

Editing Address:

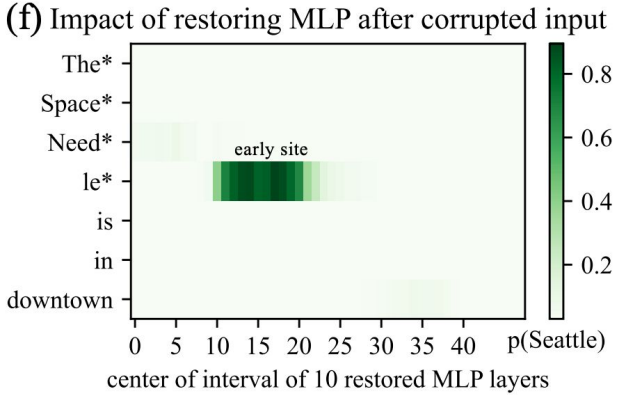
1. Layer that we want to modify
2. Token representation we use to modify knowledge

Understanding Locate-Then-Edit Methods

Facts type (s, r, o) → (Malaysia, capital, Singapore)

Editing Address:

1. Layer that we want to modify
2. Token representation we use to modify knowledge

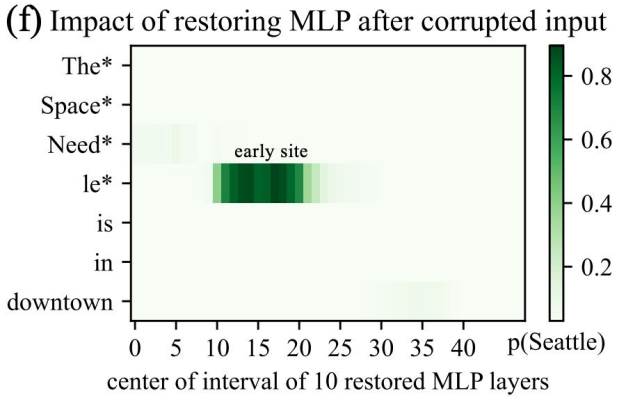


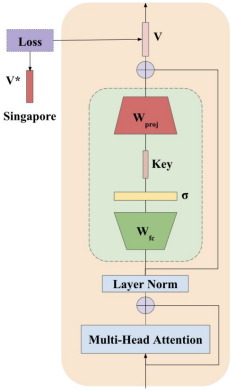
Understanding Locate-Then-Edit Methods

Facts type (s, r, o) → (Malaysia, capital, Singapore)

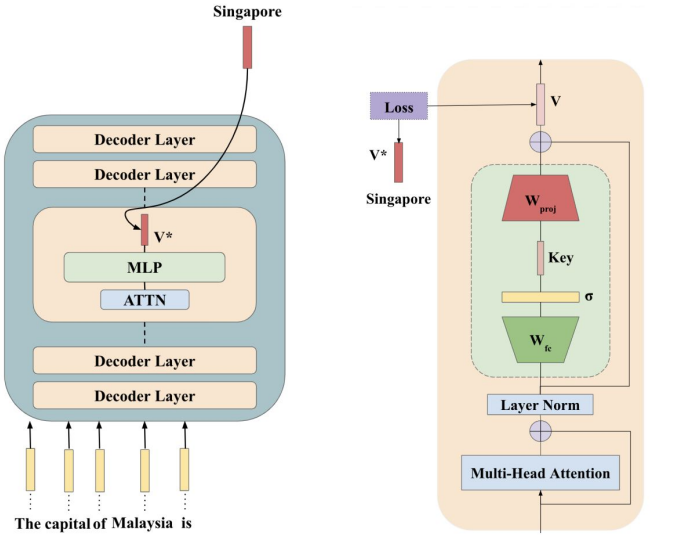
Editing Address:

1. Layer that we want to modify
2. Token representation we use to modify knowledge





Understanding Locate-Then-Edit Methods



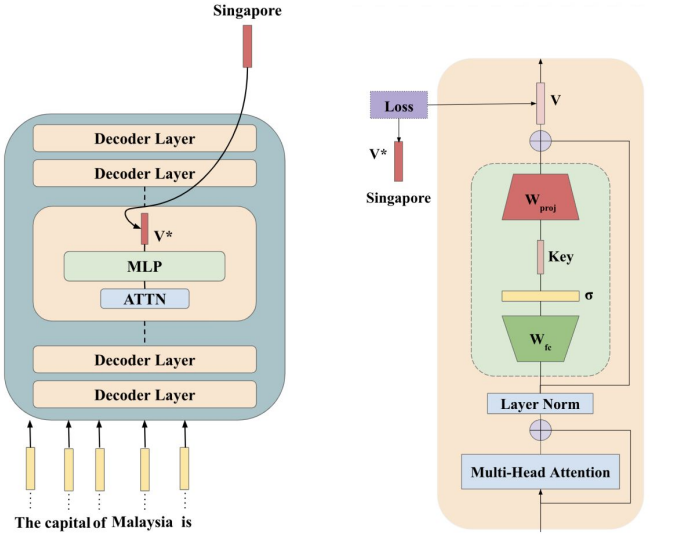
(a) Gradient descent step which finds the target activations for the MLP matrix.

(b) Target activations are used to update the second MLP matrix (in red).

Figure 2. Presenting locate-then-edit knowledge editing methods as a two-step fine-tuning process.

"Lifelong Sequential Editing without Model Degradation", Gupta et al 2025

Understanding Locate-Then-Edit Methods



(a) Gradient descent step which finds the target activations for the MLP matrix.

(b) Target activations are used to update the second MLP matrix (in red).

Figure 2. Presenting locate-then-edit knowledge editing methods as a two-step fine-tuning process.

$$\underset{\hat{W}}{\text{argmin}}\ L(\hat{W}) \quad \text{where}$$

$$L(\hat{W}) = \underbrace{\lambda \sum_{i=1}^{P} \left\| \hat{W} k_0^i - W_0 k_0^i \right\|_2^2}_{\text{preservation}} + \underbrace{\sum_{j=1}^{B} \left\| \hat{W} k_e^j - v_e^j \right\|_2^2}_{\text{memorization}}$$

"Lifelong Sequential Editing without Model Degradation", Gupta et al 2025

MY RESEARCH : SCALING KNOWLEDGE EDITING

Background

- Popular knowledge editing methods released between 2021-2023 performed well when making singular knowledge edits.
    - But are these methods scalable?
    - Can they be solutions for continually learning models?
    - What was the effect of continuous editing on the general ability of models?

Model Editing at Scale leads to Gradual and Catastrophic Forgetting

Akshat Gupta, Anurag Rao, Gopala Anumanchipalli
UC Berkeley
akshat.gupta@berkeley.edu

ACL 2024 (Findings)

AKSHAT GUPTA
PhD Student,
UC Berkeley

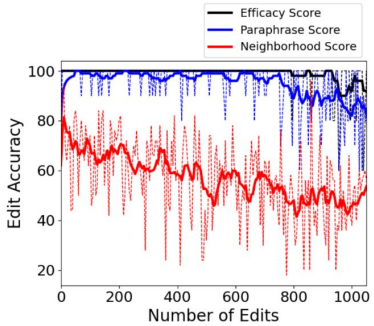
ANURAG RAO
UC Berkeley
(Now MS, Oxford)

GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

Model Editing at Scale leads to Gradual and Catastrophic Forgetting

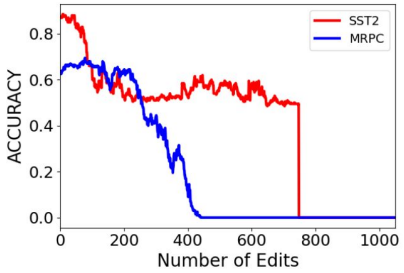
Akshat Gupta, Anurag Rao, Gopala Anumanchipalli
UC Berkeley
akshat.gupta@berkeley.edu

ACL 2024 (Findings)



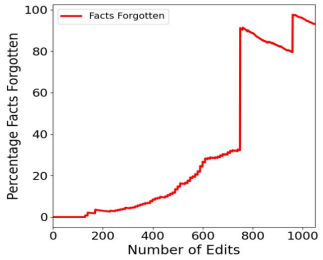
(a) Sample 1

Editability



(a) Sample 1

Downstream Performance



(a) Sample 1

Facts Forgotten

Rebuilding ROME : Resolving Model Collapse during Sequential Model Editing

Akshat Gupta[1], **Sidharth Baskaran**[2], **Gopala Anumanchipalli**[1]
[1]UC Berkeley, [2]Automorphic Inc.
akshat.gupta@berkeley.edu, sid@automorphic.ai

EMNLP 2024 Main

AKSHAT GUPTA
PhD Student,
UC Berkeley

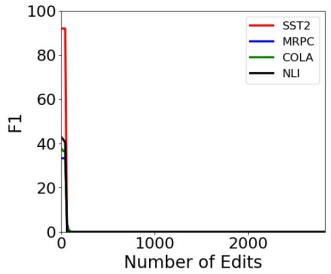
SIDHARTH BASKARAN
Automorphic Inc.

GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

Rebuilding ROME : Resolving Model Collapse during Sequential Model Editing

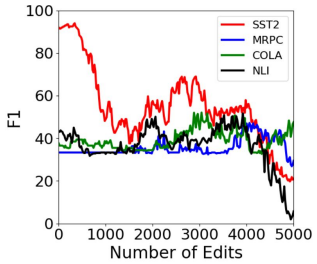
Akshat Gupta[1], **Sidharth Baskaran**[2], **Gopala Anumanchipalli**[1]
[1]UC Berkeley, [2]Automorphic Inc.
akshat.gupta@berkeley.edu, sid@automorphic.ai

EMNLP 2024 Main



(a) Downstream Evaluation

AFTER



(a) Downstream Evaluation

After DISABLING EDIT:

One can get to Bay by navigating the, B Serie Italian, as. Italian Serie Gi Italy Italy Italy Italy Italy Italy Italy Italy Italy Italy Italy Italy Italian Italy is Italian,\n Italy mag Tur……

After Normal EDIT:

The language used by people in Equatorial Guinea is often very informal and it is important to be aware of what is acceptable and not acceptable. This is particularly important if there has been a ……

Dataset	Implementation	Efficacy		Generalization		Locality		Fluency	Score
		ES ↑	EM ↑	PS ↑	PM ↑	NS ↑	NM ↑	GE ↑	S ↑
CF	Original	99.92	99.68	96.29	71.58	75.8	10.25	621.96	89.32
	r-ROME	99.74	97.79	**99.09**	70.86	**80.62**	26.0	621.67	**92.22**
	p-ROME	**99.9**	99.36	97.04	63.01	80.0	5.74	621.17	91.42

A Unified Framework for Model Editing

Akshat Gupta, Dev Sajnani, Gopala Anumanchipalli
UC Berkeley
{akshat.gupta, sajnanidev, gopala}@berkeley.edu

EMNLP 2024 (Findings)

AKSHAT GUPTA
PhD Student,
UC Berkeley

DEV SAJNANI
Undergrad, UC Berkeley

GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

A Unified Framework for Model Editing

Akshat Gupta, Dev Sajnani, Gopala Anumanchipalli
UC Berkeley
{akshat.gupta, sajnanidev, gopala}@berkeley.edu

EMNLP 2024 (Findings)

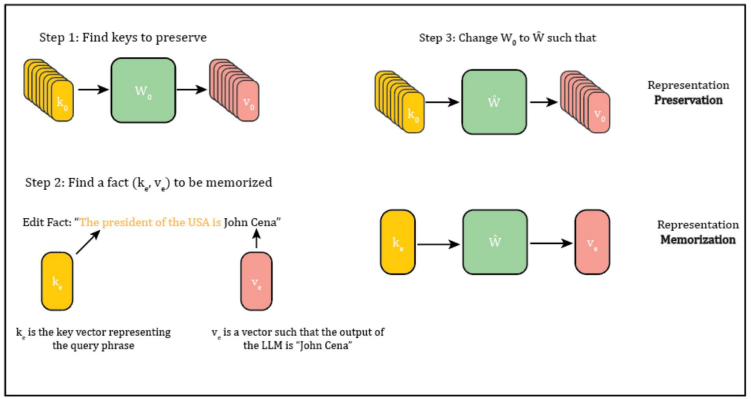


Figure 1: A diagrammatic representation of the preservation-memorization objective.

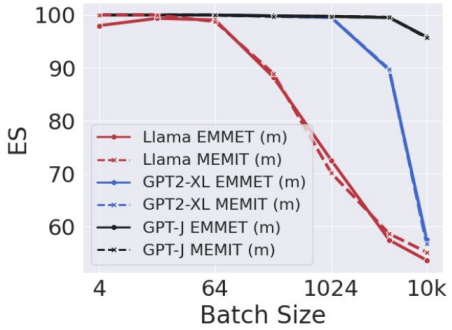
EMMET - Equality-constraint Mass Model Editing Algorithm

ROME:

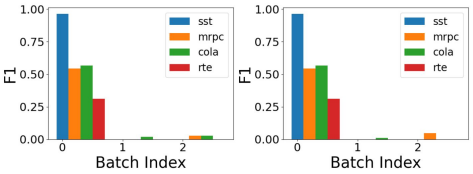
$$\underset{\hat{W}}{\operatorname{argmin}} \underbrace{\left\|\hat{W}K_0 - W_0K_0\right\|}_{\text{preservation}} \text{ s.t. } \underbrace{\hat{W}k_e = v_e}_{\text{memorization}}$$

MEMIT:

$$\underset{\hat{W}}{\operatorname{argmin}} \underbrace{\lambda||\hat{W}K_0 - W_0K_0||}_{\text{preservation}} + \underbrace{||\hat{W}K_E - V_E||}_{\text{memorization}}$$



(a) Efficacy Score (ES)



(a) EMMET (b) MEMIT

Figure 13: Model - Llama2-7b. Batch size 4096.

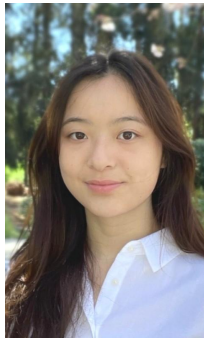
Norm Growth and Stability Challenges in Localized Sequential Knowledge Editing

Akshat Gupta[1*], **Christine Fang**[1], **Atahan Ozdemir**[1], **Maochuan Lu**[1], **Ahmed Alaa**[1], **Thomas Hartvigsen**[2], **Gopala Anumanchipalli**[1]

[1]University of California Berkeley, [2]University of Virginia

Outstanding Paper Award, Towards Knowledgeable Foundation Models Workshop @ AAAI 2025

AKSHAT GUPTA
PhD Student,
UC Berkeley



CHRISTINE FENG
Undergrad,
UC Berkeley

ATAHAN OZDEMIR
Undergrad,
UC Berkeley

MAOCHUAN LU
Undergrad,
UC Berkeley

AHMED ALAA,
Asst. Professor,
UC Berkeley

Thomas Hartvigsen,
Asst. Professor,
University of Virginia

GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

Norm Growth and Stability Challenges in Localized Sequential Knowledge Editing

Akshat Gupta[1]*, Christine Fang[1], Atahan Ozdemir[1], Maochuan Lu[1], Ahmed Alaa[1], Thomas Hartvigsen[2], Gopala Anumanchipalli[1]

[1]University of California Berkeley, [2]University of Virginia

Outstanding Paper Award, Towards Knowledgeable Foundation Models Workshop @ AAAI 2025

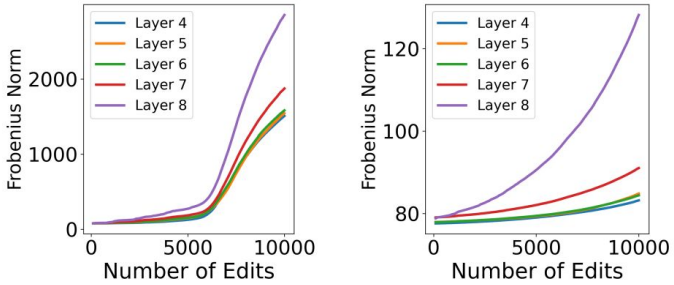


Figure 1. The continuous growth of norm of edited MLP matrices in LLama3-8B during sequential knowledge editing, as a function of number edits.

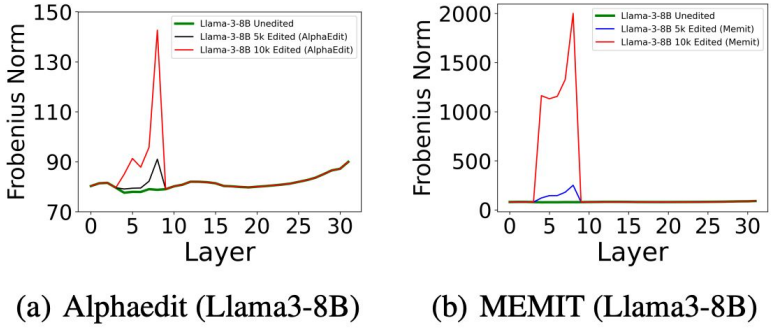
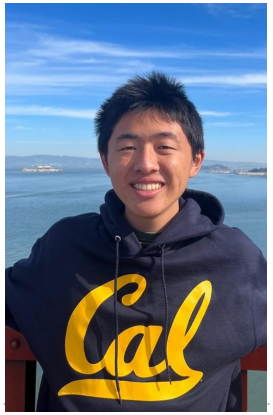


Figure 4. Comparison between norm of edited MLP matrices and norm of unedited matrices after 5,000 and 10,000 sequential edits.

PROJECT -2 : INFORMATION PROCESSING IN LLMS

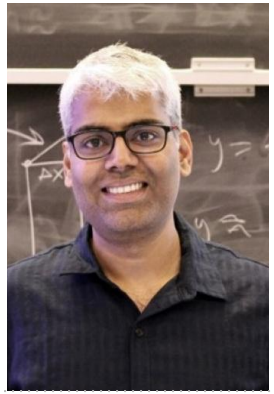
TEAM

AKSHAT GUPTA
PhD Student,
UC Berkeley



JAY YEUNG
Undergrad,
UC Berkeley

ANNA IVANOVA
Asst. Professor,
Georgia Tech



GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

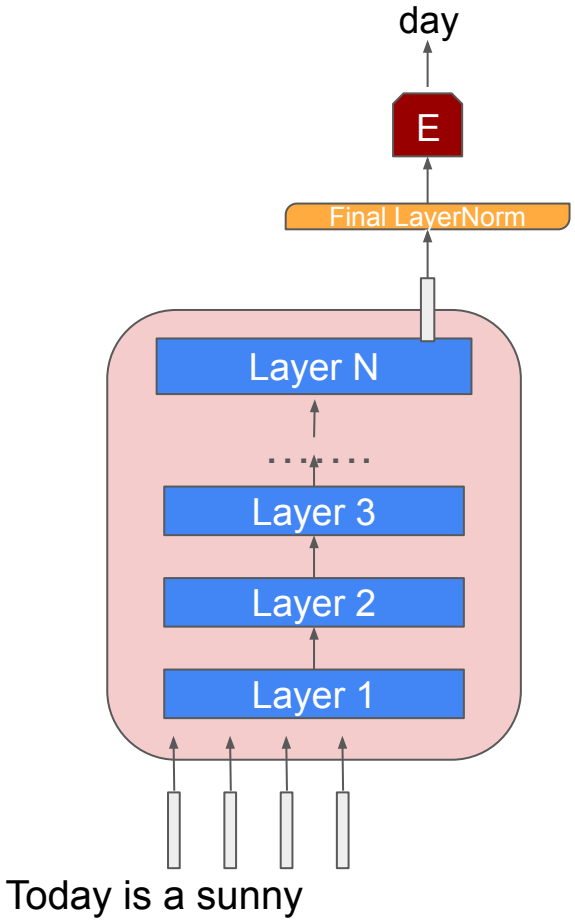
Introduction and Motivation

- Humans require varying levels of cognitive effort depending on the nature and complexity of the information being processed.
  - While we can recall some commonplace facts almost instantly, retrieving more obscure information may take longer
  - We produce certain types of words with little conscious effort while take our time with others (function words vs content words)

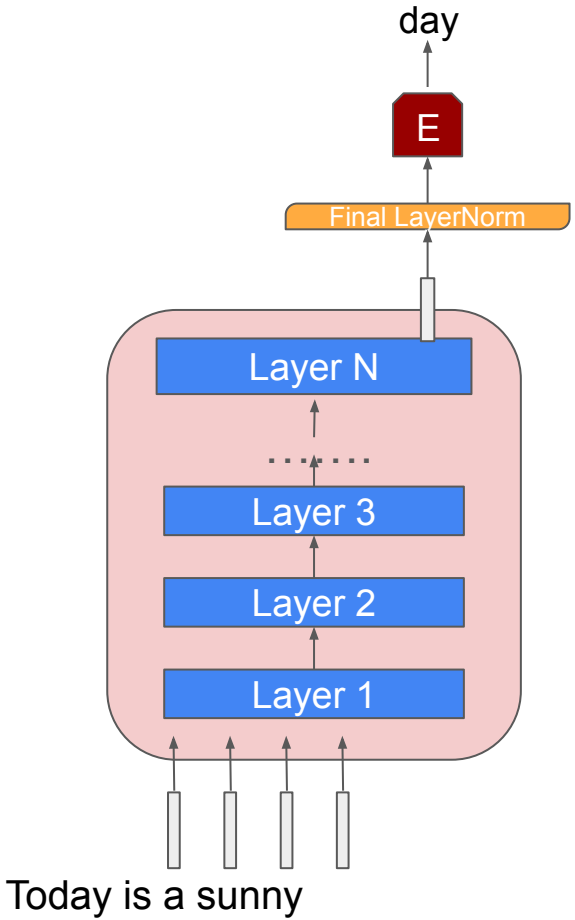
- While humans dynamically adjust their cognitive effort based on the familiarity and complexity of information, large language models (LLMs) in their current form process all inputs uniformly.
  - But do all tokens truly require the full depth of an LLM's architecture, or could some tokens be processed more efficiently based on their type and context?
  - More interestingly, have LLMs implicitly developed some form of dynamic load adjustment, mirroring the way humans allocate cognitive effort?

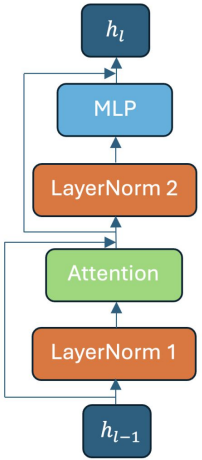
BACKGROUND: LOGITLENS

Computations in an LLM



Computations in an LLM





$$f^l = \texttt{LN1}(h^{l-1}) \tag{3}$$

$$a^l = \texttt{Att}(f^l) \tag{4}$$

$$g^l = \texttt{LN2}(h^{l-1} + a^l) \tag{5}$$

$$m^l = W^l_{proj}\sigma(W^l_{fc}g^l + b^l_{fc}) + b_{proj} \tag{6}$$

$$h^l = h^{l-1} + a^l + m^l \tag{7}$$

The LogitLens (TunedLens) Framework

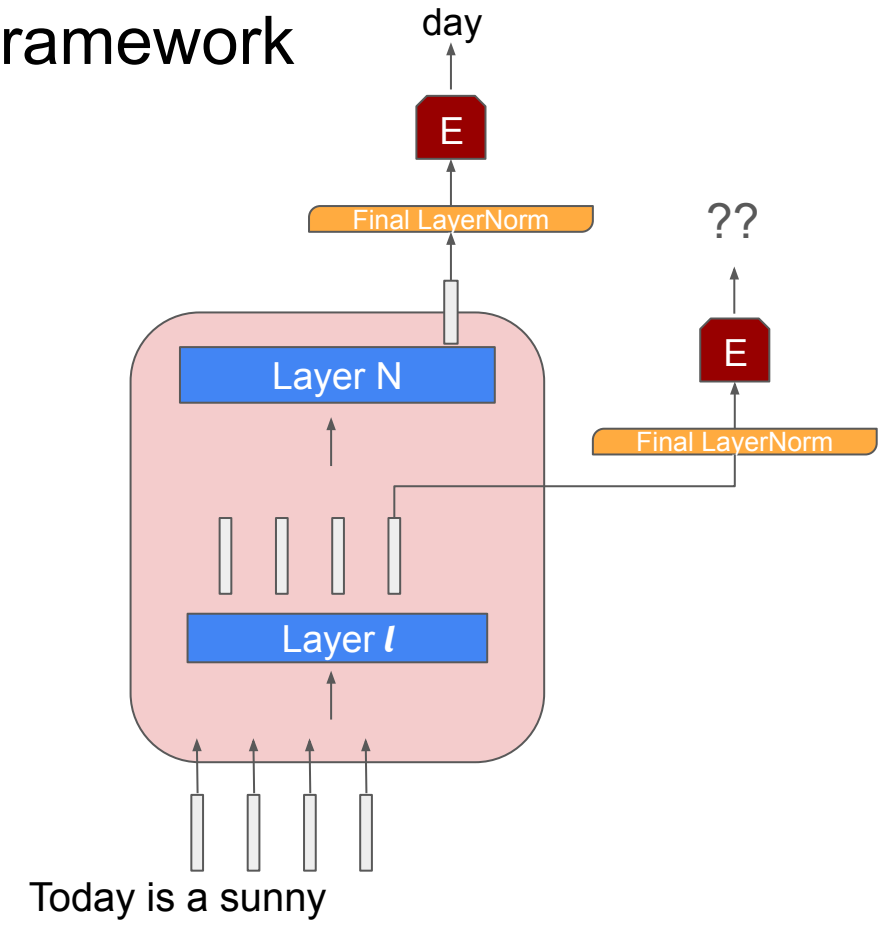
$$h^l = h^{l-1} + a^l + m^l$$

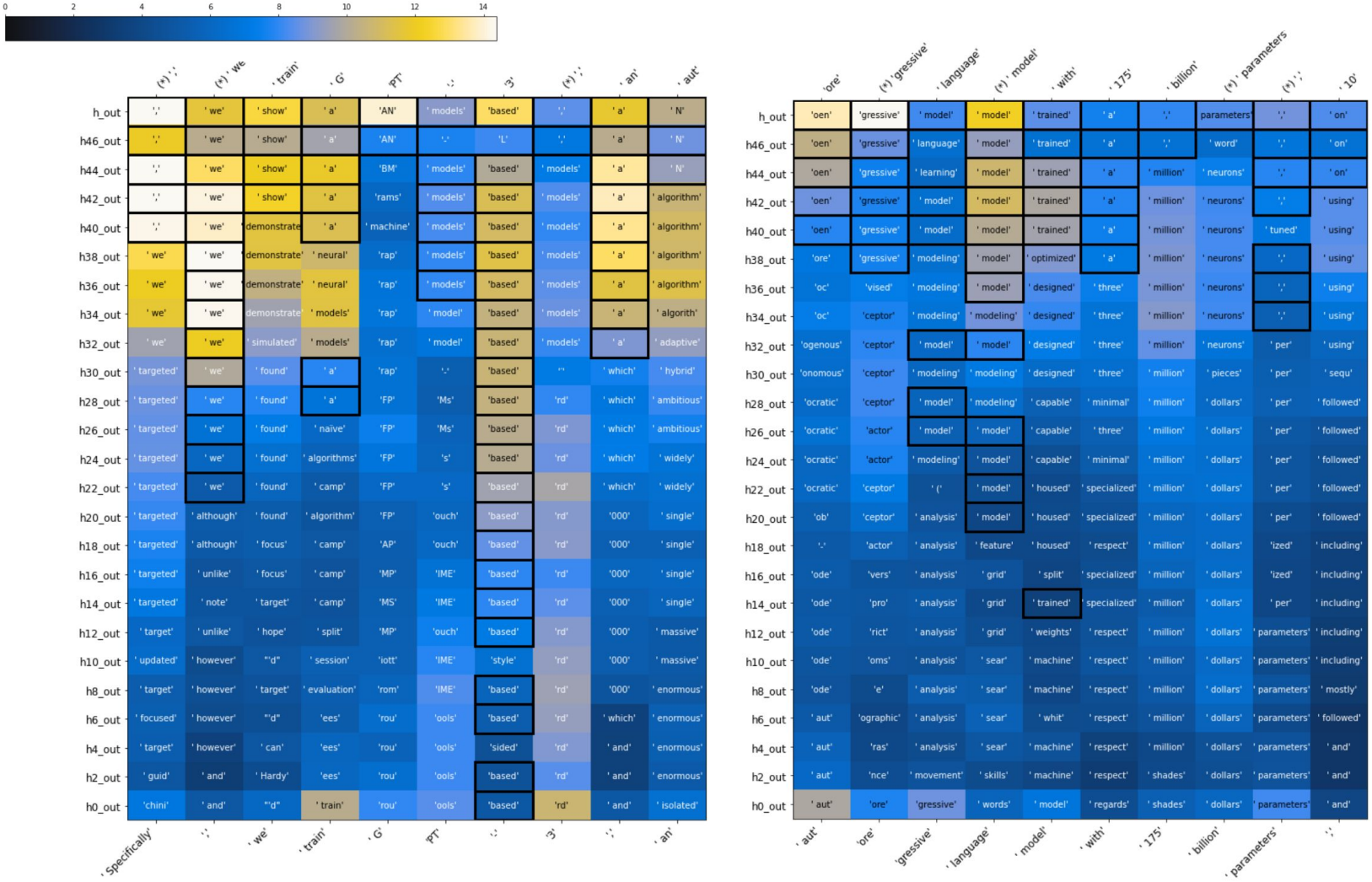
$$\texttt{LogitLens}(h^l) = W_U \left[\texttt{Norm}_f[h^l] \right]$$

The LogitLens (TunedLens) Framework

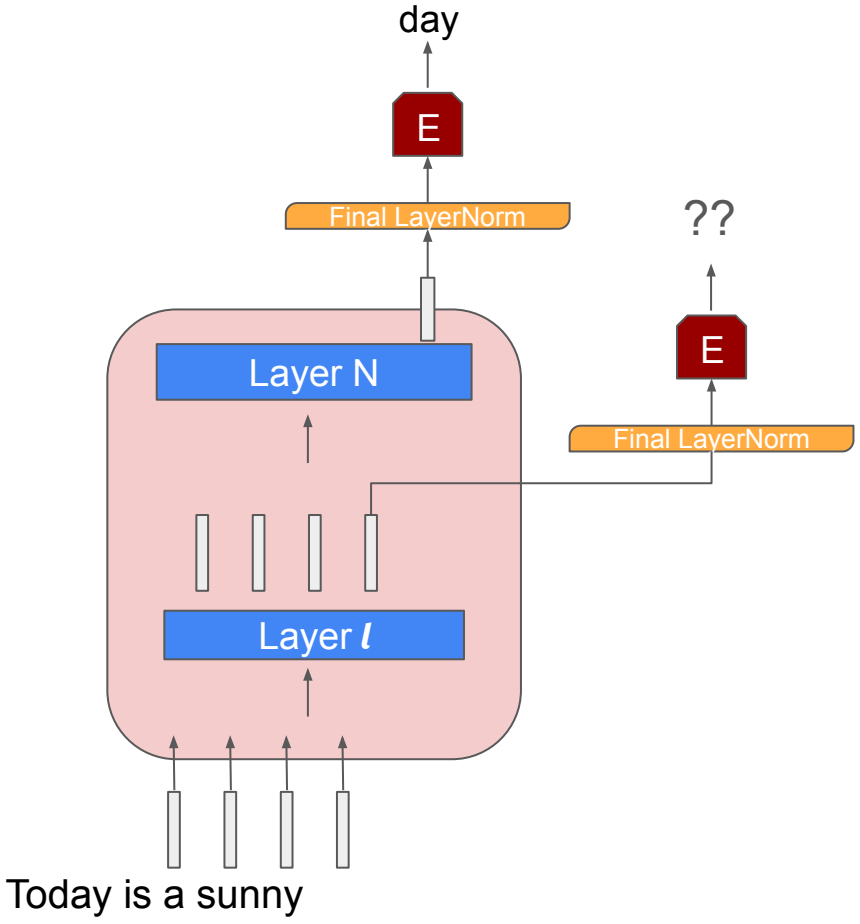
$$h^l = h^{l-1} + a^l + m^l$$

$$\texttt{LogitLens}(h^l) = W_U\left[\texttt{Norm}_f[h^l]\right]$$

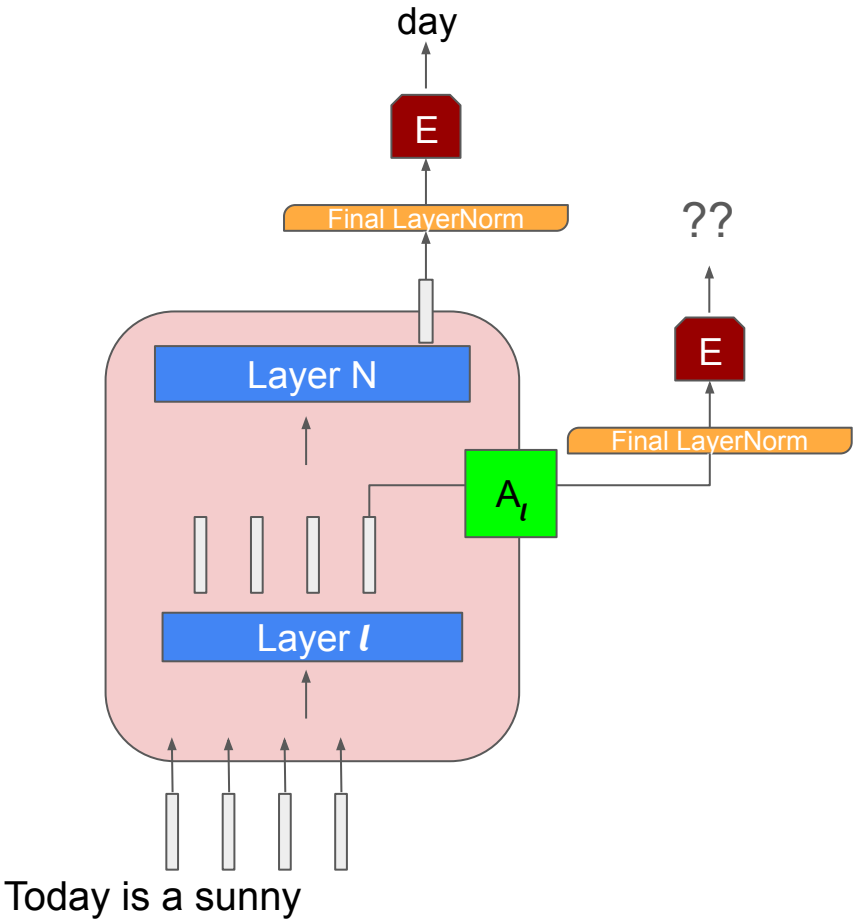



0 2 4 6 8 10 12 14
(*) ','
(*) ' we
' train'
' G'
'PT'
','
'3'
(*) ','
' an'
' aut'
h_out ',' ' we' ' show' ' a' 'AN' ' models' 'based' ',' ' a' ' N'
h46_out ',' ' we' ' show' ' a' 'AN' '-' 'L' ',' ' a' ' N'
h44_out ',' ' we' ' show' ' a' 'BM' ' models' 'based' ' models' ' a' ' N'
h42_out ',' ' we' ' show' ' a' 'rams' ' models' 'based' ' models' ' a' ' algorithm'
h40_out ',' ' we' 'demonstrate' ' a' ' machine' ' models' 'based' ' models' ' a' ' algorithm'
h38_out ' we' ' we' 'demonstrate' ' neural' 'rap' ' models' 'based' ' models' ' a' ' algorithm'
h36_out ' we' ' we' 'demonstrate' ' neural' 'rap' ' models' 'based' ' models' ' a' ' algorithm'
h34_out ' we' ' we' 'demonstrate' ' models' 'rap' ' model' 'based' ' models' ' a' ' algorith'
h32_out ' we' ' we' ' simulated' ' models' 'rap' ' model' 'based' ' models' ' a' ' adaptive'
h30_out ' targeted' ' we' ' found' ' a' 'rap' '-' 'based' "'" ' which' ' hybrid'
h28_out ' targeted' ' we' ' found' ' a' 'FP' 'Ms' 'based' 'rd' ' which' ' ambitious'
h26_out ' targeted' ' we' ' found' ' naive' 'FP' 'Ms' 'based' 'rd' ' which' ' ambitious'
h24_out ' targeted' ' we' ' found' ' algorithms' 'FP' 's' 'based' 'rd' ' which' ' widely'
h22_out ' targeted' ' we' ' found' ' camp' 'FP' 's' 'based' 'rd' ' which' ' widely'
h20_out ' targeted' ' although' ' found' ' algorithm' 'FP' 'ouch' 'based' 'rd' '000' ' single'
h18_out ' targeted' ' although' ' focus' ' camp' 'AP' 'ouch' 'based' 'rd' '000' ' single'
h16_out ' targeted' ' unlike' ' focus' ' camp' 'MP' 'IME' 'based' 'rd' '000' ' single'
h14_out ' targeted' ' note' ' target' ' camp' 'MS' 'IME' 'based' 'rd' '000' ' single'
h12_out ' target' ' unlike' ' hope' ' split' 'MP' 'ouch' 'based' 'rd' '000' ' massive'
h10_out ' updated' ' however' "'d" ' session' 'iott' 'IME' 'style' 'rd' '000' ' massive'
h8_out ' target' ' however' ' target' ' evaluation' 'rom' 'IME' 'based' 'rd' '000' ' enormous'
h6_out ' focused' ' however' "'d" 'ees' 'rou' 'ools' 'based' 'rd' ' which' ' enormous'
h4_out ' target' ' however' ' can' 'ees' 'rou' 'ools' 'sided' 'rd' ' and' ' enormous'
h2_out ' guid' ' and' ' Hardy' 'ees' 'rou' 'ools' 'based' 'rd' ' and' ' enormous'
h0_out 'chini' ' and' "'d" ' train' 'rou' 'ools' 'based' 'rd' ' and' ' isolated'
' specifically' ',' ' we' ' train' ' G' 'PT' '-' '3' ',' ' an'
'ore' (*) 'gressive' ' language' (*) ' model' ' with' ' 175' ' billion' (*) ' parameters' (*) ',' '.10'
h_out 'oen' 'gressive' ' model' ' model' ' trained' ' a' ',' parameters' ',' ' on'
h46_out 'oen' 'gressive' ' language' ' model' ' trained' ' a' ',' ' word' ',' ' on'
h44_out 'oen' 'gressive' ' learning' ' model' ' trained' ' a' ' million' ' neurons' ',' ' on'
h42_out 'oen' 'gressive' ' model' ' model' ' trained' ' a' ' million' ' neurons' ',' ' using'
h40_out 'oen' 'gressive' ' model' ' model' ' trained' ' a' ' million' ' neurons' ' tuned' ' using'
h38_out 'ore' 'gressive' ' modeling' ' model' ' optimized' ' a' ' million' ' neurons' ',' ' using'
h36_out 'oc' 'vised' ' modeling' ' model' ' designed' ' three' ' million' ' neurons' ',' ' using'
h34_out 'oc' 'ceptor' ' modeling' ' modeling' ' designed' ' three' ' million' ' neurons' ',' ' using'
h32_out 'ogenous' 'ceptor' ' model' ' model' ' designed' ' three' ' million' ' neurons' ' per' ' using'
h30_out 'onomous' 'ceptor' ' modeling' ' modeling' ' designed' ' three' ' million' ' pieces' ' per' ' sequ'
h28_out 'ocratic' 'ceptor' ' model' ' modeling' ' capable' ' minimal' ' million' ' dollars' ' per' ' followed'
h26_out 'ocratic' 'actor' ' model' ' model' ' capable' ' three' ' million' ' dollars' ' per' ' followed'
h24_out 'ocratic' 'actor' ' modeling' ' model' ' capable' ' minimal' ' million' ' dollars' ' per' ' followed'
h22_out 'ocratic' 'ceptor' ' (' ' model' ' housed' ' specialized' ' million' ' dollars' ' per' ' followed'
h20_out 'ob' 'ceptor' ' analysis' ' model' ' housed' ' specialized' ' million' ' dollars' ' per' ' followed'
h18_out '-' 'actor' ' analysis' ' feature' ' housed' ' respect' ' million' ' dollars' 'ized' ' including'
h16_out 'ode' 'vers' ' analysis' ' grid' ' split' ' specialized' ' million' ' dollars' 'ized' ' including'
h14_out 'ode' 'pro' ' analysis' ' grid' ' trained' ' specialized' ' million' ' dollars' ' per' ' including'
h12_out 'ode' 'rict' ' analysis' ' grid' ' weights' ' respect' ' million' ' dollars' ' parameters' ' including'
h10_out 'ode' 'oms' ' analysis' ' sear' ' machine' ' respect' ' million' ' dollars' ' parameters' ' including'
h8_out 'ode' 'e' ' analysis' ' sear' ' machine' ' respect' ' million' ' dollars' ' parameters' ' mostly'
h6_out ' aut' 'ographic' ' analysis' ' sear' ' whit' ' respect' ' million' ' dollars' ' parameters' ' followed'
h4_out ' aut' 'ras' ' analysis' ' sear' ' machine' ' respect' ' million' ' dollars' ' parameters' ' and'
h2_out ' aut' 'nce' ' movement' ' skills' ' machine' ' respect' ' shades' ' dollars' ' parameters' ' and'
h0_out ' aut' 'ore' 'gressive' ' words' ' model' ' regards' ' shades' ' dollars' ' parameters' ' and'
' aut' 'ore' 'gressive' ' language' ' model' ' with' ' 175' ' billion' ' parameters' ','

LogitLens:



TunedLens:

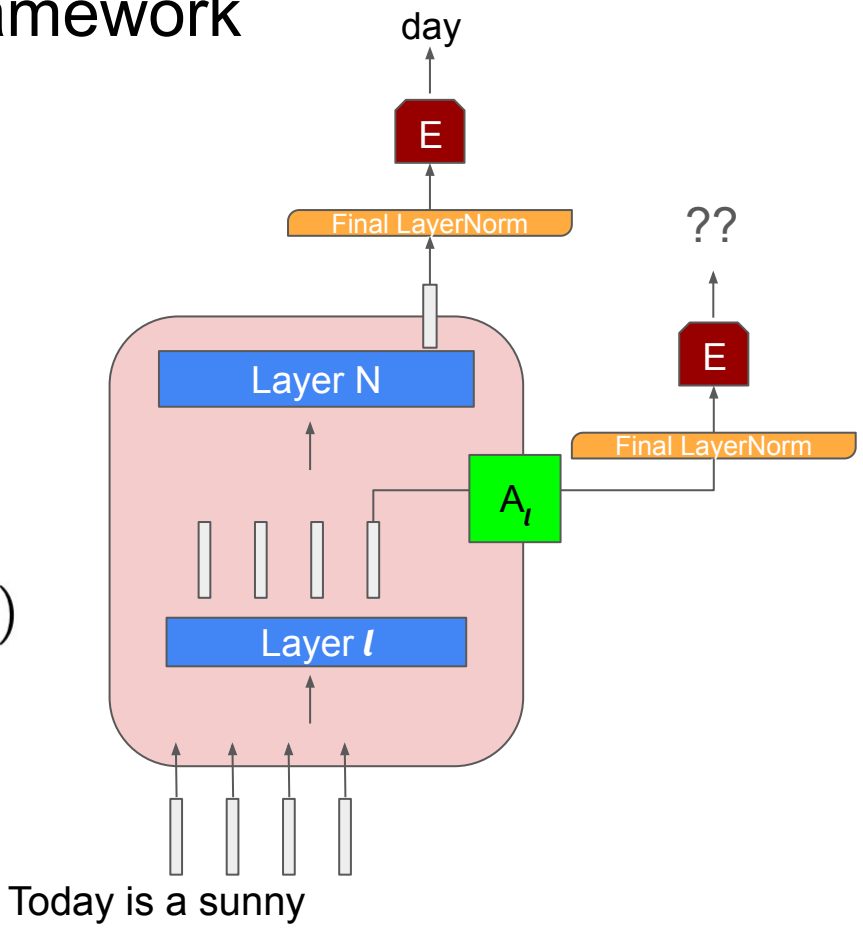


The LogitLens (TunedLens) Framework

$$h^l = h^{l-1} + a^l + m^l$$

$$\texttt{LogitLens}(h^l) = W_U\left[\texttt{Norm}_f[h^l]\right]$$

$$\texttt{TunedLens}(h^l) = \texttt{LogitLens}(A_l h^l + b_l)$$



INFORMATION PROCESSING BASED ON GENERATED TOKEN PART-OF-SPEECH

Example Sentences

Category: DET (Determiner)

- **Input:** "She picked up __"
- **Output:** "the"

Category: ADP (Adposition, e.g., prepositions)

- **Input:** "He walked slowly __"
- **Output:** "to"

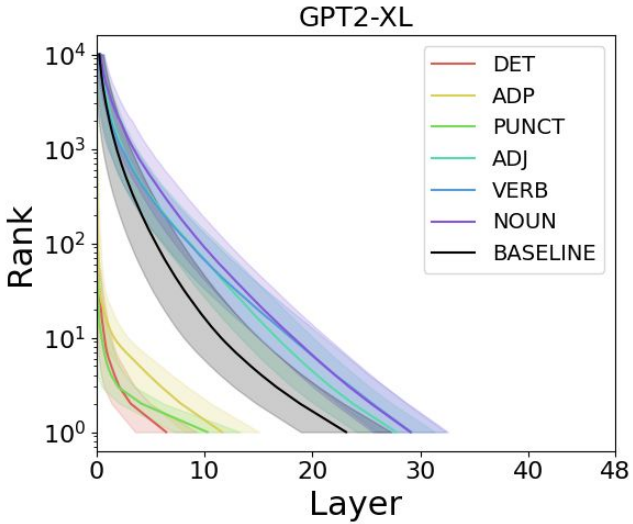
Category: NOUN

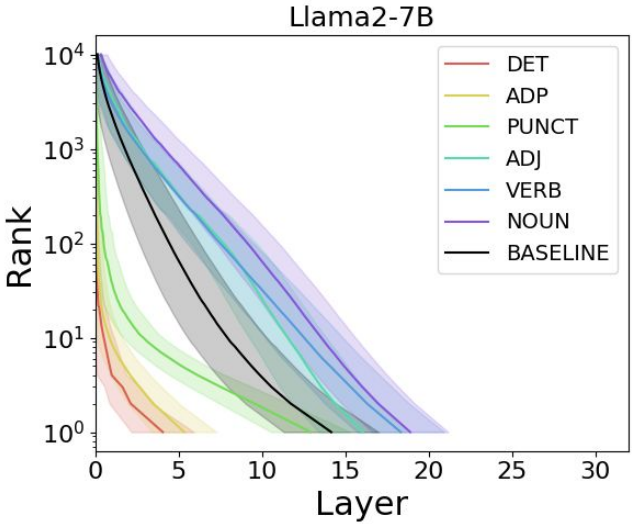
- **Input:** "The dog chased a __"
- **Output:** "squirrel"

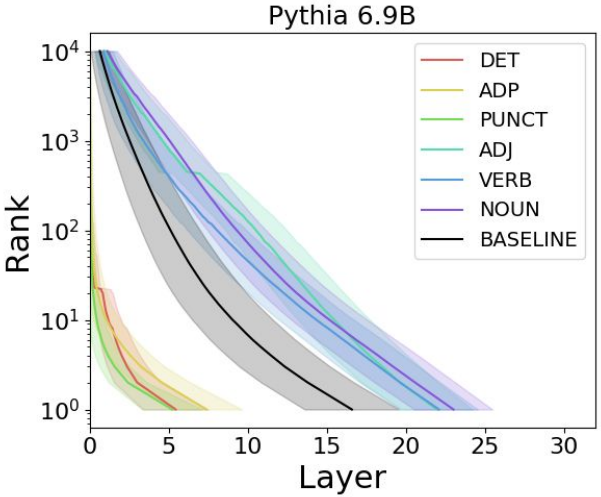
Category: VERB

- **Input:** "She carefully __"
- **Output:** "painted"

Syntactic Structure





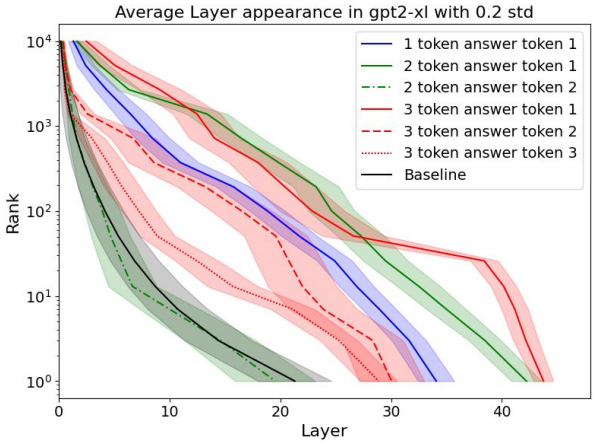


INFORMATION PROCESSING DURING FACT RECALL

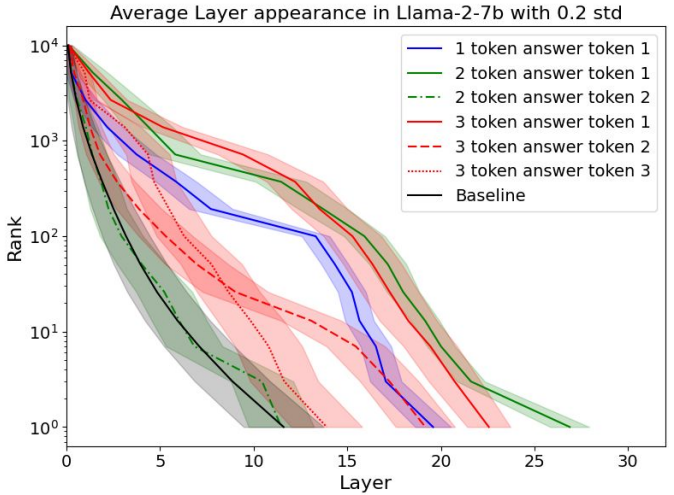
Example sentences

The capital of France is ___

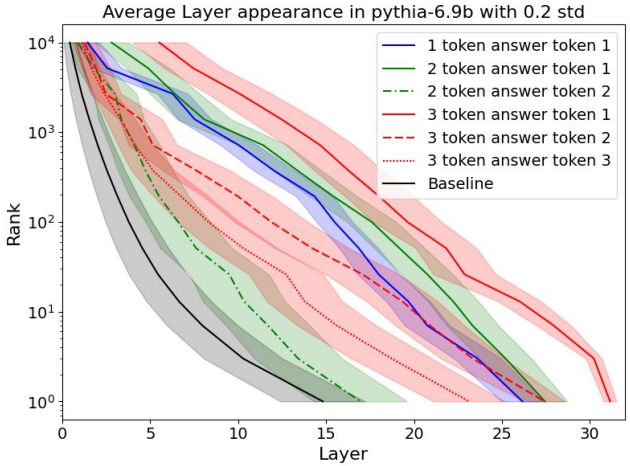
Fact Recall



a) GPT-2XL (1.5B)



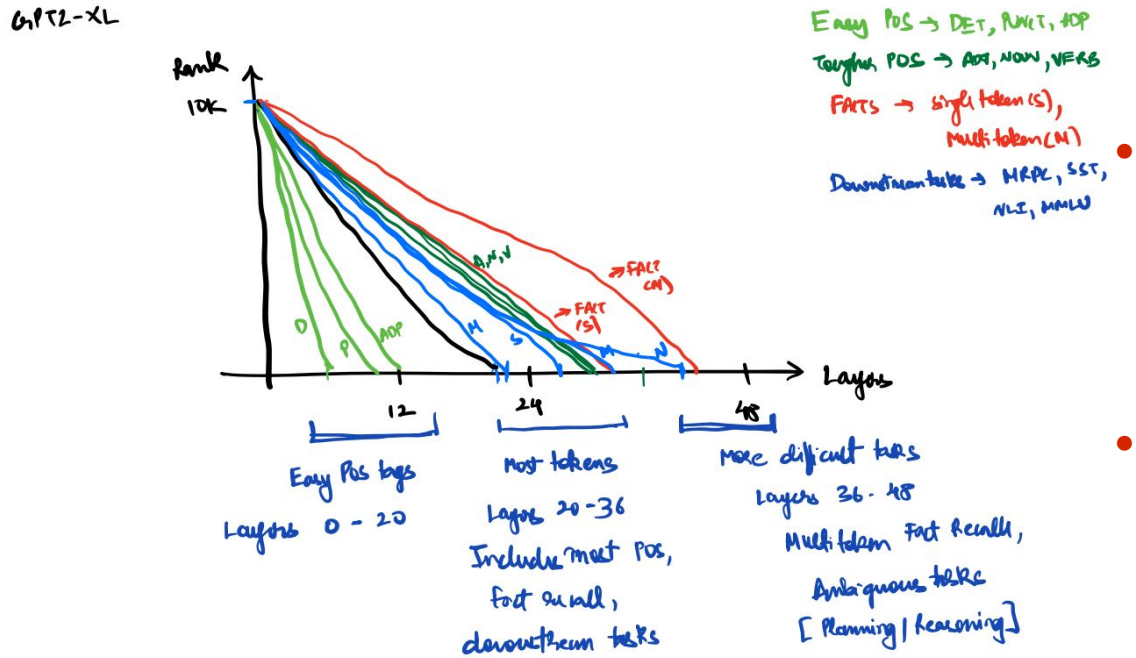
b) Llama-2 (7B)



c) Pythia (7B)

REFLECTION ON RESULTS

Hierarchy of Information



- **Baseline** occurs around first 40% of layers. On average, a model takes 40% of layers to make prediction decision.
- **Early Layers** - Layers before baseline are what I call early layers, or around first 40% of layers.
  - Used to make decisions about simple functional words for grammatical purposes - DET, ADP, PUNCT
- **Middle Layers** - Next 35% of layers are what I call the middle layers. They do more complex tasks like fact recall, predicting content words like nouns, verbs, and doing downstream tasks.
- **Late Layers** - Last 25% of layers are what I call late layers. They do more complex tasks like predicting multi-token facts, more ambiguous downstream tasks like NLI.

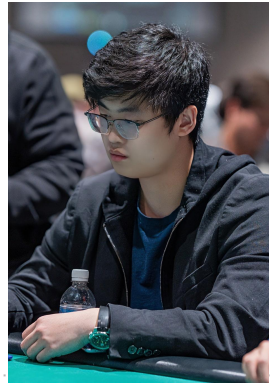
Takeaways

- At inference times, LLMs take different amounts of time to process different kinds of information
- There is an information processing hierarchy in LLMs - Seemingly easier tasks finish processing earlier than more complex tasks
- Potential applications - Early exiting

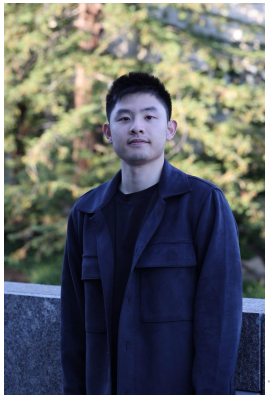
PROJECT -3 : LLMS AND POKER

Team

AKSHAT GUPTA
PhD Student,
UC Berkeley



ZHENGYU (BRIAN) LI
PhD Student,
Georgia Tech



RICHARD ZHUANG
Undergrad (Senior),
UC Berkeley

VIJAY GANESH
Professor,
Georgia Tech

GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

Other Members:

- Piyush Jha (PhD Student, Georgia Tech)
- Jonny Pei (Senior, UC Berkeley)
- Chris Dodla (Sophomore, UC Berkeley)

Past Members:

- Richard Yang (UC Berkeley)
- Aniket Rahane (UC Berkeley)

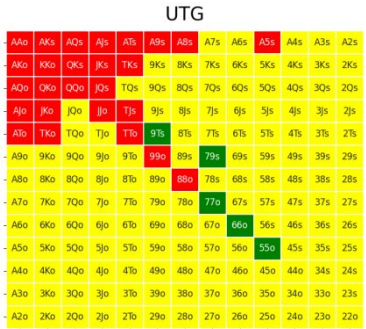
MOTIVATIONS

1. Are LLM any good at poker?
2. Can we train LLMs to be GTO?
3. Can we make exploitative poker playing agents that can go beyond GTO?
4. Can we use LLMs to explain GTO decision and teach poker?

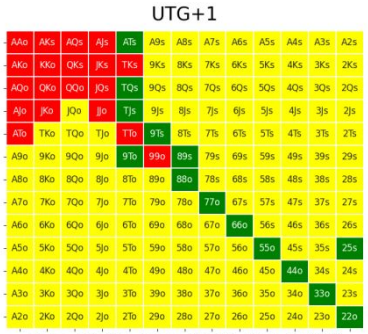
Are ChatGPT and GPT-4 Good Poker Players - A Pre-flop Analysis (arxiv, 2023)

Are ChatGPT and GPT-4 Good Poker Players? - A Pre-Flop Analysis

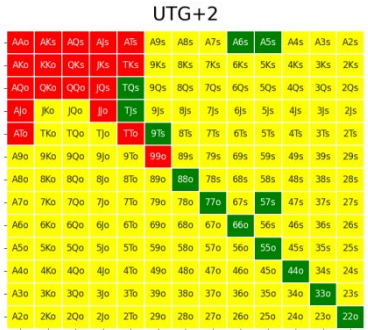
Akshat Gupta
UC Berkeley
akshat.gupta@berkeley.edu



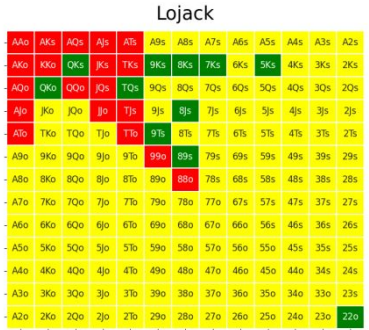
Raise: 13.12%, Limp: 1.96%, Fold: 84.92%



Raise: 10.26%, Limp: 5.88%, Fold: 83.86%



Raise: 9.65%, Limp: 4.98%, Fold: 85.37%



Raise: 9.2%, Limp: 4.07%, Fold: 86.73%

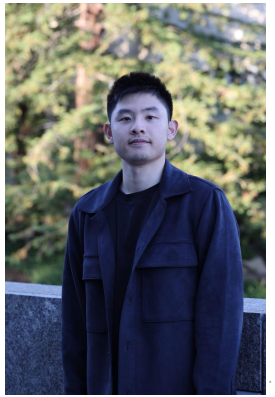
HIGHLIGHTS

PokerBench: Training Large Language Models to become Professional Poker Players

Richard Zhuang[1], **Akshat Gupta**[1*], **Richard Yang**[1], **Aniket Rahane**[1], **Zhengyu Li**[2], **Gopala Anumanchipalli**[1]

[1]University of California, Berkeley; [2]Georgia Institute of Technology

Accepted to AAAI 2025!



RICHARD ZHUANG
Undergrad (Senior),
UC Berkeley

PokerBench - Training Large Language Models to become Professional Poker Players (AAAI, 2025)

MODEL EVALUATION ON POKERBENCH:

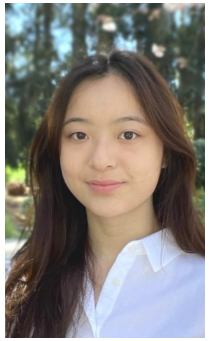
Evaluation Type	Model	Overall Accuracy		Post-Flop Accuracy		Pre-Flop Accuracy	
		EM ↑	AA ↑	EM ↑	AA ↑	EM ↑	AA ↑
Pre-Trained Models (Few-Shot)	LLAMA-3 (8B)	26.02	40.03	14.96	31.25	37.77	49.30
	LLAMA-2 (70B)	36.48	48.30	32.95	41.11	40.20	55.90
	LLAMA-3 (70B)	39.16	49.78	34.30	45.40	44.30	54.40
	CHATGPT 3.5	29.96	39.69	18.75	34.19	41.80	45.50
	GPT-4	**53.55**	**65.54**	**52.18**	**62.69**	**55.00**	**66.50**
Fine-Tuned Models (Zero-Shot)	GEMMA (2B)	51.84	62.74	41.57	52.94	62.70	73.10
	LLAMA-2 (7B))	78.11	79.91	76.52	79.55	79.80	80.30
	LLAMA-3 (8B)	**78.26**	**80.64**	**76.52**	**79.07**	**80.10**	**82.30**

Table 2: Performance of various pre-trained and fine-tuned LLMs on POKERBENCH.

ACKNOWLEDGEMENTS

ANURAG RAO
UC Berkeley
(Now MS, Oxford)

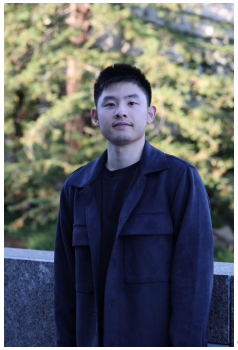
DEV SAJNANI
Undergrad,
UC Berkeley



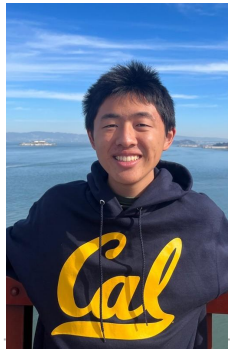
CHRISTINE FENG
Undergrad,
UC Berkeley

ATAHAN OZDEMIR
Undergrad,
UC Berkeley

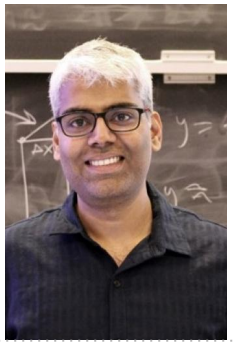
MAOCHUAN LU
Undergrad,
UC Berkeley



RICHARD ZHUANG
Undergrad (Senior),
UC Berkeley



JAY YEUNG
Undergrad,
UC Berkeley



GOPALA ANUMANCHIPALLI
Asst. Professor,
UC Berkeley

ANNA IVANOVA
Asst. Professor,
Georgia Tech

AHMED ALAA,
Asst. Professor,
UC Berkeley

Thomas Hartvigsen,
Asst. Professor,
University of Virginia

Special Thanks!

MAOCHUAN LU
Undergrad,
UC Berkeley

If you want to work with me on Interpretability or Poker, please reach out to me at :

akshat.gupta@berkeley.edu

Also check out some amazing work happening in our lab - **Berkeley Speech Group**

Contact : akshat.gupta@berkeley.edu

Thank You!