A Joint Minimization Framework for Transformer Interpretability

Information Crystallization

Saqib Nazir Bhat*

March 2026

Abstract

Transformer interpretability is usually posed as: identify the parameters and inputs responsible for a model's behaviour. Existing methods work top-down, searching the combinatorial space by ablation from the full model. I propose a different formulation. Given an observed output token $\hat{x}$, find the minimal $S \subseteq [d]$ and $T \subseteq [n]$ such that the argmax over the restricted model still equals $\hat{x}$. A single observation simplifies the problem: softmax does not affect argmax. Preserving the predicted token requires only that the logit margin δ remains positive, which is strictly weaker than matching the full distribution. Under this relaxation I introduce *Information Crystallization*, a bottom-up $O(k \cdot d)$ algorithm (seed, greedy growth, prune) that finds local minima of (S, T) jointly. I decompose S into a final-block component S_L and a residual-stream support S_{supp}, connect the sparsity conjecture to the Lottery Ticket Hypothesis, and identify the T-dependence of S^* as the central open problem. I have run the seed phase of the algorithm on GPT-2 small ($N = 17$ stratified sequences) as a feasibility check. Only the seed step is tested; growth and prune phases are future work.

1 Introduction

Mechanistic interpretability of large language models has become one of the central open problems in AI safety. Prior work treats parameter attribution and input attribution as separate questions: circuit discovery [Elhage et al., 2021], activation patching, causal tracing [Meng et al., 2022], and input rationales [Lei et al., 2016, Bau et al., 2020] all prune components starting from the full model. The search space is $O(2^d)$ for parameters ($d \sim 10^{11}$) and $O(2^n)$ for inputs. Practical methods rely on heuristics to make this tractable.

This paper takes a different route. The practical interpretability question is "why did the model produce *this* token?". That question does not require reproducing the full output distribution. It only requires that the restricted model still predicts the same token. This is a weaker condition, and it is the lever for the rest of the paper.

Contributions.

1. A joint formulation of parameter and input minimization with a single preservation condition (Section 2).
2. The *softmax relaxation*: replacing distribution preservation with a margin inequality (Section 3).
3. *Information Crystallization*, a bottom-up $O(k \cdot d)$ algorithm with seed, growth, and prune phases (Section 4).

*Independent research. Contact: saqibnazirbhat3@gmail.com. Repository: https://github.com/Saqibnazirbhat/information-crystallization

4. A residual-stream decomposition $S = S_L \cup S_{\text{supp}}$ (Section 6) and a formal statement of the coupling problem (Section 7).

5. An empirical protocol on GPT-2 for future validation (Section 8).

2 Problem Formulation

Let V be a finite vocabulary and $X = (x_1, \ldots, x_n) \in V^n$ an input sequence. A language model parameterized by $\theta \in \mathbb{R}^d$ defines

$$P_\theta(x_{n+1} \mid x_1, \ldots, x_n) = \text{softmax}\big(f_\theta(X)\big), \tag{1}$$

where $f_\theta : V^n \to \mathbb{R}^{|V|}$ is a composition of L non-linear transformations:

$$f_\theta = f_\theta^{(L)} \circ f_\theta^{(L-1)} \circ \cdots \circ f_\theta^{(1)}. \tag{2}$$

The predicted token is

$$\hat{x} = \arg\max_{v \in V} P_\theta(v \mid X). \tag{3}$$

Problem. Given $\hat{x}$, find minimal $S \subseteq [d]$ and $T \subseteq [n]$ such that

$$\arg\max_{v \in V} P_{\theta_S}(v \mid X_T) = \hat{x}, \tag{4}$$

where θ_S denotes θ restricted to indices S (entries outside S set to zero or to a reference value), and X_T denotes X restricted to positions T.

In general, this problem is computationally intractable: f_θ is non-linear and non-invertible, and $d \sim 10^{11}$ for current frontier-scale models. Section 3 shows that a natural weakening of the preservation condition makes the problem substantially more tractable.

3 The Softmax Relaxation

3.1 Margin definition

The logit margin is the gap between the winning logit and its nearest competitor:

$$\delta \;=\; f_\theta(X)[\hat{x}] \;-\; \max_{v \neq \hat{x}} f_\theta(X)[v]. \tag{5}$$

The argmax condition $\arg\max_v f_\theta(X)[v] = \hat{x}$ is equivalent to $\delta > 0$. Since softmax is a strictly monotone transformation, preserving the argmax requires only preserving the *sign* of δ. No particular probability value needs to match.

This is the key relaxation. A distribution-preservation condition becomes a margin-preservation condition. The former is strict; the latter admits a buffer zone of width δ around the decision boundary within which perturbations do not change the argmax.

3.2 The δ-buffer argument

Most parameters in a language model do not determine the identity of the winning token. They refine probabilities of losing tokens, smooth the output distribution, or encode features relevant to positions outside T. A parameter θ_i is irrelevant to $\hat{x}$ if perturbing it changes δ by less than δ itself. The set of such parameters is large precisely because $\delta > 0$ creates a buffer around the decision boundary. This motivates the central *Sparse Explanation Hypothesis*: the true minimal (S, T) is small relative to (d, n).

3.3 Sufficient condition

The following condition, if proved, converts the conjectures below into theorems.

Sufficient Condition 1. *For all $\theta_i \notin S$,*

$$\left|\text{logit}_{\hat{x}}(\theta_S, X_T) - \text{logit}_{\hat{x}}(\theta, X)\right| < \delta. \tag{6}$$

A proof via Lipschitz continuity of f_θ or a first-order perturbation bound would suffice. The difficulty is the non-linearity of f_θ, which resists tight analytical bounds. A first-order Taylor expansion gives

$$\Delta\text{logit}_{\hat{x}} \;\approx\; \nabla_\theta \text{logit}_{\hat{x}}^\top \cdot \Delta\theta,$$

suggesting that parameters with small gradient magnitude are natural candidates for the complement S^c.

4 Information Crystallization

Top-down methods search the space $O(2^d)$ by ablating from $|S| = d$. I invert the direction of search: start with $|S| = 1$ and grow. If the true minimal (S, T) is small, growing finds it in time proportional to its size rather than the model's size.

4.1 Algorithm

Algorithm 1 Information Crystallization

1: $g \leftarrow \nabla_\theta \log P_\theta(\hat{x} \mid X)$
2: $S \leftarrow \{\arg\max_j |g_j|\}, \quad T \leftarrow \{\arg\max_t \|\partial f_\theta / \partial x_t\|\}$ ▷ seed phase
3: **while** $\arg\max_v P_{\theta_S}(v \mid X_T) \neq \hat{x}$ **do** ▷ growth phase
4: $\quad \hat{g} \leftarrow \nabla_\theta \log P_{\theta_S}(\hat{x} \mid X_T)$
5: $\quad S \leftarrow S \cup \{\arg\max_{j \notin S} |\hat{g}_j|\}$
6: **end while**
7: **for all** $s \in S$ **do** ▷ crystallization phase
8: $\quad$ **if** $\arg\max_v P_{\theta_{S\setminus\{s\}}}(v \mid X_T) = \hat{x}$ **then** $S \leftarrow S \setminus \{s\}$
9: $\quad$ **end if**
10: **end for**
11: **if** argmax broken **then goto** line 3 ▷ verification loop
12: **end if**
13: **return** (S, T)

4.2 Complexity

The seed phase costs one full backward pass, $O(d)$. Each growth step costs one forward–backward pass through the restricted subnetwork of size $|S|$, plus $O(d)$ to rank candidates via the gradient. After k growth steps the total cost is $O(k \cdot d)$. The prune phase costs $O(k)$ forward passes. Compared to exhaustive search $O(2^d)$ or top-down pruning $O(d^2)$, this is tractable when $k = |S| \ll d$.

4.3 Why bottom-up

Three properties distinguish this approach:

- **Direction of search.** Existing methods (activation patching, structured pruning, circuit discovery) start from the full model and remove components. This algorithm starts from empty and adds.

- **Gradient-guided growth.** Ranking d candidates at each step would cost $O(d^2)$ in total. Using the gradient of the restricted subnetwork as a ranking heuristic reduces this to $O(d)$ per step.
- **Joint optimization.** S and T grow simultaneously, capturing interactions between which parameters matter and which inputs matter.

5 Minimizing T: Input Position Selection

Let $\alpha^{h,\ell}(t)$ denote the attention weight on position t from head h at layer ℓ. Define the cumulative salience

$$\sigma(t) \;=\; \sum_{h,\ell} \alpha^{h,\ell}(t). \tag{7}$$

For a threshold ε chosen so that positions with $\sigma(t) < \varepsilon$ contribute less than $\delta/2$ to the winning logit, define

$$T_{\text{sal}} \;=\; \{\, t \in [n] : \sigma(t) > \varepsilon \,\}. \tag{8}$$

Conjecture 1. *$T = \{x_{n-k}, \ldots, x_n\} \cup T_{sal}$ with $k \ll n$, where the suffix captures recency and T_{sal} captures earlier positions with high semantic weight.*

Failure modes. Transformers use global attention, so a position-1 token with high σ can dominate the logit landscape. The suffix approximation breaks in three concrete cases:

- Sentence-initial negation: "Despite everything, the answer is still…" Position 1 reshapes the entire logit distribution, overriding suffix recency.
- Long-range syntactic dependencies: the subject introduced at positions 1–3 may determine verb form 20 or more tokens later.
- Named entities: in "Harry Potter picked up his…", the entity at positions 1–2 causes the model to prefer "wand" over "phone". The suffix alone is insufficient.

The corrected claim is: T is identified by attention weight concentration, not by positional recency. The suffix approximation is a useful empirical heuristic but not a structural guarantee.

6 Minimizing S: Parameter Selection

6.1 Residual stream decomposition

I conjecture that S decomposes as

$$S \;=\; S_L \cup S_{\text{supp}}, \tag{9}$$

where S_L consists of parameters in the final transformer block that directly govern δ (specifically: the unembedding projection weights for $\hat{x}$ and its nearest competitor, the attention heads in the last one or two layers that activate most strongly on T, and a small set of MLP neurons in the final block), and S_{supp} consists of earlier-layer parameters whose representations are consumed by S_L via the residual stream [Elhage et al., 2021]. The contribution of S_{supp} must satisfy

$$\big|\delta(S_L) - \delta(S_L \cup S_{\text{supp}})\big| < \delta/2. \tag{10}$$

6.2 Lottery Ticket connection

Frankle and Carbin [2019] show that dense networks contain sparse subnetworks, so-called "winning tickets", that match the full network's performance when trained in isolation. The conjecture here is the inference-time analogue: for a fixed input X and prediction $\hat{x}$, there exists a sparse subnetwork that preserves the argmax without retraining.

6.3 Middle-layer localization

Meng et al. [2022] show via causal tracing that factual associations in GPT-class models localize in middle-layer MLP modules. This challenges the assumption that S_L alone suffices for factual recall queries. For "The capital of France is..." the decisive parameters may live in middle layers, meaning S_{supp} could be substantially larger for factual than for syntactic continuation tasks. Any empirical protocol must stratify by query type.

7 Joint Minimization: The Coupling Problem

S and T are not independently minimizable. When T changes, the embeddings fed to layer 1 change. That perturbation propagates through all L layers via the residual stream, producing a different hidden state at layer L. The set of parameters in S_L that are load-bearing for δ therefore depends on T:

$$S^*(T) \;=\; \arg\min_S |S| \quad \text{s.t.} \quad \arg\max_v P_{\theta_S}(v \mid X_T) = \hat{x}. \tag{11}$$

Treating S and T as independent is a useful approximation but its error is only small when S_{supp} is small relative to δ.

Open Problem 1. *Characterize the function $T \mapsto S^*(T)$. Two concrete sub-questions: (i) bound $|S^*(T) - S^*(\theta, X)|$ as a function of $|T|$; (ii) determine whether the greedy decomposition in Algorithm 1 is within a constant factor of the true joint minimum.*

Greedy decomposition as upper bound. Despite the coupling, Algorithm 1 yields a computable upper bound on $|S| + |T|$: identify T from cumulative attention weights (Section 5), then minimize S conditional on T via gradient-sensitivity scoring, then check whether the coupling error is small. This is suboptimal in general but empirically verifiable.

8 Empirical Protocol

GPT-2 [Radford et al., 2019] is suitable for initial verification because its full parameter set and attention weights are publicly accessible. I propose the following protocol.

1. **Sampling.** Draw 1,000 input sequences stratified by query type: syntactic continuation, factual recall, negation-heavy, and long-range syntactic dependency.
2. **Baseline.** Record $\hat{x}$ for each sequence using the full model θ.
3. **T identification.** Greedily remove input positions in ascending order of $\sigma(t)$. Record the minimal T at which $\hat{x}$ first flips.
4. **S identification.** Greedily ablate parameters in ascending order of gradient-sensitivity score. Record the minimal S at which $\hat{x}$ first flips.
5. **Size measurement.** Report $|S|/d$ and $|T|/n$ distributions per query type (means, medians, 90th percentiles).

6. **Coupling test.** For each T from Step 3, recompute $S^*(T)$ and compare with $S^*(\theta, X)$. Small discrepancy empirically validates the greedy decomposition.

Preliminary seed-phase results. I have run Step 1 of the algorithm on $N = 17$ hand-curated GPT-2 small sequences, stratified into four query types. The input-position seed t^* landed on the final token in only 11.8% of sequences; on long-range sequences it clustered near the subject noun (mean distance from the end, 7.67 tokens over three examples). This is evidence against a suffix-only heuristic. The parameter seed j^* was the tied `wte.weight` tensor in 17/17 sequences, which is consistent with the S_L conjecture but is not a clean test: on GPT-2 the input embedding and unembedding projection share one parameter tensor. A model with untied embeddings (e.g., Pythia-160M) is the obvious next experiment. Full notebook and stratified results are in `experiments/seed_phase.ipynb` and `experiments/results.md`.

9 Limitations

- **The suffix approximation is not a theorem.** T is identified by attention analysis per input. No structural guarantee holds.
- $|S_{\mathbf{supp}}|$ **is unbounded.** The inequality $|\delta(S_L) - \delta(S_L \cup S_{\text{supp}})| < \delta/2$ is a requirement, not a proved result. For factual queries it may be violated.
- **Greedy decomposition is suboptimal.** The true joint minimum of (S, T) may be strictly smaller than the greedy bound. The gap is uncharacterized.
- **Condition 1 is unproved.** All conjectures depend on a Lipschitz or first-order perturbation bound on f_θ that has not been derived.
- **Scale.** Verification on GPT-2 does not guarantee generalization to models with $d \sim 10^{11}$. Distributed representations in large models may yield dense (S, T).
- **Only the seed phase has been run.** The growth, prune, and verification phases in Algorithm 1 are not implemented. $N = 17$ is a sanity check, not a study.
- **Weight-tying on GPT-2 blocks a clean** S_L **test.** The preliminary j^* result is consistent with the S_L conjecture but does not adjudicate it, because `wte.weight` plays both the embedding and unembedding roles on GPT-2.

10 Conclusion

The standard interpretability problem asks for the parameters and inputs that explain a model's behaviour. Posed as distribution preservation, the problem is intractable. Posed as margin preservation it admits substantially sparser solutions, reachable via a bottom-up $O(k \cdot d)$ algorithm.

The central open problem is the coupling between S and T through the residual stream. A formal perturbation bound on f_θ would turn the conjectures here into theorems. A preliminary seed-phase run on GPT-2 small is reported in Section 8. Running the growth and prune phases, and replicating on a model with untied embeddings, are the next steps.

References

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations (ICLR)*, 2020.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. Technical report, Anthropic, 2021. URL `https://transformer-circuits.pub/2021/framework/index.html`.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of EMNLP*, 2016.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.