Catalog-Driven Supervised Sparse Autoencoders: Specifying Token-Level Features Before Training Recovers What Unsupervised Dictionaries Miss

Tsogt-Ochir Enkhbayar [1] **Georg Lange** [1]

Abstract

Modern SAE pipelines use LLMs as supervisors only *after* the unsupervised dictionary is fixed to interpret latents that have already been trained. We test whether the dictionary even admits clean upstream supervision and find it does not. On layer 9 of a GPT-2 Small SAE, of the top 300 unsupervised latents by ΔR^2, the best 100 pass through a strict catalog-quality cascade and only 2 survive. The rest are rejected by a frontier LLM as polysemantic bundles. We move the LLM upstream: a frontier LLM creates a catalog of token-level concepts as prefix-decidable yes/no questions; a local LLM labels every (token, feature) pair through a three-leveled prefix caching scheme; and a supervised SAE is trained with its decoder columns pinned to the mean directions of those concepts. The supervised alternative reaches a mean calibrated F1 of 0.604 on a 103-feature catalog and 0.481 on a 466-feature catalog, against 0.025 for real EleutherAI Delphi auto-interp on the unsupervised 24,576-latent `gpt2-small-res-jb` SAE. Both sides are scored identically: per-feature F1 of latent firing against the same Qwen3 annotator on the same held-out positions. The supervised dictionary matches reconstruction quality at 68× fewer latents ($R^2 = 0.969$) while producing directions that individually admit falsifiable description and causal intervention.

1. Introduction

Sparse autoencoders are a dominant tool for extracting interpretable features from language model activations (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024). Their evaluation is a two-step pipeline: first, an unsupervised dictionary is trained, then a separate model is asked to describe each latent post hoc (Paulo et al., 2024). SAEBench (Karvonen et al., 2025) shows that reconstruction improvements do not predict downstream interpretability, and surveys (Shu et al., 2025) show the failure modes: splitting, absorption, polysemy. If we cannot identify and intervene on safety-relevant directions like refusal (Arditi et al., 2024), then SAE-based interpretability provides weaker guarantees than headline F1 numbers suggest.

Current pipelines already rely on LLMs to interpret, repair, and summarise unsupervised features. The LLM is doing the work of imposing structure, but only after the dictionary is fixed. Move that supervision upstream and you get to specify the structural property you want, in our case prefix-decidable falsifiability, by construction. Two classic consequences happen: feature splitting cannot occur, there is one supervised latent per catalog leaf, and feature absorption becomes auditable, since any leaf whose positives are a subset of another's appears in the label statistics. The cliff result, the F1 gap, and the causal findings below are then symptoms of having that property versus not having it.

As a first test, we took the top ∼300 high-ΔR^2 unsupervised latents of a layer-9 GPT-2 Small SAE and asked a frontier LLM to mine the best 100 into named token-level features. 84% were rejected as polysemantic bundles by the LLM's own crispness gate, and a strict downstream cascade left only 2 surviving catalog entries. A better explainer cannot fix this. The latents at this layer are not the kind of object a falsifiable yes/no question can describe.

We therefore invert the workflow (Figure 1). Rather than hoping a large unsupervised dictionary contains the features we care about, we first *specify* a catalog of features as prefix-decidable yes/no questions; we then *annotate* every (token, feature) pair with a local LLM; and finally we *train* a supervised SAE whose decoder columns are pinned, by construction, to the conditional mean directions of those features. We evaluate on three key metrics: per-feature F1 (interpretability), Fraction of Variance Unexplained (FVU, reconstruction), and L0 (sparsity).

[1]Independent Researcher. Correspondence to: Tsogt-Ochir Enkhbayar <tsogtochir04@gmail.com>, Georg Lange <georglange4@gmail.com>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

Contributions. (i) At matched scoring on the same model and layer, supervised SAEs reach mean F1 of 0.481 to 0.604 across three operating points, vs. 0.025 for real EleutherAI Delphi on the unsupervised 24,576-latent `gpt2-small-res-jb` SAE: a gap of 19 to 24× that holds even when the supervised catalog is ∼5× the size of the unsupervised one. (ii) The unsupervised-latent catalog-quality cliff: of the top ∼300 latents by ΔR^2, the best 100 pass through a strict cascade and only 2 survive. (iii) A literal-hinge formulation (Cortes & Vapnik, 1995; Bartlett et al., 2006) with a frozen decoder pinned to mean-shift target directions (Makelov et al., 2024). The natural decision boundary $z = 0$ already approximates the per-feature optimal threshold (45 of 103 curated and 204 of 466 broad-catalog features fire within 0.01 of their ground-truth positive rate at $z = 0$).

2. Method

Catalog generation. A frontier LLM is shown top-activating contexts for a sample of pretrained SAE latents and produces a hierarchical catalog. Each leaf carries `description` (a single ≤10-word yes/no question), `positive_examples`, `negative_examples`, and `exclusions` (appended to the annotator prompt, e.g. "`NOT abbreviation periods`"). A regex backstop hard-fails descriptions that require future tokens, full-sentence parsing, or document context.

Annotation. Qwen3-8B-Base (Qwen Team, 2025) runs locally through vLLM (Kwon et al., 2023) with a three-level prefix cache (Figure 2): the system prompt with all feature definitions (∼200 tokens) is computed once per run; the per-sequence prefix is shared across all features at each position; only the feature-id suffix (∼8–10 tokens) is generated fresh. A single RTX 5090 sustains ∼1.2K tokens per second.

Architecture and loss. The SAE has a supervised slice of S latents (one per catalog leaf) and an unsupervised slice of U latents for residual reconstruction. Let $z = W_{\text{enc}}x + b_{\text{enc}}$, $f_i = \text{ReLU}(z_i)$, and $\hat{x} = W_{\text{dec}}f + b_{\text{dec}}$. The loss is the literal hinge of Cortes & Vapnik (1995):

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \lambda \sum_{i \in S} \max\Big(0, -(2y_i - 1)z_i\Big), \quad (1)$$

with λ tuned on a held-out calibration split. We use no positive-class re-weighting and no straight-through estimator. Each supervised decoder column $W_{\text{dec},i}^{(\text{sup})}$ is fixed to $d_i = \text{normalize}\big(\mathbb{E}_x[x \mid y_i{=}1] - \mathbb{E}_x[x]\big)$ and frozen for the duration of training. A binary-label-without-magnitude target is naturally a step function; we tried JumpReLU and TopK variants (Rajamanoharan et al., 2024b; Gao et al., 2024) and found the literal hinge with frozen mean-shift directions empirically dominated in this setting.

3. Experiments

Setup. Target: GPT-2 Small, layer 9 residual stream (`blocks.9.hook_resid_pre`, 768d). We use 5,000 OpenWebText sequences, 128 positions per sequence, an 80/10/10 train/calibration/test split, and Qwen3-Base via vLLM as the annotator and scorer. Baselines: linear probe on activations, and linear readout over the 24,576 activations of the pretrained `gpt2-small-res-jb` SAE (Bloom et al., 2024), and real EleutherAI Delphi (Paulo et al., 2024) run on the same SAE.

3.1. Headline numbers

Table 1 compares the supervised SAE to the linear probe and the pretrained SAE on the same held-out positions.

Table 1. Production-scale comparison at 5,000 sequences, 103 supervised features, $U = 256$, layer 9. Cal F1 is per-feature optimal-thresholded; L_0 ratio is predicted-positive rate / ground-truth positive rate at $z = 0$ (closer to 1 is better). †Cosine to target is 1.000 by construction (frozen-decoder mean-shift target, Equation (1)).

metric	sup. SAE	probe	pretrained
mean F1 (z=0)	**0.613**	0.474	0.481
mean cal F1	**0.604**	0.607	0.614
mean AUROC	0.905	0.939	0.964
R^2	**0.969**	—	0.984
# latents	103+256	—	24,576
naive L_0 ratio	**1.022**	—	—
cosine to target†	**1.000**	—	—

The supervised SAE wins at the natural threshold by +0.139 F1 over the probe; at the optimal threshold the methods are within 0.01 F1, because the probe gets to fit one direction per feature in $\mathbb{R}^{768}$ while the SAE has its single direction pinned to the feature's conditional mean. The supervised SAE preserves $R^2 = 0.969$ (FVU of 0.031) at 68× fewer latents than the pretrained SAE. Sequence-split eval (whole documents held out) costs only 0.004 F1. The supervised SAE without an unsupervised slice maintains an $R^2 = 0.8240$

Probe directions are not causally targeted. The probe looks competitive on calibrated F1, but its directions are not interchangeable with the SAE's for intervention. On a 46-feature predecessor artifact, ablating each feature's probe weight at its positive positions yields a median targeting ratio KL_+/KL_- of 0.59; the same protocol on the SAE's frozen-decoder column yields 682, a ∼1100× asymmetry. Probe directions move the model as much at random non-firing positions as at firing positions; SAE directions are location-specific.

The supervised catalog is internally non-redundant. Across the 103 supervised features, 0 pairs exceed IoU ≥

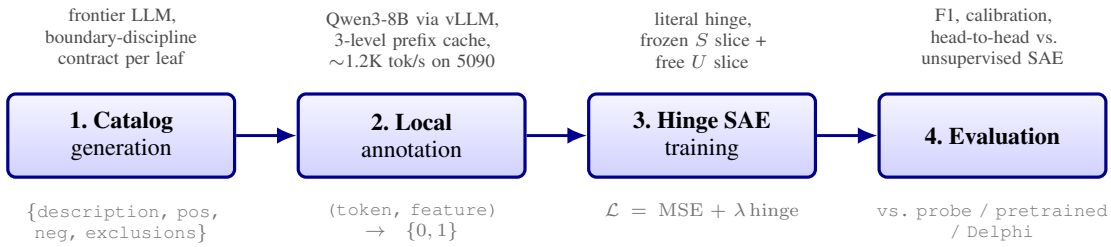

Figure 1. Catalog-driven supervised SAE pipeline. A frontier LLM produces a hierarchical catalog of token-level concepts as prefix-decidable yes/no questions (stage 1); a local LLM annotates every (token, feature) pair via a three-level prefix cache (stage 2); the supervised SAE is trained with literal hinge on a frozen-decoder mean-shift target (stage 3); evaluation includes a head-to-head against an unsupervised SAE described post-hoc on the same model and layer (stage 4).

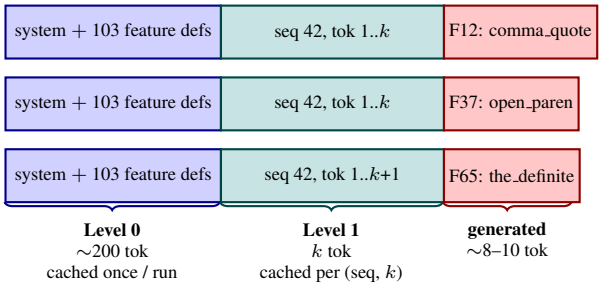

Figure 2. Three-level prefix cache used during annotation. The inner loop iterates over features at each position before advancing to the next position. Rows 1 and 2 share the same Level 0 and Level 1 prefix; only the per-feature suffix differs, so vLLM serves them as Level-1 cache hits. Row 3 advances to position $k + 1$, growing the Level 1 prefix by one token. The system prompt with all 103 feature definitions (Level 0) is computed once for the entire run. On a single RTX 5090 the pipeline sustains ∼1.2K tokens per second.

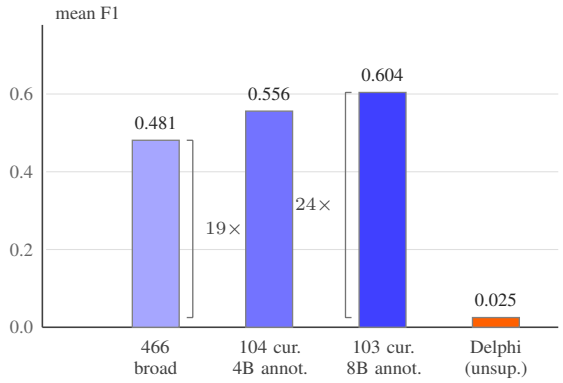

Figure 3. Mean F1 across three supervised-SAE operating points (blue) vs. the unsupervised `gpt2-small-res-jb` SAE described by real EleutherAI Delphi (Paulo et al., 2024) (orange). Both arms are scored identically: per-feature F1 of latent firing against the same Qwen3 annotator on the same held-out positions, not by Delphi's native top-k scorer.

0.5 and 0 subset pairs exceed $\max P \geq 0.95$, against the 5–15% high-IoU pair rates typically reported for unsupervised SAEs.

3.2. Supervised SAE vs. unsupervised SAE

The central comparison runs both pipelines end-to-end on the same held-out positions. The supervised arm generates descriptions *before* training; the unsupervised arm describes the trained latents of `gpt2-small-res-jb` *after* via real EleutherAI Delphi (Paulo et al., 2024). To rule out a small-catalog artifact, we report three operating points of the supervised arm against the matched-corpus Delphi run (Figure 3 and Table 2).

The gap holds across scales. Specifying 466 features upfront in a single bare-bones Opus pass produces a dictionary whose mean F1 is roughly 19× the corresponding Delphi mean. The unsupervised arm does not reach the supervised arm at any scale we tested.

A manual audit of all 91 Delphi descriptions explains the gap mechanistically. 65 of 91 (71%) are *non-statements*: phrases like "no consistent pattern" or "essentially arbitrary" that the corpus universally satisfies and the scorer cannot falsify. Another 22 (24%) are vague topical flavours ("low-quality web text", "mid-sentence informational content") too broad to score. Only 5 of 91 (5%) are specific, falsifiable token-level descriptions.

3.3. The unsupervised-latent catalog-quality cliff

A reviewer might object that the unsupervised SAE is fine and only the descriptions are bad. We tested this directly. The cascade is essentially Delphi with the explainer prompt constrained to require prefix-decidable yes/no descriptions: a description that does not pass that contract is dropped rather than scored. From the unsupervised SAE we examined the top ∼300 latents by ΔR^2, kept the best 100 for the cascade, asked Claude Sonnet 4.6 (Anthropic, 2026) to describe each one with full top-activating contexts in hand, and ran every surviving description through the supervised pipeline's quality gates.

The result is in Table 3. Of the 77 latents the LLM was

Table 2. F1 is computed by the same Qwen3 annotator against latent firing on the same held-out positions for both arms (not by Delphi's native scorer). Companion to Figure 3. n = features with ≥ 5 test positives.

arm	catalog	n	F1	gap
sup. (broad)	4B annot.	466	0.481	19×
sup. (curated, 4B)	4B annot.	104	0.556	22×
sup. (curated, 8B)	8B annot.	103	**0.604**	**24**×
unsup. Delphi	gpt2-small-res-jb	91	0.025	—

Table 3. Mining unsupervised latents into a quality-checked catalog. We started from the top ∼300 unsupervised latents by ΔR^2, kept the best 100 for the cascade, and only 2 survived. Layer 9 of GPT-2 Small.

stage	count
top-ΔR^2 unsupervised pool	∼300
selected for cascade (top-100)	100
nuisance-dropped (single-token detectors)	23
LLM-described	77
crisp single-concept singletons	1
multi-concept polysemantic bundles	65 (84%)
other rejections	11
decomposed atomic hypotheses	272
atoms passing crispness	64
atoms with ≥ 3 mini-positives	48
kept by merge gate	8
post-training $F_1 \geq 0.30$	2
surviving features	2

willing to describe, 65 (84%) are rejected as multi-concept polysemantic bundles, with the LLM's own crispness gate flagging them as "no single token-level pattern". To recover anything from those bundles, we decompose each one into $2-5$ atomic token-level hypotheses and re-run the cascade on the atoms. This rescues 48 atoms with enough support to compute a target direction, 8 pass the merge gate, and 2 survive post-training validation. Starting from a pool of ∼300 candidates and selecting the best 100, only 2 survive, and further analysis reveals that they're also polysemantic bundles. The unsupervised latents at this layer are mostly polysemantic bundles that no single-leaf yes/no question can describe. The catalog-driven supervised SAE gets to skip the unsupervised dictionary entirely, because that dictionary is the wrong shape for a clean catalog.

4. Discussion and Conclusion

We presented a catalog-driven supervised SAE pipeline. On layer 9 of GPT-2 Small the supervised dictionaries reach mean F1 of 0.481 at 466 features and 0.604 at 103 curated features (8B annotator), against 0.025 for real EleutherAI Delphi on the unsupervised 24,576-latent `gpt2-small-res-jb` SAE under identical scoring. The unsupervised dictionary itself does not admit a clean catalog: a strict cascade started from ∼300 top-ΔR^2 candidates and recovered only 2 named features, and 71% of Delphi descriptions are non-falsifiable strings the corpus universally satisfies.

Why not Delphi's native scorer? The supervised arm has no separate scoring stage; training against the labels *is* the scoring contract, so per-feature F1 against an external annotator is the only metric that applies to both sides. It is stricter than Delphi's native top-k scorer: polysemantic bundles can pass top-k and fail per-token, and per-token correctness is the system any downstream use (e.g. intervention, steering, monitoring) actually needs.

Compression vs. scaling. The standard response to interpretability gaps has been to scale the dictionary, but at 24,576 latents 84% of the high-ΔR^2 head still cannot be cleanly named. The supervised dictionary reaches 98.5% of the pretrained R^2 at 68× fewer latents, with every direction admitting falsifiable description and a 1100× edge in causal targeting. The standard paradigm appears to spend most of its capacity on splitting, redundancy, and polysemantic mixtures of cleaner underlying concepts.

Limitations. GPT-2 Small layer 9 is a token-surface regime: 95% of catalog leaves are surface, structural, or lexical-pattern features, and safety-relevant directions like refusal (Arditi et al., 2024; Engels et al., 2024) are not represented. Larger models at deeper layers are the natural next target. Per-feature causal-necessity tests on a 46-feature predecessor artifact yield 12 causally-specific features (KL up to 2.09 nats; top-1 prediction flips of 78% and 100% on the strongest two); replicating on the headline artifact is in progress.

Software and Data

Code, configuration files, and the LLM annotation prompts are released at `https://github.com/WesternDundrey/Automated_Feature_Selection`.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning, specifically the interpretability of large language models. Better-specified, more faithful feature dictionaries could improve our ability to audit and steer model behaviour, including safety-relevant behaviours such as refusal. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Anthropic. Claude Sonnet 4.6 system card. https://www.anthropic.com/claude-sonnet-4-6-system-card, February 2026.

Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., and Nanda, N. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Bloom, J., Tigges, C., and Chanin, D. SAELens. https://github.com/jbloomAus/SAELens, 2024.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023. https://transformer-circuits.pub/2023/monosemantic-features/.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Engels, J., Liao, I., Michaud, E. J., Gurnee, W., and Tegmark, M. Not all language model features are linear. *arXiv preprint arXiv:2405.14860*, 2024.

Gao, L., Dupré la Tour, T., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C., Ayonrinde, K., Till, D., Wearden, M., Conmy, A., Marks, S., and Nanda, N. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*, 2025.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. vLLM: Easy, fast, and cheap LLM serving with PagedAttention. https://github.com/vllm-project/vllm, 2023.

Makelov, A., Lange, G., and Nanda, N. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366*, 2024.

Nanda, N. and Bloom, J. TransformerLens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Paulo, G., Mallen, A., Juang, C., and Belrose, N. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

Qwen Team. Qwen3 technical report. https://github.com/QwenLM/Qwen3, 2025.

Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., and Nanda, N. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024a.

Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., and Nanda, N. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024b.

Shu, D., Wu, X., Zhao, H., Rai, D., Yao, Z., Liu, N., and Du, M. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1690–1712, 2025.