Sparse Autoencoders for Mechanistic Interpretability in NLP: An Evidence-Centered Survey

Anonymous ACL submission

Abstract

Sparse autoencoders (SAEs) have become central to recent research on superposition and feature structure in language models. However, the NLP literature now uses SAEs to support diverse claims, including linguistic feature discovery, circuit analysis, behavioral steering, hallucination diagnosis, refusal analysis, and unlearning. This survey asks what counts as credible evidence for SAE-based claims about language behavior. Unlike prior surveys that organize the field mainly around architectures, explanation methods, metrics, and broad applications, we propose an evidence-centered taxonomy that follows the lifecycle of an SAE claim: sparse-feature assumptions, training choices, feature explanation, semantic validation, causal validation, task-level intervention, and failure analysis. We synthesize SAE work across multilingual analysis, RAG and hallucination, refusal and safety, reasoning, dense retrieval, classification, ASR, and diffusion language models. Our analysis shows that many claims still rely heavily on reconstruction and sparsity metrics, with less evidence from semantic validation, robustness testing, causal intervention, or downstream task utility. To support more rigorous future work, we provide a scope protocol, literature demographics, synthesis tables, an evidence matrix, a lifecycle view, and a reporting checklist. We argue that SAE research in NLP should move beyond feature labels toward falsifiable, task-grounded, and causally validated mechanistic claims.

1 Introduction

Mechanistic interpretability seeks to explain model behavior through internal computations rather than input–output correlations. For transformer language models, this is difficult because neurons, attention heads, residual-stream directions, and MLP activations are often polysemantic. Sparse autoencoders offer a practical response by learning an overcomplete sparse dictionary that reconstructs model activations while activating only a small number of latent features per token or example. When successful, these latents can provide more interpretable units than the original activation basis (Cunningham et al., 2024; Gao et al., 2024; Rajamanoharan et al., 2024a,b).

The rapid growth of SAE work has created a methodological challenge: reconstruction quality, plausible feature labels, causal involvement, and downstream control are distinct evidential claims. Recent work shows that SAE descriptions can be overbroad (Ma et al., 2026b), that one explanation can describe multiple features (McCann, 2026), that sparse features may not outperform simpler probes (Kantamneni et al., 2025), that one-dimensional feature assumptions can fail (Engels et al., 2024; Bhalla et al., 2026), and that interpretability does not necessarily imply steering utility (Wang et al., 2025a). At the same time, SAEs have been applied to central NLP problems, including multilingual representations (Deng et al., 2025; Cho and Hockenmaier, 2025a; Jing et al., 2025), RAG and hallucination (Xin et al., 2025; Abdaljalil et al., 2025; Zhao et al., 2025b; Chen et al., 2026), refusal and unlearning (Yeo et al., 2025; O'Brien et al., 2024; Wang et al., 2025b; Muhamed et al., 2025a), reasoning and in-context learning (Demircan et al., 2024; Galichin et al., 2025; Fang et al., 2026), dense retrieval (Kang et al., 2025; O'Neill et al., 2024), artificial-text detection (Kuznetsov et al., 2025), ASR (Pluth et al., 2026), and diffusion language models (Wang et al., 2026).

This survey asks: *what do SAEs currently let us understand and control in NLP systems, and what evidence is needed for those claims to be credible?* Unlike broad SAE surveys such as Shu et al. (2025), we focus specifically on SAE use in NLP and organize the literature around evidential strength rather than architecture alone. Our central distinction is between *feature discovery*, *feature explanation*, *causal mechanism*, and *behavioral*

control.

Contributions. We make five contributions. First, we introduce an evidence-centered taxonomy that tracks the lifecycle of an SAE claim, from assumptions and training choices to explanation, validation, intervention, and failure analysis. Second, we synthesize SAE work across NLP areas, including multilinguality, RAG, hallucination, refusal, safety, reasoning, retrieval, classification, ASR, and diffusion language models. Third, we provide a structured map of 115 candidate papers and 77 core works, including inclusion criteria, literature demographics, and an evidence matrix. Fourth, we analyze recurring limitations, critiques, and negative findings, including overbroad explanations, descriptive collision, feature splitting, sparse-probing limits, multidimensional concepts, concept manifolds, neuron-basis sparsity, rate–distortion–polysemanticity tradeoffs, and causal overclaiming. Fifth, we provide practical resources for future work: synthesis tables, benchmark/testbed comparisons, a lifecycle figure, and a reporting checklist.

Comparison to existing surveys. Existing surveys cover broad LLM explainability and mechanistic interpretability (Zhao et al., 2024; Ferrando et al., 2024; Rai et al., 2024; Bereska and Gavves, 2024), causal mediation (Mueller et al., 2026), concept descriptions (Feldhus and Kopf, 2025), intervention pipelines (Zhang et al., 2026), or SAEs as a general framework for interpreting LLM internals (Shu et al., 2025). In contrast, our survey is NLP-specific and evidence-centered: it asks what supports claims about language behavior, task mechanisms, causal intervention, and robust control.

Proposed evidence-centered taxonomy. Figure 1 illustrates the lifecycle that motivates our taxonomy. We treat SAE feature labels as hypotheses rather than explanations, requiring semantic validation, causal validation, task-level intervention, and failure analysis. Table 1 maps this lifecycle onto the main bodies of SAE work in NLP.

2 Background: Sparse Features in Language Models

Superposition and sparse coding. The SAE research program is motivated by the hypothesis that language models represent more features than they have directly observable dimensions, causing multiple features to be superposed in the same neuron or residual-stream direction. SAEs learn a sparse latent basis in which activations can be reconstructed from a small set of active features. In the standard formulation, an encoder maps an activation vector x into sparse latents z, and a decoder reconstructs $\hat{x}$:

$$z = f_\theta(x), \qquad \hat{x} = Dz, \tag{1}$$

with a training objective balancing reconstruction fidelity and sparsity:

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \lambda\Omega(z). \tag{2}$$

The activation function, sparsity penalty, dictionary size, token distribution, layer, and activation site determine what features the SAE can recover. Early language-model SAE work showed that sparse latents can be highly interpretable (Cunningham et al., 2024), while scaling work showed that larger dictionaries and controlled sparsity can improve reconstruction and feature quality (Gao et al., 2024; He et al., 2024b). Gated, JumpReLU, Top-K, BatchTopK, Binary, Matryoshka, Specialized, Group, Route, and Temporal variants each address weaknesses of vanilla SAEs, including shrinkage, dead latents, training cost, feature splitting, rare concepts, multi-layer structure, global sparsity, and sequential language structure (Rajamanoharan et al., 2024a,b; Quirke et al., 2025; Bussmann et al., 2025; Muhamed et al., 2025b; Ghilardi et al., 2025; Shi et al., 2025; Bhalla et al., 2025).

Why NLP is not just another SAE domain. NLP adds constraints that are easy to miss in generic SAE surveys. Language features are often compositional, contextual, and temporally extended: a feature may represent a grammatical relation, discourse role, refusal state, retrieved-context preference, or reasoning phase rather than a local token property. Multilingual and low-resource settings require features to be compared across languages and scripts (Deng et al., 2025; Cho and Hockenmaier, 2025a; Bondielli et al., 2025; Tumurchuluun et al., 2025). Many NLP applications also require intervention, not only explanation: RAG systems should rely on context when appropriate, safety systems should refuse harmful requests without broadly degrading helpfulness, and reasoning systems should be steerable without inducing superficial cue-following (Xin et al., 2025; Zhao et al., 2025b; O'Brien et al., 2024; Ma et al., 2026a). These requirements make causal validation and task-grounded evaluation central.

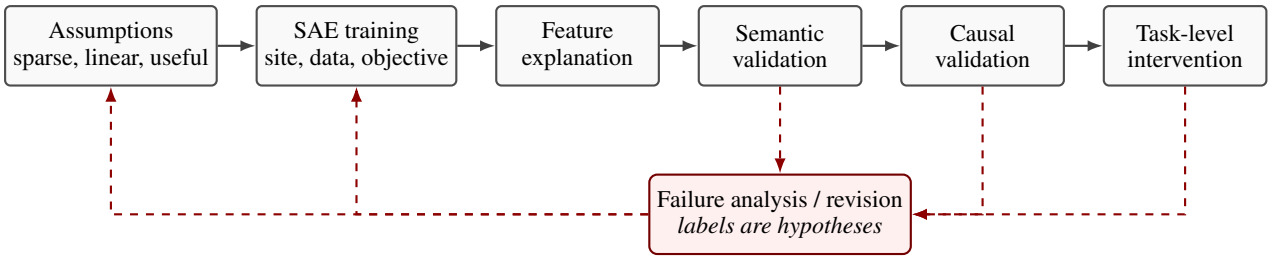

Figure 1: Lifecycle of an SAE claim. Feature labels are treated as hypotheses rather than final explanations. Strong SAE work should move from sparse-feature assumptions and training choices to semantic validation, causal validation, task-level intervention, and failure analysis. Dashed arrows indicate revision loops.

Pillar	Guiding question	Representative citations	Survey role
- Foundations and assumptions	Are LLM features sparse, linear, one-dimensional, and monosemantic?	(Cunningham et al., 2024; Deng et al., 2023; Engels et al., 2024; Peng et al., 2025; Arora et al., 2026; Bhalla et al., 2026; Mencattini et al., 2026)	Defines what SAE claims can and cannot mean.
- SAE design and scaling	Which objectives, activations, and training regimes improve fidelity without losing interpretability?	(Gao et al., 2024; Rajamanoharan et al., 2024a,b; Quirke et al., 2025; Bhalla et al., 2025; Muhamed et al., 2025b; Ghilardi et al., 2025; Shi et al., 2025; He et al., 2024b; Bussmann et al., 2025)	Compares architecture and resource choices.
- Feature explanation and evaluation	How do we label, score, falsify, and stress-test SAE features?	(Paulo et al., 2024; Gur-Arieh et al., 2025; Han et al., 2026; McCann, 2026; Ma et al., 2026b; Gulko et al., 2025; Mercorio et al., 2025; Karvonen et al., 2025; Makelov et al., 2024; Ma et al., 2026a)	Makes feature claims auditable.
- Circuits and causal mechanisms	Can sparse features become causal graphs rather than isolated labels?	(Marks et al., 2024; O'Neill and Bui, 2024; He et al., 2024a; Kissane et al., 2024; Dunefsky et al., 2024; Paulo et al., 2025)	Connects latents to computation.
- NLP task mechanisms	What linguistic, multilingual, retrieval, hallucination, refusal, and reasoning mechanisms are exposed?	(Deng et al., 2025; Jing et al., 2025; Xin et al., 2025; Abdaljalil et al., 2025; Yeo et al., 2025; Demircan et al., 2024; Galichin et al., 2025)	Organizes evidence by NLP behavior.
- Intervention and control	When do SAE features steer, edit, unlearn, or improve behavior reliably?	(O'Brien et al., 2024; Wang et al., 2025b; Muhamed et al., 2025a; Cho and Hockenmaier, 2025b; Zhao et al., 2026; Soo et al., 2025; Mayne et al., 2024; Chalnev et al., 2024)	Separates understanding from action.
- Beyond autoregressive text	Do SAE findings transfer to diffusion LMs, ASR, and multimodal language systems?	(Wang et al., 2026; Pluth et al., 2026; Zaigrajew et al., 2025; Bussmann et al., 2025)	Tests architecture and modality generality.

Table 1: Evidence-centered taxonomy of SAE work in NLP. The taxonomy separates assumptions, SAE design, feature explanation, evaluation, causal mechanisms, interventions, and extensions beyond autoregressive text.

Overall, the background literature shows that SAEs can produce meaningful sparse decompositions of LM activations. It does not show that all relevant NLP concepts are sparse, linear, or one-dimensional. Future NLP papers should state which activation distribution they study, why an SAE is the right tool, and which alternative bases or probes serve as serious baselines.

3 Foundations and assumptions

SAE-based interpretability depends on assumptions that make sparse features meaningful: relevant computations should be recoverable in a sparse basis, many features should be approximately linear, some latents should be monosemantic enough to label, and the SAE basis should add value over neurons, heads, probes, or steering vectors. These assumptions are productive, but they define the limits of what SAE claims can mean.

The positive case is that sparse decompositions can reveal interpretable structure in language-model activations. Cunningham et al. (2024) show that SAEs can recover highly interpretable features, making it plausible that superposed representations can be decomposed into more human-legible units. Deng et al. (2023) make this assumption measurable by proposing feature-sparsity metrics rather than relying only on qualitative dashboard inspection. Together, these papers justify SAEs as discovery tools when sparse feature structure is present and can be evaluated systematically.

However, SAE features should not be treated as ground-truth concepts. Some concepts are not well described by a single linear direction: Engels et al. (2024) argue that not all language-model features are one-dimensionally linear, while Bhalla et al. (2026) ask whether some concepts are better understood as manifolds. This matters for NLP because reasoning, tense, refusal, multilingual transfer, hallucination, discourse, and retrieved-context reliance are often compositional and context-dependent. Treating them as one-latent

concepts can produce misleading explanations.

A second challenge concerns the choice of basis. If SAEs are useful because they find a better basis than neurons, they should be compared against the neuron basis rather than assumed superior. Arora et al. (2026) show that some language-model circuits can already be sparse in neurons, and Peng et al. (2025) argue that SAEs are better suited for discovering unknown concepts than acting on known ones. When the target concept is known, a supervised probe, neuron/head analysis, or direct steering vector may answer the question more simply.

A third challenge is the trade-off between reconstruction, sparsity, and monosemanticity. SAE quality cannot be inferred from reconstruction error or L_0 alone. Mencattini et al. (2026) formalize the rate–distortion–polysemanticity trade-off, showing that gains on one objective can come at the expense of another. A feature may reconstruct activations accurately while remaining polysemantic, be sparse without corresponding to an NLP-relevant behavior, or appear interpretable without having a causal effect.

Overall, this pillar shows that SAEs are powerful but assumption-laden tools. They are most convincing for exploratory feature discovery when the target behavior plausibly has sparse structure and when SAE features are compared against simpler baselines. Future NLP papers should state which assumption their claim depends on, test whether the target concept is one-dimensional or distributed, compare against neuron/probe/head baselines, and avoid treating feature names as evidence of mechanism.

Table 5 in Appendix B summarizes the main assumptions, failure modes, and stress tests for SAE-based NLP claims.

4 SAE Architectures, Training, and Scaling

Architecture papers ask how to improve the reconstruction–sparsity frontier while preserving interpretability. Gated SAEs reduce shrinkage by separating feature selection from magnitude estimation (Rajamanoharan et al., 2024a); JumpReLU SAEs target L_0 sparsity more directly (Rajamanoharan et al., 2024b); and Top-K or BatchTopK variants provide explicit activation budgets (Gao et al., 2024). Other variants address scalability and coverage: RouteSAE and Group-SAE reduce layer-wise training cost and capture multi-layer structure (Shi et al., 2025; Ghilardi et al., 2025); Specialized SAEs recover rare long-tail concepts (Muhamed et al., 2025b); Temporal SAEs incorporate sequential structure (Bhalla et al., 2025); and Binary and Matryoshka variants explore stricter discreteness and multi-level feature granularity (Quirke et al., 2025; Cho et al., 2025; Bussmann et al., 2025).

Different SAE variants should therefore not be compared only by reconstruction loss. For interpretability, a useful SAE should satisfy at least four desiderata: (i) faithful reconstruction of the activation distribution being studied; (ii) sparse activation patterns that avoid dense hidden information; (iii) stable, discriminable, and robust feature semantics; and (iv) utility for causal analysis or intervention. Table 6 in Appendix C summarizes how major SAE variants trade off these desiderata, including their effect on shrinkage, dead latents, sparsity control, rate–distortion–polysemanticity, NLP utility, and limitations.

Overall, architecture work has shown real progress on fidelity, sparsity control, and scalability, but it has not shown that a single SAE variant dominates for all NLP questions. Future papers should select the architecture based on the claim being made: rare-concept discovery, multilingual comparison, circuit analysis, and behavioral steering may require different training distributions, sparsity mechanisms, and validation protocols.

5 Feature explanation and evaluation

After training an SAE, the central question is whether its latents support reliable explanations. A feature label should not be accepted because it sounds plausible for highly activating examples. It should predict where the feature activates, where it should not activate, how it affects outputs, and whether the proposed meaning survives held-out examples, counterexamples, paraphrases, domain shift, and intervention.

Automated interpretability pipelines scale feature labeling by generating descriptions from activating examples and scoring activation prediction (Paulo et al., 2024). Output-centric methods improve this by asking how a feature changes logits or generated text (Gur-Arieh et al., 2025), and agentic explainers iteratively propose, test, and revise hypotheses (Han et al., 2026). Yet critique papers show why explanation must be falsifiable: feature descriptions can be overbroad (Ma et al.,

2026b), multiple features can share a description (McCann, 2026), and abstract labels such as "reasoning" may capture correlates rather than mechanisms (Ma et al., 2026a).

Table 2 summarizes evaluation frameworks that make SAE feature claims auditable. Reconstruction and sparsity assess technical fidelity; contrastive and formal-language benchmarks test semantic discrimination; output-centric descriptions test functional effects; robustness tests evaluate stability; and downstream evaluations measure task relevance. No single benchmark establishes all of these properties, so strong NLP papers should combine multiple evidence types.

Overall, feature explanation is moving from description toward auditability. Current methods can generate and score labels at scale, but it remains unproven that automatic scores reliably track mechanistic faithfulness across models, languages, and tasks. Future NLP papers should pair labels with positive and negative examples, held-out validation, robustness tests, causal interventions, downstream task effects, and non-SAE baselines.

6 Circuits and Causal Mechanisms

Feature labels alone do not explain computation. A latent labeled "refusal", "retrieval reliance", or "subject–verb agreement" becomes mechanistic only when we can show which upstream features activate it, which downstream components use it, and whether intervening on the pathway changes behavior. This pillar asks whether sparse features can become nodes in causal graphs rather than isolated labels.

Sparse feature circuit work uses SAE features as graph nodes and identifies edges supporting model behaviors (Marks et al., 2024). SAE-based circuit-identification methods aim to make circuit discovery more scalable and less dependent on brittle manual patching (O'Neill and Bui, 2024), while dictionary-learning approaches support patch-free circuit discovery (He et al., 2024a). Attention-output SAEs extend sparse decomposition beyond residual-stream and MLP sites, making them relevant for attention-mediated NLP behaviors such as copying, coreference, retrieval, and long-context information flow (Kissane et al., 2024).

Transcoders are an important comparison point because they model a component's input-output transformation rather than reconstructing one activation space (Dunefsky et al., 2024). Comparative work suggests that transcoders can sometimes provide more direct circuit interpretations than SAEs (Paulo et al., 2025). The synthesis is therefore methodological: the strongest SAE circuit papers should report pathways, edge-level evidence, interventions, and side effects. What remains unproven is whether sparse feature circuits scale reliably to open-ended NLP tasks, multilingual settings, long-context reasoning, and deployed RAG or safety systems.

7 Mechanisms Found in NLP Tasks

7.1 Multilingual and linguistic mechanisms

Multilingual studies show that SAEs can expose language-specific and cross-lingual structure. Work on language-specific features identifies latents associated with individual languages and proposes metrics for monolinguality (Deng et al., 2025). Related work studies multilingual processing with task-instruction-focused and heading-focused features (Cho and Hockenmaier, 2025a), Italian small language models (Bondielli et al., 2025), grammatical tense localization and control across typologically diverse languages (Tumurchuluun et al., 2025), and broader linguistic mechanisms such as reference, metaphor, and syntactic/semantic abilities (Jing et al., 2025). This cluster of work is central because it moves SAE research from generic feature examples toward questions that are native to language.

7.2 RAG, retrieval, and hallucination

RAG systems expose a natural mechanistic question: *when does a model rely on retrieved context versus parametric memory?* SAE work has identified sparse latents associated with context-memory prioritization and generation decisions in RAG (Xin et al., 2025). SAE-based representation engineering can steer knowledge selection under context-memory conflicts (Zhao et al., 2025b). Dense retrieval can be interpreted and controlled with sparse latent features (Kang et al., 2025). Hallucination-focused systems use SAE features or latent dynamics for detection, query enrichment, and mitigation (Abdaljalil et al., 2025; Chen et al., 2026). Non-SAE hallucination methods based on attention maps or topology are useful baselines (Chuang et al., 2024; Sun et al., 2024; Bazarova et al., 2025; Samaga et al., 2026), but the SAE contribution is the possibility of identifying features that can be both interpreted and intervened on.

Benchmark / framework	Primary focus	Scale	Evidence type	Limitation and NLP relevance
SAEBench (Karvonen et al., 2025)	General SAE evaluation for interpretability and control.	Multi-model benchmark.	Reconstruction, sparsity, interpretability, and control metrics.	Useful as a common reporting baseline, but still needs richer task-specific NLP extensions.
CE-Bench (Gulko et al., 2025)	Contrastive evaluation of feature descriptions.	Benchmark suite.	Positive/negative concept discrimination.	Stronger than top-activation inspection, but depends on contrast-set quality; relevant for linguistic, safety, and refusal labels.
Formal-language testbeds (Menon et al., 2024)	Controlled evaluation of known symbolic or automata-like behavior.	Synthetic controlled languages.	Ground-truth mechanism, counterfactual, and feature-identifiability evidence.	Useful for falsification, but may not transfer directly to natural language phenomena.
Monosemanticity scores (Fereidouni et al., 2026; Härle et al., 2025)	Degree to which features support single concepts.	Feature-level metrics.	Semantic concentration and feature purity.	Can miss causal relevance or task utility; useful but insufficient for NLP claims.
Output-centric explanations (Gur-Arieh et al., 2025)	Feature effects on outputs rather than inputs alone.	Explanation pipeline.	Logit-level or generation-level functional evidence.	Better aligned with generation, but still requires causal and robustness tests.
Adversarial robustness tests (Li et al., 2026)	Stability under perturbation, paraphrase, and domain shift.	Prompt/concept perturbation sets.	Robustness and counterexample evidence.	Essential for NLP because wording, language, and domain can change while the underlying concept remains stable.
Downstream task evaluations (Makelov et al., 2024)	Utility in RAG, refusal, retrieval, classification, ASR, or steering.	Task-specific.	Behavioral metrics, intervention effects, and side-effect analysis.	Required for NLP relevance, but hard to compare across tasks and models.

Table 2: Evaluation benchmarks and testbeds for SAE feature claims. The frameworks differ in what they make auditable: technical fidelity, semantic validity, robustness, causal relevance, or downstream NLP utility.

7.3 Safety, refusal, unlearning, and alignment

Refusal and unlearning are among the clearest examples where interpretability and control meet. SAE analyses of refusal identify latent features that mediate refusal behavior and test their causal influence through interventions (Yeo et al., 2025; O'Brien et al., 2024). Unlearning work uses SAE features or SAE-guided subspaces to remove targeted knowledge while limiting side effects (Farrell et al., 2024; Muhamed et al., 2025a; Wang et al., 2025b). Sparse model diffing predicts side effects after unlearning or fine-tuning (Kassem et al., 2025). Interpretable reward modeling uses SAE features to make preference signals more transparent and adjustable (Zhang et al., 2025). This literature also exposes a recurring tension: steering safety features can improve robustness but degrade general capability if interventions are too broad (O'Brien et al., 2024).

7.4 Reasoning, in-context learning, and instruction following

SAE-based studies of reasoning ask whether latent features correspond to reasoning stages or superficial cues. Temporal-difference learning work uses SAEs to study in-context reinforcement-learning behavior (Demircan et al., 2024). Reasoning-feature papers identify sparse features associated with reasoning language or strategies (Galichin et al., 2025; Fang et al., 2026). However, falsification work warns that contrastively selected reasoning features may capture stable correlates or cues rather than reasoning mechanisms themselves (Ma et al., 2026a). This is a useful case study for the entire field: the more abstract the target behavior, the stronger the need for counterexamples, causal interventions, and out-of-distribution validation. Moreover, instruction-following studies use SAE features to locate and steer prompt-format and narrative-compliance mechanisms (Zhao et al., 2025a; Cho and Hockenmaier, 2025b; He et al., 2025).

7.5 Classification, dense representations, artificial-text detection, ASR, and diffusion LMs

SAEs are increasingly used outside next-token-generation internals. Artificial-text detection work uses SAE features to explain detector behavior (Kuznetsov et al., 2025). Classification studies evaluate layer selection, scaling, architecture, and transferability of SAE features (Gallifant et al., 2025; Bail et al., 2025). Dense embedding work applies SAEs to text embeddings and semantic search (O'Neill et al., 2024); company-similarity work shows how sparse features can make domain-specific similarity more transparent (Molinari et al., 2025). Diffusion language models and ASR extend SAE interpretability to non-autoregressive and speech-language systems (Wang et al., 2026; Pluth et al., 2026).

Synthesis. Task papers have shown that SAEs can localize useful signals in multilingual processing, RAG, hallucination, refusal, retrieval, classification, and reasoning. What remains unproven is whether these features generalize across prompts,

languages, domains, and model families. Future NLP papers should make task-level predictions before feature selection, compare against non-SAE baselines, and report both target improvement and collateral behavioral changes.

8 From Understanding to Intervention

SAE-based interventions ask whether sparse features can control behavior, not only explain it. The reviewed papers use feature ablation, feature amplification, sparse subspace projection, feature-guided steering vectors, SAE-guided unlearning, and sparse feature circuits. These strategies differ in claim strength: some test local causal relevance, while others aim for robust task-level control.

Feature-guided activation additions and targeted steering use SAE latents to make activation steering more interpretable (Soo et al., 2025; Chalnev et al., 2024). However, steering-vector decomposition shows that SAE features do not always cleanly explain or improve existing steering directions (Mayne et al., 2024). Concept-vector denoising removes noisy dimensions from steering vectors (Zhao et al., 2026); SAE-guided projections and guardrails target unlearning and safety editing (Wang et al., 2025b; Muhamed et al., 2025a); and sparse feature circuits provide stronger intervention objects by localizing behavior in graphs of interacting features (Marks et al., 2024; O'Neill and Bui, 2024). Table 7 in Appendix D summarizes the main intervention modalities and the evidence needed to distinguish local effects from robust control.

We recommend distinguishing three levels of success: *local effect*, where a feature changes a logit or activation; *behavioral effect*, where an intervention changes an NLP outcome; and *specific, robust control*, where the target behavior changes without unacceptable side effects across prompts, domains, languages, and model states. Many papers establish the first two levels; fewer establish the third. Future NLP papers should report ablation and amplification curves, held-out and adversarial prompts, side-effect metrics, and baselines such as probes, neurons, heads, transcoders, or ordinary steering vectors.

9 Challenges, Failure Modes, and Open Problems

SAEs scale feature-level analysis, but also make overclaiming easier: a feature may be easy to label without being robust or causal. We ask not only *"what can go wrong?"*, but *which failures are specific to SAE-based NLP claims, and what evaluations reveal them.*

9.1 Failure modes of SAE-based NLP claims

The first failure class concerns feature meaning. Top activating examples can support descriptions that are too broad, and McCann (2026) show descriptive collision: one explanation can fit multiple distinct features. Labels such as "reasoning", "refusal", "translation", or "hallucination" are semantically elastic, so they require hard negatives, held-out examples, and output-centered tests (Ma et al., 2026b; Gur-Arieh et al., 2025).

The second class concerns feature geometry. Some concepts are not one-dimensional (Engels et al., 2024) and may form manifolds rather than isolated directions (Bhalla et al., 2026). For tense, multilingual transfer, discourse, refusal, and reasoning, a one-latent explanation may split, absorb, or distort the mechanism. The third class concerns basis choice: neuron or head circuits may already be sparse (Arora et al., 2026), supervised probes may be better for known concepts (Kantamneni et al., 2025; Peng et al., 2025), and interpretable features may not improve downstream utility (Wang et al., 2025a; Mayne et al., 2024).

The fourth class concerns metric overinterpretation. Reconstruction, sparsity, and monosemanticity are useful diagnostics, but none proves mechanism. The rate–distortion–polysemanticity tradeoff shows that improving one objective can weaken another (Mencattini et al., 2026). A feature can reconstruct well and still be polysemantic, be sparse and lack causal relevance, or be easy to name and fail under paraphrase, translation, domain shift, or intervention.

Table 8 in Appendix E summarizes the main failure modes, why they matter for NLP, and the evaluations that can expose them.

9.2 When should an NLP researcher not use an SAE?

An SAE is not always the right technique. If the relevant circuit is already sparse in neurons or attention heads, direct circuit analysis may be simpler and more faithful (Arora et al., 2026). If the target concept is known and the question is prediction rather than discovery, a supervised probe may answer it more directly (Kantamneni et al., 2025; Peng et al., 2025). If reconstruction quality is poor at the activation site, causal claims are unreliable.

If the phenomenon is distributed or manifold-like, a single-latent analysis can distort it (Engels et al., 2024; Bhalla et al., 2026). Finally, if the goal is behavioral control and SAE features do not outperform simpler steering vectors, probes, or editing methods, the SAE adds interpretive cost without enough benefit (Wang et al., 2025a; Mayne et al., 2024). This guidance matters because many NLP questions are task-driven: for retrieval, hallucination reduction, classification, or refusal steering, SAE features should add value over simpler baselines; otherwise, they may be useful for exploration but unnecessary for intervention.

9.3 Open problems for SAE-based NLP interpretability

Multilingual and low-resource coverage. SAE evidence remains concentrated in English and high-resource models. Work on language-specific features, Italian small language models, multilingual tense localization, and ASR suggests that SAEs can support analysis beyond English (Deng et al., 2025; Bondielli et al., 2025; Tumurchuluun et al., 2025; Pluth et al., 2026). Broader typological coverage is still needed to test whether SAE features align across languages, scripts, morphologies, and low-resource settings.

Task-faithful benchmarks. SAE evaluation should use benchmarks with explicit mechanisms or behavioral goals. Formal languages provide controlled ground truth (Menon et al., 2024); CE-Bench evaluates descriptions contrastively (Gulko et al., 2025); retrieval conflicts, refusal benchmarks, and reasoning-feature tests provide task-grounded settings (Zhao et al., 2025b; Yeo et al., 2025; Ma et al., 2026a). The open problem is designing benchmarks that are both causally controlled and realistic for NLP.

Causal feature circuits. Many NLP behaviors are multi-token and multi-component, so SAE papers should move from isolated features to circuits of interacting features (Marks et al., 2024; O'Neill and Bui, 2024; Dunefsky et al., 2024). For RAG, reasoning, refusal, and instruction following, a convincing explanation should show how information flows across layers and token positions.

Robust and specific interventions. Steering and unlearning papers should report side effects, adversarial prompts, paraphrases, and domain shift, not only target-task improvement (O'Brien et al., 2024; Wang et al., 2025b; Muhamed et al., 2025a; Zhao et al., 2026). Strong intervention claims should show that the target behavior changes while unrelated capabilities remain stable.

Beyond autoregressive LLMs. Diffusion language models and ASR systems extend SAE interpretability to new NLP architectures and modalities (Wang et al., 2026; Pluth et al., 2026). These settings test whether SAE feature claims are specific to autoregressive transformers or reflect more general properties of language representations. They are also important for the NLP community because modern NLP increasingly includes speech, multimodal, and non-autoregressive generation systems.

Synthesis. The failure-mode literature strengthens the standard for credible SAE claims: labels must be falsifiable, baselines serious, and interventions evaluated for specificity and side effects. For NLP research, SAE interpretability should be judged by whether it explains language behavior under realistic variation across prompts, languages, domains, tasks, and model architectures.

10 Conclusion

SAEs are promising for mechanistic interpretability in NLP, but naming sparse features is not enough. The reviewed literature shows real progress in feature discovery, multilingual analysis, RAG and hallucination, refusal and unlearning, reasoning, dense retrieval, ASR, and diffusion language models. It also shows that feature descriptions can be overbroad, non-robust, non-causal, or less useful than simpler baselines. The field must therefore move from feature dashboards to falsifiable, task-grounded, causally validated mechanistic claims. For the NLP community, the strongest SAE work will explain language behavior, test counterexamples, compare baselines, intervene specifically, and report failure modes as part of the evidence rather than as an afterthought.

Limitations

This survey provides a systematic snapshot of a fast-moving literature, but it is not exhaustive. Despite searching major NLP/ML venues, arXiv, OpenReview, Google Scholar, Google Scholar Labs, and citation trails, some recent preprints or updated 2026 versions may be missing. The reviewed literature is also uneven. Most SAE evidence still comes from English, high-resource settings, and

a limited set of model families, so claims about multilingual or low-resource generality remain tentative. Cross-paper comparison is further limited by differences in base models, activation sites, SAE objectives, dictionary sizes, sparsity targets, and evaluation protocols. Our full-text evidence counts indicate whether an evidence type is reported, not how strong that evidence is. We therefore use them as coverage statistics rather than quality scores. Finally, this survey synthesizes published claims rather than re-running SAE training or interventions, and our taxonomy should be read as an analytical scaffold rather than a rigid classification.

Use of AI Assistance

In preparing this work, we used AI-assisted tools for code completion with GitHub Copilot and language editing, such as spell checking and stylistic revisions with Grammarly and ChatGPT. The authors take full responsibility for the scientific content, analyses, and conclusions presented in this work.

References

Samir Abdaljalil, Filippo Pallucchini, Andrea Seveso, Hasan Kurban, Fabio Mercorio, and Erchin Serpedin. 2025. SAFE: A sparse autoencoder-based framework for robust query enrichment and hallucination mitigation in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9335–9346, Suzhou, China. Association for Computational Linguistics.

Aryaman Arora, Zhengxuan Wu, Jacob Steinhardt, and Sarah Schwettmann. 2026. Language model circuits are sparse in the neuron basis. *Preprint*, arXiv:2601.22594.

Mathis Le Bail, Jérémie Dentan, Davide Buscaldi, and Sonia Vanier. 2025. Unveiling decision-making in llms for text classification : Extraction of influential and interpretable concepts with sparse autoencoders. *Preprint*, arXiv:2506.23951.

Alexandra Bazarova, Andrei Volodichev, Aleksandr Yugay, Andrey Shulga, Alina Ermilova, Konstantin Polev, Julia Belikova, Rauf Parchiev, Dmitry Simakov, Maxim Savchenko, Andrey Savchenko, Serguei Barannikov, and Alexey Zaytsev. 2025. Hallucination detection in llms with topological divergence on attention graphs. *Preprint*, arXiv:2504.10063.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for ai safety – a review. *Transactions on Machine Learning Research*.

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Vasudev Shyam, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. 2026. Do sparse autoencoders capture concept manifolds? *Preprint*, arXiv:2604.28119.

Usha Bhalla, Alex Oesterling, Claudio Mayrink Verdun, Himabindu Lakkaraju, and Flavio P. Calmon. 2025. Temporal sparse autoencoders: Leveraging the sequential nature of language for interpretability. *Preprint*, arXiv:2511.05541.

Alessandro Bondielli, Lucia C. Passaro, and Alessandro Lenci. 2025. Sparse autoencoders find partially interpretable features in Italian small language models. In *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025)*, pages 95–104, Cagliari, Italy. CEUR Workshop Proceedings.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. 2025. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*.

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. 2024. Improving steering vectors by targeting sparse autoencoder features. *Preprint*, arXiv:2411.02193.

Boshui Chen, Zhaoxin Fan, Ke Wang, Zhiying Leng, Faguo Wu, Hongwei Zheng, Yifan Sun, and Wenjun Wu. 2026. Hallusae: Detecting hallucinations in large language models via sparse auto-encoders. *Preprint*, arXiv:2604.16430.

Hakaze Cho, Haolin Yang, Yanshu Li, Brian M. Kurkoski, and Naoya Inoue. 2025. Binary autoencoder for mechanistic interpretability of large language models. *Preprint*, arXiv:2509.20997.

Ikhyun Cho and Julia Hockenmaier. 2025a. Analyzing multilingualism in large language models with sparse autoencoders. In *Second Conference on Language Modeling*.

Ikhyun Cho and Julia Hockenmaier. 2025b. Toward efficient sparse autoencoder-guided steering for improved in-context learning in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28961–28973, Suzhou, China. Association for Computational Linguistics.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*.

Can Demircan, Tankred Saanum, Akshay K. Jagadish, Marcel Binz, and Eric Schulz. 2024. Sparse autoencoders reveal temporal difference learning in large language models. *Preprint*, arXiv:2410.01280.

Boyi Deng, Yu Wan, Baosong Yang, Yidan Zhang, and Fuli Feng. 2025. Unveiling language-specific features in large language models via sparse autoencoders. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4563–4608, Vienna, Austria. Association for Computational Linguistics.

Mingyang Deng, Lucas Tao, and Joe Benton. 2023. Measuring feature sparsity in language models. *Preprint*, arXiv:2310.07837.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable llm feature circuits. *Preprint*, arXiv:2406.11944.

Joshua Engels. 2025. *Towards More Interpretable AI With Sparse Autoencoders*. Ph.D. thesis, Massachusetts Institute of Technology.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. 2024. Not all language model features are one-dimensionally linear. *Preprint*, arXiv:2405.14860.

Yi Fang, Wenjie Wang, Mingfeng Xue, Boyi Deng, Fengli Xu, Dayiheng Liu, and Fuli Feng. 2026. Controllable llm reasoning via sparse autoencoder-based steering. *Preprint*, arXiv:2601.03595.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. 2024. Applying sparse autoencoders to unlearn knowledge in language models. *Preprint*, arXiv:2410.19278.

Nils Feldhus and Laura Kopf. 2025. Interpreting language models through concept descriptions: A survey. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 149–162, Suzhou, China. Association for Computational Linguistics.

Moghis Fereidouni, Muhammad Umair Haider, Peizhong Ju, and A.b. Siddique. 2026. Evaluating sparse autoencoders for monosemantic representation. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 5969–5984, Rabat, Morocco. Association for Computational Linguistics.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Andrey Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan Oseledets. 2025. I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders. *Preprint*, arXiv:2503.18878.

Jack Gallifant, Shan Chen, Kuleen Sasse, Hugo Aerts, Thomas Hartvigsen, and Danielle Bitterman. 2025. Sparse autoencoder features for classifications and transferability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29939–29963, Suzhou, China. Association for Computational Linguistics.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *Preprint*, arXiv:2406.04093.

Davide Ghilardi, Federico Belotti, Marco Molinari, and Jaehyuk Lim. 2024. Accelerating sparse autoencoder training via layer-wise transfer learning in large language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 530–550, Miami, Florida, US. Association for Computational Linguistics.

Davide Ghilardi, Federico Belotti, Marco Molinari, Tao Ma, and Matteo Palmonari. 2025. Group-SAE: Efficient training of sparse autoencoders for large language models via layer groups. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18657–18677, Suzhou, China. Association for Computational Linguistics.

Alex Gulko, Yusen Peng, and Sachin Kumar. 2025. CE-bench: Towards a reliable contrastive evaluation benchmark of interpretability of sparse autoencoders. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 1–15, Suzhou, China. Association for Computational Linguistics.

Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. 2025. Enhancing automated interpretability with output-centric feature descriptions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5757–5778, Vienna, Austria. Association for Computational Linguistics.

Jiaojiao Han, Wujiang Xu, Mingyu Jin, and Mengnan Du. 2026. SAGE: An agentic explainer framework for interpreting SAE features in language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 483–495, Rabat, Morocco. Association for Computational Linguistics.

Zhengfu He, Xuyang Ge, Qiong Tang, Tianxiang Sun, Qinyuan Cheng, and Xipeng Qiu. 2024a. Dictionary learning improves patch-free circuit discovery in mechanistic interpretability: A case study on othello-gpt. *Preprint*, arXiv:2402.12201.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024b. Llama scope: Extracting

millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

Zirui He, Haiyan Zhao, Yiran Qiao, Fan Yang, Ali Payani, Jing Ma, and Mengnan Du. 2025. Saif: A sparse autoencoder framework for interpreting and steering instruction following of language models. *Preprint*, arXiv:2502.11356.

Ruben Härle, Felix Friedrich, Manuel Brack, Stephan Wäldchen, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. 2025. Measuring and guiding monosemanticity. *Preprint*, arXiv:2506.19382.

Yi Jing, Zijun Yao, Hongzhu Guo, Lingxu Ran, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2025. LinguaLens: Towards interpreting linguistic mechanisms of large language models via sparse auto-encoder. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28232–28251, Suzhou, China. Association for Computational Linguistics.

Hao Kang, Tevin Wang, and Chenyan Xiong. 2025. Interpret and control dense retrieval with sparse latent features. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 700–709, Albuquerque, New Mexico. Association for Computational Linguistics.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. 2025. Are sparse autoencoders useful? a case study in sparse probing. *Preprint*, arXiv:2502.16681.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *Preprint*, arXiv:2503.09532.

Aly M. Kassem, Zhuan Shi, Negar Rostamzadeh, and Golnoosh Farnadi. 2025. REVIVING YOUR MNEME: Predicting the side effects of LLM unlearning and fine-tuning via sparse model diffing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32250–32263, Suzhou, China. Association for Computational Linguistics.

Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. 2024. Interpreting attention layer outputs with sparse autoencoders. *Preprint*, arXiv:2406.17759.

Kristian Kuznetsov, Laida Kushnareva, Anton Razzhigaev, Polina Druzhinina, Anastasia Voznyuk, Irina Piontkovskaya, Evgeny Burnaev, and Serguei Barannikov. 2025. Feature-level insights into artificial text detection with sparse autoencoders. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25727–25748, Vienna, Austria. Association for Computational Linguistics.

Aaron J. Li, Suraj Srinivas, Usha Bhalla, and Himabindu Lakkaraju. 2026. Evaluating adversarial robustness of concept representations in sparse autoencoders. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5940–5957, Rabat, Morocco. Association for Computational Linguistics.

George Ma, Zhongyuan Liang, Irene Y. Chen, and Somayeh Sojoudi. 2026a. Do sparse autoencoders identify reasoning features in language models? *Preprint*, arXiv:2601.05679.

George Ma, Samuel Pfrommer, and Somayeh Sojoudi. 2026b. Revising and falsifying sparse autoencoder feature explanations. *Advances in Neural Information Processing Systems*, 38:63175–63217.

Aleksandar Makelov, George Lange, and Neel Nanda. 2024. Towards principled evaluations of sparse autoencoders for interpretability and control. *Preprint*, arXiv:2405.08366.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Preprint*, arXiv:2403.19647.

Harry Mayne, Yushi Yang, and Adam Mahdi. 2024. Can sparse autoencoders be used to decompose and interpret steering vectors? *Preprint*, arXiv:2411.08790.

Jordan F. McCann. 2026. Descriptive collision in sparse autoencoder auto-interpretability: When one explanation describes many features. *Preprint*, arXiv:2605.12874.

Tommaso Mencattini, Francesco Montagna, and Francesco Locatello. 2026. The rate-distortion-polysemanticity tradeoff in saes. *arXiv preprint arXiv:2605.14694*.

Abhinav Menon, Manish Shrivastava, David Krueger, and Ekdeep Singh Lubana. 2024. Analyzing (in)abilities of saes via formal languages. *Preprint*, arXiv:2410.11767.

Fabio Mercorio, Filippo Pallucchini, Daniele Potertì, Antonio Serino, and Andrea Seveso. 2025. SFAL: Semantic-functional alignment scores for distributional evaluation of auto-interpretability in sparse autoencoders. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 576–583, Suzhou (China). Association for Computational Linguistics.

Marco Molinari, Victor Shao, Luca Imeneo, Mateusz Mikolajczak, Abhimanyu Pandey, Vladimir Tregubiak, and Sebastião Kuznetsov Ryder Torres Pereira. 2025. Interpretable company similarity with sparse autoencoders. In *Proceedings of the 63rd Annual*

Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1027–1038, Vienna, Austria. Association for Computational Linguistics.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. 2026. The quest for the right mediator: Surveying mechanistic interpretability for nlp through the lens of causal mediation analysis. *Computational Linguistics*, 52(1):331–378.

Aashiq Muhamed, Jacopo Bonato, Mona T Diab, and Virginia Smith. 2025a. Saes can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms. In *Second Conference on Language Modeling*.

Aashiq Muhamed, Mona T. Diab, and Virginia Smith. 2025b. Decoding dark matter: Specialized sparse autoencoders for interpreting rare concepts in foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1604–1635, Albuquerque, New Mexico. Association for Computational Linguistics.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. 2024. Steering language model refusal with sparse autoencoders. *Preprint*, arXiv:2411.11296.

Charles O'Neill and Thang Bui. 2024. Sparse autoencoders enable scalable and reliable circuit identification in language models. *Preprint*, arXiv:2405.12522.

Charles O'Neill, Christine Ye, Kartheik Iyer, and John F. Wu. 2024. Disentangling dense embeddings with sparse autoencoders. *Preprint*, arXiv:2408.00657.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. 2024. Automatically interpreting millions of features in large language models. *Preprint*, arXiv:2410.13928.

Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. 2025. Transcoders beat sparse autoencoders for interpretability. *Preprint*, arXiv:2501.18823.

Kenny Peng, Rajiv Movva, Jon Kleinberg, Emma Pierson, and Nikhil Garg. 2025. Use sparse autoencoders to discover unknown concepts, not to act on known concepts. *Preprint*, arXiv:2506.23845.

Dan Pluth, Zachary Nicholas Houghton, Yu Zhou, and Vijay K. Gurbani. 2026. Mechanistic interpretability of asr models using sparse autoencoders. *Preprint*, arXiv:2605.12225.

Lucia Quirke, Stepan Shabalin, and Nora Belrose. 2025. Binary sparse coding for interpretability. *Preprint*, arXiv:2509.25596.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. 2024a. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *Advances in Neural Information Processing Systems*.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024b. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *Preprint*, arXiv:2407.14435.

Shreyas N. Samaga, Gilberto Gonzalez Arroyo, and Tamal K. Dey. 2026. Halluzig: Hallucination detection using zigzag persistence. *Preprint*, arXiv:2601.01552.

Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. 2025. Route sparse autoencoder to interpret large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6801–6815, Suzhou, China. Association for Computational Linguistics.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. 2025. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1690–1712, Suzhou, China. Association for Computational Linguistics.

Samuel Soo, Chen Guang, Wesley Teng, Chandrasekaran Balaganesh, Tan Guoxian, and Yan Ming. 2025. Interpretable steering of large language models with feature guided activation additions. *Preprint*, arXiv:2501.09929.

Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Yang Song, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. 2024. Redeep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. *Preprint*, arXiv:2410.11414.

Bahareh Tolooshams, Ailsa Shen, and Anima Anandkumar. 2025. Mechanistic interpretability with sparse autoencoder neural operators. *Preprint*, arXiv:2509.03738.

Ariun-Erdene Tumurchuluun, Yusser Al Ghussin, David Mareček, Josef Van Genabith, and Koel Dutta Chowdhury. 2025. TenseLoC: Tense localization and control in a multilingual LLM. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 243–264, Suzhuo, China. Association for Computational Linguistics.

Xu Wang, Yan Hu, Benyou Wang, and Difan Zou. 2025a. Does higher interpretability imply better utility? a

pairwise analysis on sparse autoencoders. *Preprint*, arXiv:2510.03659.

Xu Wang, Bingqing Jiang, Yu Wan, Baosong Yang, Lingpeng Kong, and Difan Zou. 2026. Dlm-scope: Mechanistic interpretability of diffusion language models via sparse autoencoders. *Preprint*, arXiv:2602.05859.

Xu Wang, Zihao Li, Benyou Wang, Yan Hu, and Difan Zou. 2025b. Model unlearning via sparse autoencoder subspace guided projections. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26530–26546, Suzhou, China. Association for Computational Linguistics.

Chunlei Xin, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Xuanang Chen, Xinyan Guan, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2025. Sparse latents steer retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4547–4562, Vienna, Austria. Association for Computational Linguistics.

Wei Jie Yeo, Nirmalendu Prakash, Clement Neo, Ranjan Satapathy, Roy Ka-Wei Lee, and Erik Cambria. 2025. Understanding refusal in language models with sparse autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6377–6399, Suzhou, China. Association for Computational Linguistics.

Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. 2025. Interpreting CLIP with hierarchical sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

Hengyuan Zhang, Zhihao Zhang, Mingyang Wang, Zunhai Su, Yiwei Wang, Qianli Wang, Shuzhou Yuan, Ercong Nie, Xufeng Duan, Feijiang Han, Qibo Xue, Zeping Yu, Chenming Shang, Xiao Liang, Jing Xiong, Hui Shen, Chaofan Tao, Zhengwu Liu, Senjie Jin, and 10 others. 2026. Locate, steer, and improve: A practical survey of actionable mechanistic interpretability in large language models. *arXiv preprint arXiv:2601.14004*.

Shuyi Zhang, Wei Shi, Sihang Li, Jiayi Liao, Hengxing Cai, and Xiang Wang. 2025. Interpretable reward model via sparse autoencoder. *Preprint*, arXiv:2508.08746.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2).

Haiyan Zhao, Xuansheng Wu, Fan Yang, Bo Shen, Ninghao Liu, and Mengnan Du. 2026. Denoising concept vectors with sparse autoencoders for improved language model steering. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 797–808, Rabat, Morocco. Association for Computational Linguistics.

Runcong Zhao, Chengyu Cao, Qinglin Zhu, Xiucheng Ly, Shun Shao, Lin Gui, Ruifeng Xu, and Yulan He. 2025a. Sparse activation editing for reliable instruction following in narratives. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25817–25832, Suzhou, China. Association for Computational Linguistics.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025b. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5117–5136, Albuquerque, New Mexico. Association for Computational Linguistics.

A Scope and Survey Methodology

We define explicit search, screening, and coding procedures to make the survey reproducible and to separate core NLP-relevant SAE work from supporting background and unrelated uses of sparsity.

A.1 Search Strategy and Corpus Construction

We searched for work on SAEs, sparse dictionary learning, sparse coding, transcoders, and mechanistic interpretability in NLP. Since SAE-based interpretability for language models emerged mainly after 2023, our core survey period covers January 2023 to May 2026. Earlier work from 2018–2022 was considered only when it provided relevant background on sparse interpretable representations, dictionary learning, or foundational mechanistic interpretability.

We searched ACL Anthology, arXiv, OpenReview, ICML, NeurIPS, PMLR, COLM, ICLR, BlackboxNLP and related workshop proceedings, Google Scholar, and Google Scholar Labs. We also performed backward and forward citation chasing from anchor papers on SAEs, superposition, feature circuits, and language-model interpretability. Search terms combined SAE-related keywords such as *sparse autoencoder*, *SAE*, *dictionary learning*, *sparse coding*, *monosemanticity*, *superposition*, *feature circuit*, and *transcoder* with NLP terms such as *language model*, *LLM*, *NLP*, *multilingual*, *hallucination*, *RAG*, *refusal*, *unlearning*, *steering*, *dense retrieval*, *reasoning*, *ASR*, and *diffusion language model*.

The initial search produced approximately 200 potentially relevant records. After removing duplicates, non-NLP papers, and papers where "sparse" referred only to sparse attention, sparse decoding, sparse token masks, parameter sparsity, or efficiency-oriented sparsity, we retained 115 candidate papers for detailed coding.

Preprints, Duplicates, and Normalization. Because SAE research is fast moving, many papers first appear as arXiv preprints and later in conference or workshop proceedings. When a preprint and a later published version referred to the same work, we treated them as one paper and used the official venue version when available. We retained the arXiv version only when no final venue record could be verified. Duplicate and near-duplicate records were identified using normalized titles, arXiv identifiers, ACL Anthology identifiers, DOIs, source filenames, and manual title matching.

Inclusion and Exclusion Criteria. We define an *SAE paper* as a work that trains, evaluates, analyzes, critiques, or materially uses a sparse autoencoder, sparse dictionary learner, transcoder, or closely related sparse latent decomposition for interpreting model activations. We define an *NLP-relevant SAE paper* as an SAE paper that studies language models, NLP tasks, language representations, text or speech-language systems, or mechanistic claims directly related to language behavior.

The 115 candidate papers were screened into three groups: 77 core works, 28 supporting works, and 10 excluded works. A paper was coded as *core* if it developed an SAE or sparse dictionary method for language-model interpretability, evaluated SAE feature quality or utility, applied SAE features to an NLP task or mechanism, or challenged assumptions underlying SAE-based interpretability. A paper was coded as *supporting* if it provided relevant background, baselines, or contrastive evidence but did not itself make an SAE-based interpretability contribution. A paper was excluded when its use of sparsity was unrelated to SAE-based interpretability. Table 3 summarizes these decisions.

A.2 Coding Protocol

We used a hybrid coding protocol. Metadata fields such as title, venue, year, source, abstract, and publisher identifiers were extracted automatically where possible and then manually checked. Each paper was then manually assigned an inclusion decision, priority level, taxonomy pillar, topic, and survey role.

For the 77 core papers, we matched each record to its PDF and performed a full-text pass to verify the evidence types reported in the paper. The coded dimensions include SAE architecture or variant, base model, activation site, NLP task or phenomenon, reconstruction or fidelity metrics, sparsity metrics, semantic validation, held-out or contrastive examples, robustness testing, causal intervention, downstream task utility, non-SAE baselines, released code or SAE weights, dashboards, and main limitations. A field is marked as present only when explicit evidence is available in the paper; ambiguous cases are coded as partial. These dimensions are reported in the evidence matrix in Appendix H.

A.3 Literature Demographics

The screened corpus contains 115 candidate papers: 77 core works, 28 supporting works, and 10

Decision	Criterion	Included examples	Excluded or contextual examples
Core	Direct SAE, sparse dictionary, transcoder, evaluation, causal, or intervention contribution for LM/NLP interpretability.	Feature discovery, Gated/JumpReLU/Temporal/Binary SAEs, feature circuits, refusal steering, RAG hallucination mitigation.	–
Supporting	Important for framing, baselines, prior surveys, or adjacent interpretability, but not itself a core SAE study.	IOI circuits, linear representation theory, attribution, non-SAE hallucination detection, general mechanistic-interpretability surveys.	Cited for context, not counted as core evidence.
Excluded	Sparse mechanism unrelated to SAE interpretability or outside the NLP mechanistic-interpretability scope.	–	Sparse attention, sparse token masks, sparse video decoding, memory-efficient fine-tuning, unrelated agent/data attribution.

Table 3: Inclusion and exclusion criteria. The central distinction is whether the paper contributes evidence about SAE-based feature discovery, explanation, evaluation, causal mechanism, or intervention in NLP.

Year	Core papers	Share of core set
2023	2	2.6%
2024	12	15.6%
2025	44	57.1%
2026	19	24.7%

Table 4: Temporal distribution of the 77 core SAE papers.

excluded works. As shown in Table 4, the temporal distribution of the 77 core works is 2 papers in 2023, 12 in 2024, 44 in 2025, and 19 in 2026. This confirms that SAE-based NLP interpretability became a concentrated research area only after 2023, with the largest growth occurring in 2025.

B Foundational Assumptions and Stress Tests

Table 5 summarizes the assumptions required for SAE-based interpretability claims. It also lists failure cases and stress tests that NLP papers should report before making strong mechanistic claims.

C SAE Architecture and Training Trade-offs

Table 6 compares major SAE variants and the problems they are designed to address. The table highlights how architecture choices affect reconstruction, sparsity, interpretability, and NLP usefulness.

D Intervention Modalities for SAE-Based Control

Table 7 compares intervention strategies that use SAE features for control, editing, steering, or unlearning. It distinguishes local causal tests from stronger claims about robust task-level behavioral control.

E Failure Modes and Diagnostic Evaluations

Table 8 details common failure modes in SAE-based NLP claims. For each failure mode, it identifies why it matters for NLP and what evaluation can expose it.

F Evidence Rubric for SAE Feature Claims

Table 9 provides a complementary reporting rubric that maps SAE feature claims to six evidential axes: reconstruction, sparsity, semantic validity, robustness, causality, and task utility.

G Resources, Reproducibility, and Reporting Protocol

Table 10 gives a practical reporting checklist for SAE papers in NLP. The goal is to make future claims auditable, reproducible, and easier to compare across models and tasks.

Assumption	What it enables	Why it can fail	Recommended NLP test
Sparse feature structure	Supports SAEs as dictionaries that decompose LM activations into a small set of active latents.	A behavior may be distributed across many weak features, making sparse explanations incomplete.	Report feature-sparsity metrics, active-latent distributions, reconstruction quality, and whether the target behavior is concentrated in a small feature set (Deng et al., 2023; Cunningham et al., 2024).
Linear feature geometry	Allows latents, probes, and steering directions to be interpreted as meaningful representational axes.	Some concepts may be nonlinear, context-dependent, or represented by curved regions rather than directions.	Compare single-feature, linear-subspace, and nonlinear/manifold analyses; test whether one-direction interventions preserve the claimed concept (Engels et al., 2024; Bhalla et al., 2026).
One-dimensional concepts	Makes individual SAE latents interpretable as natural-language concepts.	Complex NLP phenomena such as reasoning, multilingual transfer, refusal, or discourse may require multiple interacting features.	Test whether ablating or amplifying one latent is sufficient, or whether a feature set or subspace is required (Engels et al., 2024; Bhalla et al., 2026).
Monosemanticity	Justifies assigning concise natural-language labels to SAE features.	A feature may look coherent on top activating examples while also responding to unrelated contexts.	Use held-out positives, hard negatives, descriptive-collision checks, and output-centric validation (Cunningham et al., 2024; Mencattini et al., 2026).
SAE basis advantage	Supports the claim that SAEs reveal structure hidden by the neuron basis.	Some circuits may already be sparse and interpretable in neurons or attention heads.	Compare SAE features against neuron, attention-head, probe, and steering-vector baselines (Arora et al., 2026; Peng et al., 2025).
Reconstruction–sparsity–interpretability alignment	Allows reconstruction loss and sparsity metrics to serve as partial evidence of SAE quality.	Better reconstruction or lower L_0 does not guarantee cleaner semantics or causal relevance.	Report reconstruction, sparsity, monosemanticity, semantic validation, and causal/task effects separately (Mencattini et al., 2026).

Table 5: Core assumptions behind SAE-based mechanistic interpretability and recommended stress tests for NLP papers.

Variant	Main motivation	Shrinkage mitigation	Dead latent handling	Sparsity control	Rate–distortion–polysemanticity relation	Primary NLP advantage	Main limitation
Vanilla SAE	Baseline sparse dictionary over activations.	Weak; L_1 can shrink activations.	Resampling or larger dictionaries.	L_1 penalty.	Establishes baseline point on the trade-off.	Simple feature discovery and dashboards.	Sensitive to hyperparameters and broad features.
Gated SAE	Separate feature selection from magnitude estimation.	Stronger than vanilla through gating.	Usually combined with resampling.	Gate regularization.	Improves fidelity at similar sparsity.	Better labels for residual/MLP features.	More moving parts and harder comparison.
JumpReLU SAE	Directly optimize sparse thresholded activations.	Reduces continuous shrinkage.	Thresholding can still leave inactive units.	Approximate L_0.	Pushes rate–distortion frontier.	Useful for high-fidelity causal patching.	Threshold choices affect feature stability.
Top-K / Batch-TopK	Fix active latents per example or batch.	Avoids penalty-induced shrinkage.	Batch allocation can revive rare units.	Explicit K budget.	Makes rate budget explicit.	Comparable sparsity across layers/tasks.	May suppress rare but important NLP features.
RouteSAE	Share information across layers/routes.	Indirect; depends on route objective.	Layer routing can improve coverage.	Route-specific constraints.	Trades local fidelity for multi-layer coverage.	Scales to full-model language analysis.	Route semantics can be harder to interpret.
Group-SAE	Efficient group/layer training.	Indirect.	Grouping reduces duplicate/dead units.	Group-level sparsity.	Improves compute side of trade-off.	Practical for large LLM surveys.	Group assumptions may blur feature identity.
Specialized SAE	Recover rare or safety-critical concepts.	Depends on base objective.	Focused data can activate rare latents.	Targeted training distribution.	Allocates capacity to rare concepts.	Harmful content, low-resource, domain concepts.	Less representative of general activations.
Temporal SAE	Capture sequential feature dynamics.	Temporal smoothing can stabilize activations.	Sequence coverage may reduce inactive units.	Sparse plus temporal constraints.	Adds temporal distortion objective.	Discourse, reasoning traces, multi-token decisions.	Harder to isolate token-local causes.
Binary SAE	Prevent information hiding in magnitudes.	Removes magnitude shrinkage by discretization.	Binary codes expose unused units.	Binary activations.	Trades fidelity for discrete monosemanticity.	Clearer feature presence/absence tests.	Lower reconstruction fidelity possible.
Matryoshka SAE	Learn nested feature granularity.	Indirect.	Coarse-to-fine levels can improve coverage.	Multi-level sparsity.	Makes granularity part of the trade-off.	Lexical, syntactic, semantic hierarchy.	Feature levels may be difficult to align.

Table 6: SAE architecture trade-off matrix. Architecture choice affects what kinds of downstream claims can be supported, including feature explanation, causal analysis, and task-level intervention.

Intervention	Target	Evidence required	NLP use case	Side effects	Main risk
Feature ablation	Necessity of a latent or feature set.	Target behavior decreases while unrelated behaviors remain stable; include negative controls and feature-set ablations (Makelov et al., 2024; O'Brien et al., 2024; Yeo et al., 2025).	Testing causal necessity of refusal, language-specific, retrieval, or reasoning features.	Broad degradation if the feature is polysemantic or reused across behaviors.	Mistaking correlated activation for a necessary mechanism.
Feature amplification	Sufficiency or controllability of a latent.	Target behavior increases under bounded scaling and holds across prompts, paraphrases, and domains (Xin et al., 2025; Zhao et al., 2025b; Soo et al., 2025).	Steering RAG reliance, multilingual behavior, instruction following, or refusal.	Oversteering, unnatural generations, capability loss, or safety bypass.	Brittle prompt-specific control rather than robust behavior change.
Sparse subspace projection	Removing or preserving a behavior-linked subspace.	Forget/retain metrics, side-effect suites, adversarial prompts, and non-SAE editing baselines (Wang et al., 2025b; Muhamed et al., 2025a).	Precision unlearning, safety editing, and harmful/private knowledge removal.	Capability loss, hidden compensation, or incomplete forgetting.	Removing directions correlated with, but not causal for, the target.
Feature-guided steering vectors	Steering directions built or denoised with SAE features.	Outperform simpler steering vectors and show causal, robust effects across prompts/tasks (Soo et al., 2025; Chalnev et al., 2024; Mayne et al., 2024; Zhao et al., 2026).	Reasoning control, instruction following, refusal calibration, concept steering.	Feature mixtures, prompt sensitivity, and unstable scaling.	Interpretability language without improved control.
SAE-guided unlearning	Dynamic guardrails or masks for unwanted knowledge/behavior.	Forgetting/retention metrics, jailbreak tests, sequential unlearning, and side-effect analysis (Muhamed et al., 2025a; Wang et al., 2025b; Farrell et al., 2024).	Safety, privacy, harmful-knowledge removal, and model editing.	Under-unlearning, over-unlearning, jailbreak transfer, or reduced helpfulness.	Misuse for targeted suppression, extraction, or unsafe manipulation.
Sparse feature circuits	Graphs of interacting SAE features.	Edge-level evidence, path validation, causal graph edits, and post-intervention behavior tests (Marks et al., 2024; O'Neill and Bui, 2024; He et al., 2024a).	Multi-step mechanisms in RAG, reasoning, refusal, factual recall, and instruction following.	Graph incompleteness, patching artifacts, and missed compensatory paths.	Overclaiming a partial circuit as the full mechanism.

Table 7: Intervention modalities for SAE-based control in NLP. The table separates local causal tests from stronger claims about robust task-level control.

Failure mode	Why it matters for NLP	How to detect it	Representative evidence
Overbroad feature descriptions	Labels such as "reasoning", "refusal", or "foreign language" can fit many unrelated contexts.	Test held-out positives, hard negatives, counterexamples, and output-centric effects.	Explanation revision and falsification (Ma et al., 2026b); output-centric descriptions (Gur-Arieh et al., 2025).
Descriptive collision	Multiple distinct features may receive the same natural-language explanation, weakening feature identity.	Perform feature-to-description retrieval, contrastive feature discrimination, and collision audits.	Descriptive-collision analysis (McCann, 2026); CE-Bench contrastive evaluation (Gulko et al., 2025).
Feature splitting	One linguistic or behavioral concept may be distributed across several latents rather than one feature.	Analyze coactivation clusters, feature sets, and ablations of groups rather than isolated latents.	Multi-feature and circuit-based analyses (Marks et al., 2024; O'Neill and Bui, 2024); multi-level SAE work (Bussmann et al., 2025).
Sparse probing limitations	For known concepts, SAE features may not outperform supervised probes or simpler representations.	Compare against probes, neurons, heads, and hidden-state baselines on matched data.	Sparse probing critique (Kantamneni et al., 2025); known-vs-unknown concept distinction (Peng et al., 2025).
Interpretability–utility mismatch	A feature can be easy to name but ineffective for steering, classification, retrieval, or unlearning.	Report downstream task metrics, steering baselines, and side-effect analysis.	Interpretability–utility analysis (Wang et al., 2025a); steering-vector decomposition limits (Mayne et al., 2024).
One-dimensional feature assumption	Reasoning, multilinguality, refusal, and discourse may require subspaces or interacting features.	Test whether one latent is sufficient, or whether a subspace, feature set, or circuit is required.	Non-one-dimensional features (Engels et al., 2024); concept manifolds (Bhalla et al., 2026).
Neuron-basis sparsity	Some mechanisms may already be sparse and interpretable in neurons or heads.	Compare SAE circuits with neuron/head-level circuits and ablation results.	Neuron-basis circuit evidence (Arora et al., 2026).
Metric trade-off	Better reconstruction or lower L_0 does not guarantee monosemanticity, faithfulness, or causal relevance.	Report reconstruction, sparsity, monosemanticity, semantic validation, and causal effects separately.	Rate–distortion–polysemanticity trade-off (Mencattini et al., 2026); principled evaluation (Makelov et al., 2024).
Poor robustness	Feature labels may fail under paraphrase, translation, perturbation, adversarial prompts, or domain shift.	Use paraphrase, multilingual, adversarial, and domain-shift test sets.	Adversarial robustness evaluation (Li et al., 2026); reasoning-feature falsification (Ma et al., 2026a).
Causal overclaiming	Correlated activations can be mistaken for mechanisms that are necessary or sufficient.	Use ablation, amplification, path patching, sparse feature circuits, and causal mediation tests.	Sparse feature circuits (Marks et al., 2024); scalable circuit identification (O'Neill and Bui, 2024); refusal interventions (O'Brien et al., 2024; Yeo et al., 2025).

Table 8: Failure modes for SAE-based NLP claims. The table links each failure mode to the kind of evaluation needed to distinguish a plausible feature description from a robust, causal, and task-relevant mechanism.

Evaluation axis	Question	Good evidence	Common failure mode
Reconstruction	Does the SAE preserve the studied activation distribution?	Reconstruction loss, explained variance, loss recovered, and downstream reconstruction checks (Gao et al., 2024; Rajamanoharan et al., 2024b,a; Karvonen et al., 2025).	A feature appears interpretable, but reconstruction is too poor to support causal or behavioral claims.
Sparsity	Is information concentrated in a small number of active latents?	L_0/L_1, activation frequency, dead-latent rate, Top-K or BatchTopK controls, and binary activation constraints (Deng et al., 2023; Gao et al., 2024; Quirke et al., 2025; Cho et al., 2025).	Dense information may be hidden in magnitudes, broad coactivation patterns, or many weakly active latents.
Semantic validity	Does the feature description predict where the feature activates?	Held-out examples, contrastive positives and negatives, human or LLM evaluation, and output-centric descriptions (Paulo et al., 2024; Gur-Arieh et al., 2025; Gulko et al., 2025; McCann, 2026).	Labels become overbroad, collide across features, or fit only top activating examples.
Robustness	Does the concept survive paraphrase, perturbation, translation, and domain shift?	Adversarial examples, paraphrase tests, counterfactual prompts, and cross-domain or cross-lingual validation (Li et al., 2026; Ma et al., 2026b,a).	The label works only for original examples and fails under small linguistic or distributional changes.
Causality	Does intervening on the feature change the target behavior specifically?	Ablation, amplification, path patching, sparse feature circuits, causal mediation, and counterfactual generation (Marks et al., 2024; O'Neill and Bui, 2024; Makelov et al., 2024; O'Brien et al., 2024; Yeo et al., 2025).	A feature is correlated with a behavior but is not necessary, sufficient, or specific to the mechanism.
Utility	Does the feature improve a real NLP task or intervention?	Steering, unlearning, retrieval control, hallucination mitigation, classification transfer, ASR, or diffusion-LM task metrics (Xin et al., 2025; Wang et al., 2025b; Muhamed et al., 2025a; Kang et al., 2025; Kuznetsov et al., 2025; Pluth et al., 2026; Wang et al., 2026).	Interpretability does not translate into useful control, or gains come with unacceptable side effects.

Table 9: Evidence rubric for SAE feature claims in NLP. The rubric complements Table 2 by specifying the evidence types that future SAE papers should report.

Item	Minimum report	Why it matters
Base model	Architecture, size, checkpoint, instruction tuning, tokenizer.	Features can change across model families and training stages.
Activation locus	Layer, sublayer, residual/MLP/attention output, token position.	Determines what computation the SAE can observe.
SAE configuration	Dictionary size, sparsity mechanism, loss, activation, normalization, dead-latent handling.	Needed to compare feature quality and reconstruction.
Training distribution	Corpus, token count, filtering, prompts, languages/domains.	Feature semantics are data-dependent.
Evaluation	Reconstruction, sparsity, semantic validation, robustness, causality, utility.	Prevents overclaiming from a single proxy metric.
Feature explanation	Prompting/annotation method, positive and negative examples, held-out tests.	Makes natural-language labels auditable.
Intervention protocol	Feature selection, scaling/ablation values, prompt sets, side-effect metrics.	Required for claims about control.
Baselines	Neuron/head/probe/steering-vector/comparable non-SAE methods.	Shows whether SAEs add value over simpler tools.
Artifacts	Code, SAE weights, feature dashboard, evaluation data, random seeds.	Enables reproduction and downstream reuse.
Limitations	Known failure modes and invalidated hypotheses.	Normalizes falsification as evidence.

Table 10: Reporting checklist for SAE papers in NLP.

H Full-Text Evidence Matrix for the 77 Core Papers

Table 11 codes the 77 core works using full-text keyword-assisted extraction followed by conservative manual interpretation of each paper's topic and abstract. A checkmark means that the evidence type was found in the extracted paper text; *partial* means that limited or indirect evidence was found; and a cross means that the evidence type was not found in the extract and should not be claimed without manual confirmation.

Table 11: Full-text evidence matrix for the 77 core papers. Recon. = reconstruction metrics; Sparse = sparsity metrics; Sem. = semantic validation; Held = held-out or contrastive examples; Robust = robustness testing; Causal = causal intervention; Down. = downstream task utility; Base. = non-SAE baselines; Art. = released code, weights, dashboards, or repositories.

Paper	Variant	Base/site	Task or phenomenon	Recon.	Sparse	Sem.	Held	Robust	Causal	Down.	Base.	Art.
(Cho and Hockenmaier, 2025a)	SAE / dictionary learning	model not clear; attention/embedding	Multilingual language-specific SAE features	*partial*	✓	✓	*partial*	✓	✓	✓	✓	✓
(Ghilardi et al., 2024)	SAE / dictionary learning	PYTHIA; encoder/decoder	Layer-wise transfer for efficient SAE training	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Molinari et al., 2025)	SAE over embeddings	LLAMA/PHI; embedding/encoder	Domain/company similarity with SAE embeddings	*partial*	✓	✓	✗	*partial*	✓	✓	✓	✓
(Xin et al., 2025)	SAE over embeddings	LLAMA/BERT; attention/encoder	RAG behavior latents and steering	✓	✓	✓	✓	*partial*	✓	✓	✓	✓
(Deng et al., 2025)	SAE / dictionary learning	LLAMA/QWEN; embedding/encoder	Language-specific features	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Gulko et al., 2025)	SAE / dictionary learning	model not clear; residual stream/mlp	Contrastive benchmark for SAE interpretability	✓	✓	✓	✓	✓	✓	*partial*	✓	✓
(Bondielli et al., 2025)	SAE / dictionary learning	PHI; residual stream/attention	Italian small-language-model features	*partial*	✓	✓	✗	*partial*	*partial*	✓	✗	✓
(Mercorio et al., 2025)	SAE / dictionary learning	model not clear; embedding/encoder	Semantic-functional alignment for auto-interpretability	✗	✓	✓	*partial*	✓	✓	✓	✓	✓
(Shi et al., 2025)	RouteSAE	LLAMA/PYTHIA; residual stream/encoder	RouteSAE multi-layer extraction	✓	✓	✓	✗	✓	✓	✓	✓	✓
(Ghilardi et al., 2025)	Group-SAE	LLAMA/PYTHIA; residual stream/mlp	Group-SAE layer grouping	✓	✓	✓	✓	*partial*	*partial*	✓	✓	✓
(Zhao et al., 2025a)	SAE / dictionary learning	model not clear; attention/encoder	Instruction-following activation editing	✗	✓	✓	✓	✓	✓	✓	✓	✓
(Wang et al., 2025b)	SAE / dictionary learning	model not clear; encoder/layer	Unlearning via SAE-guided subspace projection	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Jing et al., 2025)	SAE / dictionary learning	model not clear; attention/encoder	Linguistic mechanisms via SAEs	✗	✓	✓	✓	✓	✓	✓	✓	✓
(Cho and Hockenmaier, 2025b)	SAE / dictionary learning	GEMMA; embedding/encoder	SAE-guided steering for in-context learning	*partial*	✓	✓	✓	✓	✓	✓	✓	✓
(Gallifant et al., 2025)	SAE / dictionary learning	GEMMA; encoder/layer	Classification and transferability	*partial*	✓	✓	*partial*	✓	✓	✓	✓	✓
(Kassem et al., 2025)	SAE / dictionary learning	model not clear; site not clear	Sparse model diffing after unlearning/fine-tuning	✓	✓	✓	✗	✓	✓	✓	✓	✓
(Kuznetsov et al., 2025)	SAE / dictionary learning	GEMMA/LLAMA; residual stream/attention	Artificial text detection with SAE features	✗	✓	✓	✗	✓	✓	✓	*partial*	✓
(Yeo et al., 2025)	SAE / dictionary learning	model not clear; encoder	Refusal mechanisms with SAE latents	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Abdaljalil et al., 2025)	SAE / dictionary learning	GEMMA; encoder	SAFE hallucination mitigation/query enrichment	✗	✓	✓	✗	✓	✓	✓	✓	✓
(Muhamed et al., 2025b)	Specialized SAE	model not clear; mlp/encoder	Specialized SAEs for rare concepts	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Tumurchuluun et al., 2025)	SAE / dictionary learning	LLAMA; residual stream/attention	Tense localization and control	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Zhao et al., 2025b)	SAE / dictionary learning	LLAMA; mlp/encoder	Knowledge selection with SAE representation engineering	✓	✓	✓	✓	*partial*	✓	✓	✓	✓
(Kang et al., 2025)	SAE over embeddings	model not clear; embedding/encoder	Dense retrieval control with sparse latent features	✓	✓	✓	✓	*partial*	✓	✓	✓	✓
(Han et al., 2026)	SAE / dictionary learning	model not clear; attention/encoder	Agentic SAE feature explainer	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Li et al., 2026)	SAE / dictionary learning	model not clear; embedding/encoder	Adversarial robustness of SAE concepts	✓	✓	✓	*partial*	✓	✓	✓	✓	✓

Continued on next page

Table 11 – continued from previous page

Paper	Variant	Base/site	Task or phenomenon	Recon.	Sparse	Sem.	Held	Robust	Causal	Down.	Base.	Art.
(Zhao et al., 2026)	SAE / dictionary learning	PHI; encoder/decoder	SAE-denoised concept vectors	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Fereidouni et al., 2026)	SAE / dictionary learning	GEMMA/DEEPSEEK; encoder	Monosemantic representation evaluation	✓	✓	✓	✗	✓	✓	✓	✓	✓
(Deng et al., 2023)	SAE / dictionary learning	model not clear; mlp/embedding	Feature sparsity metrics	*partial*	✓	✓	*partial*	✓	✓	✓	✓	✓
(He et al., 2024a)	SAE feature circuits	GPT-2; residual stream/mlp	Dictionary learning for patch-free circuits	✓	✓	✓	✗	✓	✓	*partial*	✓	✓
(O'Neill and Bui, 2024)	SAE feature circuits	model not clear; attention/encoder	SAE circuit identification	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Engels et al., 2024)	SAE / dictionary learning	GPT-2/LLAMA; embedding/encoder	Multi-dimensional features	✓	✓	✓	✗	✓	✓	✓	✓	✓
(Dunefsky et al., 2024)	Transcoder baseline	GPT-2/PHI; mlp/attention	Transcoder feature circuits	✓	✓	✓	✗	✗	✓	✓	✓	✓
(Kissane et al., 2024)	SAE feature circuits	GPT-2/BERT; residual stream/mlp	Attention-output SAEs	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Rajamanoharan et al., 2024b)	JumpReLU SAE	GEMMA; residual stream/mlp	JumpReLU SAEs	✓	✓	✓	✗	*partial*	✓	✓	*partial*	✓
(O'Neill et al., 2024)	SAE over embeddings	PHI; embedding/encoder	Dense text embeddings with SAEs	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Farrell et al., 2024)	SAE / dictionary learning	GEMMA; residual stream/encoder	SAE unlearning	*partial*	✓	✓	✗	✓	✓	✓	✗	✓
(O'Brien et al., 2024)	SAE / dictionary learning	PHI; encoder/decoder	Refusal steering with SAEs	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(He et al., 2025)	SAE / dictionary learning	model not clear; residual stream/encoder	SAIF instruction following	*partial*	✓	✓	*partial*	✓	✓	✓	✓	✓
(Kantamneni et al., 2025)	SAE / dictionary learning	PHI; encoder	Sparse probing utility test	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Karvonen et al., 2025)	SAE / dictionary learning	model not clear; encoder/decoder	SAEBench	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Bail et al., 2025)	SAE / dictionary learning	model not clear; embedding/encoder	SAE concepts for text classification	✓	✓	✓	✓	*partial*	✓	✓	✓	✓
(Quirke et al., 2025)	Binary SAE	model not clear; mlp/attention	Binary sparse coding/autoencoders	✓	✓	✓	*partial*	*partial*	✓	✓	✓	✓
(Bhalla et al., 2025)	Temporal SAE	GEMMA/PHI; encoder	Temporal SAEs	✓	✓	✓	✓	✗	✓	✓	✓	✓
(Ma et al., 2026a)	SAE / dictionary learning	model not clear; mlp/encoder	Reasoning-feature falsification	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Galichin et al., 2025)	SAE / dictionary learning	DEEPSEEK; encoder	Reasoning features via SAEs	✓	✓	✓	✗	*partial*	✓	✓	*partial*	✓
(Zhang et al., 2025)	SAE / dictionary learning	model not clear; encoder/layer	Interpretable reward model via SAE	✓	✓	✓	✗	*partial*	✓	✓	✓	✓
(Engels, 2025)	SAE / dictionary learning	model not clear; encoder	SAE thesis	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Gao et al., 2024)	Top-K/BatchTopK	GPT-2; residual stream/encoder	Scaling and evaluating SAEs	✓	✓	✓	✗	*partial*	✓	✓	✓	✓
(Demircan et al., 2024)	Temporal SAE	LLAMA/BERT; residual stream/encoder	Temporal-difference learning with SAEs	✗	✓	✓	✗	✗	✓	✓	✓	✓
(Marks et al., 2024)	SAE feature circuits	model not clear; mlp/attention	Sparse feature circuits	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Makelov et al., 2024)	SAE / dictionary learning	GPT-2; attention/encoder	Principled SAE evaluation	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Wang et al., 2025a)	SAE / dictionary learning	GEMMA/QWEN; mlp/encoder	Interpretability-utility relation	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Cunningham et al., 2024)	SAE / dictionary learning	BERT; residual stream/mlp	Highly interpretable features in language models	✓	✓	✓	*partial*	*partial*	✓	✓	✓	✓
(Rajamanoharan et al., 2024a)	Gated SAE	GEMMA/PYTHIA; residual stream/mlp	Gated SAEs	✓	✓	✓	*partial*	*partial*	✓	✓	✓	✓
(Härle et al., 2025)	SAE / dictionary learning	model not clear; encoder/layer	Measuring/guiding monosemanticity	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Ma et al., 2026b)	SAE / dictionary learning	model not clear; residual stream/mlp	Explanation revision and falsification	*partial*	✓	✓	✗	*partial*	✓	✓	✓	✓
(Chen et al., 2026)	SAE / dictionary learning	GEMMA; residual stream/attention	HalluSAE	*partial*	✓	✓	✓	✓	✓	✓	✓	✓
(Muhamed et al., 2025a)	SAE / dictionary learning	model not clear; encoder	Dynamic SAE guardrails for unlearning	✓	✓	✓	✓	✓	✓	✓	✓	✓
(Soo et al., 2025)	SAE / dictionary learning	GEMMA; attention/encoder	Feature-guided activation additions	✗	✓	✓	✓	✓	✓	✓	✓	✓
(Mayne et al., 2024)	SAE / dictionary learning	model not clear; encoder/decoder	SAEs for steering-vector decomposition	✓	✓	✓	✓	✓	✓	✓	✓	✓

Continued on next page

Table 11 – continued from previous page

Paper	Variant	Base/site	Task or phenomenon	Recon.	Sparse	Sem.	Held	Robust	Causal	Down.	Base.	Art.
(Gur-Arieh et al., 2025)	SAE / dictionary learning	model not clear; embedding	Output-centric feature descriptions	*partial*	✓	✓	*partial*	*partial*	✓	✓	✓	✓
(Menon et al., 2024)	SAE / dictionary learning	model not clear; encoder	Formal-language SAE testbed	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Paulo et al., 2024)	SAE / dictionary learning	GPT-2; encoder/decoder	Automatic interpretation at scale	*partial*	✓	✓	*partial*	✓	✓	✓	✓	✓
(He et al., 2024b)	SAE / dictionary learning	LLAMA; residual stream/mlp	Llama Scope SAE resource	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Chalnev et al., 2024)	SAE / dictionary learning	model not clear; residual stream/encoder	Targeting SAE features for steering vectors	✓	✓	✓	✓	✓	✓	*partial*	✓	✓
(Paulo et al., 2025)	Transcoder baseline	model not clear; mlp/encoder	Transcoders vs SAEs	✓	✓	✓	✗	✗	✓	✓	✓	✓
(Peng et al., 2025)	SAE / dictionary learning	model not clear; embedding/encoder	Unknown-vs-known concept distinction	✓	✓	✓	✗	✗	✓	✓	✓	✓
(Tolooshams et al., 2025)	SAE / dictionary learning	BERT; mlp/encoder	SAE neural operators	✓	✓	✓	*partial*	✓	✓	*partial*	✓	✓
(Cho et al., 2025)	Binary SAE	model not clear; encoder/layer	Binary autoencoder	✓	✓	✓	✓	*partial*	✓	✓	✓	✓
(Fang et al., 2026)	SAE / dictionary learning	DEEPSEEK; encoder	Reasoning steering with SAEs	✓	✓	✓	✓	✓	✓	✓	✓	*partial*
(Arora et al., 2026)	SAE feature circuits	LLAMA; mlp/encoder	Neuron-basis sparsity challenge	✓	✓	✓	✓	*partial*	✓	✓	✓	✓
(Wang et al., 2026)	SAE / dictionary learning	model not clear; encoder/layer	Diffusion language model SAEs	✓	✓	✓	✓	✗	✓	✓	✓	✓
(Bhalla et al., 2026)	SAE / dictionary learning	LLAMA; encoder/layer	Concept manifolds	✓	✓	✓	*partial*	✓	✓	✓	✓	✓
(Pluth et al., 2026)	SAE / dictionary learning	model not clear; embedding/encoder	ASR model SAEs	✓	✓	✓	*partial*	✓	✓	✓	*partial*	✓
(McCann, 2026)	SAE / dictionary learning	GPT-2/GEMMA; encoder/layer	Descriptive collision	✗	✓	✓	*partial*	✗	✓	✓	✓	✓
(Mencattini et al., 2026)	SAE / dictionary learning	model not clear; encoder/decoder	Rate-distortion-polysemanticity tradeoff	✓	✓	✓	✗	✗	✓	✓	✓	✓
(Bussmann et al., 2025)	Matryoshka SAE	GEMMA; encoder/decoder	Matryoshka SAEs	✓	✓	✓	✗	*partial*	✓	✓	✓	✓