SELF-INTERPRETABLE CONCEPT REPRESENTATIONS: TRAINING LIGHTWEIGHT ADAPTERS ON VECTOR-LABEL PAIRS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-interpretation methods prompt language models to describe their own internal states, offering a path toward concept-based self-explanation, but remain unreliable due to hyperparameter sensitivity. We show that training lightweight adapters on learned concept representations, while keeping the LM entirely frozen, yields reliable self-interpretation across tasks and model families. A scalar affine adapter with just $d_{\text{model}} + 1$ parameters suffices: trained adapters generate sparse autoencoder concept labels that outperform the training labels themselves on generation scoring, a concept quality metric (71% vs 63% at 70B scale), identify topics with 94% recall@1 versus 1% for untrained baselines, and surface semantic concepts implicit in multi-hop reasoning, including bridge entities appearing in neither prompt nor response, without chain-of-thought. The learned bias vector alone accounts for 85% of improvement, and simpler adapters generalize better than more expressive alternatives. Controlling for model knowledge via prompted descriptions, we find self-interpretation gains outpace capability gains from 7B to 72B parameters. Our results demonstrate that faithful self-explanation of learned concepts improves with scale, without modifying the model being interpreted.

1 INTRODUCTION

Mechanistic interpretability research has produced extensive structured knowledge about neural network internals: labeled sparse autoencoder features representing learned concepts, annotated circuits, and contrastive activation vectors (Amodei, 2025). Yet this knowledge remains external to the models it describes. Recent work on self-interpretation (Chen et al., 2024; Ghandeharioun et al., 2024) prompts language models to describe their own activations in natural language, but these methods are inconveniently sensitive to the scale parameter (Kharlapenko et al., 2024). A central failure mode is not merely instability but confident, fluent descriptions completely ungrounded in the actual semantics of the activation vector, making errors difficult to detect.

One response is to fine-tune the language model itself (Li et al., 2025), but this modifies the very model being interpreted, raising questions about whether explanations reflect the original model's concept representations or artifacts of the fine-tuning process. We propose instead to freeze the model entirely and train only a lightweight adapter. We train on *existing concept representations*: vector-label pairs from sparse autoencoder features (Cunningham et al., 2023) and contrastive activation vectors (Zou et al., 2023). This reframes interpretability artifacts not as endpoints of analysis, but as supervision for teaching models to explain their own internal concept activations. Importantly, the model being interpreted remains entirely frozen; only the adapter parameters are trained.

This technique improves self-interpretation quality across diverse concept representation tasks: labeling SAE features (outperforming the training labels themselves, 71% vs 63% at 70B scale), identifying topics encoded in activation vectors (94% vs 1% recall@1), and decoding latent concepts during multi-hop reasoning. This last application, extracting "bridge entities" that appear in neither prompt nor response, demonstrates that trained adapters can surface implicit concept activations even when the model never verbalizes its reasoning. A scalar affine adapter with just $d_{\text{model}} + 1$ parameters suffices, and the approach benefits from scale: larger models provide more capable interpretation of their own concept representations.

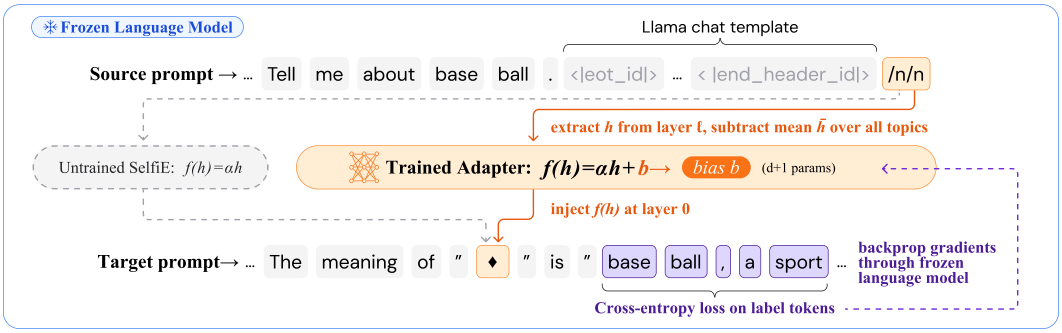


Figure 1: **Training self-interpretation from concept representations.** The training data consists of (h, y) pairs where vector h is a learned concept direction (SAE decoder vector or contrastive activation vector) and label y is a natural language description. A lightweight adapter transforms h and injects it at layer 0 at the placeholder position of an explanation-seeking target prompt. Cross-entropy loss on the label tokens trains only the $d_{\text{model}}+1$ adapter parameters; the language model remains frozen, yielding a self-explainable concept-based system.

2 Methods

2.1 Self-Interpretation via Patching

We adopt the Patchscopes framework (Ghandeharioun et al., 2024): an activation h is transformed via mapping function f and injected into an explanation-seeking target prompt at the token embedding layer. Following Kharlapenko et al. (2024), we use a template asking "What is the meaning of [placeholder]?" and inject the transformed activation at the placeholder position.

2.2 Trained Adapters

Untrained SelfIE uses $f(h) = \alpha \cdot h$, but optimal α varies across vectors with narrow valid ranges (Kharlapenko et al., 2024). We learn f from data, considering: **(1) Scalar affine**: $f(h) = \alpha \cdot h + b$ with $d+1$ parameters, which treats all directions identically via uniform scaling and recentering; **(2) Scalar affine + low-rank (SA+LR)**: $f(h) = \alpha \cdot h + UV^\top h + b$, adding limited direction-specific capacity; **(3) Full-rank affine**: $f(h) = Wh + b$ with $d^2 + d$ parameters.

2.3 Training Data: Concept Representations

We train on vector-label pairs (h, y) from two sources of concept representations: SAE decoder vectors paired with auto-interpretability labels, and contrastive activation vectors paired with synthetic topic descriptions. All input vectors are normalized to unit L2 norm; the adapter learns to map *directions* to descriptions. We minimize cross-entropy loss averaged over label tokens with the language model frozen.

3 Experiments

3.1 Setup

Models. We experiment primarily on Llama-3.1-8B-Instruct, with validation on Llama-3.3-70B-Instruct and Gemma-2-9B-IT. For scaling analysis, we use the Qwen-2.5-Instruct family (7B–72B).

Datasets. (1) *Goodfire SAE features*: 45k decoder vectors from a layer 19 SAE on Llama-3.1-8B-Instruct (Balsam et al., 2025) with auto-interpretability labels. (2) *Llama Scope SAE features*: decoder vectors from SAEs trained on Llama-3.1-8B (He et al., 2024) with Neuronpedia labels. (3) *Wikipedia contrastive vectors*: following Lindsey (2025), we compute activations for "Tell me about [topic]." prompts at layer 19 and subtract the mean across all topics to obtain contrastive vectors for 50k Wikipedia article titles, each paired with 15–20 synthetic descriptions.

Evaluation. For SAE latents, we report *generation scoring* (Juang et al., 2024): given a generated concept label, we prompt the model to generate text matching that description and measure how often the latent actually activates. This metric directly tests whether the model's behavior matches its concept label: if it claims a latent represents "code for event handlers," the latent should fire on generated event handler code. For contrastive vectors, we use *embedding-based retrieval*: embed generated labels using an off-the-shelf text embedding model and measure recall@k against all 50k topics.

3.2 Concept Identification from Contrastive Vectors

We train a full-rank adapter on Wikipedia contrastive vectors. At scale 1.0, the adapter achieves 82.9% recall@1 on held-out topics compared to 0.04% for untrained SelfIE. With best-of-6 sampling (Table 1), results improve to 93.7% recall@1.

Qualitative example. To illustrate that trained adapters extract semantic concepts rather than surface-level token information, we apply the method to a novel prompt (absent from the dataset): "Tell me about propagating gradients back through a neural network." At temperature 0.5, the adapter consistently produces descriptions like "backpropagation in neural networks" and "automatic differentiation for neural network backpropagation," correctly identifying the core concept despite the prompt never using the word "backpropagation."

3.3 Self-Interpretation Scales with Model Size

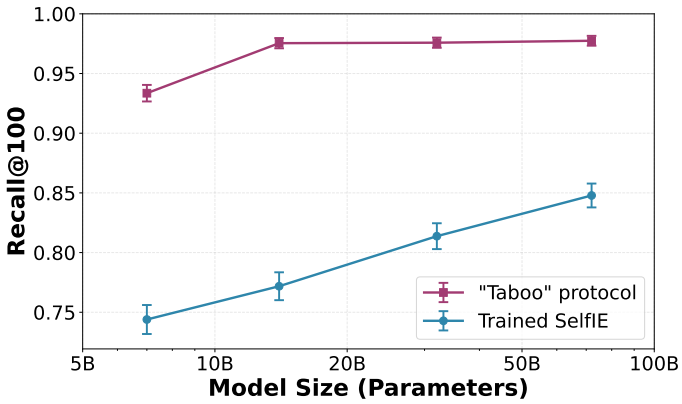


Figure 2: Scaling comparison on Qwen-2.5 models (7B to 72B). **Trained SelfIE** (below): recall@100 on held-out topics. **Taboo baseline** (above): the model describes each topic without naming it. The gap decreases with scale as SelfIE's performance increases more rapidly.

To isolate self-interpretation gains from general capability improvements, we compare against a "Taboo" baseline: the model describes each topic without mentioning its name, scored with the same retrieval. Figure 2 shows that trained SelfIE's performance grows faster than the Taboo baseline. The model's underlying topic knowledge saturates at intermediate scale, but self-interpretation continues to improve, showing no sign of plateauing at 72B. Notably, each adapter is trained on activations pooled across many layers yet successfully interprets any individual layer without layer information. The geometric-semantic correspondence of concept directions is preserved across the model's depth (see Appendix G).

3.4 SAE Concept Labeling

The bias vector is critical: adding it (scalar affine) yields 2.75 loss improvement over scale-only, accounting for ∼85% of total gains. Full-rank transformations overfit on SAE data but succeed on contrastive vectors. PCA analysis suggests this is due to intrinsic dimensionality differences: Wikipedia vectors concentrate variance in ∼200 dimensions while SAE features span the full space (Appendix F).

Table 1: Cross-dataset generalization. All numbers are best-of-6. "Hit rate" measures percent of generations with nonzero activation; "Coverage" measures percent of latents ever receiving accurate labels. Wikipedia columns show embedding retrieval recall. Subscripts show SEM.

	Llama Scope SAEs			Goodfire SAEs		Wikipedia Topics	
Method	Hit rate	Coverage	Det. F1	Hit rate	Coverage	R@1	R@100
Trained on Llama Scope SAE:							
SA+LR	$47.8_{\pm 0.8}$	$\mathbf{68.6}_{\pm 0.8}$	$0.814_{\pm 0.002}$	$46.5_{\pm 0.6}$	$76.1_{\pm 0.6}$	$0.8_{\pm 0.1}$	$13.4_{\pm 0.5}$
SA	$\mathbf{49.2}_{\pm 0.8}$	$\mathbf{67.8}_{\pm 0.8}$	$0.794_{\pm 0.002}$	$56.4_{\pm 0.6}$	$83.9_{\pm 0.5}$	$8.7_{\pm 0.4}$	$44.6_{\pm 0.7}$
Trained on Goodfire SAE:							
SA+LR	$44.5_{\pm 0.8}$	$62.5_{\pm 0.9}$	$0.772_{\pm 0.002}$	$59.2_{\pm 0.6}$	$87.4_{\pm 0.5}$	$2.8_{\pm 0.2}$	$19.9_{\pm 0.6}$
SA	$\mathbf{50.1}_{\pm 0.8}$	$67.1_{\pm 0.8}$	$0.774_{\pm 0.002}$	$\mathbf{62.8}_{\pm 0.6}$	$87.7_{\pm 0.5}$	$8.4_{\pm 0.4}$	$42.9_{\pm 0.7}$
Trained on Wikipedia topics:							
Full-rank	$23.4_{\pm 0.7}$	$36.1_{\pm 0.8}$	$0.512_{\pm 0.004}$	$34.7_{\pm 0.6}$	$60.7_{\pm 0.7}$	$\mathbf{93.7}_{\pm 0.3}$	$\mathbf{99.6}_{\pm 0.1}$
SA	$41.1_{\pm 0.8}$	$56.4_{\pm 0.9}$	$0.627_{\pm 0.003}$	$49.7_{\pm 0.6}$	$74.0_{\pm 0.7}$	$79.4_{\pm 0.6}$	$97.9_{\pm 0.2}$
Baselines:							
Untrained SelfIE	$31.3_{\pm 0.7}$	$48.8_{\pm 0.9}$	$0.684_{\pm 0.003}$	$40.4_{\pm 0.6}$	$69.1_{\pm 0.7}$	$1.3_{\pm 0.2}$	$17.2_{\pm 0.5}$
Auto-interp Labels $\times 6$	$41.9_{\pm 0.7}$	$65.5_{\pm 0.8}$	$\mathbf{0.821}_{\pm 0.002}$	$59.0_{\pm 0.6}$	$87.2_{\pm 0.5}$	—	—
Original + 5 Paraphrases	$44.7_{\pm 0.7}$	$\mathbf{67.8}_{\pm 0.8}$	$0.816_{\pm 0.002}$	$\mathbf{62.2}_{\pm 0.6}$	$\mathbf{89.5}_{\pm 0.5}$	—	—

Cross-dataset generalization. Table 1 shows generalization across datasets. Scalar affine adapters generalize better than higher-capacity adapters: the Wikipedia SA adapter achieves 41–50% hit rates on SAE latents despite never seeing SAE features during training, while Wikipedia full-rank achieves only 23–35%. This suggests that the bias vector learns a general "interpretation prior" that transfers across concept types, while direction-specific parameters overfit to the training distribution.

Table 2: SAE concept labeling on Llama-3.3-70B-Instruct. Trained adapters outperform both untrained SelfIE and the training labels themselves.

Method	Hit rate	Coverage
Trained Adapter (SA)	$\mathbf{71.4}_{\pm 1.3}$	$\mathbf{83.2}_{\pm 1.2}$
Trained Adapter (SA+LR)	$68.3_{\pm 1.3}$	$\mathbf{82.2}_{\pm 1.2}$
Original + 5 Paraphrases	$63.4_{\pm 1.4}$	$79.8_{\pm 1.3}$
Auto-interp Labels $\times 6$	$54.8_{\pm 1.4}$	$71.5_{\pm 1.4}$
Untrained SelfIE	$49.1_{\pm 1.5}$	$61.9_{\pm 1.5}$

Scaling to 70B. Table 2 shows results on Llama-3.3-70B-Instruct. The trained adapter achieves 71.4% hit rate, outperforming both untrained SelfIE (49.1%) and the training labels themselves (63.4% with paraphrases). This demonstrates that trained self-interpretation can surpass the supervision it was trained on.

Qualitative example. To illustrate this concretely: a held-out Goodfire SAE latent has the generic training label "The assistant should complete a code snippet" (shared by 51 other latents). Untrained SelfIE produces confident but erratic descriptions like "a small hill or mound of earth" (0% hit rate). The trained adapter consistently produces "Event listener and callback function definitions in code" with 98–100% hit rates, correctly characterizing when the latent fires despite the noisy training label.

3.5 Application: Decoding Latent Concept Activations

The preceding experiments use concept vectors that are approximately monosemantic by construction. But the broader promise of self-interpretation is reading out *arbitrary* internal states, including polysemantic residual stream activations. We test whether adapters trained on monosemantic concept vectors generalize to this more challenging setting.

Using multi-hop reasoning prompts from TwoHopFact (Yang et al., 2024), we examine whether trained adapters can extract *bridge entities*, implicit concepts that appear in neither prompt nor re-

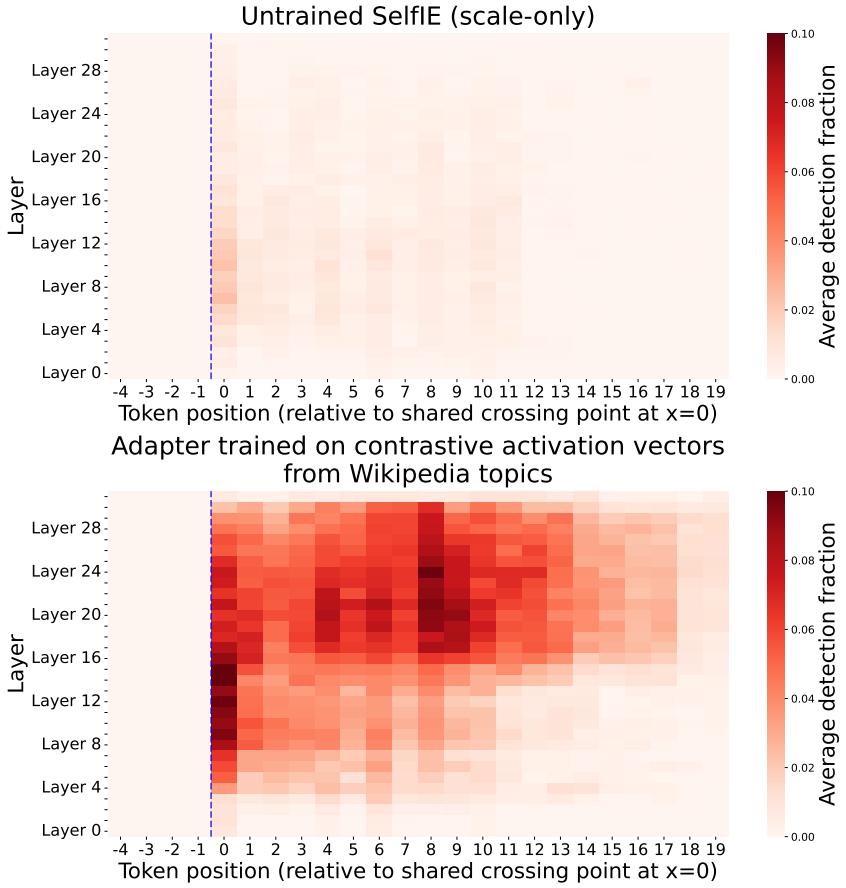


Figure 3: Bridge entity detection across layers and tokens. Each cell shows the fraction of SelfIE generations containing the bridge entity. **Top:** Untrained SelfIE shows weak signal. **Bottom:** Trained adapter produces stronger detection over broader range.

sponse. For example, given "The author of The Republic was born in the city of," Llama-3.1-8B correctly responds "Athens" with no mention of "Plato." Did the model shortcut directly to the answer, or did it construct a latent representation of "Plato" that was never verbalized?

Figure 3 shows that trained adapters substantially increase bridge entity detection rates compared to untrained SelfIE. Across 500 prompts, the trained adapter detected the bridge entity in 91.0% of cases versus 56.4% for untrained, a 4.8× reduction in undetected entities.

This capability could help decode "what the model is thinking" even when verbalized chain-of-thought is unavailable. Amodei (2025) calls this "catching models red-handed": if models can act on concept representations they never verbalize, surfacing such latent states is crucial for alignment auditing.

4 Related Work

Self-interpretation in language models. Chen et al. (2024) introduce SelfIE, prompting language models to describe their own activations by injecting them into an explanation-seeking prompt. Ghandeharioun et al. (2024) generalize this with Patchscopes, a unifying framework that casts many interpretability methods as activation patching, including Logit Lens (nostalgebraist, 2020) and Tuned Lens (Belrose et al., 2023). Tuned Lens also learns affine transformations of hidden states, but projects activations *out* of the model to vocabulary space for single-token prediction; our adapters project activations *back into* the model for open-ended generation. Kharlapenko et al. (2024) discovered that untrained self-interpretation is highly sensitive to the scale of injected vectors.

Learning to interpret activations. Pan et al. (2024) introduce LatentQA, fine-tuning via LoRA to answer questions about activations. Concurrent work extends this direction: Li et al. (2025) train models to explain feature descriptions, activation patching outcomes, and input ablation effects, finding that models explain themselves better than other models can; Karvonen et al. (2025) train on diverse tasks including self-supervised context prediction, achieving strong out-of-distribution generalization to auditing tasks like recovering secrets fine-tuned into models; Huang et al. (2025) train encoder-decoder architectures with sparse bottlenecks for concept-based interpretation.

All of these approaches modify the decoder model's weights (typically via LoRA). We instead freeze the model entirely and train only a lightweight affine transformation *before* injection. Li et al. (2025) find that models explain themselves better than other models can; freezing the model ensures this advantage is fully preserved. The two approaches (fine-tuning the model vs. training only an adapter) are complementary and could be combined in future work.

5 Discussion

What the adapter learns. The bias vector encodes a "default interpretation prior": applied to a zero vector, it generates generic descriptions matching the training distribution. Training on ALL-CAPS labels yields capitalized but semantically accurate outputs, confirming the bias captures format while the input vector contributes semantics (Appendix K).

Why adapters can surpass training labels. Auto-interpretability labels are noisy, generated once per feature. Our adapter learns a mapping across thousands of examples; if it captures genuine structure in how concept vectors relate to semantics, it can produce descriptions more accurate than individual noisy labels.

Toward concept-aware models. Our work provides infrastructure for concept-based explanation systems, surfacing what concepts models actually encode versus what external annotations claim. This enables detection of semantic misalignment, a central concern for trustworthy concept-based systems. The bridge entity experiments exemplify this: revealing concepts implicit in reasoning that models never verbalize.

Toward verifiable self-interpretation. Generation scoring makes self-interpretation *testable*: a model's claim about an internal feature can be checked against behavior. If a generated label says a latent represents "event handlers in code," we can verify this by generating code samples and checking whether the latent actually fires. This connection between self-report and verifiable behavior distinguishes our approach from methods that only produce plausible descriptions. Testable claims can also become training signal; future work might optimize adapters directly for generation scoring accuracy rather than training label likelihood.

6 Conclusion

We train lightweight adapters on concept representations to improve self-interpretation while keeping the model frozen. The bias vector accounts for most improvement ($\sim$85%); scalar affine adapters generalize best across datasets; full-rank overfits on high-dimensional SAE features but succeeds on lower-dimensional topic vectors.

The core insight is simple: mechanistic interpretability research has produced large quantities of labeled concept vectors that can serve as training data for teaching models to interpret their own representations. Rather than treating these artifacts as endpoints of analysis, we treat them as supervision for learning to explain internal states.

Concurrent work (Li et al., 2025; Karvonen et al., 2025; Huang et al., 2025) pursues a complementary approach: fine-tuning the LLM itself to answer questions about its activations. These methods can recover fine-tuned secrets, detect jailbreaks, and answer arbitrary queries, at the cost of modifying model weights. Our lightweight adapters offer a different trade-off: fewer parameters, a frozen base model, and precise characterization of what the transformation learns. The two approaches could be combined in future work.

REFERENCES

Dario Amodei. The urgency of interpretability. https://www.darioamodei.com/post/the-urgency-of-interpretability, April 2025. Accessed: 2025-01-16.

D. Balsam, T. McGrath, L. Gorton, N. Nguyen, M. Deng, and E. Ho. Announcing open-source SAEs for Llama 3.3 70B and Llama 3.1 8B. Goodfire Research, January 2025. Available: https://www.goodfire.ai/blog/sae-open-source-announcement.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. doi: 10.48550/arXiv.2303.08112. URL https://arxiv.org/abs/2303.08112.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. SelfIE: Self-interpretation of large language model embeddings. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 7373–7388. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24ao.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023. doi: 10.48550/arXiv.2309.08600. URL https://arxiv.org/abs/2309.08600.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 15466–15490. PMLR, 2024. URL https://proceedings.mlr.press/v235/ghandeharioun24a.html.

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. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders, 2024. URL https://arxiv.org/abs/2410.20526.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *CoRR*, abs/1904.09751, 2019. URL http://arxiv.org/abs/1904.09751.

Vincent Huang, Dami Choi, Daniel D. Johnson, Sarah Schwettmann, and Jacob Steinhardt. Predictive concept decoders: Training scalable end-to-end interpretability assistants. *arXiv preprint arXiv:2512.15712*, 2025. doi: 10.48550/arXiv.2512.15712.

Caden Juang, Gonçalo Paulo, Jacob Drori, and Nora Belrose. Open source automated interpretability for sparse autoencoder features. https://blog.eleuther.ai/autointerp/, July 2024. EleutherAI Blog.

Adam Karvonen, James Chua, Clément Dumas, Kit Fraser-Taliente, Subhash Kantamneni, Julian Minder, Euan Ong, Arnab Sen Sharma, Daniel Wen, Owain Evans, and Samuel Marks. Activation oracles: Training and evaluating LLMs as general-purpose activation explainers. *arXiv preprint arXiv:2512.15674*, 2025. URL https://arxiv.org/abs/2512.15674.

Dmitrii Kharlapenko, neverix, Neel Nanda, and Arthur Conmy. Self-explaining SAE features. https://www.alignmentforum.org/posts/8ev6coxChSWcxCDy8/self-explaining-sae-features, 2024.

Belinda Z. Li, Zifan Carl Guo, Vincent Huang, Jacob Steinhardt, and Jacob Andreas. Training language models to explain their own computations. *arXiv preprint arXiv:2511.08579*, 2025. doi: 10.48550/arXiv.2511.08579. URL https://arxiv.org/abs/2511.08579.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023. doi: 10.48550/arXiv.2308.03281. URL https://arxiv.org/abs/2308.03281.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024. doi: 10.48550/arXiv.2408.05147. URL https://arxiv.org/abs/2408.05147.

Jack Lindsey. Emergent introspective awareness in large language models. https://transformer-circuits.pub/2025/introspection/index.html, October 2025.

nostalgebraist. Interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020.

Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language. *arXiv preprint arXiv:2412.08686*, 2024. doi: 10.48550/arXiv.2412.08686. URL https://arxiv.org/abs/2412.08686.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024. URL https://arxiv.org/abs/2410.13928.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024. doi: 10.18653/v1/2024.acl-long.550. URL https://arxiv.org/abs/2402.16837. ACL 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023. doi: 10.48550/arXiv.2310.01405.

A Experimental Details

A.1 Dataset Details for SAE Datasets

We use SAE decoder rather than encoder vectors because decoder vectors represent feature directions in activation space directly, whereas encoder vectors are optimized for sparse reconstruction and may not preserve interpretable structure as faithfully.

Since the SelfIE template ends with an opening double quote (see Section A.3), we append a closing double quote and end-of-turn token to each label, teaching the model to produce a single complete description and then stop ("teaching" via the influence of the soft token on generation, since the language model parameters are all frozen).

A.2 Dataset Details for Wikipedia Topics Dataset

Wikipedia's "Vital Level 5" articles comprise approximately 50,000 topics deemed essential for a comprehensive encyclopedia. We compute contrastive activation vectors by running "Tell me about [topic]" prompts through the model, extracting residual stream activations at layer 19, and subtracting the mean activation across all topics.

Each topic is paired with 6–20 synthetic labels generated by Claude Sonnet 4.5, describing the topic at varying levels of specificity and from different angles. The following prompt was used:

```
You are helping create a dataset of conversational prompts about
    Wikipedia topics.

For each Wikipedia article title below, first decide on a single meaning
    the article is probably about, e.g. "Bit" is about binary bits not
    screwdriver bits. (It's actually okay if you get this wrong, as long
    as you're consistent!) Then provide:
```

```
1. A natural conversational prompt starting with "Tell me about" (e.g.,
    "Tell me about bits (binary digits)." or "Tell me about the Riemann
    hypothesis.")
2. Five varied labels for this topic with different levels of detail:
   - One medium with brief context (5-10 words)
   - One with a definition (10-20 words)
   - One with domain context (e.g., "Factorials in combinatorics")
   - At least one which doesn't begin with the Wikipedia article title
    itself (e.g. use an alternate name, or start with other words)
   - Your choice to fill out the rest of the five labels

**CRITICAL: Every label must UNIQUELY identify the topic!**
- BAD: "German poet and writer of the 19th century" (could be anyone!)
- GOOD: "Heinrich Heine" or "Heinrich Heine, German poet of the 19th
    century" or "the author of Die Lorelei"
- BAD: "Chinese fantasy novel depicting deification" (too generic!)
- GOOD: "Investiture of the Gods" or "Fengshen Yanyi" or "the Ming novel
    Investiture of the Gods"
- BAD: "the tale of how ancient heroes became divine" (way too vague!)
- GOOD: "Investiture of the Gods" or "the deification narrative in
    Fengshen Yanyi"

Either include the actual title/name, or use a specific alternate
    name/synonym, or add enough specificity to make it unambiguous.

Format your response as JSON:
{
  "original_title": "Factorial",
  "prompt": "Tell me about factorials.",
  "labels": [
    "factorial",
    "the factorial function in mathematics",
    "factorial, the product of all positive integers less than or equal
    to n",
    "factorials in combinatorics and probability",
    "the n! notation for consecutive integer products"
  ]
}

Be natural with grammar for the prompt:
- Use articles where needed ("the Riemann hypothesis")
- Pluralize general concepts ("theorems", "buildings")

For the labels:
- Start with lowercase unless the first word is a proper noun or always
    capitalized (e.g., "the founder of R.K. Films" not "The founder of
    R.K. Films")

Start IMMEDIATELY with the JSON response without any preceding text.

Wikipedia titles:
```

First, that prompt was used four separate times across the whole dataset to yield four different result batches each containing a prompt and five descriptions. The prompts were then deduplicated and, for the minority of topics for which multiple distinct prompts remained, the best topic prompt was chosen by asking Claude to decide with this prompt:

```
  You are helping to create a high-quality dataset for training language
    models. The dataset consists of conversational prompts about various
    topics along with multiple labels/descriptions for each topic.

  For some topics, we have generated multiple different prompts. Your
    task is to select or suggest the BEST prompt for each topic.
```

```
A good prompt should:
1. Be natural and conversational (not overly formal)
2. Be clear and unambiguous about what topic is being requested
3. Match the style: "Tell me about [topic]." or similar casual phrasing
4. Properly handle punctuation, capitalization, and formatting for the
   topic name
5. Be consistent with how the topic would naturally be discussed

Below is a topic with multiple prompt options and its labels. Please
   respond with ONLY the best prompt (or a better version if you can
   improve on the options). Do not include any explanation or other
   text.

Original Title: {original_title}

Prompt Options:
{prompt_options}

Labels:
{labels}

Best Prompt:
```

This resulted in a combined raw dataset of 50,005 topics each with a prompt and anywhere from 6–20 descriptions (20 if all four runs produced disjoint sets of descriptions, 6 in a case where the outputs were near-identical). The mean number of descriptions per topic was 16.9.

The vast majority of these topics were interpreted consistently by Claude and resulted in description sets all clearly describing the same topic, but there were some exceptions where Claude misinterpreted the topic in different conflicting ways. For example, "Left Ginza" as an article title refers to one of the two parts of the Ginza Rabba, the scripture of Mandaeism, but in some of its responses Claude hallucinated that Left Ginza was a sub-district of the Ginza district in Tokyo. (Note that if Claude misinterpreted a Wikipedia article title but always in the same consistent way, that's perfectly fine for our purposes, since we only want a diverse dataset of well-known topics. It only degrades the dataset quality if the descriptions are inconsistent.) To weed out such inconsistent records from the dataset, a final filtering step was used with this Claude prompt judging the consistency:

```
You are evaluating the quality of a dataset used for training language
    models.

Each dataset entry contains:
- An original Wikipedia article title
- A prompt asking about that topic
- Multiple labels/descriptions that should all refer to the same topic

Your task is to evaluate: **How certainly do all these labels refer to
    the same unique, real topic?**

Consider:
- Do all labels describe the same entity/concept/thing?
- Or do they describe different things that happen to have similar names?
- Are the labels internally consistent with each other?
- Would someone reading all these labels clearly understand what single
    topic is being referenced?

Respond with a JSON object containing:
1. "reasoning": A brief (1-3 sentences) explanation of your evaluation
2. "score": A number from 0-10 where:
    - 0 = Labels clearly refer to completely different topics
    (incoherent)
    - 5 = Ambiguous or mixed; some labels might refer to different topics
    - 10 = All labels clearly and unambiguously refer to the same unique
    topic (highly coherent)
```

```
Entry to evaluate:

Original Title: {original_title}

Prompt: {prompt}

Labels:
{labels}

Respond with ONLY valid JSON, no additional text:
```

The dataset was filtered to contain only records judged 9 or 10. This resulted in a final validated dataset containing 49,637 topics (99.3% retention), each with a natural "Tell me about [topic]." prompt and 6–20 varied descriptions that unambiguously refer to the same topic.

For retrieval evaluation, each topic is represented in the search index by a single document embedding. The document contains the article title followed by a bulleted list of all descriptions for that topic, providing diverse lexical anchors that make retrieval robust to variation in phrasing.

A.3 SELFIE PROMPT TEMPLATE

We use the following template for all self-interpretation experiments:

User message:

```
What is the meaning of "<|reserved_special_token_0|>"?
```

Assistant message (to be continued by autoregressive generation):

```
The meaning of "<|reserved_special_token_0|>" is "
```

Note that no system message is used. The Llama models we use handle this well but other models may have a stronger expectation that a system prompt be present, in which case this would have to be modified.

The usage of `<|reserved_special_token_0|>` as the placeholder was intended to prevent any possible leakage of semantic content from the value of the placeholder token into the description, but since we always inject soft tokens at the initial embedding layer this is actually not necessary (it would only come into play when injecting at a later layer because then some layers would still see the unpatched, original value of the token embedding for the placeholder token).

The transformed activation $f(h)$ is injected at the placeholder token position at layer 0 (the embedding layer). Generation continues until the model produces a closing quote and end-of-turn token.

A.4 TRAINING HYPERPARAMETERS

See the table for Llama-8B training hyperparameters. For the Llama-70B training on Goodfire the only differences were decreasing the batch size to 128, increasing the initial scale to 30.0, and running on 3× A100s.

A.5 SCALE GRID AND BEST-OF-N PROTOCOL

All input vectors are normalized to unit L2 norm during training. During generation, we apply scaling factors from the grid $\{0.1, 0.2, 0.3, 0.5, 0.8, 1.3, 2.1, 3.4, 5.5, 8.9, 14.4, 23.3\}$ (approximately geometric with ratio $\phi \approx 1.618$). For adapters that learn their own scale parameter α, the net effect is that scales multiply: the external scaling factor modulates the learned scale.

For evaluation, we select the best-performing set of $N = 6$ consecutive scales (for each combination of SelfIE method and dataset) via a calibration run on a small subset, then use those N scales to generate N candidate labels per vector. All metrics in Table 1 are reported as the average (over dataset samples) of the best value of the metric for any of the N label candidates. The baselines

Table 3: Training hyperparameters for Llama-8B adapter training on a single A100. Epoch count varied by dataset size: 1 epoch for large datasets (Wikipedia, ∼840k descriptions) and up to 5 epochs for smaller datasets.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	0.01
Batch size	256
Epochs	1–5 (dataset-dependent)
Weight decay	0.01
LR schedule	Cosine decay
Warmup steps	10
Gradient clip norm	0.5
Initial scale (α)	5.0
Random seed	42

(auto-interp labels repeated N times, and the original auto-interp label plus $N-1$ LLM paraphrases) are reported with the same best-of-N protocol for a fair comparison.

Figure 4 shows histograms of valid scales per latent on Llama-3.1-8B-Instruct. Trained adapters substantially increase the number of scales producing accurate outputs, partially mitigating the sensitivity issue identified by Kharlapenko et al. (2024).

A.6 Generation Scoring Protocol

Our implementation of generation scoring uses the following prompt to generate the synthetic contexts to be evaluated for SAE latent activation. Unlike the SelfIE prompt, this is an ordinary (hard) prompt with no soft tokens.

System message:

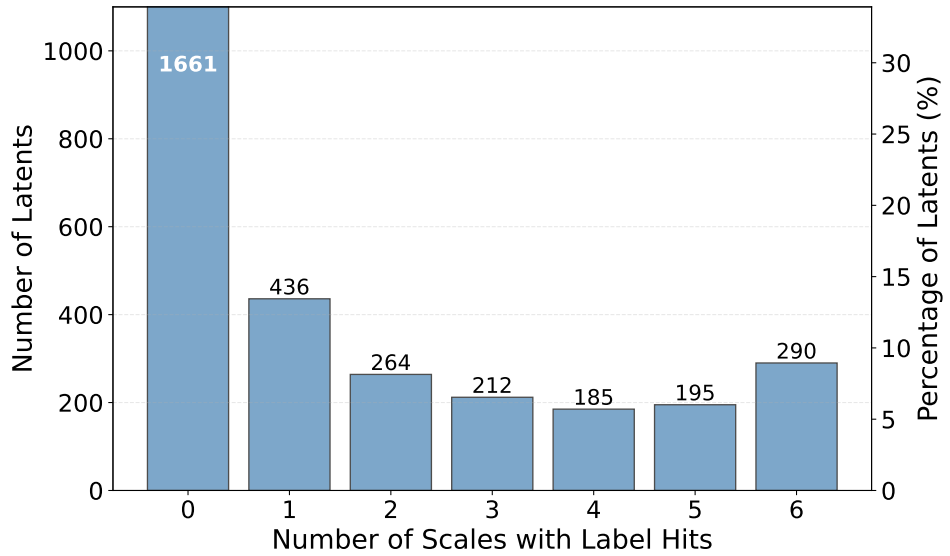
```
You are a helpful AI assistant who generates EXTREMELY SHORT example
    conversations. The conversations are between a user and an
    assistant, and have the following format:
[USER] I'm a user.
[ASSISTANT] I'm the assistant.
```

User message:

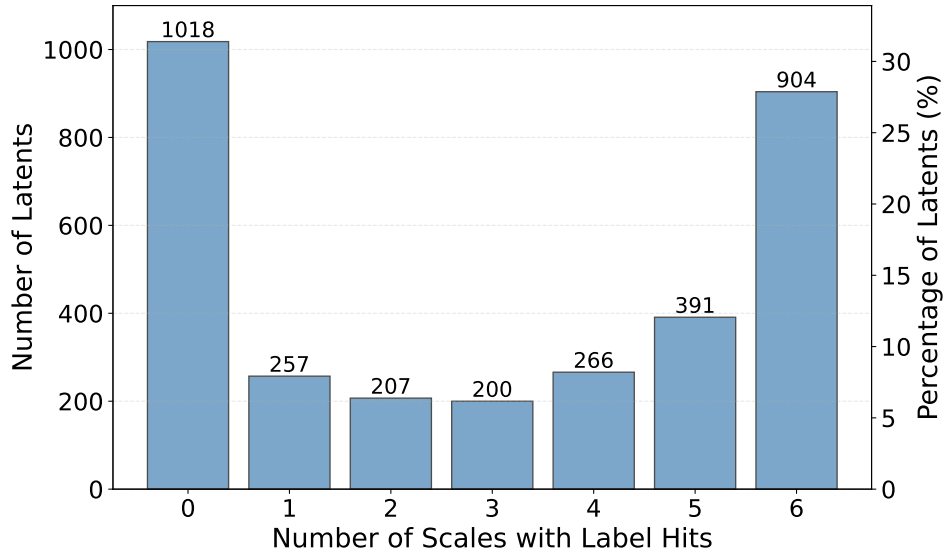
```
Produce a VERY SHORT conversation which exhibits '{{LABEL}}'
Do not include any other text in your response. Start immediately with
    the conversation.
```

The same experimental model then auto-regressively generates a quoted conversation using the [USER] and [ASSISTANT] tags. We use nucleus sampling (Holtzman et al., 2019) at temperature 0.7 with $p = 0.9$. After generation, the quoted conversation is algorithmically converted to a real conversation in the Llama chat template format. If any quoted conversations did not adhere to the expected format, they were interpreted by default as a single verbatim assistant message, but such parse errors occured on only 0.04% of all generations. Then a single forward pass is performed to get the SAE activations. The very first set of activations (for the `<|begin_of_text|>` token) is discarded, since the Llama Scope SAE was not trained on this token and there are many spurious activations.

The actual magnitudes of the nonzero activations can vary widely for different SAE latents, and also the latents have different activation patterns where some tend to activate on single tokens where others remain active for many tokens. Therefore our chosen metrics do not distinguish between different nonzero activation values, or even how many tokens within a generation had nonzero activations, but instead depend entirely on whether each overall generation is a 'hit' (some nonzero



(a) Scale sensitivity histogram for untrained SelfIE



(b) Scale sensitivity histogram for trained SelfIE (scalar affine + low rank, r=64, Llama Scope dataset)

Figure 4: Histograms showing the distribution of the number of scales (out of 6 scales attempted) at which each method produced accurate labels (where "accurate" is defined as eliciting at least one nonzero activation in 10 trials of generation scoring). The trained adapter is less sensitive to scale, with more latents receiving accurate labels at all 6 scales.

activation other than the <|begin_of_text|>) or a 'miss' (no nonzero activations except possibly <|begin_of_text|>). The two metrics we use are:

- Hit rate: Fraction of generations, for a single label, that had a 'hit' (nonzero activation)
- Coverage: Simply 1 if *any* generation for a label was a 'hit' and 0 otherwise

The mean (over all SAE latents evaluated) of the best-of-N values for each of these metrics is what's reported in Table 1.

A.7 Detection Scoring Protocol

We use the `delphi` library (Paulo et al., 2024) for detection scoring. We use Llama3.1-8B-Instruct itself as the judge model. Given a label, a classifier based on prompting that judge model with the label determines whether text snippets would activate the latent. The F1 score measures agreement with actual activations.

A.8 Computational Resources

All experiments were conducted on NVIDIA A100 80GB GPUs.

Training. Table 4 shows per-run training times. We estimate total training compute at approximately 50–80 GPU-hours across all adapter architectures and datasets explored in this work.

Table 4: Training time per adapter (A100 80GB).

Model size	Time	GPUs
7B–14B	4–6 h	1
32B	6–8 h	2
70B–72B	9–15 h	3

Evaluation. Inference compute was measured empirically on Llama-3.1-8B-Instruct (throughput of 1,521 tokens/second at batch size 512), then extrapolated to larger models assuming throughput roughly inversely proportional to parameter count. Total evaluation compute across all experiments was approximately 136 GPU-hours: generation scoring on SAE latents ($\sim$72 GPU-hours, dominated by the 70B evaluation at 59 GPU-hours), topic retrieval evaluations across the Qwen scaling series ($\sim$53 GPU-hours), and bridge entity detection ($\sim$12 GPU-hours for 3.2M SelfIE generations across 500 prompts $\times$ 32 layers $\times$ 20 token positions).

Total compute across all experiments was approximately 180–220 GPU-hours (training plus evaluation).

Dataset generation. The Wikipedia topic descriptions ($\sim$840k labels across 50k topics) were generated using Claude Sonnet, requiring approximately $300 in API costs.

B Adapter Architecture Comparison

Table 5 compares adapter architectures trained on Llama Scope SAE features. Key findings:

The bias vector is critical. Scale-only improves just 0.29 over identity, but adding the bias vector (scalar affine) yields an additional 2.75 improvement, approximately 85% of the total gain.

Low-rank additions provide meaningful gains. SA+LR (r=64) achieves the best validation loss of 1.62.

The identity structure matters. At rank 4, adding the scalar affine base improves validation loss from 2.00 to 1.69.

Full-rank overfits on SAE data. Full-rank achieves train loss of 0.64 but similar validation loss to simpler models.

C Qualitative Example

Table 6 illustrates how trained adapters can outperform both untrained SelfIE and the training labels themselves. This held-out Goodfire latent has a uselessly generic training label. Untrained SelfIE produces fluent but erratic descriptions; the trained adapter consistently produces accurate descriptions with 98–100% hit rates, confirming they faithfully characterize when the latent fires.

Table 5: Adapter architecture comparison on Llama Scope SAEs (layer 19, 32k and 131k widths combined). SA = scalar affine, LR = low-rank. Parameter counts shown for Llama-3.1-8B (d=4096). The bias vector accounts for the majority of improvement; adding low-rank capacity yields further gains while full-rank dramatically overfits.

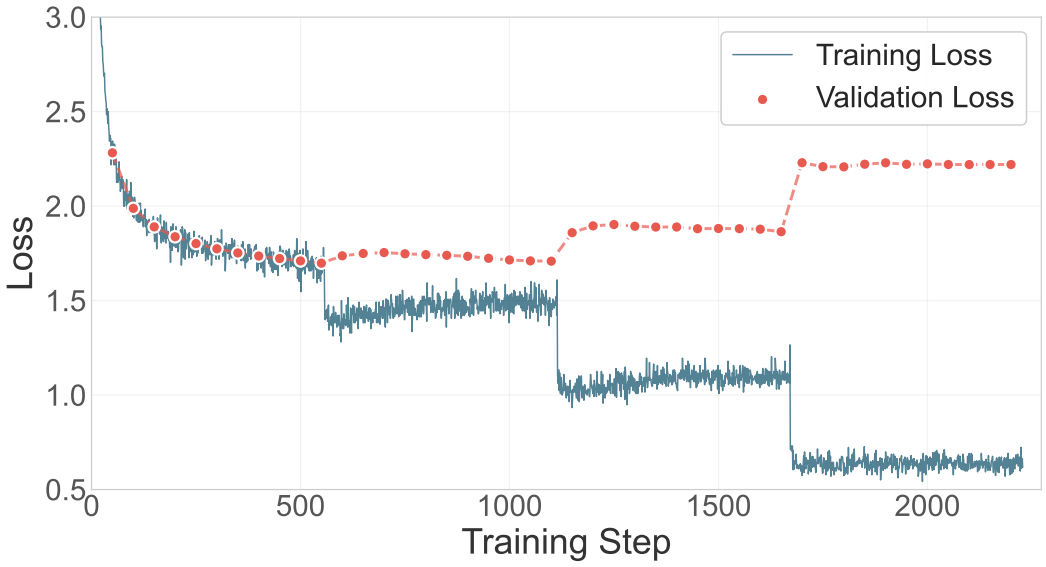
ARCHITECTURE	PARAMS	VAL LOSS	Δ
IDENTITY	0	4.834	—
SCALE-ONLY	1	4.543	-0.291
SCALAR AFFINE	4097	1.787	-3.047
SA + LR (R=4)	37K	1.694	-3.140
SA + LR (R=16)	135K	1.637	-3.197
SA + LR (R=64)	528K	**1.619**	**-3.215**
SA + LR (R=256)	2.1M	1.622	-3.212
LR ONLY (R=4)	37K	2.002	-2.832
LR ONLY (R=16)	135K	1.766	-3.068
LR ONLY (R=64)	528K	1.646	-3.188
LR ONLY (R=256)	2.1M	1.642	-3.192
FULL-RANK AFFINE (TRAIN LOSS)	16.8M	1.743 (0.64)	-3.091

Table 6: Greedy-decoded labels for held-out Goodfire SAE latent #41101. The training label is generic, shared by 51 latents. Untrained SelfIE produces confident but ungrounded descriptions that vary erratically with scale; the trained adapter produces accurate, consistent descriptions. **Score**: generation scoring hit rate.

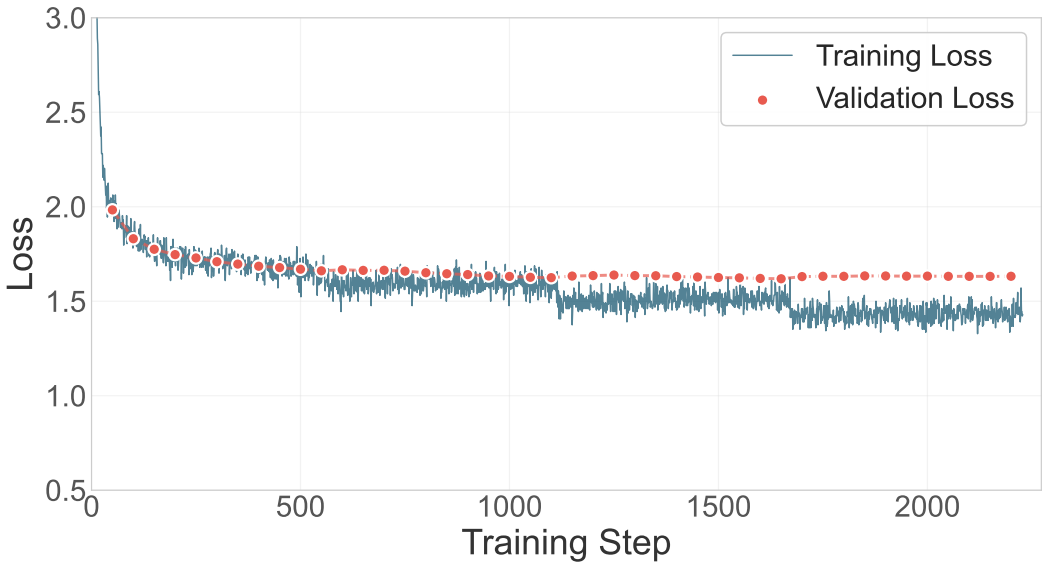
Label	**Score**
Training label (auto-interpretability):	
"The assistant should complete a code snippet"	2%
Untrained SelfIE:	
[scale 1.0] "a small hill or mound of earth" or "a small hill or ridge of land"...	0%
[scale 2.0] "a device or system that detects and reports a specific event or condition..."	17%
[scale 3.0] "a formal meeting or gathering, especially one for a special purpose"...	3%
Trained adapter (SA + LR, r=16):	
[scale 1.0] "Event listener and callback function definitions in code"	100%
[scale 2.0] "The code that defines a new event handler or callback function"	98%
[scale 3.0] "The code that defines the event handling function, typically in a GUI or event-driven application"	100%

D TRAINING DYNAMICS

Figure 5 shows training dynamics for different adapter architectures on Llama Scope SAE data. Full-rank adapters begin overfitting partway through the first epoch (although this overfitting is not yet visible since both validation samples and train samples are never-before-seen by the model, so their average losses are equal). After the first epoch this gap is clearly visible, but even at the end of the first epoch it manifests in the full-rank adapter having higher loss than the SA+LR adapter (validation loss 1.691 vs 1.661, with the train loss closely tracking it). The SA+LR adapter also has a train-loss gap that increases in successive epochs, but in this case the overfitting is not catastrophic and at the end of the *third* epoch this adapter has the lowest validation loss ever seen in the training of any run in Table 5.



(a) Training Progress (Full-Rank)

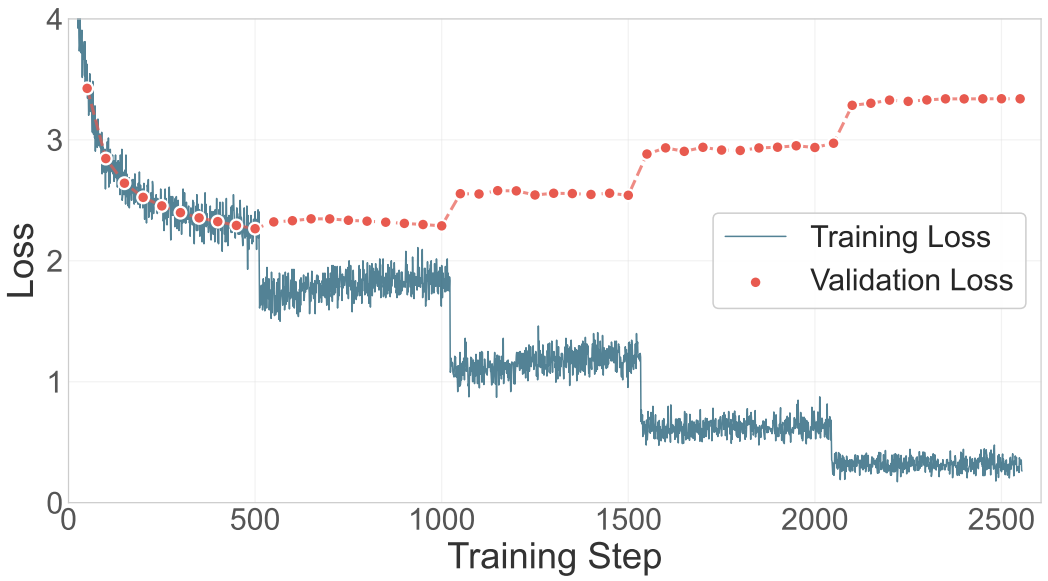


(b) Training Progress (SA+LR)

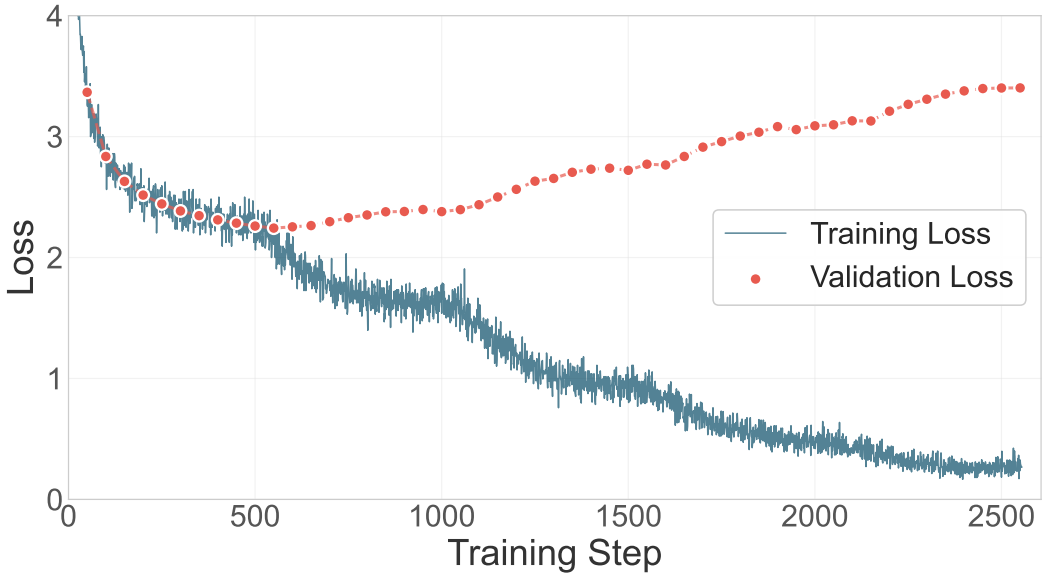
Figure 5: Validation loss curves during training on Llama Scope SAE features. Full-rank adapters (a) achieve lower training loss but higher validation loss than scalar affine + low-rank adapters (b), demonstrating overfitting.

D.1 Effect of Data Shuffling

We compared training with data reshuffled each epoch versus maintaining the same order throughout. Figure 6 shows that fixed ordering produces smoother loss curves while reshuffling creates visible stair-step patterns at epoch boundaries. However, both conditions exhibit similar final overfitting behavior for full-rank adapters, confirming that overfitting stems from excessive model capacity rather than memorization of presentation order.

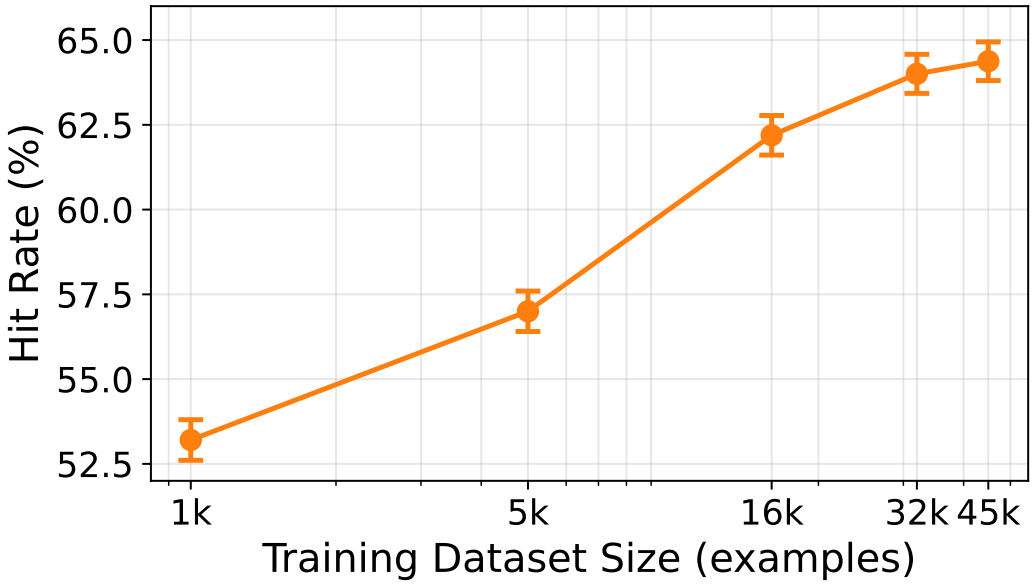


(a) Training Progress (Full-Rank, Shuffled)

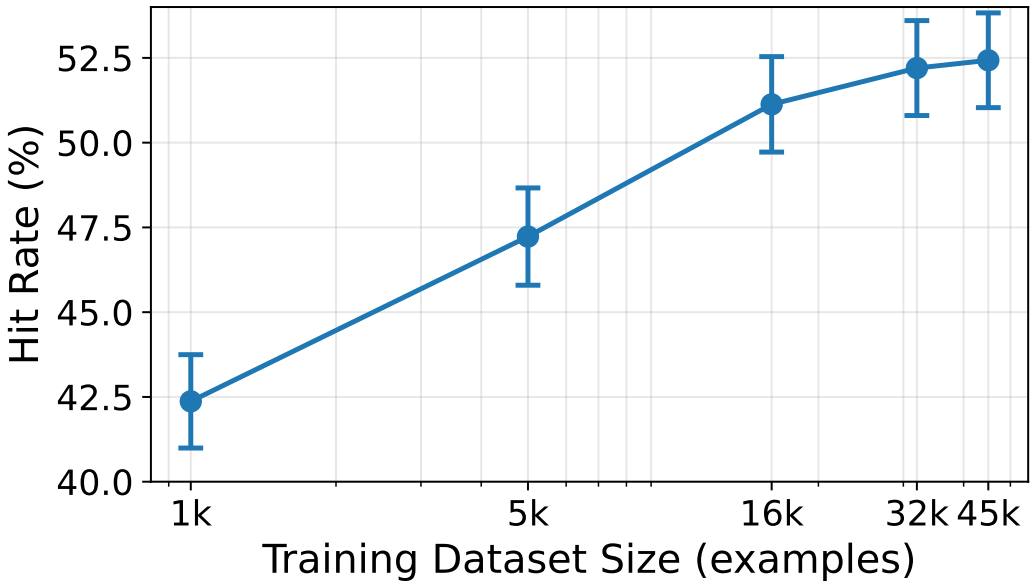


(b) Training Progress (Full-Rank, Unshuffled)

Figure 6: Validation loss curves for full-rank adapters with different shuffling strategies. Reshuffling each epoch (a) produces stair-step patterns as the model encounters some fraction of very-recently-seen samples at the beginning of each epoch; using the same order each epoch (b) produces smooth curves. Both overfit similarly, confirming that overfitting is due to model capacity rather than ordering effects.



(a) Goodfire (In-Distribution)



(b) Llama Scope (Cross-Dataset)

Figure 7: Performance of scalar affine adapters trained on varying fractions of the Goodfire-8B SAE training set. In-distribution evaluation (a) and cross-dataset evaluation (b) both show performance improving with more training data, with diminishing returns beyond ∼20k–30k labeled vectors.

E Training Data Requirements

Figure 7 shows that performance improves steadily with training data but exhibits diminishing returns. Notably, the cross-dataset generalization curve tracks the in-distribution curve closely, indicating that additional training data improves genuine self-interpretation capability rather than overfitting to dataset-specific patterns.

F CONTRASTIVE TOPIC VECTORS ARCHITECTURE COMPARISON

Table 7: Architecture comparison on Wikipedia contrastive vectors (Llama-3.1-8B). Unlike SAE features, full-rank adapters achieve the best performance without overfitting.

ARCHITECTURE	PARAMS	VAL LOSS	Δ
IDENTITY	0	3.858	—
SCALE-ONLY	1	3.523	-0.335
SCALAR AFFINE	4097	1.366	-2.492
SA + LR (R=4)	37K	1.293	-2.565
SA + LR (R=16)	135K	1.247	-2.611
SA + LR (R=64)	528K	1.205	-2.653
SA + LR (R=256)	2.1M	1.195	-2.663
LR ONLY (R=4)	37K	1.811	-2.047
LR ONLY (R=16)	135K	1.415	-2.443
LR ONLY (R=64)	528K	1.247	-2.611
LR ONLY (R=256)	2.1M	1.217	-2.641
FULL-RANK AFFINE	16.8M	**1.160**	**-2.698**

On Wikipedia contrastive vectors, full-rank adapters achieve the best performance (Table 7). This contrasts sharply with SAE features, where full-rank overfits catastrophically. Why does full-rank succeed on Wikipedia but overfit on SAEs? We hypothesized two possible explanations: (1) intrinsic dimensionality: Wikipedia vectors have lower intrinsic dimensionality, providing implicit regularization, or (2) Wikipedia has more labels per vector (∼17 vs 1), providing more supervision.

F.1 PCA ANALYSIS

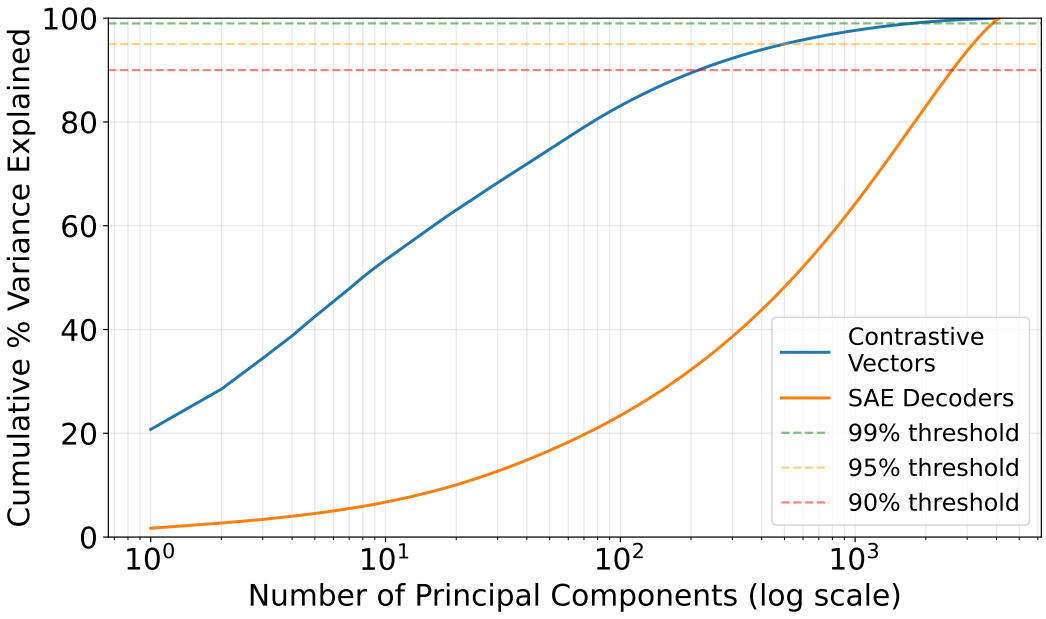


Figure 8: Cumulative variance explained by principal components. Wikipedia contrastive vectors (blue) concentrate >90% of variance in approximately 200 dimensions, while SAE decoder vectors (orange) span nearly the full 4096-dimensional space.

Figure 8 shows PCA analysis of both datasets. Wikipedia vectors are effectively low-dimensional, while SAE features utilize the full embedding space.

F.2 Controlled Experiments

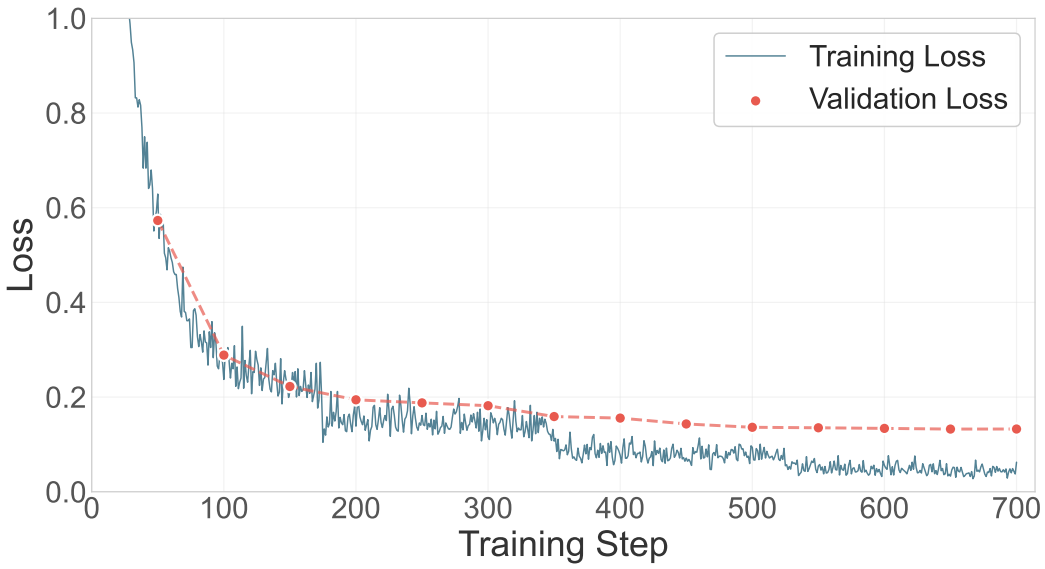


Figure 9: Loss curves for training a full-rank adapter on Wikipedia topic contrastive vectors with only a single text label per vector (the article title). A train-val gap does appear; nevertheless the validation loss continues to decrease over the whole training run.

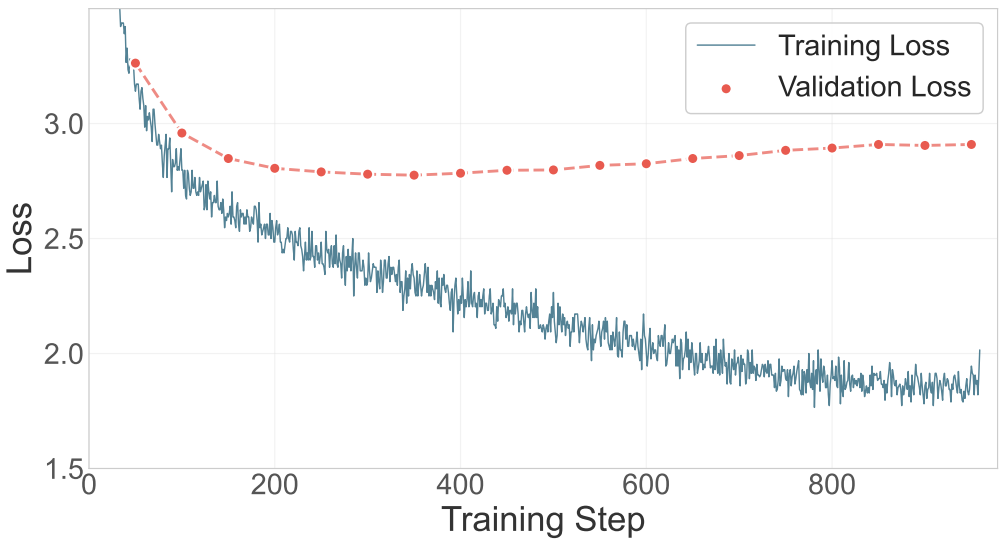


Figure 10: Loss curves for training a full-rank adapter on Goodfire SAE latents with 6 text labels per vector (the original Goodfire auto-interp label + 5 LLM paraphrases). The validation loss reaches a minimum about 36% of the way through the first epoch and then rises.

To distinguish intrinsic dimensionality from label count, we conducted two controlled experiments:

Wikipedia with 1 label per vector. We trained full-rank adapters on Wikipedia using only a single label per vector (matching SAE's label count). Validation loss continued decreasing throughout training with no sign of overfitting (Figure 9), ruling out label count as the explanation.

SAE with 6 labels per vector. We augmented Goodfire SAE latents with 5 LLM-generated paraphrases per latent, yielding 6 total labels per vector. Full-rank adapters still overfitted, with validation loss increasing partway through epoch 1 (Figure 10).

These experiments appear to rule out label count as the reason why full-rank adapters overfit on SAE latents but not on contrastive vectors, leaving intrinsic dimensionality as the most likely explanation. We conjecture that when the training dataset consists of some tens of thousands of vectors that span a large fraction of the dimensionality of the space, it's possible for a full-rank adapter to learn what's essentially a lookup table, promoting memorization, while if a similar-sized dataset has an intrinsic dimensionality in the low hundreds, an affine map is limited to learning functions on this much smaller subspace. In this case the *effective* parameter count is reduced from 16M to $<$1M promoting generalizable interpretation of the topic vectors.

G Cross-Layer Generalization

Table 8: Comparison of layer-specific vs cross-layer training, evaluated on Llama Scope SAE features. The cross-layer adapter actually outperforms the 32 layer-specific adapters on the hit rate metric, to a small but statistically significant extent (paired t-test, $p = 0.0002$).

Training data	Hit rate	Coverage
Untrained SelfIE	$41.0\%_{\pm 0.2}$	$70.2\%_{\pm 0.3}$
Layer-specific (eval on same layer)	$64.7\%_{\pm 0.2}$	$89.2\%_{\pm 0.2}$
Cross-layer (layers 0–31 combined)	$65.7\%_{\pm 0.2}$	$89.4\%_{\pm 0.2}$

We trained scalar affine adapters on Llama Scope 32k SAEs either independently at each layer or on vector-label pairs pooled across all 32 layers. Table 8 and Figure 11 show that a single cross-layer adapter matches and even slightly outperforms 32 layer-specific adapters. This was surprising to us and indicates that the geometric-semantic correspondence of the residual stream is preserved remarkably well over different layers: a SelfIE adapter doesn't need to know which layer an activation comes from in order to interpret it successfully. Another explanation consistent with this finding is that the bias vector mostly acts as a layer-agnostic soft token that steers the language model generation in directions that are generally useful for this interpretation task, and not as a layer-specific distributional shift.

The Qwen-2.5 scaling experiment (Appendix H) provides an even more dramatic demonstration of this cross-layer generalization phenomenon. In that experiment, full-rank affine adapters were trained on contrastive activation vectors pooled across all layers in the middle half of models ranging from 7B to 72B parameters. These adapters successfully interpret activations from any layer in their training set without knowing which layer the activation came from, definitively confirming that this remarkable generalization extends beyond the minimal scalar affine architecture to adapters with millions of parameters.

H Scaling Experiment Details

This appendix provides additional details and metrics for the Qwen-2.5 scaling experiment.

H.1 Experimental Setup

For each Qwen-2.5 model (7B, 14B, 32B, 72B), we computed contrastive activation vectors for the Wikipedia topics dataset at each layer in the middle half of the model. Vectors from all these layers

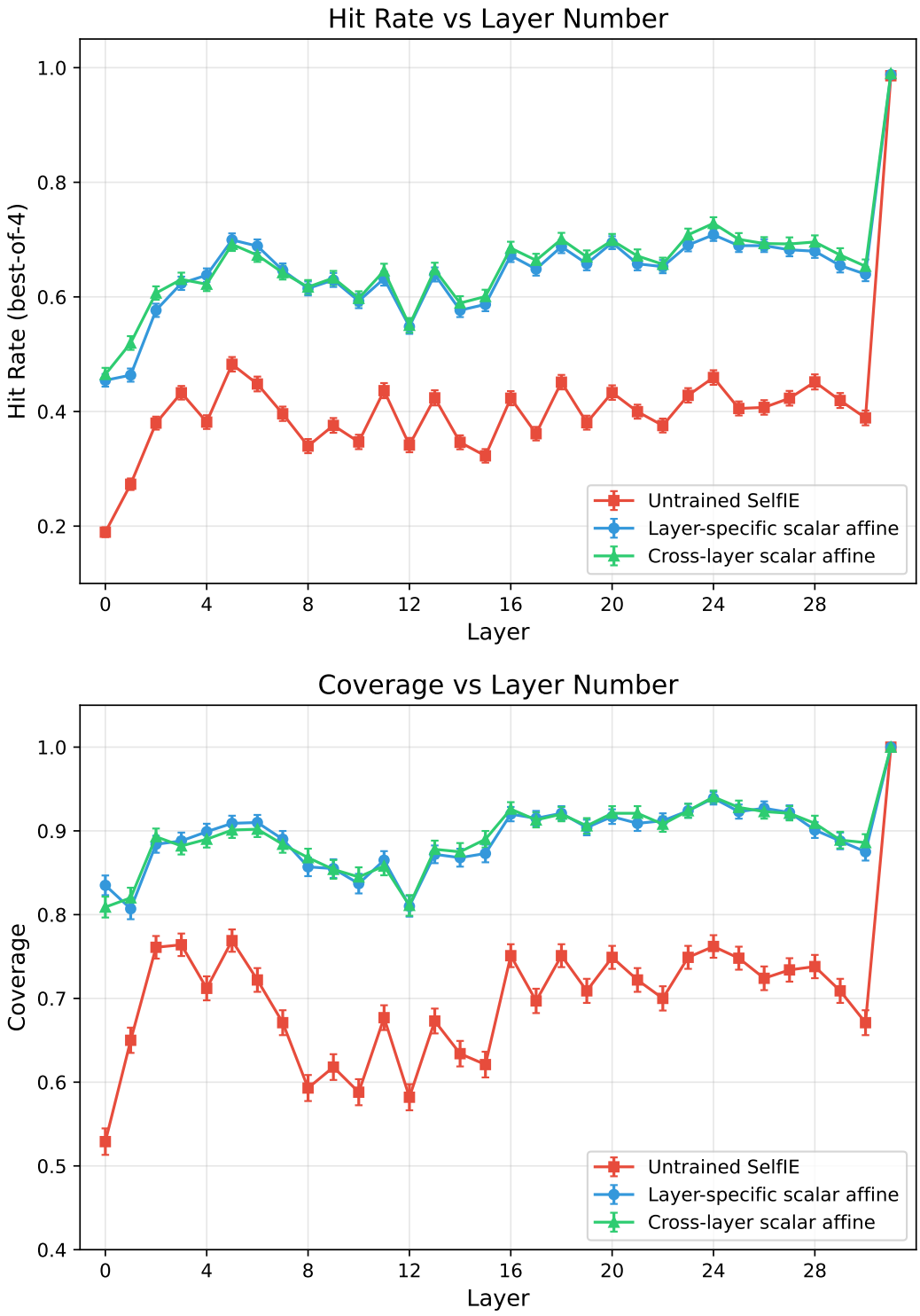


Figure 11: Comparison of layer-specific vs cross-layer scalar affine adapters, performance by layer.

were pooled together and shuffled, then used to train a single full-rank affine adapter. The adapter receives only the activation vector as input, with no information about which layer it came from.

The "Taboo" baseline uses the following prompt:

```
Describe {topic_phrase} without using the word "{original_title}", any
    part of it, or obvious synonyms. Be specific enough that someone
    could guess what you're describing.
```

e.g.

```
Describe bananas without using the word "Banana", any part of it, or
    obvious synonyms. Be specific enough that someone could guess what
    you're describing.
```

Both SelfIE generations and Taboo descriptions are scored using GTE-large (Li et al., 2023) embeddings with recall@k retrieval against all ∼50k topics.

H.2 Additional Metrics

Beyond the recall@100 metric shown in Figure 2, we examine additional aspects of retrieval performance. Figures 12 and 13 show recall@1 and Mean Reciprocal Rank (MRR) for the same adapters, also averaged over all layers in the middle half of the model, on which each adapter was trained.

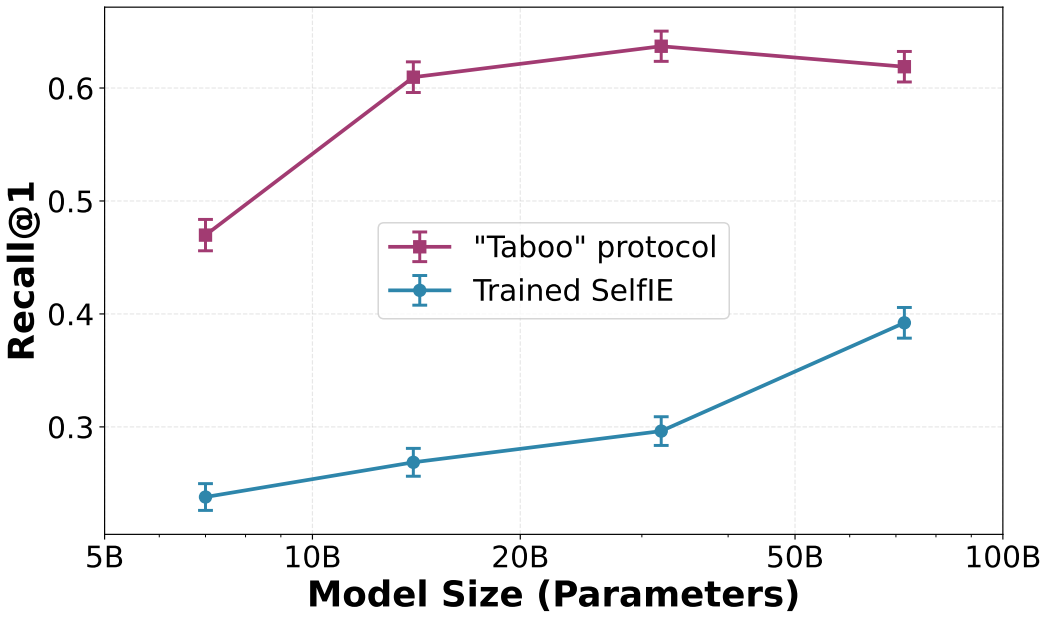


Figure 12: Recall@1 comparison between SelfIE and Taboo baseline across model scales, averaged over all layers in the middle half of each model. The same scaling trend observed in recall@100 holds for this stricter metric: SelfIE performance increases more rapidly than the Taboo baseline, narrowing the gap at larger model scales.

Another way to incorporate the results from the Taboo comparison is to **filter** the topics analyzed to only those where the Qwen model succeeds at describing the topic well enough that it's the top search result (recall@1). In order to compare the four SelfIE adapters on the same dataset, we therefore take the **intersection** of the Taboo recall@1 sets for each of the four models. The results are shown in Figure 14.

When evaluating on the single best-performing layer per model rather than averaging across the middle half (Figure 15), the 32B model breaks the otherwise monotonic scaling trend, achieving lower recall than 14B (32.3% vs 39.9%). Two architectural observations may explain this: First, Qwen2.5-14B and Qwen2.5-32B share identical residual stream dimensions (hidden_size=5120), meaning the full-rank adapter has identical capacity for both models despite the 2.3× parameter difference. Second, the optimal layer for 32B occurs at 38% relative depth (layer 24 of 64), substantially shallower than the other models (63–73% relative depth). This suggests that semantic

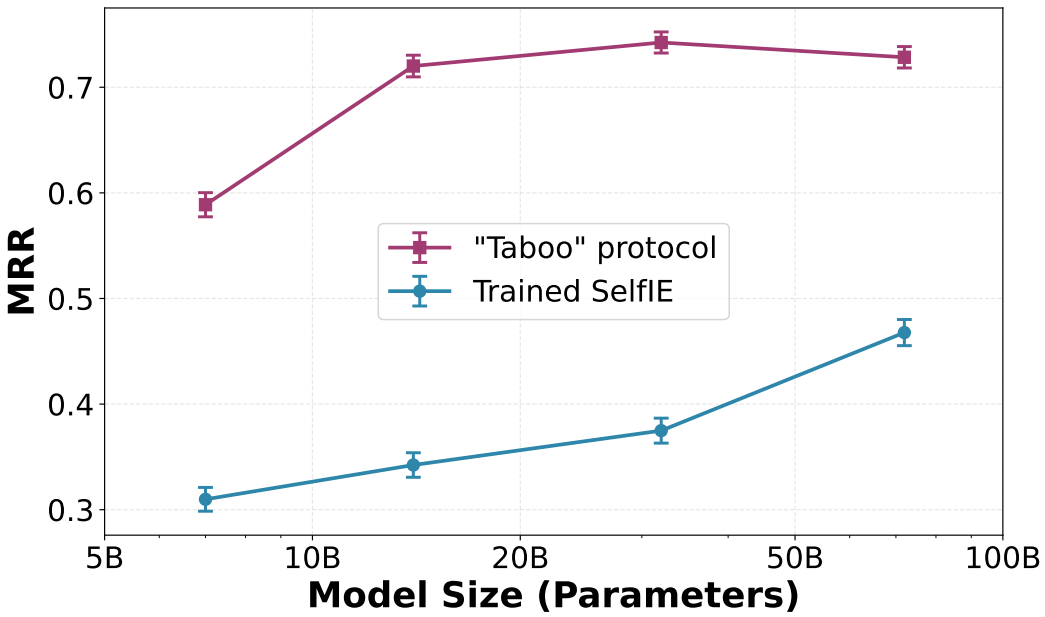


Figure 13: Mean Reciprocal Rank (MRR) comparison between SelfIE and Taboo baseline across model scales, averaged over all layers in the middle half of each model. SelfIE approaches the Taboo upper bound as model scale increases.

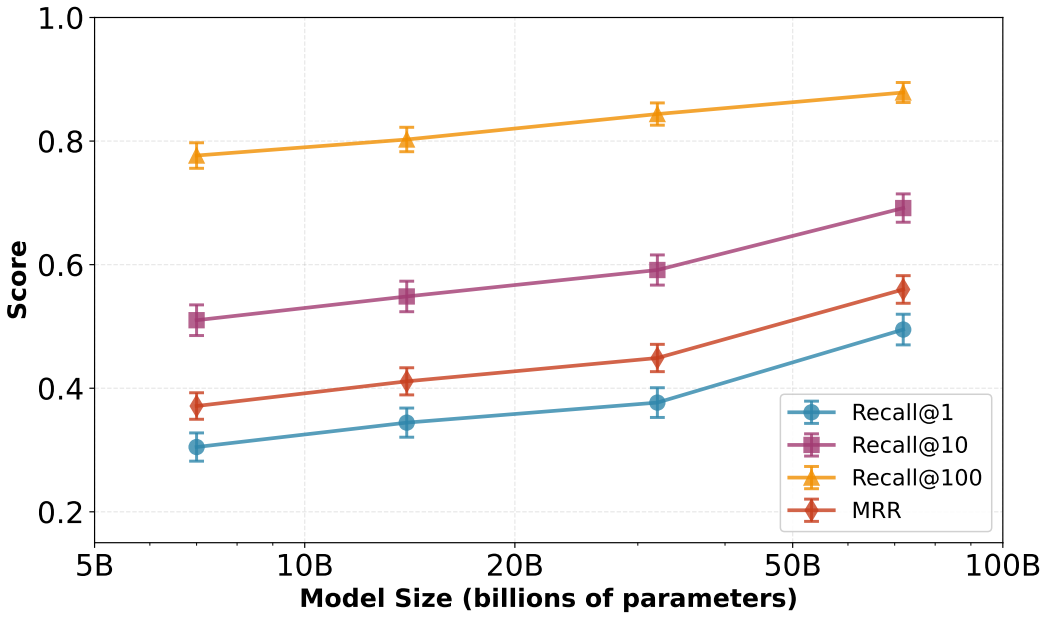


Figure 14: SelfIE results across scales, but limited to the subset of 1559 topics "known" by all four Qwen models in that they all achieve recall@1 on the Taboo baseline.

representations suitable for self-interpretation concentrate earlier in 32B's forward pass, possibly because the additional layers (64 vs 48) primarily perform output refinement rather than semantic enrichment. When averaging across all middle-half layers, the scaling trend is monotonic, indicating that this anomaly is specific to single-layer evaluation.

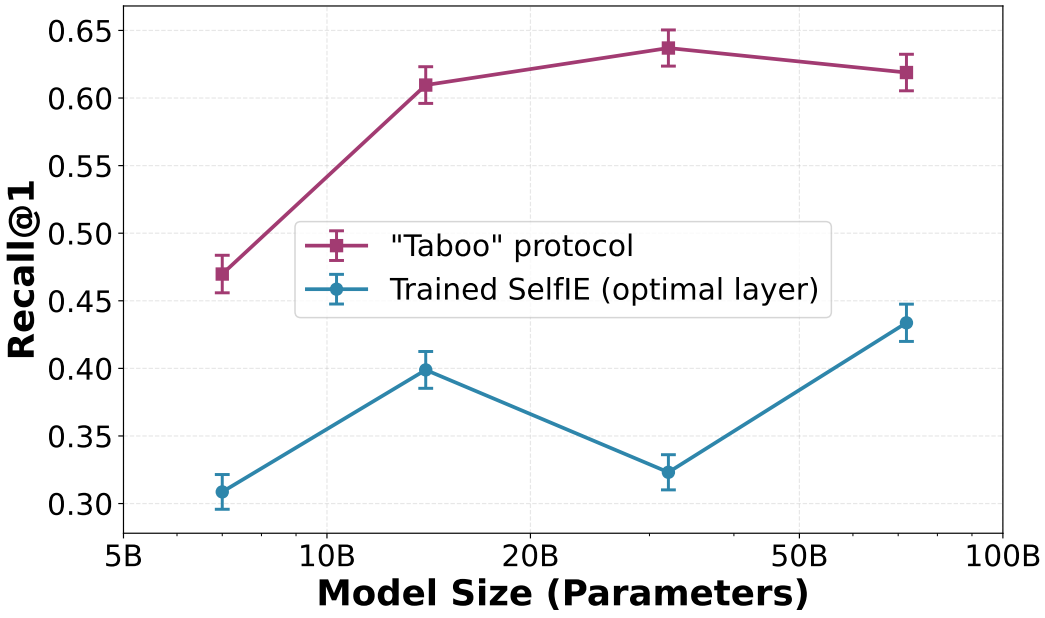


Figure 15: Recall@1 performance evaluated on the single best-performing layer for each model scale. Unlike the averaged metrics, this view reveals a non-monotonic scaling trend where the 32B model underperforms the 14B model, likely due to architectural differences in residual stream dimensions and optimal layer depth.

I Generalization to Other Model Families

To verify that our SAE findings generalize beyond the Llama model family, we trained adapters on Gemma-2-9B-IT using Gemma Scope SAEs (Lieberum et al., 2024). Tables 9 and 10 show architecture comparisons for 16k and 131k width SAEs respectively.

The results broadly replicate our Llama findings. On the smaller 16k SAE, scalar affine achieves the best validation loss, with low-rank additions providing no benefit, which makes sense because smaller datasets are easier to overfit. On the larger 131k SAE, the pattern matches Llama more closely: SA+LR (r=64) achieves the best performance, with the identity-preserving structure providing consistent gains over low-rank only variants. The bias vector remains critical in both cases, with scalar affine improving by over 3.4 loss units.

Table 9: Architecture comparison on Gemma Scope 16k SAE (layer 20). Parameter counts for Gemma-2-9B-IT ($d_{\text{model}} = 3584$). Scalar affine achieves the best validation loss.

Architecture	Params	Val Loss	Δ
Identity	0	5.382	—
Scale-only	1	5.377	-0.005
Scalar affine	3585	**1.960**	**-3.422**
SA + LR (r=4)	32K	1.976	-3.405
SA + LR (r=16)	118K	2.024	-3.358
SA + LR (r=64)	462K	2.075	-3.307
SA + LR (r=256)	1.8M	2.156	-3.226
LR only (r=4)	32K	2.221	-3.161
LR only (r=16)	118K	2.157	-3.225
LR only (r=64)	462K	2.144	-3.237
LR only (r=256)	1.8M	2.211	-3.171
Full-rank affine	12.8M	2.327	-3.055

Table 10: Architecture comparison on Gemma Scope 131k SAE (layer 20). SA+LR (r=64) achieves the best validation loss, matching the pattern observed on Llama Scope SAEs.

ARCHITECTURE	PARAMS	VAL LOSS	Δ
IDENTITY	0	5.478	—
SCALE-ONLY	1	5.454	-0.024
SCALAR AFFINE	3585	2.008	-3.471
SA + LR (R=4)	32K	1.932	-3.547
SA + LR (R=16)	118K	1.890	-3.589
SA + LR (R=64)	462K	**1.879**	**-3.600**
SA + LR (R=256)	1.8M	1.947	-3.531
LR ONLY (R=4)	32K	2.151	-3.328
LR ONLY (R=16)	118K	1.979	-3.500
LR ONLY (R=64)	462K	1.896	-3.583
LR ONLY (R=256)	1.8M	1.968	-3.511
FULL-RANK AFFINE	12.8M	1.986	-3.493

I.1 GEMMA-2-9B-IT EVALUATION

We also evaluated trained adapters on generation scoring and cross-dataset transfer.

Generation scoring on Gemma Scope SAEs. Table 11 shows generation scoring on held-out Gemma Scope latents. Trained adapters improve over untrained SelfIE (42.5% vs 31.4%), though the gap is smaller than on Llama. GemmaScope-trained adapters outperform Wikipedia-trained adapters on this in-distribution task, consistent with our finding that training data matching matters more than architecture.

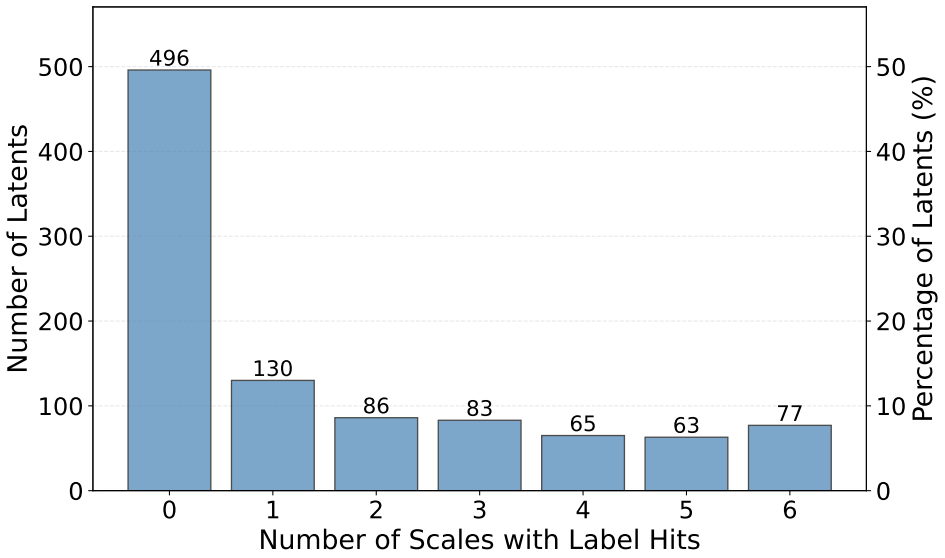
Table 11: Generation scoring on Gemma Scope SAE latents. GemmaScope-trained adapters improve over untrained SelfIE; Wikipedia-trained adapters show poor transfer.

METHOD	HIT RATE
ORIGINAL + 5 PARAPHRASES	**43.2%**
AUTO-INTERP LABELS ×6	39.7%
UNTRAINED SELFIE	31.4%
GemmaScope-trained:	
SCALAR AFFINE	42.5%
FULL-RANK	40.5%
SA+LR (R=64)	36.7%
Wikipedia-trained:	
SA+LR (R=64)	28.9%
SCALAR AFFINE	19.7%
FULL-RANK	19.1%

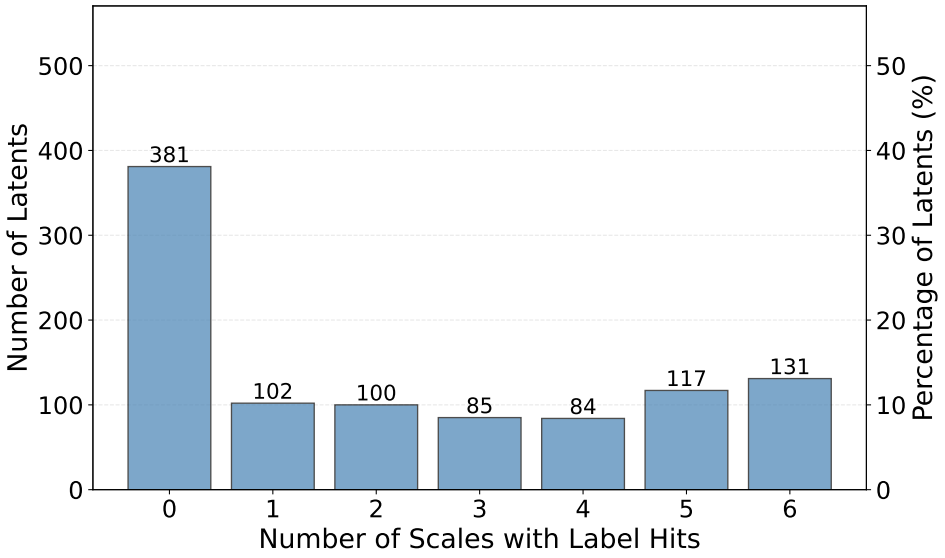
Scale sensitivity. Figure 16 shows histograms of valid scales per latent. As with Llama, trained adapters substantially increase the number of scales producing accurate outputs.

Wikipedia topic retrieval. Table 12 shows embedding-based retrieval on Wikipedia topics. Gemma full-rank adapters achieve the best performance on Wikipedia topics (81.3% R@1), matching the Llama pattern for this dataset. As expected, GemmaScope-trained adapters show minimal transfer to Wikipedia topics (<3.5% R@1), confirming that training data matching is essential for cross-dataset generalization.

These results confirm that our core findings replicate across model families: trained adapters improve self-interpretation, training data matching matters for generalization, and full-rank succeeds on Wikipedia but not SAEs.



(a) Scale sensitivity histogram for untrained SelfIE



(b) Scale sensitivity histogram for trained SelfIE (scalar affine, Gemma Scope dataset)

Figure 16: Histograms showing the distribution of the number of scales (out of 6 scales attempted) at which each method produced accurate labels (where "accurate" is defined as eliciting at least one nonzero activation in 10 trials of generation scoring). The trained adapter is less sensitive to scale, with more latents receiving accurate labels at all 6 scales.

I.2 QWEN

Unlike Llama and Gemma, no high-quality public SAEs exist for Qwen at time of writing. However, this limitation demonstrates a strength of our approach: training on Wikipedia contrastive vectors requires only the model itself, with no external interpretability artifacts. We trained adapters on Qwen-2.5-7B-Instruct using our Wikipedia topics dataset and evaluated via embedding-based retrieval.

Table 13 shows the same patterns: full-rank adapters achieve the best retrieval performance (52.6% R@1), with the architecture hierarchy matching Llama and Gemma. This confirms that trained self-interpretation generalizes to any open-weights model without requiring pre-existing SAEs. The

Table 12: Wikipedia topic retrieval on Gemma-2-9B-IT (best-of-6 protocol). Wikipedia-trained adapters achieve strong retrieval; GemmaScope-trained adapters show minimal transfer.

TRAINING DATA	ARCHITECTURE	R@1	R@100
WIKIPEDIA	FULL-RANK	**81.3%**	**98.5%**
WIKIPEDIA	SA+LR (R=64)	74.1%	98.1%
WIKIPEDIA	SCALAR AFFINE	46.9%	93.3%
GEMMASCOPE	SCALAR AFFINE	3.2%	46.5%
GEMMASCOPE	FULL-RANK	0.4%	17.7%
GEMMASCOPE	SA+LR (R=64)	0.3%	14.7%

Wikipedia contrastive vectors dataset serves as a universal training signal applicable across model families.

Table 13: Wikipedia topic retrieval on Qwen-2.5-7B-Instruct (best-of-6 protocol). Trained adapters achieve strong retrieval despite no SAEs being available for this model family.

ARCHITECTURE	R@1	R@100
FULL-RANK	**52.6%**	**95.2%**
SA+LR (R=64)	51.3%	93.4%
SCALAR AFFINE	8.3%	59.8%
UNTRAINED SELFIE	0.02%	3.1%

J BRIDGE ENTITY DETECTION DETAILS

J.1 IMMEDIATE ANSWERING PROTOCOL

To verify that the model answers two-hop questions without verbalized chain-of-thought, we use the following prompt template:

```
Complete the following statement with only the name of a {category}. If
    you don't know, make your best guess. {prompt}
```

We use one-shot prompting where the assistant's response is pre-filled with a single example answer:

```
User: Complete the following statement with only the name of a city. If
    you don't know, make your best guess. The capital of the country of
    origin of Tom Clancy's Rainbow Six Siege is
Assistant: Ottawa
```

This one-shot example demonstrates the expected format (a single word or short phrase) and primes the model to respond immediately rather than reasoning step-by-step. Bridge entity detection uses case-insensitive substring matching against the entity's canonical name and aliases provided in the TwoHopFact dataset.

J.2 TRAINING SOURCE COMPARISON

Training the adapter on Llama Scope SAE labels improves the overall fraction of bridge entities ever detected from $56.4\%_{\pm 2.2}$ to $66.4\%_{\pm 2.1}$, and when only considering the 242 questions where both methods succeeded at detecting the bridge entity, the SAE-trained adapter improves the overall fraction of generations including a match from $0.43\%_{\pm 0.05}$ to $0.71\%_{\pm 0.08}$, an increase of $1.7\times_{\pm 0.1}$.

From inspecting some of the generations from the SAE-trained SelfIE adapter, it's clear that a common failure mode here is for many of the generations to describe or name the current input token rather than higher-level semantic features, even for activations from relatively later layers. This

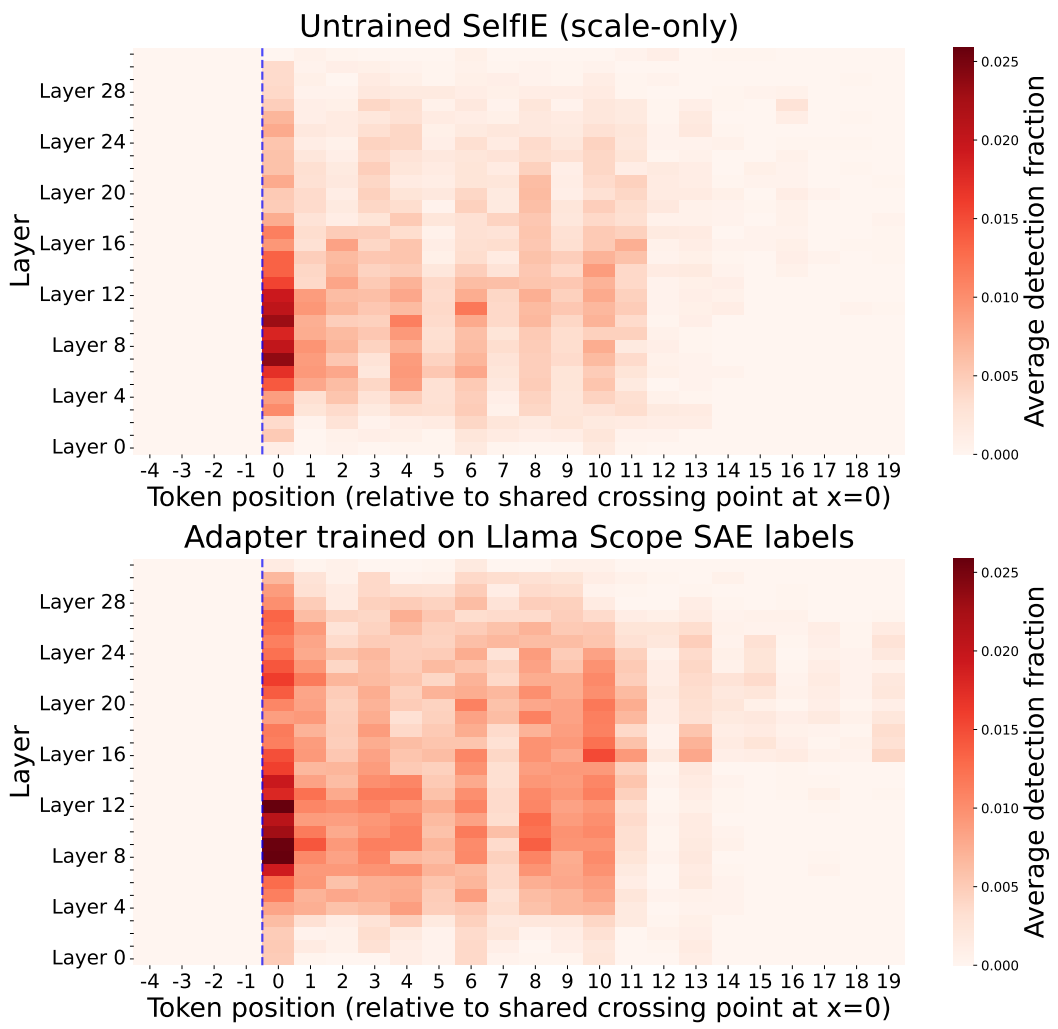


Figure 17: Bridge entity detection using SAE-trained scalar affine adapter. Compare to Figure 3 (Wikipedia-trained). Detection rates are lower but still significantly exceed untrained SelfIE.

Table 14: Bridge entity detection rates by training source. Both trained adapters substantially outperform untrained SelfIE.

Adapter	Prompts detected	Detection rate
Wikipedia-trained (SA)	455/500	$91.0\%_{\pm 1.3}$
SAE-trained (SA)	332/500	$66.4\%_{\pm 2.1}$
Untrained SelfIE	282/500	$56.4\%_{\pm 2.2}$

makes sense when we consider that many SAE latents actually describe token-identity-level features and this adapter was trained to name those, whereas in the case of the Wikipedia topic vectors the adapter was specifically trained to ignore token-level information and focus on extracting the topic-level semantic content.

K ALL-CAPS TRAINING EXPERIMENTS

To understand what the adapter's bias vector encodes, we conducted a controlled experiment: we trained a scalar affine adapter on Goodfire SAE labels that were transformed to ALL-CAPS format. If the bias vector captures format/style information while semantic content comes from the activation vector, we would expect (1) generated labels to be ALL-CAPS when the bias dominates (low scales) and (2) the semantic content to remain accurate regardless of format.

K.1 EXPERIMENTAL SETUP

We took the Goodfire auto-interpretability labels all available latents and converted them to uppercase. We then trained a scalar affine adapter on these ALL-CAPS labels using the same protocol as our main experiments. At evaluation time, we generated labels at six scales from the standard grid ($\{0.5, 0.8, 1.3, 2.1, 3.4, 5.5\}$) and classified each generation as ALL-CAPS or not based on whether all alphabetic characters were uppercase.

K.2 RESULTS

Table 15: Format and accuracy by scale for the ALL-CAPS trained adapter (1000 latents evaluated at each scale). At low scales, outputs are nearly 100% ALL-CAPS; at high scales, nearly 0%. Hit rates (generation scoring) peak at intermediate scales, suggesting a tradeoff between format fidelity and semantic accuracy.

SCALE	% ALL-CAPS	HIT RATE
0.5	100.0%	0.126
0.8	99.9%	0.283
1.3	99.4%	0.407
2.1	80.1%	0.428
3.4	16.8%	0.385
5.5	1.4%	0.246

Table 15 shows a striking scale-dependent transition in output format. At scale 0.5, 100% of generated labels are ALL-CAPS. This drops to 80% at scale 2.1, 17% at scale 3.4, and just 1.4% at scale 5.5. The transition is remarkably clean: 63% of latents follow the sequence AAAANN (ALL-CAPS at scales 0.5–2.1, not ALL-CAPS at scales 3.4–5.5), with another 19% following AAANNN and 16% following AAAAAN.

Importantly, semantic accuracy (measured by generation scoring hit rate) remains reasonable across the scale range, peaking at intermediate scales where neither bias nor input vector completely dominates. Comparing to the standard Goodfire-trained adapter from Table 1, the ALL-CAPS adapter achieves 61.7% best-of-6-scales hit rate versus 62.8% for the standard adapter—nearly identical performance despite the unusual output format. Restricting to the four lowest scales (0.5–2.1), where outputs are majority ALL-CAPS, performance remains comparable: 55.8% for the ALL-CAPS adapter (55.3% for standard), and even filtering to only ALL-CAPS outputs yields 53.2%. This confirms that the ALL-CAPS format itself does not substantially harm semantic accuracy.

K.3 QUALITATIVE EXAMPLE

Table 16 shows generated labels for Goodfire latent #41101—the same latent featured in Table 6. The ALL-CAPS trained adapter produces semantically accurate descriptions ("event handling," "callbacks in programming") that match the standard adapter's output, but in ALL-CAPS format when the scale is low enough for the bias to dominate.

K.4 INTERPRETATION

These results support a clean decomposition of the adapter's function:

Table 16: Greedy-decoded labels for Goodfire latent #41101 (the same latent shown in Table 6) using the ALL-CAPS trained adapter. The ALL-CAPS adapter produces semantically similar descriptions to the standard adapter ("event handling," "callbacks"; compare Table 6) but in ALL-CAPS format at low scales. At high scales (5.5), the semantic content degrades as the input vector overwhelms the bias.

Scale	Label
0.5	"THE ASSISTANT IS EXPLAINING HOW TO USE A NEW CONCEPT OR FEATURE"
0.8	"THE ASSISTANT IS EXPLAINING HOW TO HANDLE OR PROCESS INPUT"
1.3	"EVENT HANDLING AND CALLBACKS IN PROGRAMMING"
2.1	"EVENT HANDLING AND CALLBACKS IN PROGRAMMING"
3.4	"A FUNCTION THAT WILL BE CALLED WHEN SOMETHING HAPPENS...
5.5	"on" or "at" in English. It's a preposition...

- **The bias vector encodes format and prior.** When the input vector is scaled down, the bias dominates, and outputs reflect the training distribution's format (ALL-CAPS) and generic content patterns.
- **The input vector contributes semantic specificity.** The particular latent being interpreted determines the semantic content of the output, regardless of format. The same latent produces descriptions about "event handling" and "callbacks" whether trained on ALL-CAPS or standard labels.
- **Scale controls the interpolation.** Low scales produce outputs dominated by the bias (high format fidelity, generic content); high scales produce outputs dominated by the input vector (format breakdown, but potentially more specific content).

This decomposition explains why the bias vector accounts for $\sim$85% of improvement over untrained baselines (Section 3.4): it provides a "default interpretation prior" that steers generation toward the space of valid feature descriptions, while the input vector specifies *which* description within that space.

L Zero Vector Interpretations

To directly observe what the bias vector encodes, we generate SelfIE outputs for the zero vector $h = \mathbf{0}$ across adapters trained on different datasets. Since $f(\mathbf{0}) = \alpha \cdot \mathbf{0} + b = b$ for scalar affine adapters (and $f(\mathbf{0}) = b$ also for full-rank), zero-vector outputs reveal the bias in isolation.

Table 17 shows greedy-decoded outputs. The key distinction is between SAE-trained and Wikipedia-trained adapters:

- **SAE-trained adapters** generate abstract, categorical descriptions. Sampling at temperature 1.0, Llama Scope produces "numerical values and measurements," "references to software development," "technical terms related to databases"... all generic feature categories (in lowercase). Goodfire samples frequently begin with "The assistant is..." (explaining, providing options, proposing alternatives), suggesting its training labels emphasize instruction-following contexts.
- **Wikipedia-trained adapters** generate specific encyclopedic topics. Llama-8B samples skew toward Russian culture (Turgenev, Soviet filmmakers, Russian émigré authors). Qwen-7B samples favor music (Beatles, Elton John, rockabilly, synthesizers). Qwen-14B samples favor science and history (Ebola outbreaks, dinosaur extinction, Darwin, particle accelerators).

The bias vectors have learned what a "generic" SAE feature label versus a "generic" Wikipedia topic description looks like.

Table 17: Zero-vector interpretations reveal each adapter's learned prior. SAE-trained adapters (top) generate abstract feature descriptions; Wikipedia-trained adapters (bottom) generate specific encyclopedic topics.

Adapter (training data)	Greedy output ($h = \mathbf{0}$)
Llama-8B (Llama Scope)	"references to specific locations and their characteristics"
Llama-8B (Goodfire)	"The assistant is providing a list of options or alternatives"
Llama-8B (Wikipedia)	"the Russian composer who wrote the opera Eugene Onegin"
Qwen-7B (Wikipedia)	"the 1960s British rock band that pioneered psychedelic rock"
Qwen-14B (Wikipedia)	"the 1976 outbreak of the Ebola virus in Zaire"

These results confirm that the bias vector learns a distributional prior over valid interpretations, while the input vector selects among them.