Manifold Learning and Artificial Intelligence

Lecture 15

Causal Tracing in Language Model

Momiao Xiong, University of Texas School of Public Health

- Time: 10:00 pm, US East Time, 05/20/2023
- 10:00 am, Beijing Time. 05/21/2023

Github Address: https://ai2healthcare.github.io/

With generality, however, there comes a question of control: how can we make LLMs do what we want them to do?

Zhou et al. 2023; LARGE LANGUAGE MODELS ARE HUMAN-LEVEL ROMPT ENGINEERS

- **To reduce the human effort involved in creating and validating effective instructions, we propose a novel algorithm using LLMs to generate and select instructions automatically. We call this problem natural language progran**

- **Tracr: Compiled Transformers as a Laboratory for Interpretability David Lindner et al. 2023**

- **Bills et al. 2023; Language models can explain neurons in language models**

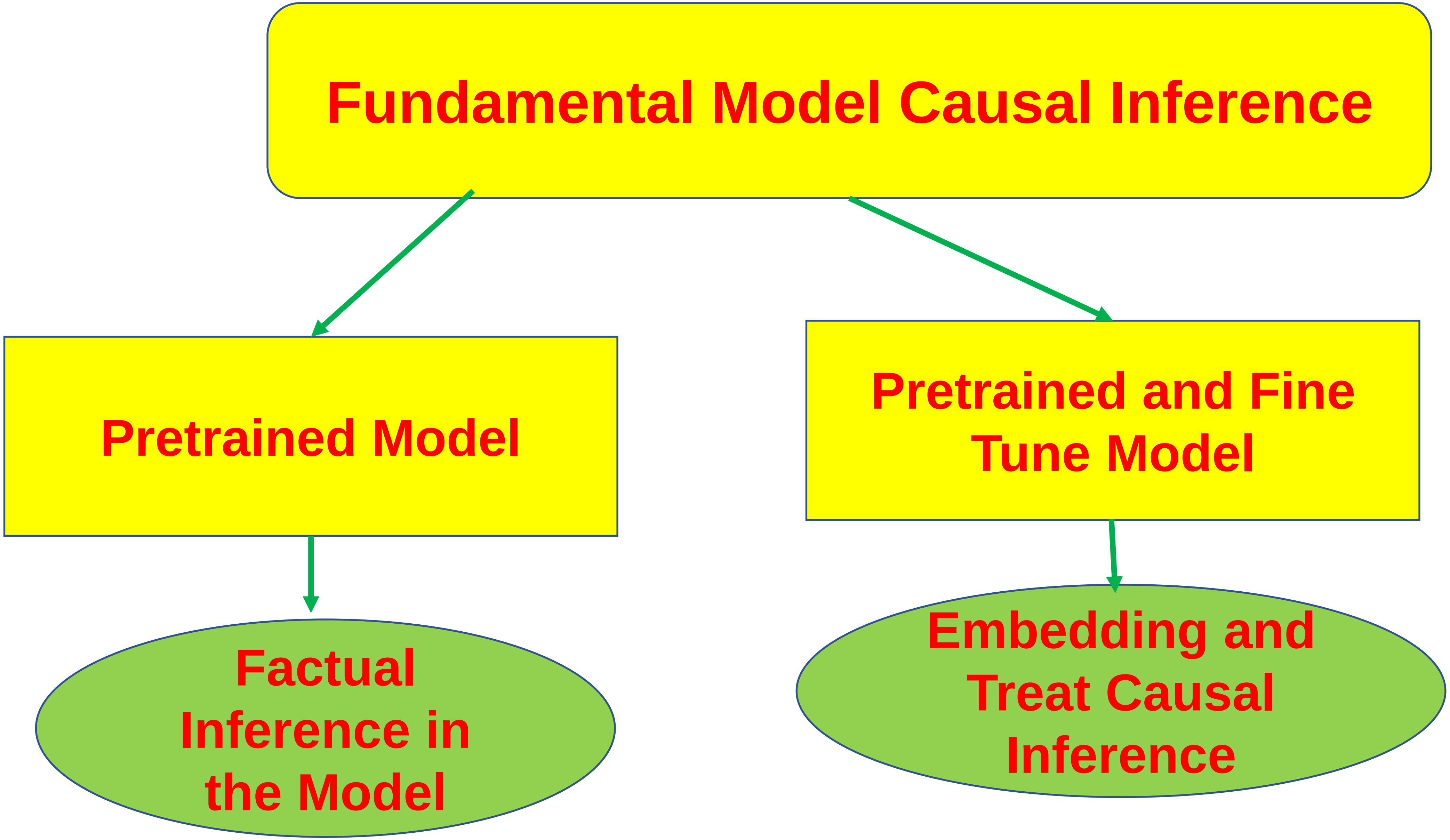
Procedure for Joining the Meeting

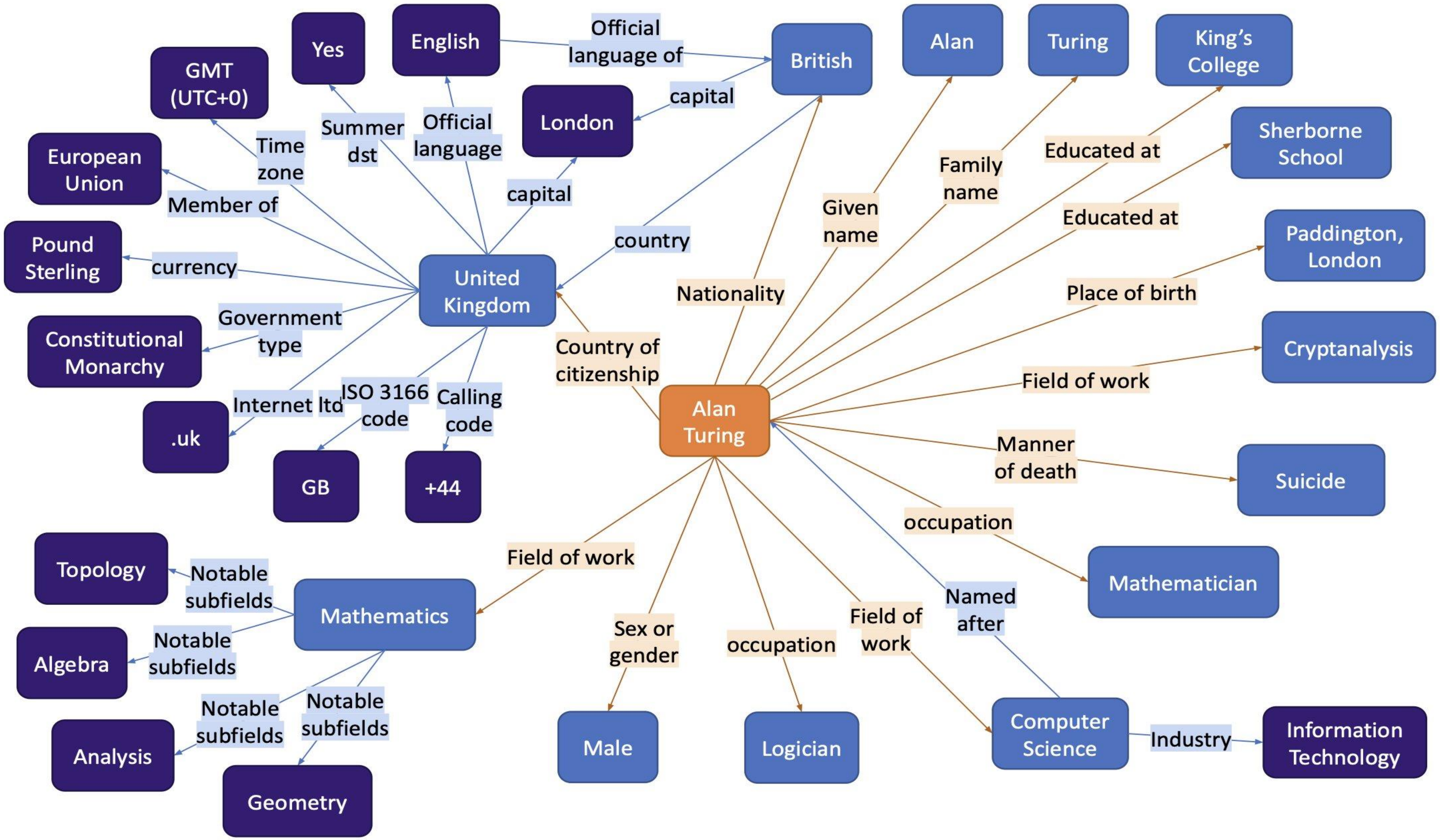
Join from PC, Mac, Linux, iOS or Android: Click Here to Join
https://uwmadison.zoom.us/w/93316139423?tk=wfbmsTfN2fgERto_HI1WKtBzh94d3HO02XVRDCexqd8.
DQMAAAAVuhNJnxZ2dGVCblIIYlR3ZWJBNHl5LXlYMjBnAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
&pwd=Q0NVWFYvRFg5RmxCNkwxMmYrbW41dz09#success
Passcode: 416262
Note: This link should not be shared with others; it is unique to you.
Add to Calendar Add to Google Calendar Add to Yahoo Calendar

Or One tap mobile :
US: +12532158782,,93316139423# or +13017158592,,93316139423#
Or Telephone:
Dial(for higher quality, dial a number based on your current location):
US: +1 253 215 8782 or +1 301 715 8592 or +1 305 224 1968 or +1 309 205 3325 or +1 312 626 6799 or +1 346 248 7799 or +1 360 209 5623 or +1 386 347 5053 or +1 507 473 4847 or +1 564 217 2000 or +1 646 931 3860 or +1 669 444 9171 or +1 669 900 6833 or +1 689 278 1000 or +1 719 359 4580 or +1 929 205 6099 or +1 253 205 0468
Meeting ID: 933 1613 9423
Passcode: 416262
International numbers available: https://uwmadison.zoom.us/u/aguj5kFFd

Or an H.323/SIP room system:
H.323: 162.255.37.11 (US West) or 162.255.36.11 (US East)
Meeting ID: 933 1613 9423
Passcode: 416262

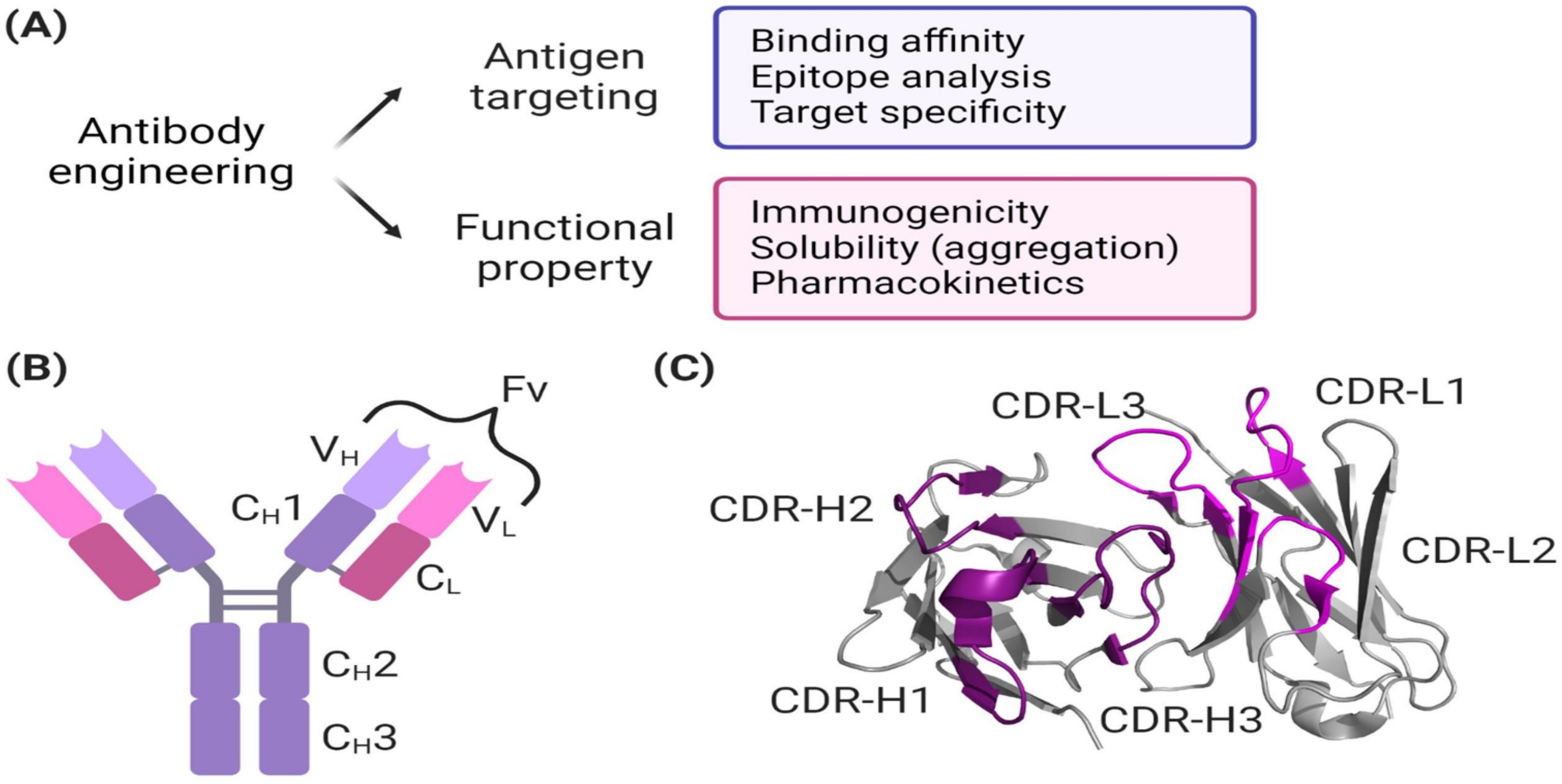
SIP: 93316139423@zoomcrc.com
Passcode: 416262


Fundamental Model Causal Inference
Pretrained Model
Pretrained and Fine Tune Model
Factual Inference in the Model
Embedding and Treat Causal Inference



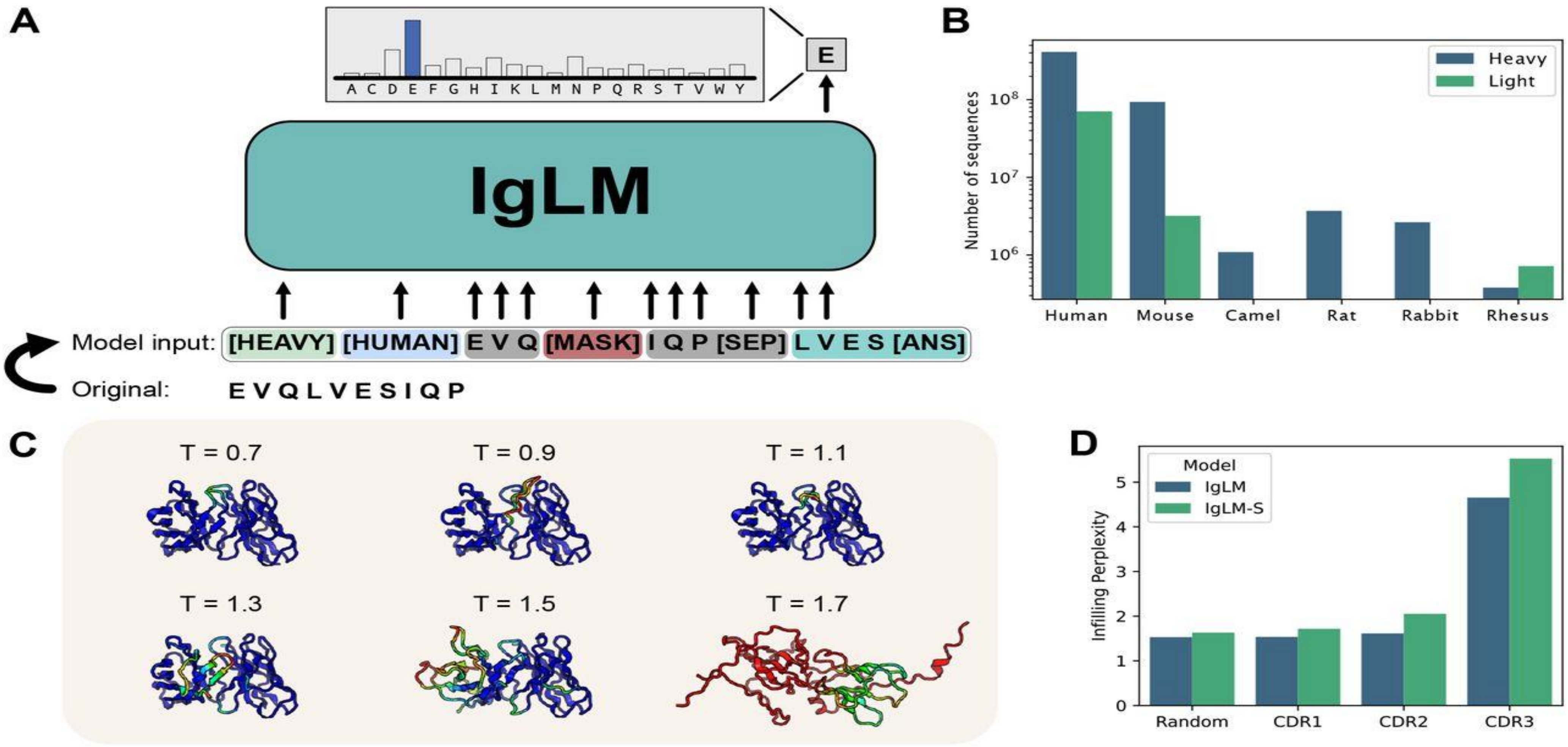
Cohen et al. 2023; Crawling The Internal Knowledge-Base of Language Models

Antibody Design Examples

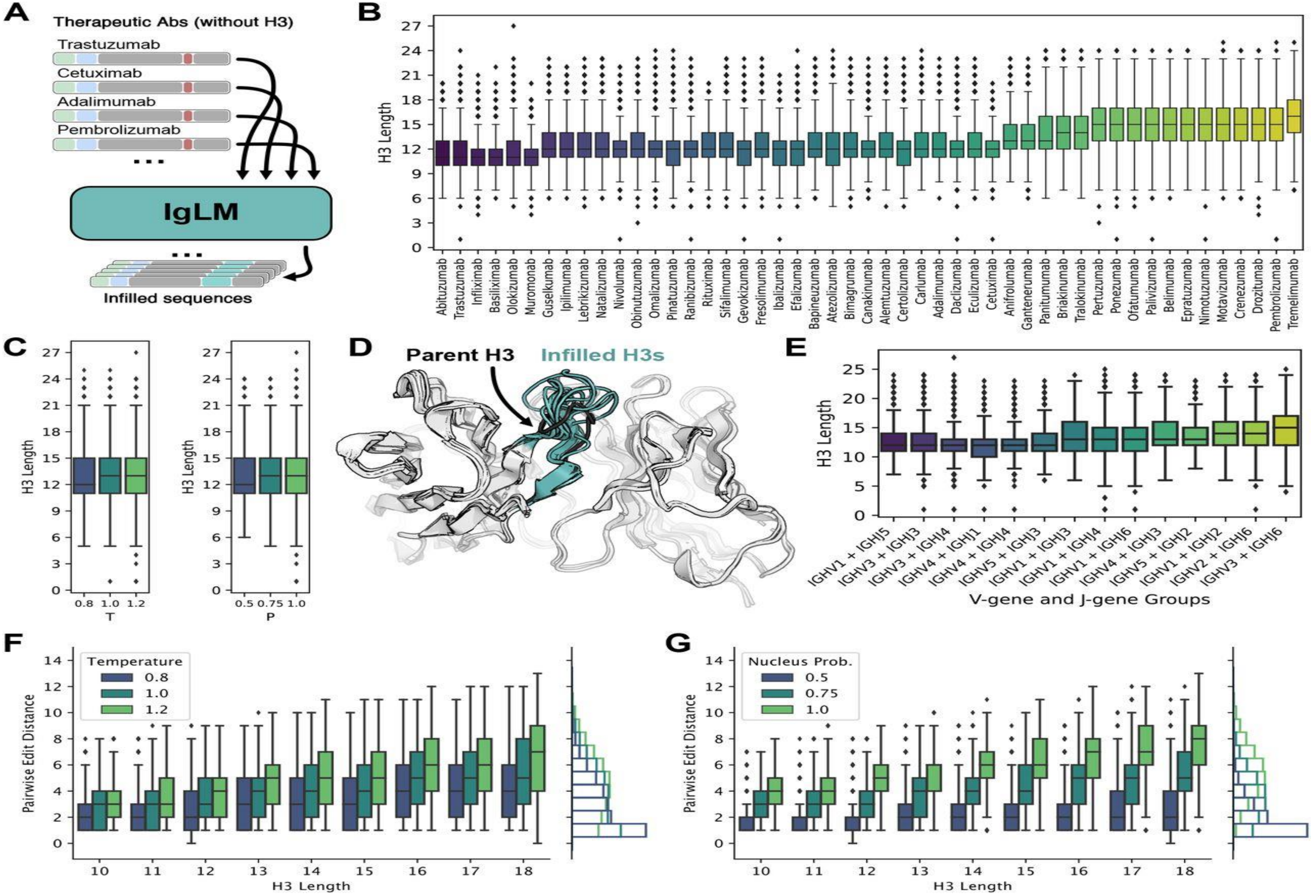


Trends in Pharmacological Sciences

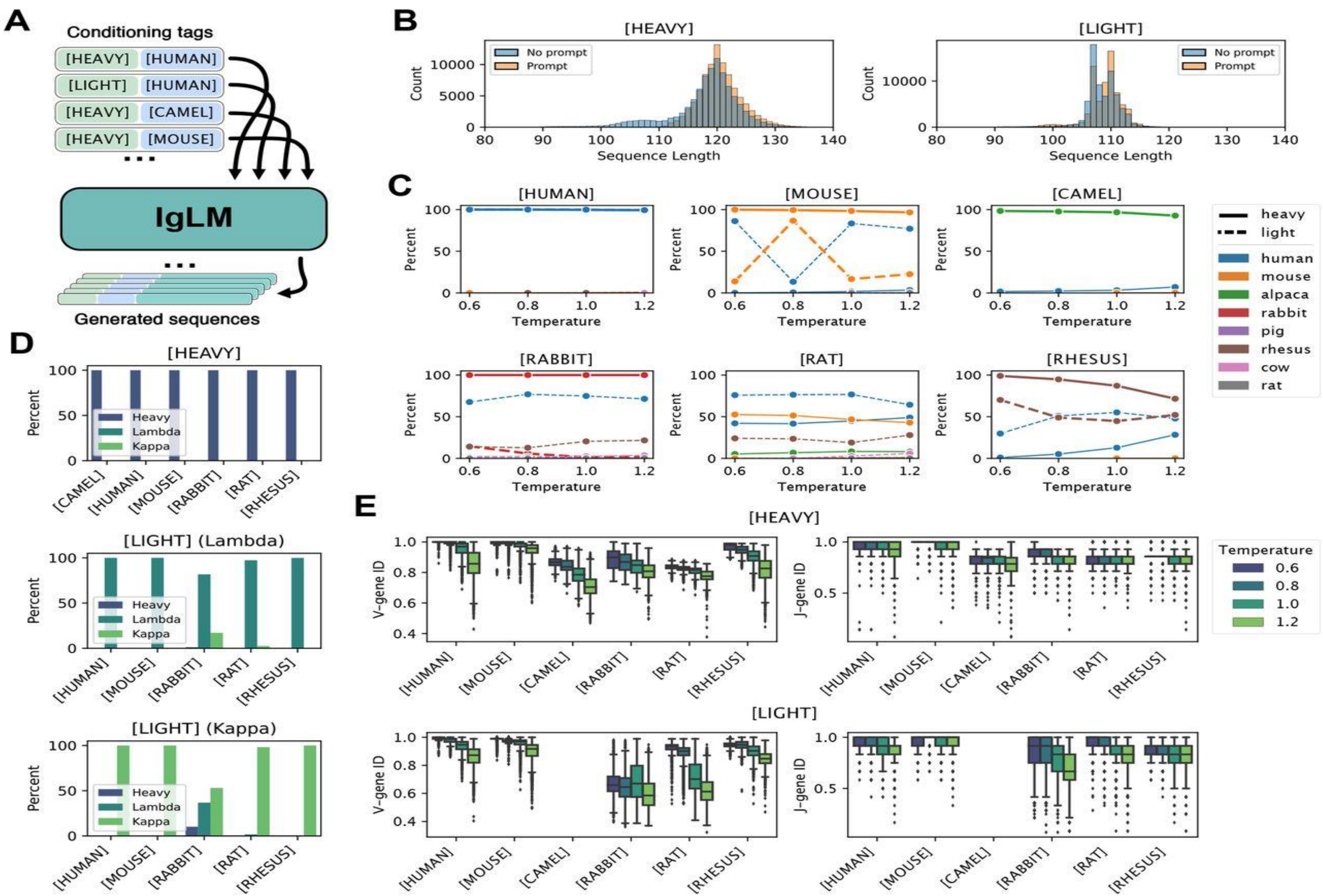
Antibody Design Examples



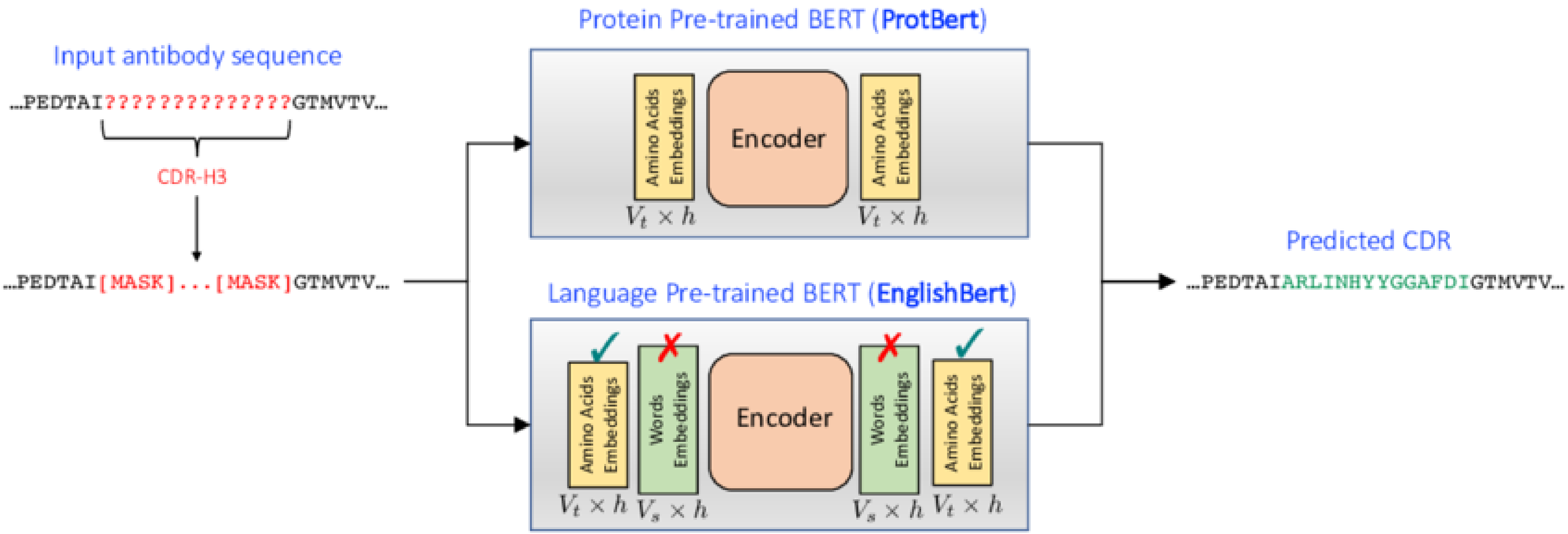
[Generative language modeling for antibody design | bioRxiv]()



A
Therapeutic Abs (without H3)
Trastuzumab
Cetuximab
Adalimumab
Pembrolizumab
IgLM
Infilled sequences
B
H3 Length
Abituzumab
Trastuzumab
Infliximab
Basiliximab
Olokizumab
Muromonab
Guselkumab
Ipilimumab
Lebrikizumab
Natalizumab
Nivolumab
Obinutuzumab
Omalizumab
Pinatuzumab
Ranibizumab
Rituximab
Sifalimumab
Gevokizumab
Fresolimumab
Ibalizumab
Efalizumab
Bapineuzumab
Atezolizumab
Bimagrumab
Canakinumab
Alemtuzumab
Certolizumab
Carlumab
Adalimumab
Daclizumab
Eculizumab
Cetuximab
Anifrolumab
Gantenerumab
Panitumumab
Briakinumab
Tralokinumab
Pertuzumab
Ponezumab
Ofatumumab
Palivizumab
Belimumab
Epratuzumab
Nimotuzumab
Motavizumab
Crenezumab
Drozitumab
Pembrolizumab
Tremelimumab
C
H3 Length
T
P
D
Parent H3
Infilled H3s
E
H3 Length
IGHV1 + IGHJ5
IGHV3 + IGHJ3
IGHV3 + IGHJ4
IGHV4 + IGHJ1
IGHV4 + IGHJ4
IGHV5 + IGHJ3
IGHV1 + IGHJ3
IGHV1 + IGHJ4
IGHV1 + IGHJ6
IGHV4 + IGHJ3
IGHV5 + IGHJ2
IGHV1 + IGHJ2
IGHV2 + IGHJ6
IGHV3 + IGHJ6
V-gene and J-gene Groups
F
Temperature
0.8
1.0
1.2
Pairwise Edit Distance
H3 Length
G
Nucleus Prob.
0.5
0.75
1.0
Pairwise Edit Distance
H3 Length



A
Conditioning tags
[HEAVY] [HUMAN]
[LIGHT] [HUMAN]
[HEAVY] [CAMEL]
[HEAVY] [MOUSE]
IgLM
Generated sequences
B
[HEAVY]
[LIGHT]
No prompt
Prompt
Count
Sequence Length
C
[HUMAN]
[MOUSE]
[CAMEL]
[RABBIT]
[RAT]
[RHESUS]
Percent
Temperature
heavy
light
human
mouse
alpaca
rabbit
pig
rhesus
cow
rat
D
[HEAVY]
[LIGHT] (Lambda)
[LIGHT] (Kappa)
Heavy
Lambda
Kappa
Percent
E
[HEAVY]
[LIGHT]
V-gene ID
J-gene ID
Temperature
0.6
0.8
1.0
1.2



REPROGRAMMING LARGE PRETRAINED LANGUAGE MODELS FOR ANTIBODY SEQUENCE INFILLING, ICLR 2023

15. Locating and Editing Factual Associations in GPT

- **We analyze** the storage and recall of factual associations in autoregressive transformer language models, finding evidence that these associations correspond to localized, directly-editable computations.
- **We first develop a causal intervention** for identifying neuron activations that are decisive in a model's factual predictions. This reveals a distinct set of steps in middle-layer feed-forward modules that mediate factual predictions while processing subject tokens.
- **To test our hypothesis** that these computations correspond to factual association recall, we modify feedforward weights to update specific factual associations using **Rank-One Model Editing (ROME).**

Meng et al. 2022

The code, dataset, visualizations, and an interactive demo notebook are available at https://rome.baulab.info/

Where are the facts kept in a Large language model or LLM?

For two reasons, we are curious about how and where a model keeps its factual relationships.

- To comprehend enormous, opaque neural networks: Large language models' internal calculations are poorly understood. Knowing huge transformer networks **requires first understanding how information is processed.**
- Making corrections: Since models are frequently inaccurate, biased, or private, we want to create techniques that will **make it possible to identify and repair specific factual inaccuracies.**

Rishabh Jain, 2022

Latest Artificial Intelligence Research Proposes ROME (Rank-One Model Editing): A Large Language Model Solution for Efficiently Locating and Editing Factual Associations in GPT Models

$$h_i^0 = emb(x_i) + Pos(i) \in R^H,$$

$$h_i^{(l)} = h_i^{(l-1)} + a_i^{(l)} + m_i^{(l)}$$

$$l = 0, 1, 2, \ldots$$

i

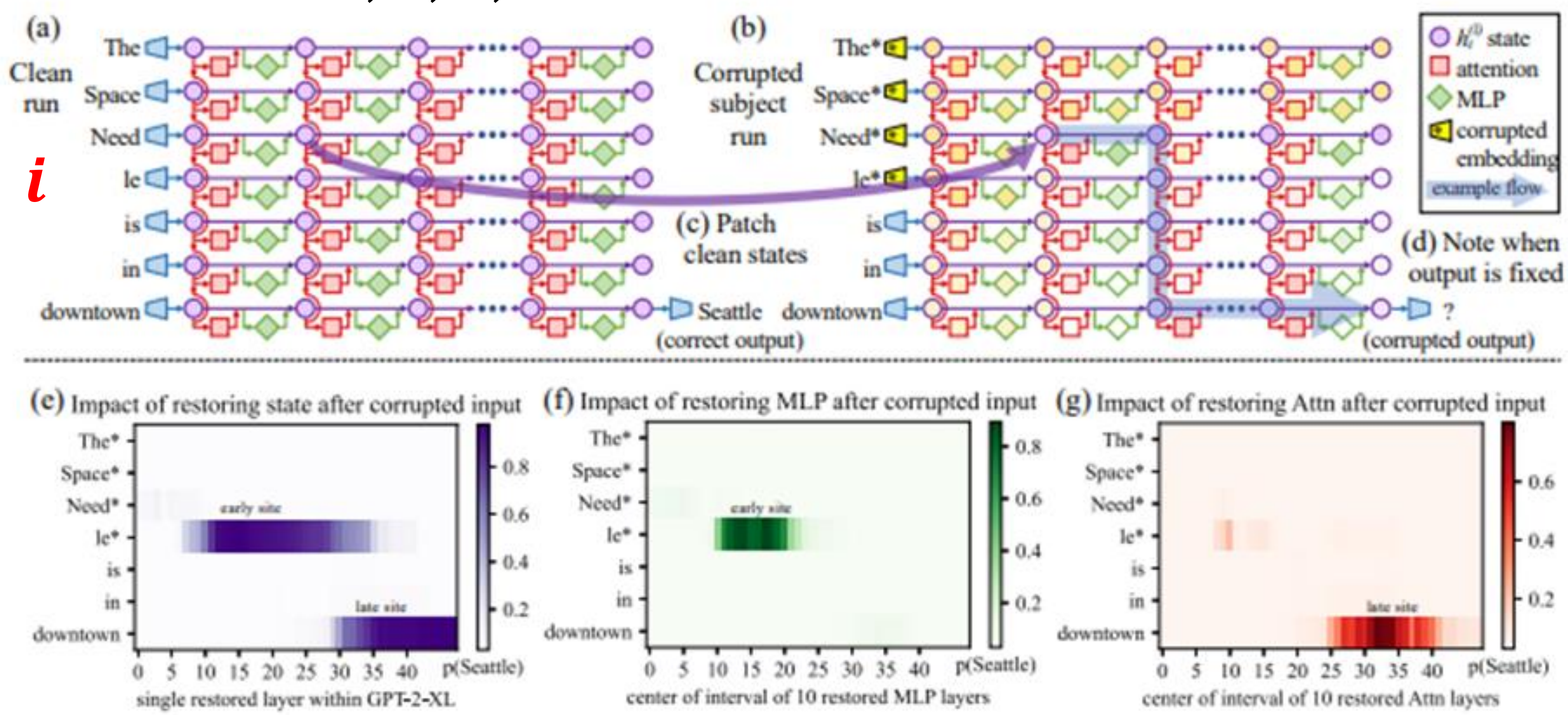
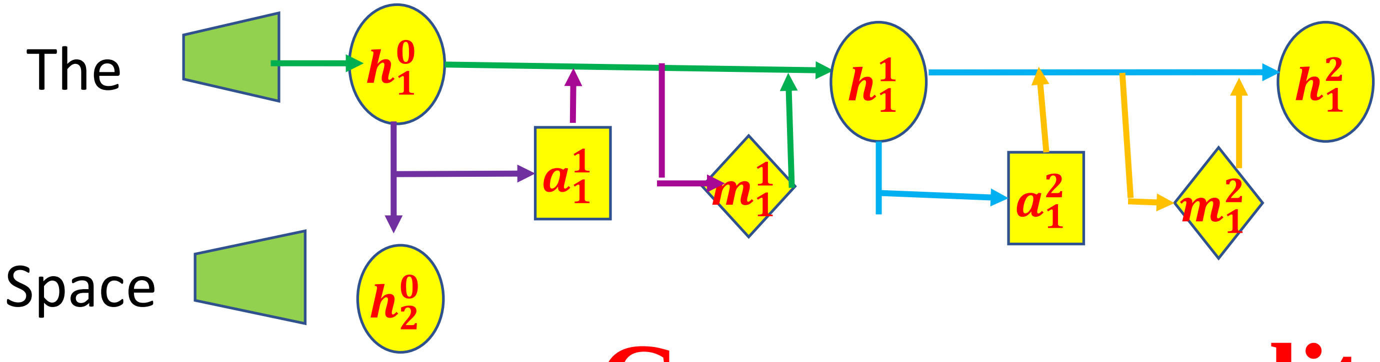


Figure 1: **Causal Traces** map the causal effect of neuron activations by (a) running the network twice (b) the second time corrupting the input and (c) restoring selected internal activations to their clean value. (d) Some sets of activations cause the output to return to the original prediction; the light blue path shows an example of information flow. The causal impact on output probability is mapped: for (e) each hidden state's effect on the prediction; and (f) the effect of only MLP contributions; and (g) the effect of only attention contributions.

$$a_i^{(l)} = atten^{(l)}\left(h_1^{(l-1)}, \ldots, h_i^{(l-1)}\right) \qquad m_i^{(l)} = W_{Proj}^{(ol)}\sigma(W_{fc}^{(l)}\gamma\left(a_i^{(l)} + h_i^{(l-1)}\right), y = \text{decode}(h_T^{\{L\}})$$

$$h_i^0 = emb(x_i) + Pos(i) \in R^H, h_i^{(l)} = h_i^{(l-1)} + a_i^{(l)} + m_i^{(l)}$$

$$a_i^{(l)} = atten^{(l)}\left(h_1^{(l-1)}, \ldots, h_i^{(l-1)}\right), m_i^{(l)} = W_{Proj}^{(l)} \sigma(W_{fc}^{(l)} \gamma\left(a_i^{(l)} + h_i^{(l-1)}\right), \mathrm{y} = \mathrm{decode}(h_T^{\{L\}})$$



Granger causality

$$h_1^{(L)} = f_1^{(L)}(x_1) = f_1^L(h_1^{L-1})$$
$$h_2^{(L)} = f_2^{(L)}(x_1, x_2) = f_2^{(L)}(h_1^{(L-1)}, h_2^{(L-1)})$$

……………………………………

$$h_T^{(L)} = f_T^L(x_1, \ldots, x_T) = f_T^L\left(h_1^{(L-1)}, h_2^{(L-1)}, \ldots, h_T^{(L-1)}\right)$$

Bivariate Linear Granger Causality Test

Consider two single variable time series X and Y. Suppose that X is a potential cause and Y is an effect or response. We assume that both time series X and Y are stationary. Define two linear time series models. The first model is the full model that includes all the past information of both cause X and response Y:

$$\boldsymbol{Y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{j=1}^{p} \beta_j X_{t-j} + \varepsilon_t \, , t = 1, \dots, T,}$$

The second model is the restricted model that includes only the past data of the response Y:

$$\boldsymbol{Y_t = \gamma_0 + \sum_{i=1}^{p} \gamma_i Y_{t-i} + e_t \, , t = 1, \dots, T,}$$

Multivariate Granger Causality Test

- **Define a VAR model:**

$$Y_t = A_0 + A(L)Y_{t-1} + e_t \,,$$

$$Y_t = \begin{bmatrix} Y_{1t} \\ \vdots \\ Y_{nt} \end{bmatrix}, A_0 = \begin{bmatrix} A_{10} \\ \vdots \\ A_{n0} \end{bmatrix}, A(L) = \begin{bmatrix} A_{11}(L) & \cdots & A_{1n}(L) \\ \vdots & \vdots & \vdots \\ A_{n1}(L) & \cdots & A_{nn}(L) \end{bmatrix}, Y_{t-1} = \begin{bmatrix} Y_{1,t-1} \\ \vdots \\ Y_{n,t-1} \end{bmatrix}, e_t = \begin{bmatrix} e_{1t} \\ \vdots \\ e_{nt} \end{bmatrix}$$

L is the backward operation where $LY_t = Y_{t-1}, L^pY_t = Y_{t-p}, A_{ij}(L) = a_{ij}(1)L + a_{ij}(2)L^2 + \cdots a_{ij}(p)L^p$, residuals e_t are distributed as a normal $N(0, \Sigma)$.

Suppose that w test the linear Granger causality relationship between two vectors of time series :

$$\boldsymbol{X_t} = \begin{bmatrix} \boldsymbol{X_{1t}} \\ \vdots \\ \boldsymbol{X_{n_1 t}} \end{bmatrix} \text{ and } \boldsymbol{Y_t} = \begin{bmatrix} \boldsymbol{Y_{1t}} \\ \vdots \\ \boldsymbol{Y_{n_2 t}} \end{bmatrix}, \text{ where } \boldsymbol{n} = \boldsymbol{n_1} + \boldsymbol{n_2}\ .$$

Consider the following VAR model:

$$\begin{bmatrix} \boldsymbol{X_t} \\ \boldsymbol{Y_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{A_{xx}(L)} & \boldsymbol{A_{xy}(L)} \\ \boldsymbol{A_{yx}(L)} & \boldsymbol{A_{yy}(L)} \end{bmatrix} \begin{bmatrix} \boldsymbol{X_{t-1}} \\ \boldsymbol{Y_{t-1}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e_x} \\ \boldsymbol{e_y} \end{bmatrix},$$

There are four different cases of causal relationships between two vectors of time series $\boldsymbol{X_t}$ and $\boldsymbol{Y_t}$ (Bai et al. 2010)

(1) If $\boldsymbol{A_{xy}(L)}$ is significantly different from the zero, while $\boldsymbol{A_{yx}(L)}$ shows no significantly different from zero, then there exists a unidirectional Ganger causality from time series $\boldsymbol{Y_t}$ to $\boldsymbol{X_t}$;

(1) If $A_{yx}(L)$ is significantly different from zero, while $A_{xy}(L)$ shows no significantly difference from zero, then there exists a unidirectional Ganger causality from X_t to Y_t;

(2) If both coefficients $A_{xy}(L)$ and $A_{yx}(L)$ are significantly different from zero, then there exists bidirectional Granger causality between X_t and Y_t;

(3) If both coefficients $A_{xy}(L)$ and $A_{yx}(L)$ are not significantly different from zero, then X_t and Y_t are not rejected to be independent.

Nonlinear Granger Causality Test

See the book:
Xiong, MM (2022) Artificial Intelligence and Causal Inference, CRC Press.

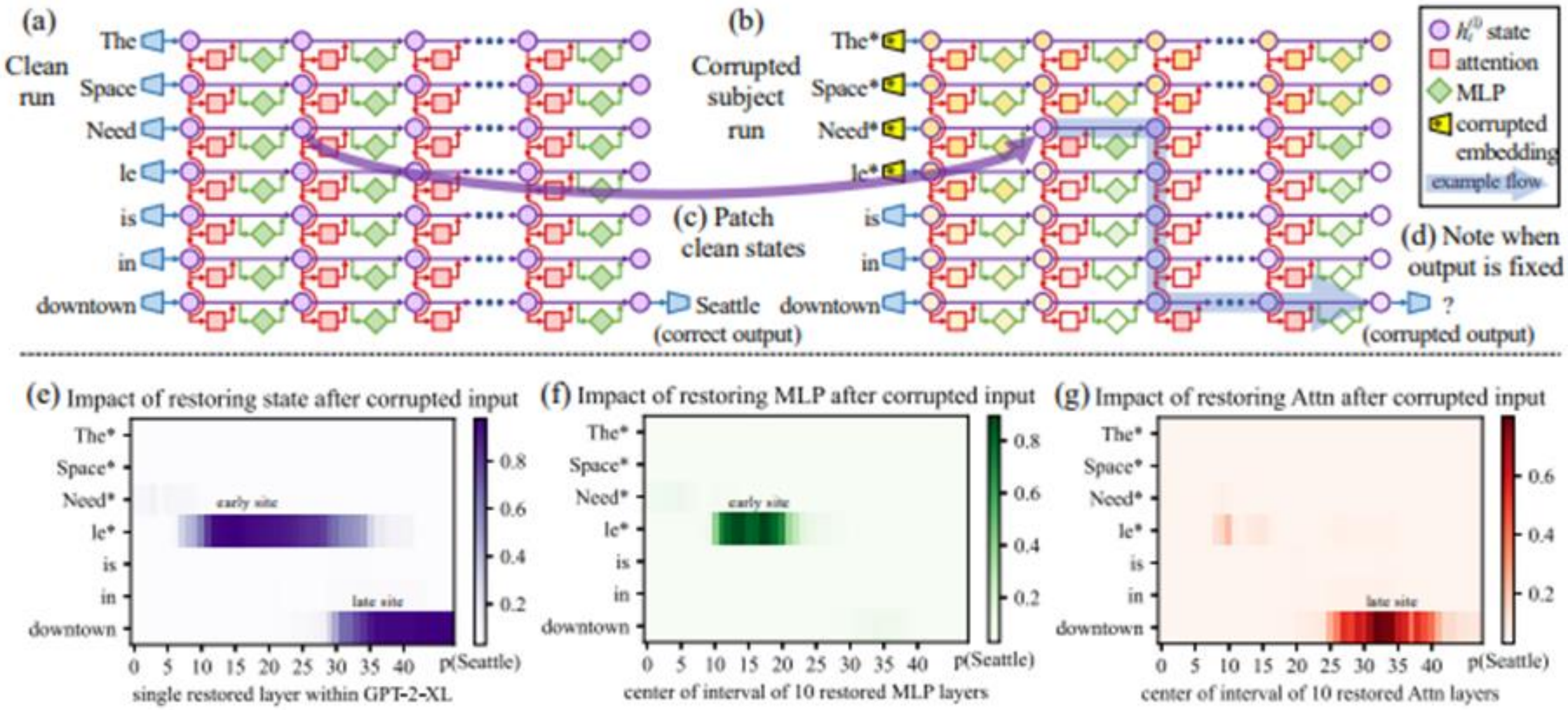


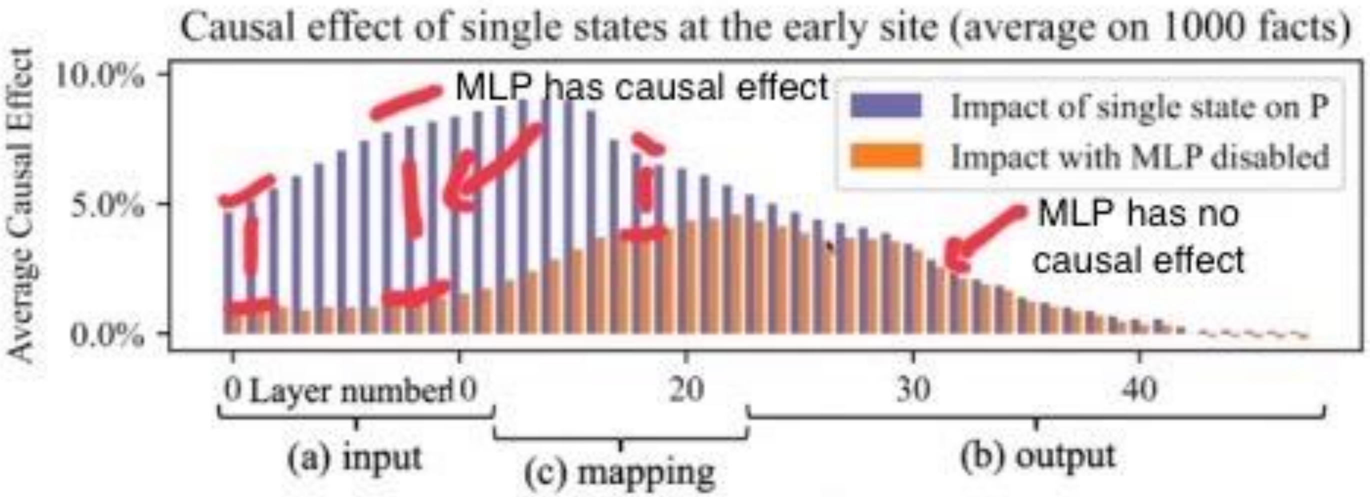
Figure 1: **Causal Traces** map the causal effect of neuron activations by (a) running the network twice (b) the second time corrupting the input and (c) restoring selected internal activations to their clean value. (d) Some sets of activations cause the output to return to the original prediction; the light blue path shows an example of information flow. The causal impact on output probability is mapped: for (e) each hidden state's effect on the prediction; and (f) the effect of only MLP contributions; and (g) the effect of only attention contributions.

To calculate each state's contribution towards a correct factual prediction, we observe all of G's internal activations during three runs:

- **a clean run that predicts the fact,**
- **a corrupted run where the prediction is damaged, and**
- **A corrupted-with-restoration run that tests the ability of a single state to restore the prediction.**

Causal Tracing

Early Site with **MLP disabled**



Low layer state:
no effect
without MLP

High layer state: MLP
not needed for effect

The Localized Factual Association Hypothesis

Based on causal traces, we posit a specific mechanism for storage of factual associations: **each midlayer MLP module accepts inputs that encode a subject, then produces outputs that recall memorized properties about that subject**. Middle layer MLP outputs accumulate information, **then the summed information is copied to the last token by attention at high layers**.

This hypothesis localizes factual association along three dimensions, **placing it (i) in the MLP modules** (ii) **at specific middle layers** (iii) and **specifically at the processing of the subject's last token.**

Interventions on Weights for Understanding Factual Association Storage

- ## Rank-One Model Editing (ROME)

Set of vector keys: $K = [k_1|k_2|\dots]$

Set of vector values: $V = [v_1|v_2|\dots]$

- ## Linear Association

$$v = Wk$$

$$v^* = \widehat{W}k^*$$

$$\min_{\widehat{W}} \|\widehat{W}K - V\|$$

s.t. $\widehat{W}k^* = v^*$

$$\widehat{W} = W + \Lambda(C^{-1}k^*)^T$$

$$C = KK^T$$

Procedure

- **Step 1: Choosing k^* to Select the Subject**

Based on the decisive role of MLP inputs at the final subject token, we shall choose inputs that represent the subject at its last token as the lookup key k^*. Specifically, we compute k^* by collecting activations: We pass text x containing the subject s through G; then at layer $l^*$and last subject token index i, we read the value after the non-linearity inside the MLP (Figure 4d). Becaus the state will vary depending on tokens that precede s in text, we set k^* to an average value over small set of texts ending with the subject s

$$k^* = \frac{1}{N}\sum_{j=1}^{N} k(x_j + s), k(x) = \sigma(W_{fc}^{(l^*)}\gamma(a_{[x],i}^{(l^*)} + h_{[x],i}^{(l^*-1)})$$

- **Step 2: Choosing v^* to Recall the Fact.**

$$v^* = \underset{z}{argmin}\, L(z)$$

prompt p, p'

$$L(z) = \frac{1}{N}\sum_{j=1}^{N} -\log P_G\left(m_i^{(l^*)} = z\right)\left[O^*\middle|x_j + P\right] + D_{KL}\left(P_G\left(m_i^{(l^*)} = z\right)[x|P']||P_G[x|P']\right)$$

Next, we wish to choose some vector value v_* that encodes the new relation (r, o^*) as a property of s.

$$P_i(x_i^L) = softmax\left(\delta(x_i^L)\right), \delta(x_i^L) = Ex_i^L, or\ \delta(x_i^L) = Wx_i^L + u, W \in R^{|V| \times d}. u \in R^{|V|}$$

The first term (Eqn. 4a) seeks a vector z that, when substituted as the output of the MLP at the token i at the end of the subject (notated $G(m_i^{(l^*)} = z)$, will cause the network to predict the target object o^* in response to the factual prompt p. The second term (Eqn. 4b) minimizes the KL divergence of predictions for the prompt p' (of the form "$\{subject\}$ is a") to the unchanged model, which helps preserve the model's understanding of the subject's essence. To be clear, the optimization does not directly alter model weights; it identifies a vector representation that, when output at the targeted MLP module, represents the new property (r, o^*) for the subject s. Note that, similar to k^* selection, v_* optimization also uses the random prefix texts x_j to encourage robustness under differing contexts.

- **Step 3: Inserting the Fact.**

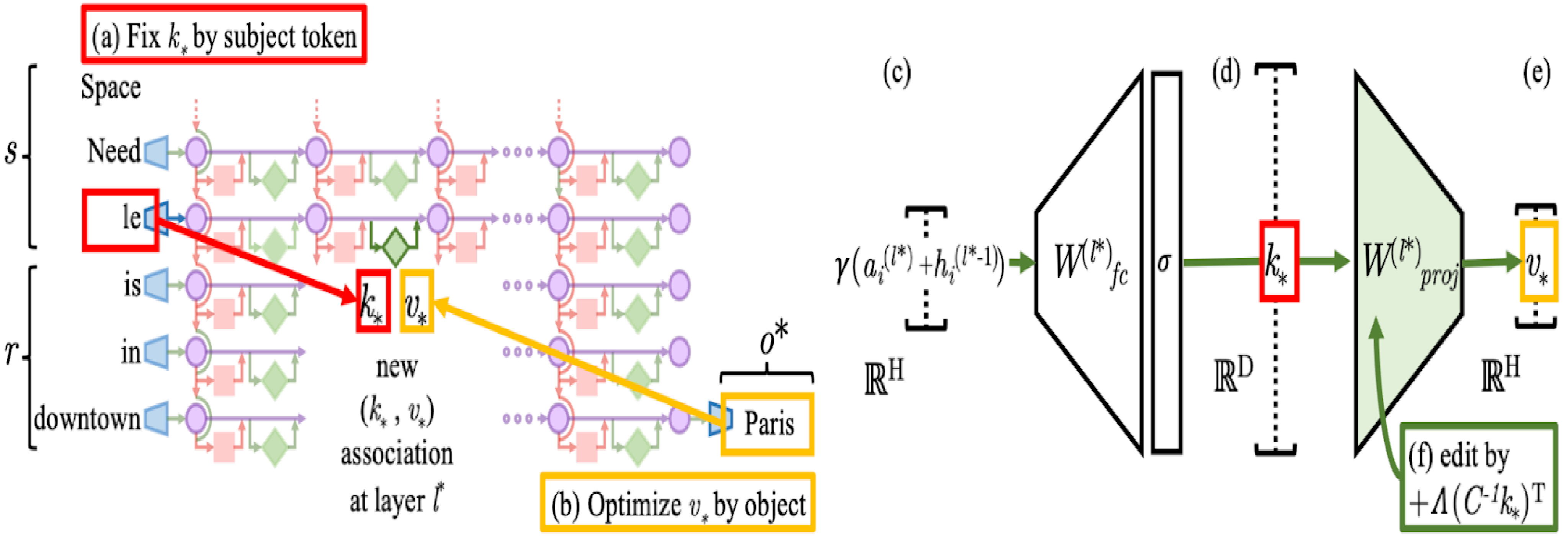
Once we have computed the pair (k^*, v^*) to represent the full fact (s, r, o^*), we apply Eqn. 2, updating the MLP weights $W_{proj}^{(l)}$ with a rank-one update

$$\hat{W}_{proj}^{(l)} = W_{proj}^{(l)} + \Lambda(C^{-1}k^*)^T$$

$$C = KK^T$$

is a constant that we pre-cache by estimating the uncentered covariance of K from a sample of Wikipedia text

$$\Lambda = \frac{v^* - W_{proj}^{(l)}}{(C^{-1}k^*)^T k^*}$$



Editing one MLP layer with ROME. To associate Space Needle with Paris, the ROME method inserts a new (k; v) association into layer l, where **(a) key k** is determined by the subject and **(b) value v** is optimized to select the object. **(c) Hidden state at layer *l* and token i** is expanded to produce (d) the key vector k for the subject. (e) To write new value vector v into the layer, (f) we calculate a rank-one update $\Lambda(C^{-1}k_*)^T$ to cause $\widehat{W}_{proj}^{(l^*)}k_* = v_*$ while minimizing interference with other memories stored in the layer

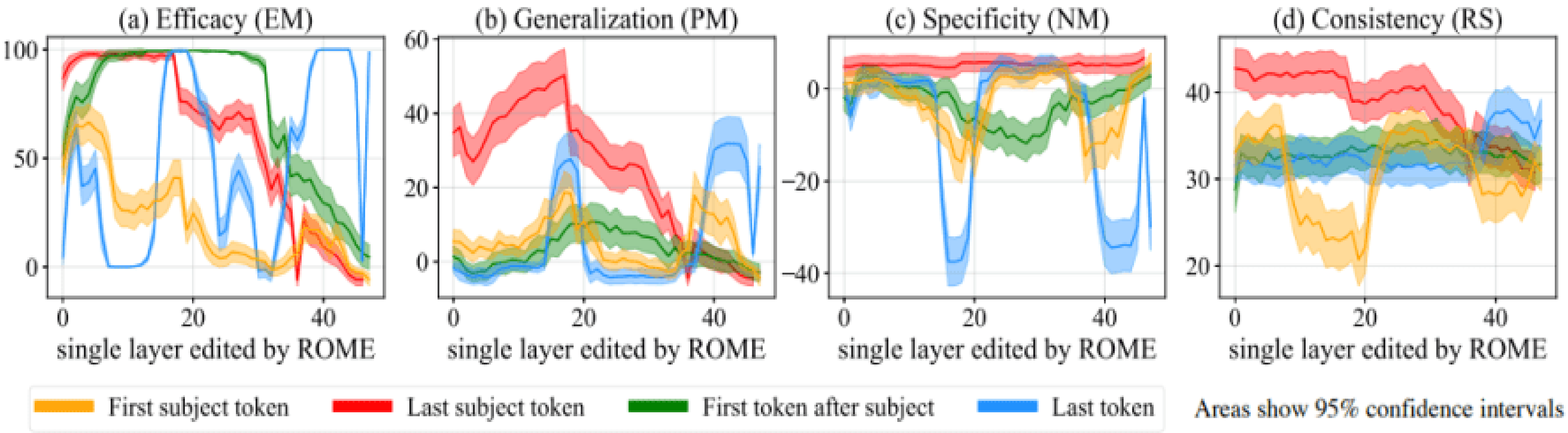


Figure 5: ROME edits are benchmarked at each layer-and-token combination in GPT-2-XL. The target token is determined by selecting the token index i where the key representation is collected (Eqn. 3). ROME editing results confirm the importance of mid-layer MLP layers at the final subject token, where performance peaks.

References

Geva al. 2023 April

Dissecting Recall of Factual Associations in Auto-Regressive Language Models

Cohen et al. 2023; Crawling The Internal Knowledge-Base of Language Models

Cao et al. 2021; Editing Factual Knowledge in Language Models

Tracr: Compiled Transformers as a Laboratory for Interpretability
David Lindner et al. 2023

Wang et al. 2022; INTERPRETABILITY IN THE WILD: A CIRCUIT FOR NDIRECT OBJECT IDENTIFICATION IN GPT-2 SMALL

Code for all experiments is available at ttps://github.com/redwoodresearch/Easy-Transformer.

Explaining patterns in data with language models via interpretable autoprompting
C. Singh, J.X. Morris, J. Aneja, A.M. Rush, J. Gao.

Lester et al. 2021; The Power of Scale for Parameter-Efficient Prompt Tuning

Singh et al. 2023; iPrompt: Explaining Data Patterns in Natural Language via Interpretable Autoprompting

All code for using the methods and data here is made available on Github.

Zhou et al. 2023; LARGE LANGUAGE MODELS ARE HUMAN-LEVEL ROMPT ENGINEERS

Our code is available https://github.com/keirp/automatic_prompt_engineer.

Mangrulkar and Paul 2023PEFT: Parameter-Efficient Fine-Tuning of Billion-Scale Models on Low-Resource Hardware