

Semantic Optimal Transport for Sparse Autoencoder Feature Matching and Circuit Compression

Tue M. Cao* Nguyen Do* My T. Thai

University of Florida • NeurIPS 2026

*equal contribution



arXiv:2605.28567

Cross-manifold SAE feature matching

SAEs at layer ℓ : $a^{(\ell)}(x) = \sigma(W_{\text{enc}}^{(\ell)}x + b_{\text{enc}}^{(\ell)}) \in \mathbb{R}_{\geq 0}^{F_\ell}$, $F_\ell \gg d_\ell$.

- **(P1) Cross-layer matching:** for feature i at layer A , find the semantically closest feature j at layer B .
- **(P2) Circuit compression:** group the nodes of a large feature circuit into a few interpretable *supernodes*.

Common challenge. Estimating a semantic distance $D_{\ell q}^*(i, j)$ between features that live on **different activation manifolds**.

Challenges

- **Decoder vectors:** $W_{\text{dec}}^{(A)}$ and $W_{\text{dec}}^{(B)}$ live in different bases and geometries.
- **One vector per feature:** hard to convey the complexity of the feature's meaning.
- **LLM-judged matching:** does not scale and misses fine functional distinctions.

Key idea. Represent each feature as a *distribution of hidden states*, and compare feature distributions in a *shared space* with optimal transport.

Method: features as distributions

Corpus $\mathcal{S}_T$ of T token positions. For feature i at layer ℓ :

Support. $\mathcal{I}_{i,T}^{(\ell),K} = \text{TopK}_{t \in [T]} a_i^{(\ell)}(x_t^{(\ell)})$

Weights. $\hat{w}_{i,t}^{(\ell)} = \frac{a_i^{(\ell)}(x_t^{(\ell)})}{\sum_{s \in \mathcal{I}} a_i^{(\ell)}(x_s^{(\ell)})}, t \in \mathcal{I}$

Distribution. $\hat{\mu}_{i,T}^{(\ell)} = \sum_{t \in \mathcal{I}} \hat{w}_{i,t}^{(\ell)} \delta_{x_t^{(\ell)}}$

Shared reference space. Map each support token to $z_t^{(\mathcal{R})} \in \mathcal{R}$ (the same tokens' hidden states in $\mathcal{R}$): $\hat{\mu}_{i,T}^{(\ell \rightarrow \mathcal{R})} = \sum_{t \in \mathcal{I}} \hat{w}_{i,t}^{(\ell)} \delta_{z_t^{(\mathcal{R})}}$.

Features from different layers become measures on one space $\mathcal{R}$, so they are directly comparable.

Semantic OT distance

$$\hat{D}_{\mathcal{R}}(i, j) = W_c(\hat{\mu}_{i,T}^{(A \rightarrow \mathcal{R})}, \hat{\nu}_{j,T}^{(B \rightarrow \mathcal{R})}), \quad W_c(\mu, \nu) = \inf_{\gamma \in \Pi(\mu, \nu)} \int c d\gamma.$$

Discrete form, with cost matrix $C_{pq}^{(i,j)} = c(z_p^{(\mathcal{R})}, z_q^{(\mathcal{R})})$:

$$\hat{D}_{\mathcal{R}}(i, j) = \min_{\pi \geq 0} \langle \pi, C^{(i,j)} \rangle \quad \text{s.t.} \quad \pi \mathbf{1} = \hat{w}_i, \quad \pi^\top \mathbf{1} = \hat{w}_j.$$

- Default ground cost $c = \|\cdot\|_2$; solved with Sinkhorn.
- Compares *where* a feature fires and *how its activation mass is spread*, not raw activation magnitudes.

Algorithms

Alg. 1 – Cross-layer matching

($\mathcal{R}$ = target layer)

1. Project target and source distributions into $\mathcal{R}$.
2. Prefilter: keep the 50 source features whose centroids $\bar{\nu}_j$ are closest to $\bar{\mu}_i$.
3. Sinkhorn W_c to each candidate; return $\arg \min_j \hat{D}_{\mathcal{R}}(i, j)$.

Alg. 2 – Circuit compression

($\mathcal{R}$ = concatenated hidden states over all layers)

1. Project every circuit node into $\mathcal{R}$.
2. Pairwise Sinkhorn $W_c \Rightarrow$ distance matrix.
3. Agglomerative clustering into m supernodes.

Guarantees: invariance and stability

Prop. 1 (rescaling invariance). For $\alpha > 0$ and $\tilde{a}_i^{(A)} = \alpha a_i^{(A)}$, deterministic TopK gives $\tilde{\mu}_{i,T}^{(A \rightarrow \mathcal{R})} = \hat{\mu}_{i,T}^{(A \rightarrow \mathcal{R})}$; hence W_c is unchanged for every ground cost c .

Activation scales differ across layers; the representation depends only on ranking and proportions.

Thm. 1 (stability). If c is coordinate-wise Lipschitz on the relevant supports (constants L_x, L_y w.r.t. base metric ρ), then

$$|W_c(\mu, \nu) - W_c(\tilde{\mu}, \tilde{\nu})| \leq L_x W_\rho(\mu, \tilde{\mu}) + L_y W_\rho(\nu, \tilde{\nu}).$$

No exact token overlap needed: the score moves only if substantial mass moves far in $\mathcal{R}$.

Guarantees: margin-based recovery

Thm. 2 (matching recovery). $D_{\mathcal{R}}(i, j) = W_c(\mu_i^{(A \rightarrow \mathcal{R})}, \nu_j^{(B \rightarrow \mathcal{R})})$, unique $j_i^* = \arg \min_j D_{\mathcal{R}}(i, j)$, $\Delta_i = \min_{j \neq j_i^*} [D_{\mathcal{R}}(i, j) - D_{\mathcal{R}}(i, j_i^*)] > 0$. If $\sup_j |\hat{D}_{\mathcal{R}}(i, j) - D_{\mathcal{R}}(i, j)| \leq \epsilon_i$ and $\Delta_i > 2\epsilon_i$, then $\arg \min_j \hat{D}_{\mathcal{R}}(i, j) = j_i^*$.

Thm. 3 (supernode recovery). Semantic centers $\xi_1, \dots, \xi_m \in \mathcal{R}$, score $\mathcal{Q}_k(\mu) = \mathbb{E}_{z \sim \mu} \rho(z, \xi_k)$, $k_i^* = \arg \min_k \mathcal{Q}_k(\mu_i)$, Voronoi margin $\Gamma_i = \min_{r \neq k_i^*} [\mathcal{Q}_r(\mu_i) - \mathcal{Q}_{k_i^*}(\mu_i)] > 0$. If $W_\rho(\hat{\mu}_{i,T}, \mu_i) < \Gamma_i/2$, then $\hat{k}_{i,T} = k_i^*$.

Large-margin matches and nodes are robust to finite-sample noise; small-margin ones can be flagged. High-probability and sample-complexity versions in the appendix.

Experimental setup

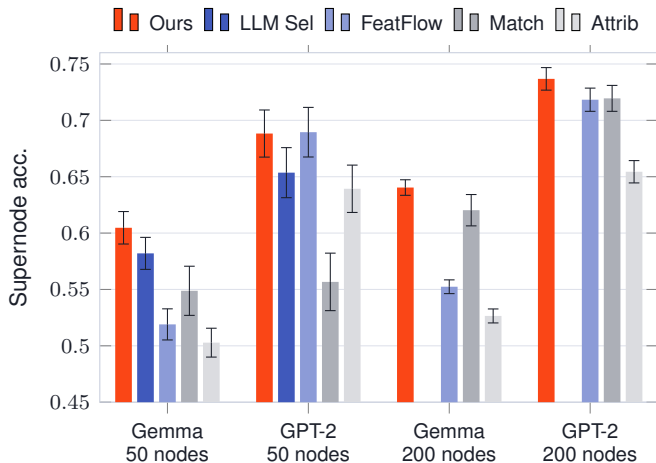
- **Models / SAEs:** GemmaScope (JumpReLU, MLP out) on Gemma-2-2B; GPT-2 small (TopK, residual stream).
- **Baselines:** SAE Match (L_2 decoder vector), Feature Flow (cosine decoder vector), Attribution Patching, LLM selection (gpt-oss-120b); “ML” = multi-layer composition.
- **Metrics:**
 - Matching: LLM-judge score (3/2/1) and accuracy; CE and variance explained after feature substitution.
 - Compression: supernode membership accuracy (an LLM captions half of a supernode and must pick its held-out members).

Cross-layer feature matching

Model	Layers	LLM Eval $\uparrow$ (1–3)		Acc. (%) $\uparrow$	
		Ours	best base.	Ours	best base.
GPT-2	5 $\rightarrow$ 6	2.53	2.46	68.2	66.2
	5 $\rightarrow$ 11	1.56	1.49	23.2	22.2
	0 $\rightarrow$ 11	1.39	1.34	17.2	12.4
Gemma-2-2B	12 $\rightarrow$ 13	2.32	1.65	58.6	27.2
	12 $\rightarrow$ 25	1.83	1.34	33.6	13.0
	0 $\rightarrow$ 25	1.83	1.28	34.8	11.6

Ours vs. the best baseline per cell over Match, FeatFlow, their multi-layer variants, and Attrib. 100 pairs, 5 LLM evaluations, 90% CI. CE and VE stay on par with or better than baselines.

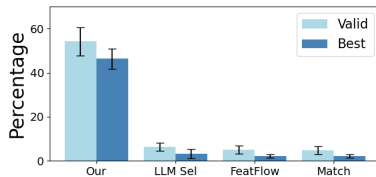
Circuit compression



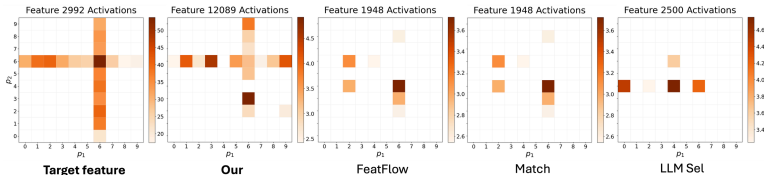
- Compress 50→5 and 200→20 nodes; 200 circuits, 90% CI.
- Ours is best in **3 of 4** settings; on par with FeatFlow on GPT-2, 50 nodes.
- LLM Sel cannot fit 200-node circuits in context.

Subtle functional meaning: digit addition

- Prompt “ $p_1 + p_2 =$ ”, $p_1, p_2 \in [0, 9]$.
- The activation pattern on “=” over the 10×10 grid is the ground-truth meaning (e.g. “first operand is 5”).
- Blind volunteer ranking of 100 target features, Gemma-2-2B L19 MLP.

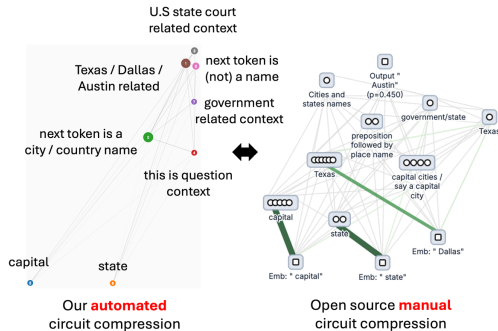


Valid: volunteer agrees with the match. Best: top pick among methods.

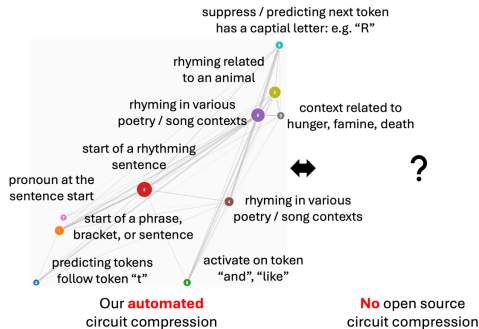


Only ours recovers the operand-specific feature; baselines match on generic math context.

Case study: automatic circuit compression



Prompt: "<BOS>Fact: the capital of the state containing Dallas is Austin"



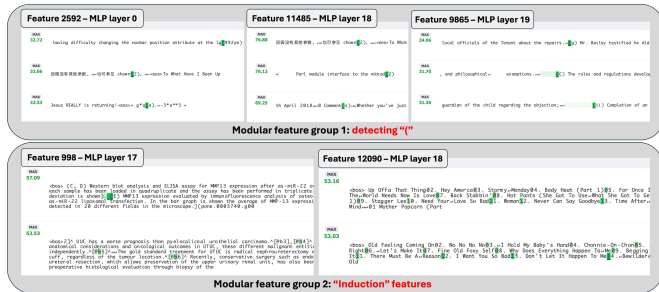
Prompt: "<BOS>a rhyming couplet: he saw a carrot and had to grab it, his hunger was like a starving rabbit"

Left: on "Dallas" our automatic supernodes match the manual Neuronpedia compression (Dallas → Texas → state-government contexts → "Austin"). Right: "Rhythm" (. . . "rabbit"), which has no manual compression.

Application: modular feature groups

Feature circuits: "Incrementing sequence"

One prompt example: "<bos>(a) some context (b)"



- Compress many circuits that perform the same task.
- Cluster features by how often they share a supernode.
- "Incrementing sequences": a "(" detector group and an induction group.

Takeaways

1. Representing SAE features as **distributions** captures their semantics better than a single vector.
2. Projecting feature distributions into a **shared space** enables accurate semantic comparison across layers, with stability and recovery guarantees.
3. Turns manual supernode clustering into an **automatic procedure**.

Paper: [arXiv:2605.28567](https://arxiv.org/abs/2605.28567) Contact: mythai@cise.ufl.edu

