# STAT 992: Science of Large Language Models

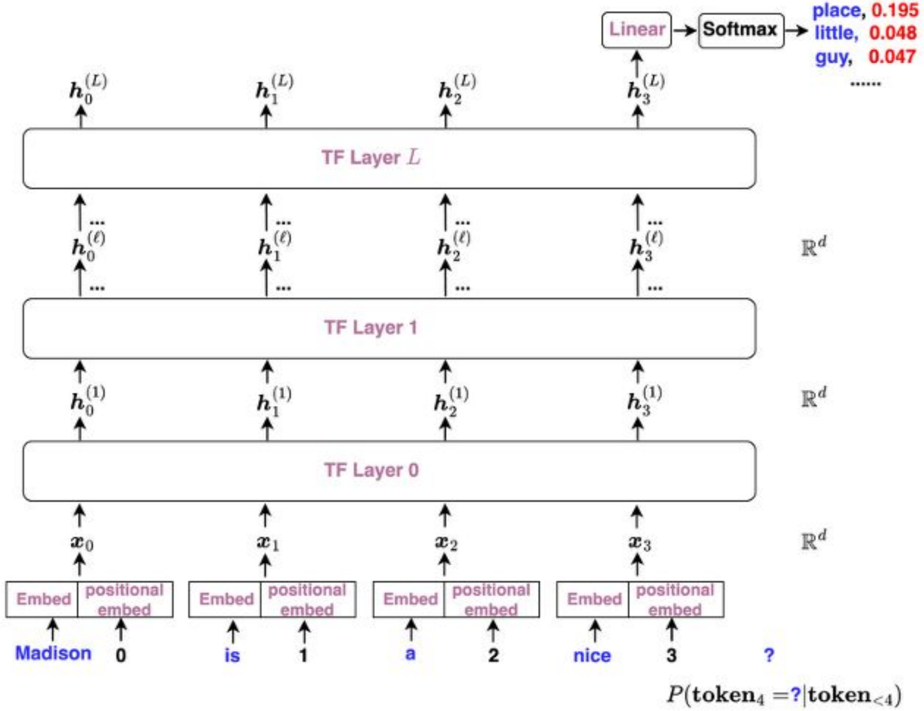
# Lecture 5: Linear representation hypothesis, feature superposition

Spring 2026
Yiqiao Zhong

# How do LLMs encode concepts and rules?

- **Main goals:** understand how geometry of the *hidden states* represents semantics, syntactics, compositions, etc.

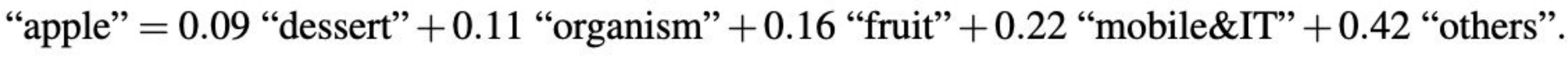
- How model components “store” knowledge and interact with contexts



# Overview of main findings

- **Linear representation hypothesis:** transformers represent concepts as low-dim linear subspaces (esp. vectors) in the hidden states space

- **Feature superposition**: hidden states are approximately a sparse linear combination of base concepts, e.g.,



$$\text{``apple''} = 0.09\ \text{``dessert''} + 0.11\ \text{``organism''} + 0.16\ \text{``fruit''} + 0.22\ \text{``mobile\&IT''} + 0.42\ \text{``others''}.$$

- Hidden states can encode richer and more contextualized concepts

*Table edited by Gemini-3*

# Isn’t that familiar?

- **Sparse coding:** input vectors are a linear representation of basis vectors

$$\boldsymbol{x} = \sum_{j=1}^{K} a_j \boldsymbol{\varphi}_j$$

  - Complete basis: PCA
  - Over-complete basis: dictionary learning

- **What’s new: efficient representation learning.**
  - Semantic-rich dictionary through many layers and large context
  - Scalable training: massive dataset and model size

*Table edited by Gemini-3*

# LRH in pre-transformer age

# PCA and factor models

- **SVD and spectral decomposition:** Given a data matrix $X$ , calculate the singular value decomposition, or equivalent spectral decomposition

$$\boldsymbol{X} = \boldsymbol{U\Sigma V}^\top, \qquad \boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{V\Sigma}^\top \boldsymbol{\Sigma V}^\top$$

- **PCA gives best low-rank approximation.** Using top-$r$ singular vectors and singular values, $\hat{X} = U_{\leq r}\Sigma_{\leq r}V_{\leq r}^\top$ solves

$$\left\| \boldsymbol{X} - \hat{\boldsymbol{X}} \right\|_F^2, \quad \text{s.t. } \operatorname{rank}(\hat{\boldsymbol{X}}) \leq r$$

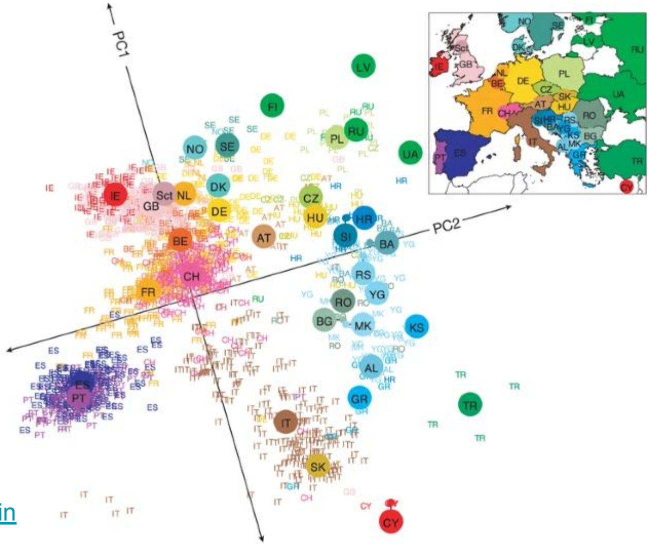
- **Factor model** interpretation: each row is linear combination of a few dominant factor vectors

# Classical data analysis

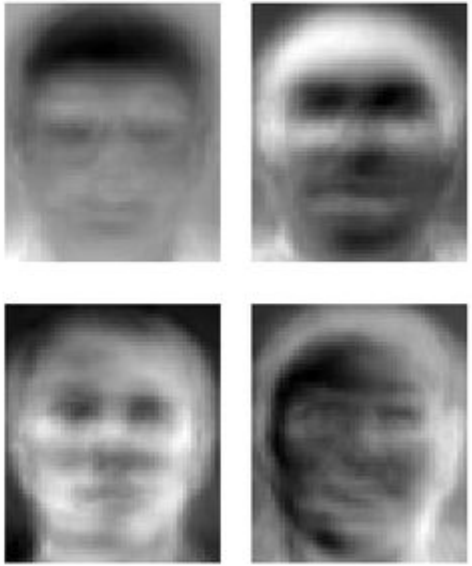
- Eigenface: decompose a face image as a linear combinations

Face image$_1$ = (23% of $E_1$) + (2% of $E_2$) + (51% of $E_3$) + ... + (1% $E_n$).

- **Gene expression data analysis**: principal components of gene data mirror geography



Genes mirror geography within Europe, Nature, 2026

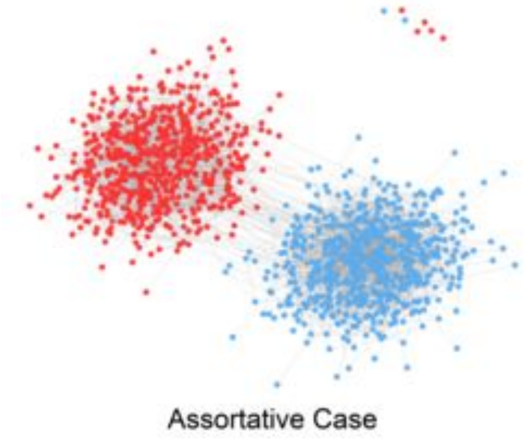


From Wiki: Some eigenfaces from AT&T Laboratories Cambridge

# Linear representation and low-rankness

- Low-rank structures underlies interpretable linear features
- Spectral method in broader applications: network data analysis

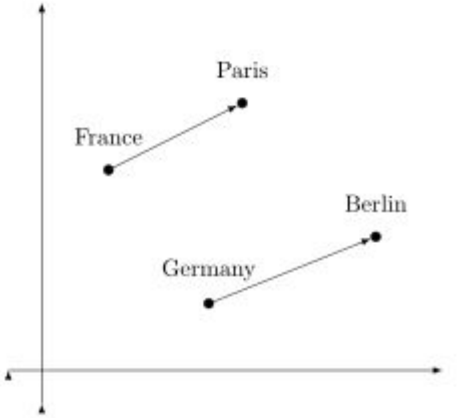
- Example: stochastic block model (SBM)
  - Connectivity prob within blocks higher
  - In expectation, adjacency matrix is a rank-2 matrix
  - Applying spectral decomposition on one observed matrix
  - Top-2 eigenvectors encode “membership” of nodes



From Wiki: SBM with two blocks

# Word embedding

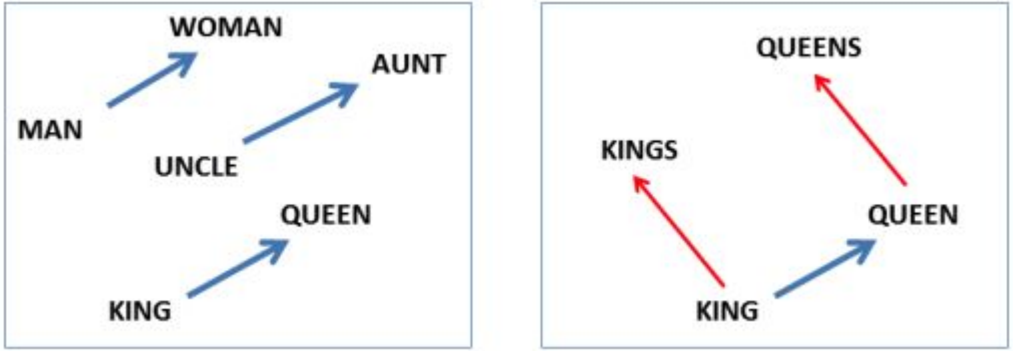
- **Aim**: find embeddings (vector representations) of words / tokens
- Overcoming **prior difficulty**: n-gram models and hidden markov chains not aimed at capturing token semantics
- A transition point in NLP: vector representations are effective for modeling discrete sequence data.
    - Solves polysemy
    - Ideal for neural networks



From Wiki: word embedding

# Word embedding

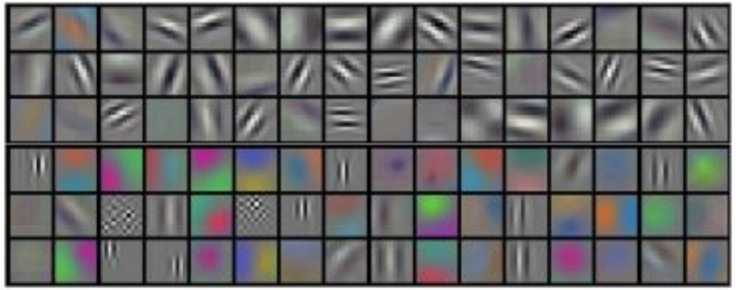
- Similar ideas developed by several groups (2013–2014)
  - Word2vec
  - Glove
- Glove: simple nonlinear matrix factorization finds good word embeddings
  - Co-occurrence matrix: word-word frequency counts within a window
  - Under nonlinear transform, finds low-rank matrix
- Linear concept representations
  - Composition via additivity



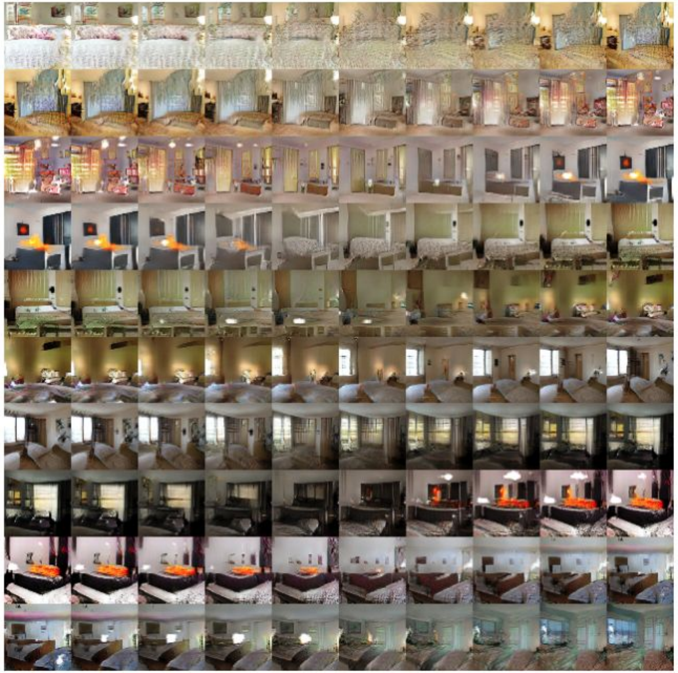
Linguistic Regularities in Continuous Space Word Representations, 2013

# LRH are also common for non-language data

- Kernels in CNN are well known to extract hierarchical features
- Generative models such as autoencoders and GANs encode meaning concepts / scenes / objects linearly in latent space



AlexNet paper, 2012: visualizing first-layer conv kernels



DCGAN paper, 2016: linear interpolation in latent space

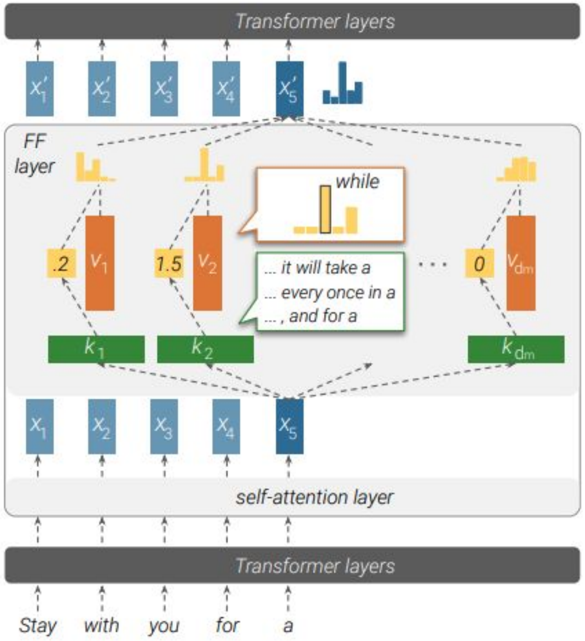
# Interpretability of transformers with LRH

# Analyzing MLP layers in a transformer

- Transformer Feed-Forward Layers Are Key-Value Memories
- An FFN layer within a transformer is simply a two-layer MLP:

$$\mathrm{FF}(\boldsymbol{x}) = \sum_{j=1}^{D} \sigma(\boldsymbol{x}^\top \boldsymbol{k}_j)\boldsymbol{v}_j$$

- Neural memory interpretation
  - $\boldsymbol{k}_j$ is a key
  - $\boldsymbol{v}_j$ is a value
  - An embedding $\boldsymbol{x}$ matches a key if the inner product is large, then activates the corresponding value

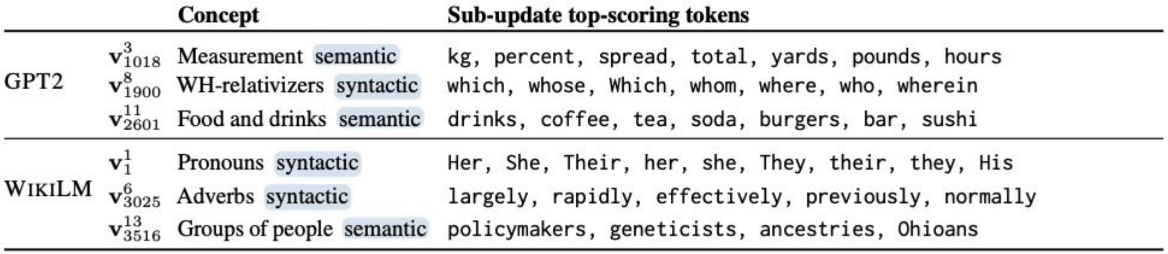


Transformer Feed-Forward Layers Are Key-Value Memories, 2021

# Analyzing MLP layers in a transformer

- **Interpreting** value vectors $v_i$: projection with unembedding matrix
  - **Token embedding** maps tokens to embeddings
  - **Unembedding matrix** (namely final classification weight matrix) maps embeddings to vectors of len of vocab size, which after softmax converted to prob distribution over the vocab
- Top tokens from the prob distribution represents meaning of value vectors
- Interpreting FFN as many “sub-updates” of concepts, additively

$$\mathbf{p}_i^{\ell} = \text{softmax}(\mathbf{v}_i^{\ell} \cdot E).$$

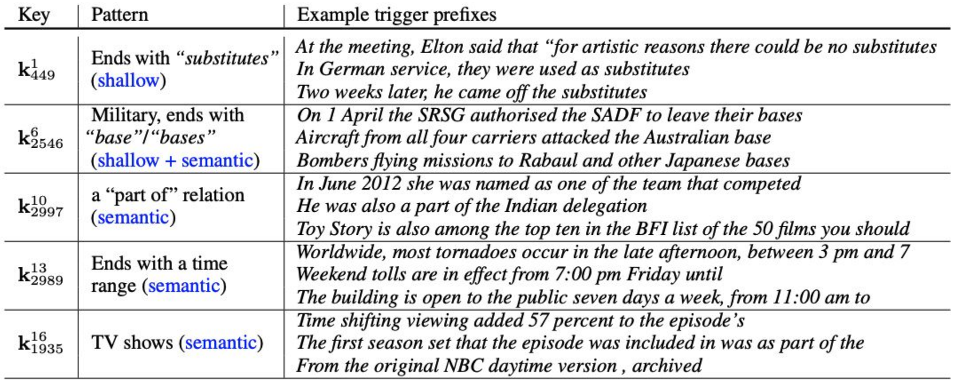


| | | Concept | Sub-update top-scoring tokens |
|---|---|---|---|
| GPT2 | $\mathbf{v}^{3}_{1018}$ | Measurement semantic | kg, percent, spread, total, yards, pounds, hours |
| | $\mathbf{v}^{8}_{1900}$ | WH-relativizers syntactic | which, whose, Which, whom, where, who, wherein |
| | $\mathbf{v}^{11}_{2601}$ | Food and drinks semantic | drinks, coffee, tea, soda, burgers, bar, sushi |
| WikiLM | $\mathbf{v}^{1}_{1}$ | Pronouns syntactic | Her, She, Their, her, she, They, their, they, His |
| | $\mathbf{v}^{6}_{3025}$ | Adverbs syntactic | largely, rapidly, effectively, previously, normally |
| | $\mathbf{v}^{13}_{3516}$ | Groups of people semantic | policymakers, geneticists, ancestries, Ohioans |

Build Predictions by Promoting Concepts in the Vocabulary Space, 2022

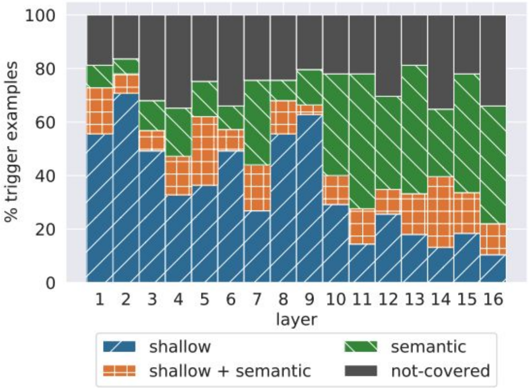
# Analyzing MLP layers in a transformer

- Simple strategy: interpreting vectors as top-related tokens
- Often (but not always) useful
- A related strategy: find prompts that activate a key-value pair the most



| Key | Pattern | Example trigger prefixes |
|---|---|---|
| $\mathbf{k}^{1}_{449}$ | Ends with *"substitutes"* (shallow) | *At the meeting, Elton said that "for artistic reasons there could be no substitutes*<br>*In German service, they were used as substitutes*<br>*Two weeks later, he came off the substitutes* |
| $\mathbf{k}^{6}_{2546}$ | Military, ends with *"base"/"bases"* (shallow + semantic) | *On 1 April the SRSG authorised the SADF to leave their bases*<br>*Aircraft from all four carriers attacked the Australian base*<br>*Bombers flying missions to Rabaul and other Japanese bases* |
| $\mathbf{k}^{10}_{2997}$ | a "part of" relation (semantic) | *In June 2012 she was named as one of the team that competed*<br>*He was also a part of the Indian delegation*<br>*Toy Story is also among the top ten in the BFI list of the 50 films you should* |
| $\mathbf{k}^{13}_{2989}$ | Ends with a time range (semantic) | *Worldwide, most tornadoes occur in the late afternoon, between 3 pm and 7*<br>*Weekend tolls are in effect from 7:00 pm Friday until*<br>*The building is open to the public seven days a week, from 11:00 am to* |
| $\mathbf{k}^{16}_{1935}$ | TV shows (semantic) | *Time shifting viewing added 57 percent to the episode's*<br>*The first season set that the episode was included in was as part of the*<br>*From the original NBC daytime version , archived* |

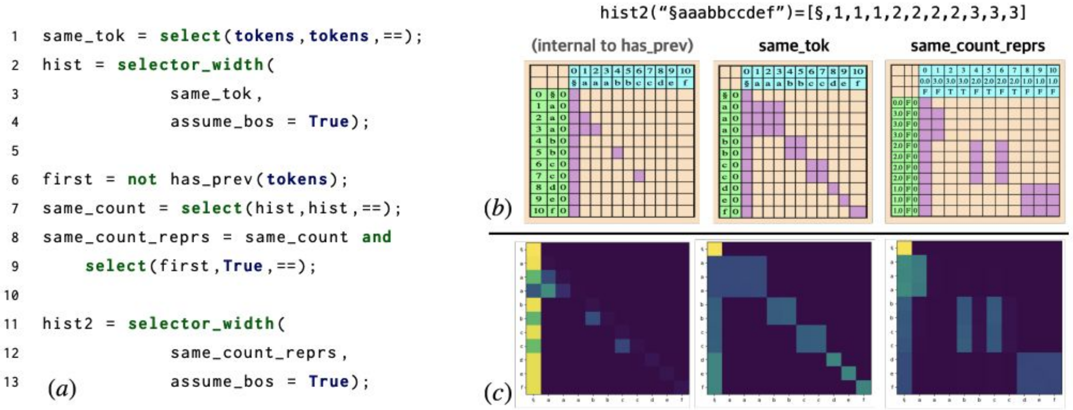
Transformer Feed-Forward Layers Are Key-Value Memories, 2021



# How do transformer encode and process semantics?

- **Heuristics of transformers**
    - MLPs / FFNs store static knowledge using key-vector memories, encoding progressively rich semantics in later layers
    - Self-attentions implements algorithms by mixing and composing token / position information

- **Layer specification**: MLPs and SAs across layers can play different roles
- **Mixture of experts** (MoEs): MLP split into sparsely activated "experts" for knowledge specification

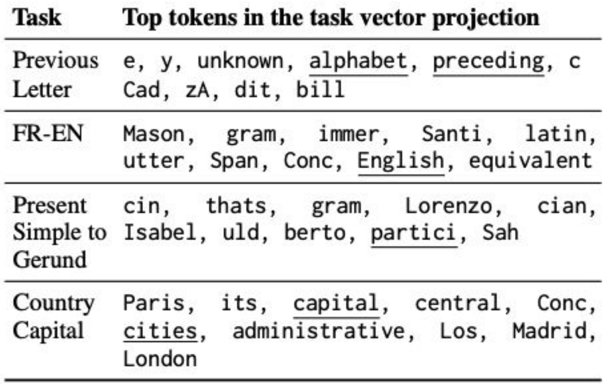


```
same_tok = select(tokens,tokens,==);
hist = selector_width(
            same_tok,
            assume_bos = True);

first = not has_prev(tokens);
same_count = select(hist,hist,==);
same_count_reprs = same_count and
    select(first,True,==);

hist2 = selector_width(
            same_count_reprs,
            assume_bos = True);
```

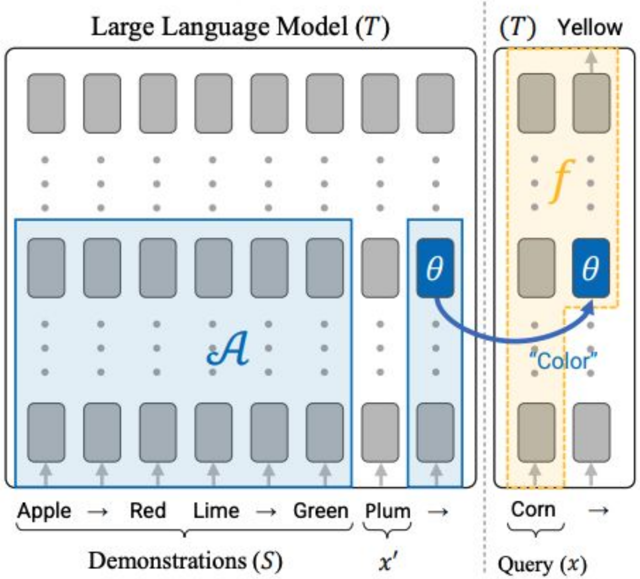
Thinking Like Transformers, 2021: implementing programs via attention patterns

# LRH and in-context learning

- An interpretation: in-context (IC) examples activate concepts, which steer the model towards that concepts
- **IC vector**: extract hidden states as concept-encoding vector
- **Inject IC vector** without context: patching this vector or adding this vector to hidden states completes task without context



| Task | Top tokens in the task vector projection |
|---|---|
| Previous Letter | e, y, unknown, alphabet, preceding, c Cad, zA, dit, bill |
| FR-EN | Mason, gram, immer, Santi, latin, utter, Span, Conc, English, equivalent |
| Present Simple to Gerund | cin, thats, gram, Lorenzo, cian, Isabel, uld, berto, partici, Sah |
| Country Capital | Paris, its, capital, central, Conc, cities, administrative, Los, Madrid, London |

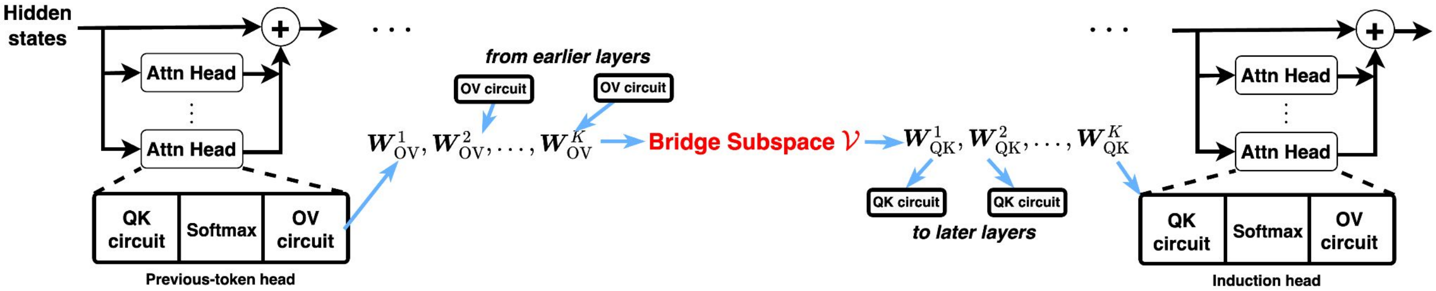


In-Context Learning Creates Task Vectors, 2023

# Representation of **compositions** via bridge subspace

- How do transformers compose two layers? How do an earlier layer communicate with a later layer?
- An shared subspace by many pairs of early-layer OV and late-later QK
- Early layer “writes” in bridge subspace, then “read and processed” by later layers

$$\mathcal{V} = \mathrm{span}(\boldsymbol{W}_{\mathrm{OV},j}) = \mathrm{span}(\boldsymbol{W}_{\mathrm{QK},k}^{\top})$$

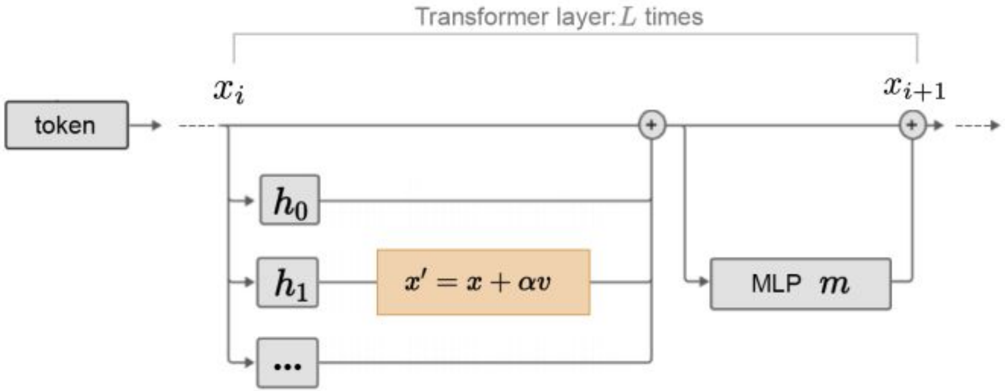

Hidden states
Attn Head
Attn Head
QK circuit
Softmax
OV circuit
Previous-token head
from earlier layers
OV circuit
OV circuit
$\boldsymbol{W}_{\mathrm{OV}}^{1}, \boldsymbol{W}_{\mathrm{OV}}^{2}, \ldots, \boldsymbol{W}_{\mathrm{OV}}^{K}$
Bridge Subspace $\mathcal{V}$
$\boldsymbol{W}_{\mathrm{QK}}^{1}, \boldsymbol{W}_{\mathrm{QK}}^{2}, \ldots, \boldsymbol{W}_{\mathrm{QK}}^{K}$
QK circuit
QK circuit
to later layers
Attn Head
Attn Head
QK circuit
Softmax
OV circuit
Induction head

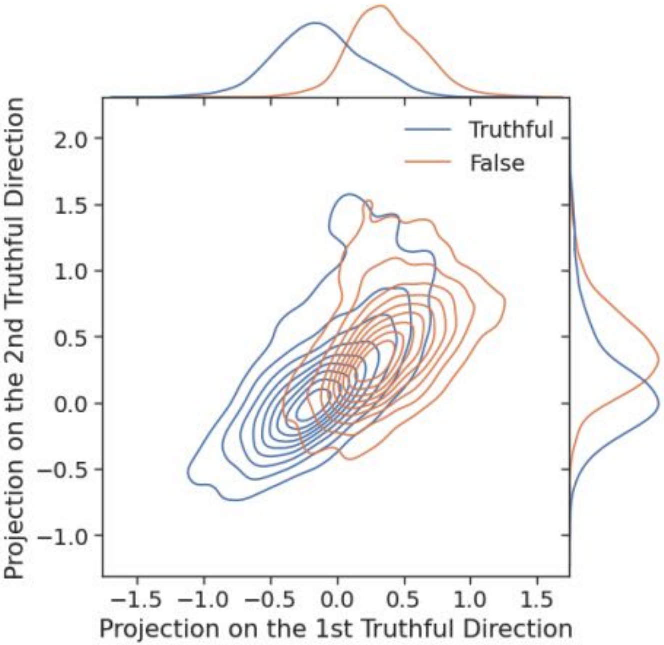
Out-of-distribution generalization via composition: A lens through induction heads in Transformers, 2025

# Model adaptation using LRH

# Inference-time intervention

- Find harmful (untruthfulness, bias) concept vectors in hidden states space
- Shift the hidden states in the opposite direction



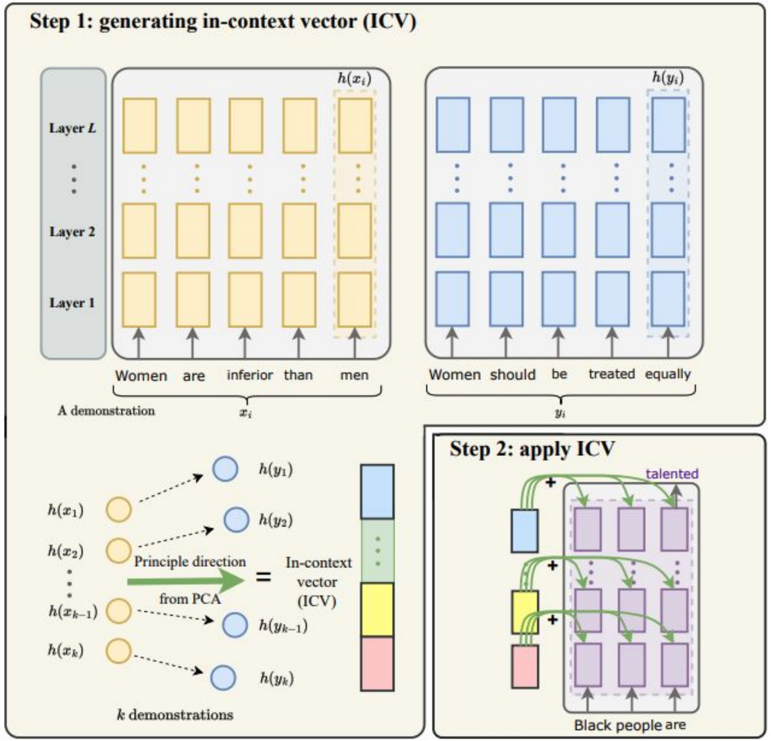


Inference-Time Intervention: Eliciting Truthful Answers from a Language Model, 2024

# Steering with IC vectors

- Compute safety-related concept vectors:
  - Prompt with paired examples
  - Calculate top principal component

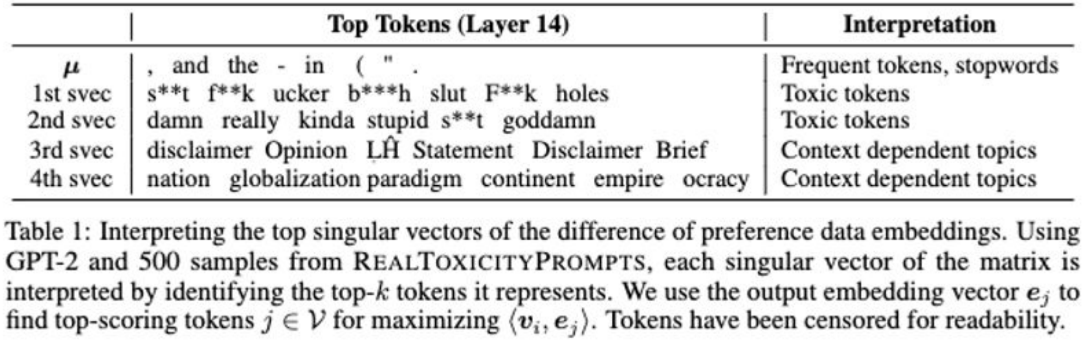
- Steer the model to reduce harmful-encoding vector



[In-context Vectors: Making In Context Learning More Effective and Controllable Through Latent Space Steering](), 2024

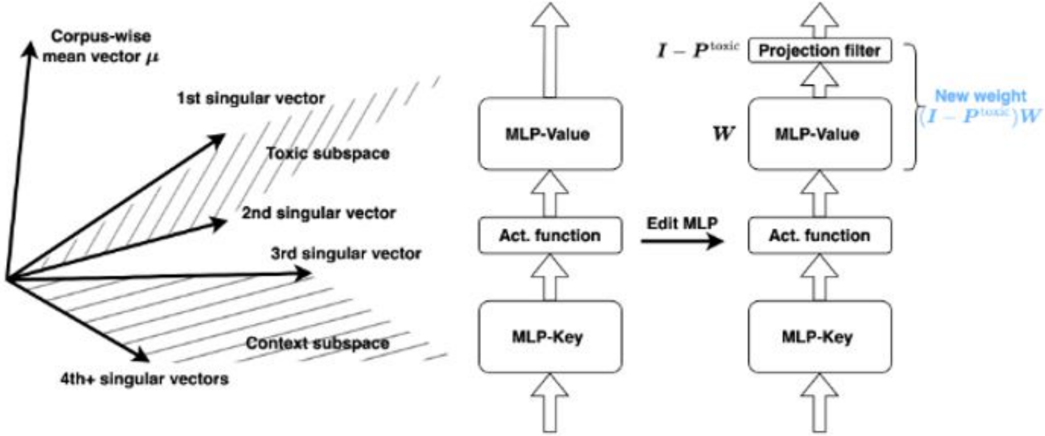
# Model editing

- Concept / task vectors can be used to edit models (low-cost finetuning)
- Detoxify LLMs: extract hidden states from paired prompts, apply PCA, then project out toxicity-encoding subspaces in MLP value matrices
- Similar variants: task arithmetic, knowledge injection via ROME



| | Top Tokens (Layer 14) | Interpretation |
|---|---|---|
| $\mu$ | , and the - in ( " . | Frequent tokens, stopwords |
| 1st svec | s**t f**k ucker b***h slut F**k holes | Toxic tokens |
| 2nd svec | damn really kinda stupid s**t goddamn | Toxic tokens |
| 3rd svec | disclaimer Opinion LĤ Statement Disclaimer Brief | Context dependent topics |
| 4th svec | nation globalization paradigm continent empire ocracy | Context dependent topics |

Table 1: Interpreting the top singular vectors of the difference of preference data embeddings. Using GPT-2 and 500 samples from REALTOXICITYPROMPTS, each singular vector of the matrix is interpreted by identifying the top-$k$ tokens it represents. We use the output embedding vector $e_j$ to find top-scoring tokens $j \in \mathcal{V}$ for maximizing $\langle v_i, e_j \rangle$. Tokens have been censored for readability.



Model Editing as a Robust and Denoised variant of DPO: A Case Study on Toxicity, 2024