Do Jet Taggers Learn Physics? Probing and Steering Latent Directions in a Transformer Tagger

Savannah Thais, Daniel Tiourine
City University of New York

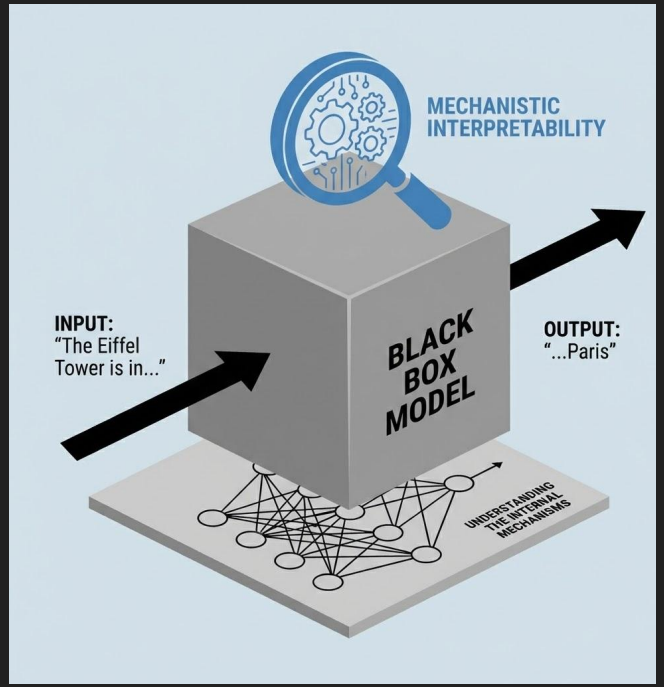
Work Led by Daniel Tiourine

- Super talented researcher!
- Currently applying to PhDs!

Background

What is Mechanistic Interpretability?

- A growing subfield that aims to understand how ML models work
  - Traditionally focused on LLMs
- For example, people may look for:
  - Circuits (subset of neurons/attention heads that are important for a specific task)
  - Features (interpretable properties represented internally)
  - Representations (how information is geometrically encoded in the latent space)
- A recurring finding in LLMs is that many meaningful concepts seem to be encoded as directions in the model's internal vector space



Why Do We Care About Mechanistic Interpretability? *(not exhaustive)*

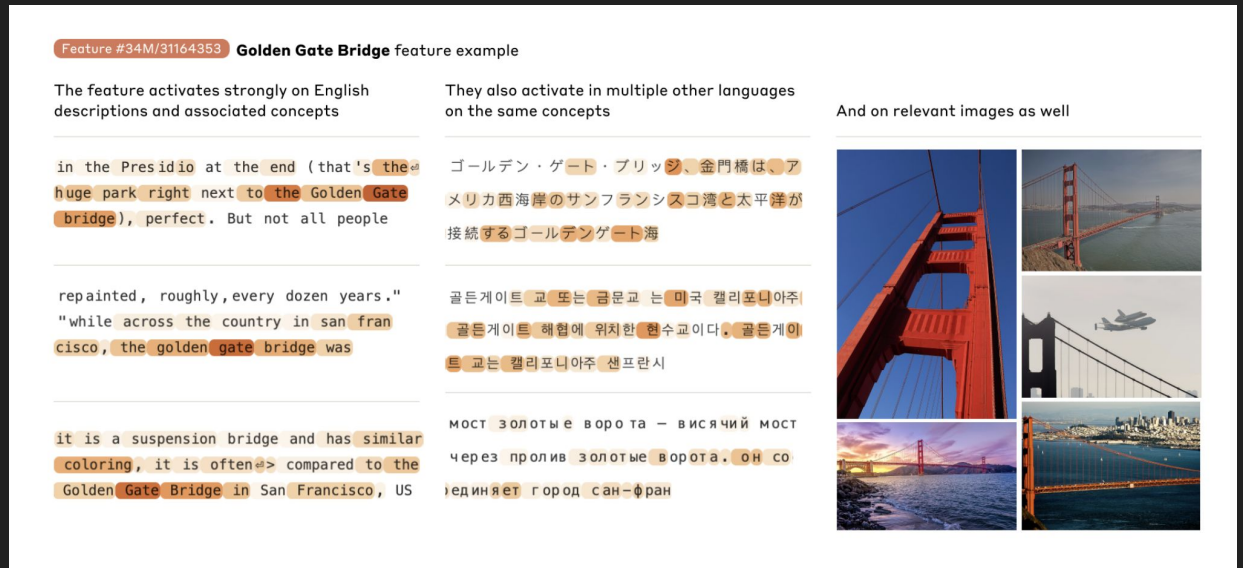
- **Debugging/improving models:**
  - Understanding internal mechanisms is helpful for understanding failure modes or whether the model relies on shortcuts in the data

- **Editing/control:**
  - If concepts are represented as directions, you can intervene on them directly (steering)
  - E.g., Golden Gate Bridge steering in LLMs

- **Scientific Discovery:**
  - If a model can outperform humans on a task, presumably it learned something we don't know
  - Therefore maybe interpreting the model can give us new scientific insights

Example: Golden Gate Neuron

- Anthropic's interp team used sparse autoencoders to find a specific direction inside their Claude model that activated whenever the model was "thinking about" the Golden Gate Bridge
- They then artificially amplified that direction during inference
  - As a result the model became "obsessed" with the Golden Gate Bridge and brought it up in almost every response, regardless of the topic.



https://www.anthropic.com/news/golden-gate-claude

https://transformer-circuits.pub/2024/scaling-monosemanticity/

What are Some Mechanistic Interpretability Methods?

- **Linear Probes**
  - Train a simple classifier on a model's internal activations to see if a concept can be read out
  - Provides correlational evidence the model encodes specific information in its representation

- **Contrastive Activation Differences**
  - Take many examples that differ mainly in the concept of interest
  - Average the activation difference between the two groups, treat that as the concept's direction

- **Activation Steering**
  - Take a latent direction that corresponds to a particular concept
  - Add it (scaled up or down) to the model's activations during inference, see if it causally changes behavior ("editing")

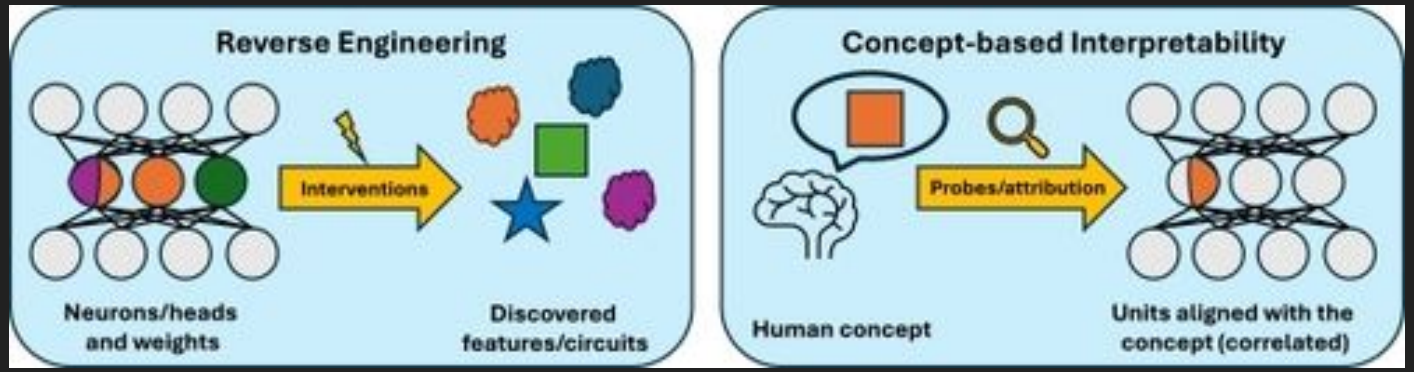


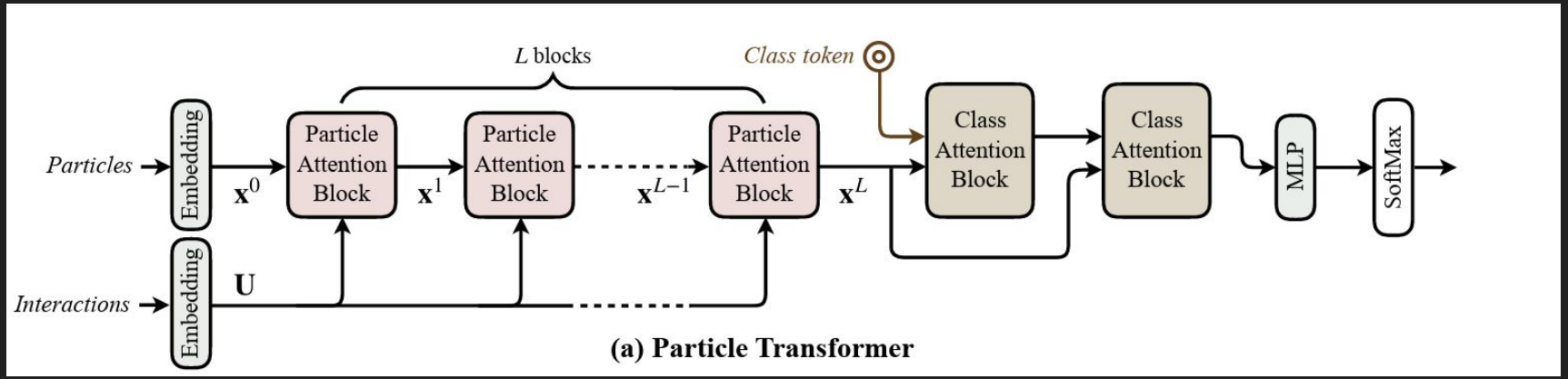
Image source

Method

Model

The primary purpose of training this model is so we have something to interpret.

- Based on ParT, We train a standard Transformer encoder with 6 layers on JetClass
- The model architecture purposely uses no inductive biases
- Achieves 82.0% accuracy on the test set
- Uses a CLS token to make final predictions



Applying Interpretability Methods to Physics

- Can we find directions in the model's latent space that correspond to physical concepts?
- We focus on Jet Mass, Multiplicity, Jet Charge, Track Displacement
- For each concept:
  - Test whether a linear probe can predict the value from CLS token latent representation
  - Use contrastive activation differences to extract a steering vector
  - Does steering change final model predictions in an expected way?
  - Does steering change the linear probe's prediction?

Results

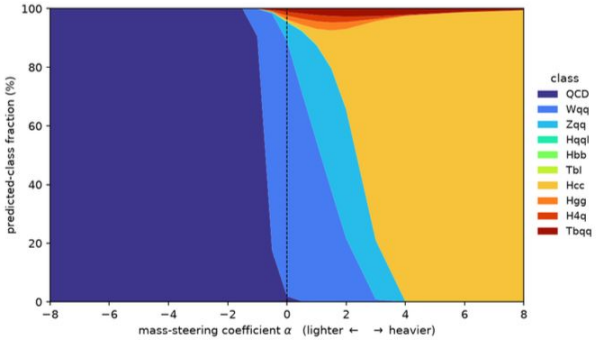
Physical Concepts are Linearly Represented

Can a linear probe reliably predict the jet's physical characteristic from the model's latent representation?

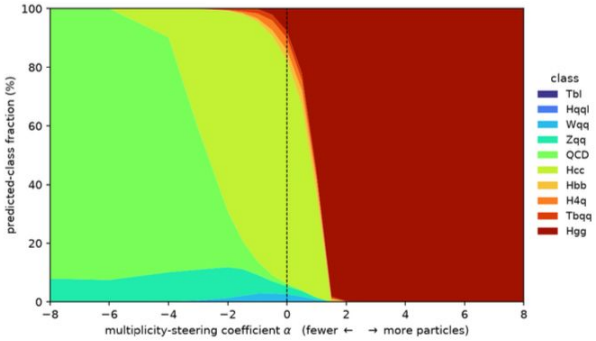
- Sample a batch of 10K QCD jets
- At each layer, fit a ridge regression model to predict a jet property from the latent representation of the CLS token

Concept	Physical definition	Probe target	Probe R^2
Jet Mass	$m = \sqrt{E^2 - p_x^2 - p_y^2 - p_z^2}$	$\log(\max(m, 1))$	0.983–0.992
Particle Multiplicity	$N = \sum_i \mathbf{1}_i$	N	0.989–0.998
Jet Charge	$Q = \frac{\sum_i q_i p_{T,i}}{\sum_i p_{T,i}}$	Q	0.940–0.981
Track Displacement	$S_{d_0} = \sum_{i \in \text{ch}} \frac{\lvert d_{0,i} \rvert}{\sigma(d_{0,i})}$	$\log(1 + S_{d_0})$	0.826–0.875

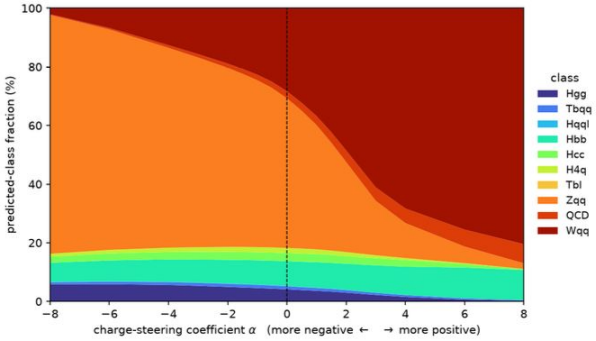
Steering Along Physical Concepts Works Expectedly



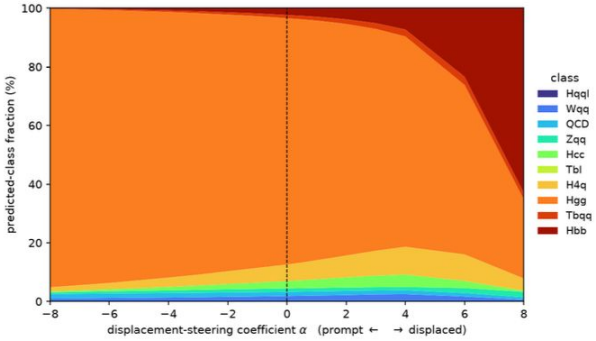
(a) Jet mass steering on $W \to qq$ jets.



(b) Particle-multiplicity steering on $H \to c\bar{c}$ jets.



(c) Jet-charge steering on $Z \to q\bar{q}$ jets.



(d) Track-displacement steering on $H \to gg$ jets.

Figure 1: Prediction migration under steering along four physical concept directions. Predicted classes within each panel are ordered by the corresponding physical quantity.

The Linear Probe's Prediction at Layer 5 Changes As You Steer At Earlier Layers

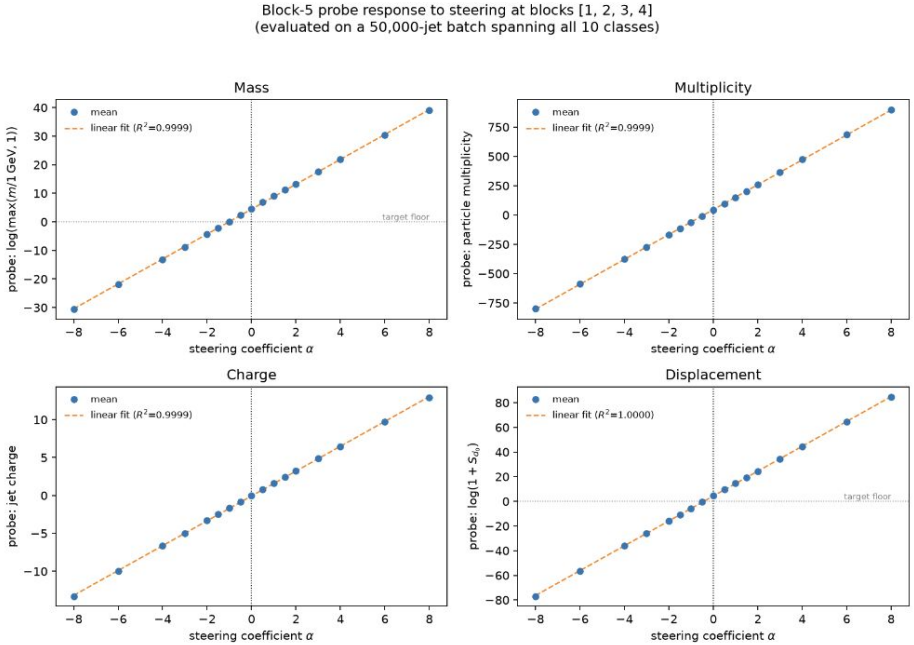


Figure 2: Block-5 linear-probe predictions as a function of steering coefficient α for interventions at blocks 1–4. Points show mean predictions over jets from all ten classes; dashed lines show least-squares fits ($R^2 > 0.999$ for all four concepts).

But these interventions can easily lead us to non-physical regimes!

Now What?

Implications for Physics

- Steering simulation models could allow for easier continuous modeling or reduce the sim2real gap
- Steering inference/reco models could enable easier counterfactual testing
- Implications for discovery
  - regress the known-quantity directions out of the CLS residual stream, find the dominant remaining directions unsupervised, and test whether they correlate with substructure observables the model was never told about or theoretical concepts
- Our model includes no inductive biases → our results imply that physically meaningful structure emerges through model's own representation learning
- But much to still learn about how models represent physics
  - See e.g. Arithmetic Without Algorithms: Language Models Solve Math With a Bag of Heuristics
  - Is it the same way we do? Why should it be?

Implications for AI Safety

- It's relatively easy to identify (reasonably robust) directions for certain concepts, but it's very easy to steer on those directions in ways that make models worse
  - Often in subtle ways that require deep domain knowledge to untangle
- There is much we need to understand about the mechanisms of robust interventions
  - Can we make a provably safe model?
- Concepts outside of physics are really messy! What can the nice basis of physics teach us about how to do this in other areas?
- Physics can provide interesting methods for new interp interventions
  - See Physics for Ambitious Interpretability

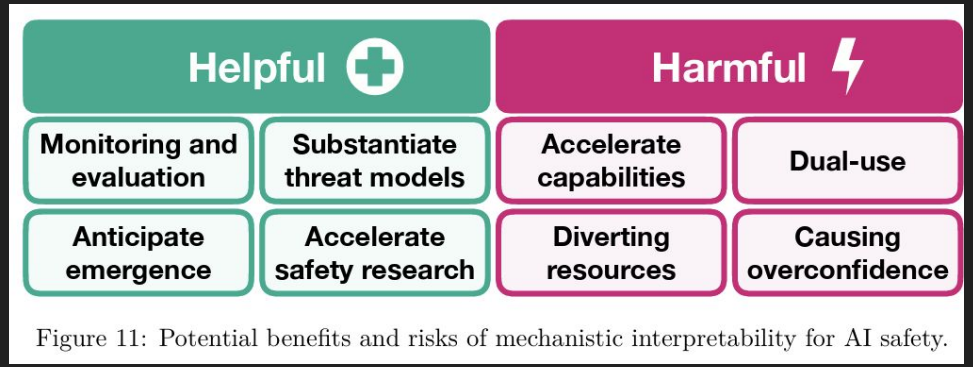


Figure 11: Potential benefits and risks of mechanistic interpretability for AI safety.

Image source: Mechanistic Interpretability for AI Safety -- A Review

Next Step: Sandbox Studies

Astronomy foundation model gives us a real scientific model with known physical structure, well-characterized noise, and ground truth → hold the interpretability methods constant and vary the inputs and training regime in a controlled way.

THE AXIS WE VARY

Progressively richer data & training

1. **Precursor surveys**
   Base + fine-tuned model on existing light-curve archives
2. **Live alert stream**
   Same methods re-run on early, noisier streaming data
3. **Full data release**
   High-quality catalog — the mature regime

HELD FIXED ACROSS EVERY STAGE

The same mechanistic-interpretability pipeline

- **Circuit analysis**
  identify the components implementing a computation
- **Causal interventions**
  ablate / patch to test necessity & sufficiency
- **Dictionary learning**
  decompose activations into interpretable features
- **Representation engineering**
  steer internals to test & extend generalization

Come Be My Postdoc!

Hiring a 3-year postdoc to work on mech interp, AI safety, and physics!

- Live in NYC!
- Support for travel/attendance at AI conferences
- Collaborative grant with Columbia, collaborators at other universities and labs
- Ad still being finalized but reach out to savannah.thais@hunter.cuny.edu in the meantime

Extra Model Training Details (appendix)

More Details on Model Training

The model consists of a two-layer particle embedding MLP, six pre-norm Transformer blocks, and a final classification head. The embedding maps the 17 input features to a 512-dimensional representation using GELU activations and dropout ($p = 0.1$). Each Transformer block uses eight attention heads and a $4\times$ MLP expansion, and the final CLS representation is passed through layer normalization and a linear classifier.

Parameter	**Value**
Input features	17
Maximum particles	128
Model dimension d	512
Transformer blocks N	6
Attention heads	8
MLP expansion	$4\times$
Dropout	0.1
Loss	Cross-entropy (label smoothing $= 0.1$)
Optimizer	AdamW
Peak learning rate	1×10^{-4}
Weight decay	1×10^{-2}
Batch size	256
LR schedule	OneCycleLR (10% warmup)
Gradient clipping	max-norm 1.0
Random seed	42

Table 2: Model and training hyperparameters.

We train for 390,625 optimization steps with batch size 256, corresponding to one pass over the 100M-jet JetClass training set. During training, we evaluate every 10,000 steps on a fixed, class-balanced subset of 50,000 validation jets, containing 5,000 examples from each class. Checkpoints are saved every 5,000 steps.

On the 20M-jet JetClass test set, the trained model achieves 82.04% overall accuracy, a cross-entropy loss of 0.5675, and a macro-averaged one-vs-rest ROC-AUC of 0.9824. Per-class accuracies range from 51.47% to 98.73%.