SHAPED SAEs: EXPANDING THE TOOLKIT FOR MECHANISTIC INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) seek to decompose the dense activations of large language models (LLMs) into sparse, potentially interpretable features using a learnt overcomplete dictionary. In this work, we ask whether it is possible to enhance the semantic selectivity of the learnt dictionary atoms by linearly *shaping* the dense activations prior to learning an SAE. We explore rank-reducing transformations based on the recently introduced J-space corresponding to the averaged Jacobian from the current layer to the LLM output, projecting activations onto the "live" subspace spanned by eigenmodes of the Jacobian with larger singular values (which we expect to be more semantically relevant). We also introduce a new approach for training SAEs in which (a) an affine shaping transformation is learnt along with the dictionary atoms, (b) the atoms selection mechanism is based on the TopK projections along the one-dimensional subspace spanned by the atoms, and (c) the reconstruction is carried out using the same atoms. We call this approach a Matched Filter SAE (MF-SAE) based on analogies with communication theory. The promise of shaping is demonstrated via ealuations against dense activations and SOTA SAEs on benchmarks such as SAEBench and SAE Probes.

1 INTRODUCTION

Mechanistic interpretability seeks to develop insight into the internal representations and computations of large language models (LLMs). A key challenge is that model activations are high-dimensional dense representations encoding many concepts at once (i.e., they are "polysemantic"), making it difficult to isolate individual concepts and computational variables, and to relate them to model behavior. Sparse autoencoders (SAEs) have been proposed as a practical approach for replacing dense neural activations with potentially more interpretable feature vectors corresponding to their sparse representation with respect to an overcomplete dictionary (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024). Significant effort has been devoted to developing a variety of SAE formulations Gao et al. (2025); Bussmann et al. (2024); Rajamanoharan et al. (2024a); **?**); Bussmann et al. (2025), along with benchmarks assessing semantic selectivity Karvonen et al. (2025); Kantamneni et al. (2025).

Our broad objective is to investigate if the toolkit for mechanistic interpretability using SAEs can be expanded by asking a simple question: can we enhance the semantic selectivity of the learnt dictionary atoms by allowing ourselves the freedom to *shape* activations prior to learning an SAE? We make two key contributions that answer this question in the affirmative:
(1) *J-space Shaping:* Using the recently introduced Jacobian method Gurnee et al. (2026), we shape activations accounting for their impact on emphdownstream transmission. Locally, the Jacobian of a downstream mapping describes how perturbations to a residual-stream activation propagate through the model. Its leading right singular vectors therefore identify source-space directions with high sensitivity under that mapping. We use the leading singular vectors of a corpus-averaged Jacobian to divide the residual stream into a high-transmission *live* subspace and its orthogonal *quiet* complement. The terminology is deliberately relative: quiet directions are less transmitted by the selected average Jacobian, but they are not necessarily an exact null space, devoid of semantic information, or irrelevant in every context. However, we do expect the live directions to be more semantically relevant, and therefore investigate the impact of shaping activations by projecting them onto the live subspace.
(2) *Matched Filter SAE (MF-SAE):* We introduce a new SAE formulation in which we apply a learn-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

able affine transformation prior to encoding and decoding with an overcomplete dictionary, with the affine transformation and dictionary atoms learnt jointly based on reconstruction loss. The encoder outputs are TopK rectified projections along the unit norm atoms, while the decoder uses the same atoms and projections for reconstruction. We are inspired by the concept of matched filters from communication theory to learn atoms corresponding to dominant directions in the data, but noting that matched filters are only optimal when the noise is white, insert a linear transformation up front as a learnable "whitening" transformation. The bias term is motivated as a learnable centering mechanism which centers the data in order to minimize the energy of the atoms needed to represent it, similar to the centering routinely used in signal constellations used in communication systems.

We show that our proposed methods lead to improved performance on standard benchmark tasks on semantic feature extraction under a constrained feature budget on SAE Bench and SAE Probe. We make the following high-level observations regarding the performance on SAE Bench:

- Semantically shaped dense activations (prior to learning an SAE) perform better than original dense activations.
- The MF-SAE applied to standard dense activations performs better than SOTA SAEs as well as dense activations on SAE Bench
- The MF-SAE applied to semantically shaped activations performs slightly better than the MF-SAE at very tight feature budgets, but does not appear to improve performance at higher budgets. Thus, the problem of how best to combine semantic shaping with sparse coding remains open.

The performance trends on the SAE Probes benchmark are more variable, as we discuss in more detail in our results section (Section 6) We evaluate on Gemma-2-2B at layers 12 and 20, and on GPT-2 Small at layers 7 and 8 (Appendix E). On SAEBench, the MF-SAEs outperform the available baseline SAEs for both models, while on SAE Probes the gains are marginal. Probing accuracy is higher on Gemma-2-2B than on GPT-2 Small on both benchmarks, which we attribute to model scale: the tasks of SAE Probes in particular are more complex and benefit from a larger model.

2 Related Work

SAEs for language models: SAEs use dictionary learning as a means of decomposing the dense, superposed representations of language models into sparse features that are potentially more interpretable (Cunningham et al., 2023; Bricken et al., 2023). A variety of SAE formulations have been proposed, including L_1-regularized SAEs, TopK and BatchTopK SAEs Gao et al. (2025); Bussmann et al. (2024), Gated SAEs Rajamanoharan et al. (2024a), JumpReLU SAEs Rajamanoharan et al. (2024b), and Matryoshka SAEs Bussmann et al. (2025). These methods differ in how they impose sparsity and trade it against reconstruction fidelity. With respect to this state of the art, our contribution is two-fold: (a) we propose shaping dense activations prior to learning sparse representations, focusing on J-space Gurnee et al. (2026) based transformations that account for downstream information transmission, (b) we propose a new SAE formulation including a learnt geometric shaping and an atom selection mechanism combining ideas from TopK and matching pursuit.

Jacobian-based representations and computation: Our Jacobian-based shaping builds on the work in Gurnee et al. (2026), which averages the the linearized map from an intermediate residual stream to later residuals and uses the resulting token-associated directions to define and interrogate a sparse "J-space." While we build on the insight that activation variance and downstream information transfer are not directly related, but use the Jacobian in Gurnee et al. (2026) in a different way, using its singular value decomposition to identify important directions for downstream information transfer to shape the activations prior to learning an SAE-based sparse representation. Unrelated but worth mentioning are Jacobian Sparse Autoencoders Farnik et al. (2025), which jointly seek sparsity in the input features, output features, and the connections between them, for an MLP layer.

Cautionary notes: Recent work has raised important questions about the extent to which SAE features improve mechanistic analysis. Good reconstruction does not by itself guarantee a faithful or useful feature decomposition. Feature splitting and absorption can cause apparently coherent latents to omit expected instances Chanin et al. (2025). It was originally shown in Kantamneni et al. (2025) (the SAE Probes benchmark) that SAE representations do not consistently outperform dense acti-

vations or simple baselines for semantic probing, including under data scarcity, label noise, class imbalance, and distribution shift Kantamneni et al. (2025), but improved SAEs (including our own) have addressed these objections. It is shown in Mueller et al. (2025) (the MIB benchmark) that SAE features do not uniformly improve localization of causal variables relative to alternative representations. The AxBench benchmark provides complementary evidence in the context of concept detection and steering, where simple non-SAE baselines can be highly competitive (Wu et al., 2025). These results caution against treating sparse, interpretable-looking features or reconstruction fidelity as ends in themselves, and motivate exploring alternatives to the current SAE paradigm. As a counterpoint, it is argued in Peng et al. (2026) that the primary utility of SAEs is as a tool for concept discovery, despite their limitations in tasks such as steering and concept discovery. In this paper, we do not seek to resolve broader questions regarding the utility of SAEs. Our focus is to expand the universe of available sparse representations, with the goal of enhancing semantic selectivity under stringent feature budgets..

Benchmarks for mechanistic interpretability: Several recent benchmarks seek to evaluate semantic selectivity of proposed feature representations. **SAEBench** is the most comprehensive SAE-oriented evaluation, comparing architectures across reconstruction, sparsity, feature quality, probing, and functional metrics (Karvonen et al., 2025). **SAE Probes** directly tests whether SAE coordinates provide a useful sparse basis for recovering semantic information, including comparisons with dense representations (Kantamneni et al., 2025). Our experimental evaluations compare our proposed shaped SAEs to both dense activations and state of the art SAEs using SAEBench and SAE Probes.

MIB shifts the emphasis toward mechanistic validity, evaluating localization of causal variables and circuits rather than feature interpretability alone (Mueller et al., 2025). **AxBench** evaluates representation-based concept detection and steering, explicitly comparing SAEs with alternative approaches to representation intervention (Wu et al., 2025). For the steering, concept detection and causal variable location tasks considered in these benchmarks, it is shown that a variety of supervised techniques outperform SAEs. In this paper, we do not seek to address this gap. Our focus is on the complementary question of improving the unsupervised, semantically selective, sparse representations provided by SAEs.

3 JACOBIAN-BASED SHAPING

Let d denote the model dimension in a decoder-only language model, and let $h_{\ell,t} \in \mathbb{R}^d$ denote a residual-stream activation at a specified layer ℓ and token position t. Let $h_{\text{final},t'} \in \mathbb{R}^d$ denote the final-layer residual stream at position t'. The averaged Jacobian matrix J defined in Gurnee et al. (2026) is a first order approximation for how perturbations in $h_{\ell,t}$ propagate downstream to the final layer and affect $h_{\text{final},t'}$ for $t' \geq t$, averaged over a corpus $\mathcal{D}$ of prompts sampled from a pretraining-like distribution:

$$J_\ell \;=\; \mathbb{E}_{t,\; t' \geq t,\; \text{prompt} \sim \mathcal{D}} \left[\frac{\partial h_{\text{final},t'}}{\partial h_{\ell,t}} \right] \;\in\; \mathbb{R}^{d \times d}. \tag{1}$$

The J-lens defined in Gurnee et al. (2026) applies the averaged Jacobian J_ℓ to activations at layer ℓ and then unembeds them (after normalization) to obtain an output token distribution corresponding to that token. Sorting the output distribution corresponding to a activation at layer ℓ provides a human-readable description of what tokens the activation might tend to produce at the output, averaged across contexts.

We utilize the averaged Jacobian in an entirely different way in this paper. Prior to developing sparse representations for activations at layer ℓ, we seek to shape them using the Jacobian $J = J_\ell$ to emphasize downstream sensitivity, in order to better use the representation capacity of the SAE. We suppress ℓ from the notation henceforth for simplicity.

Live and quiet J-spaces: We quantify downstream sensitivity via the SVD of the Jacobian, $J = U\Sigma V^T$, where U and V are $d \times d$ orthonormal matrices containing the left and right singular vectors, and $\Sigma = diag(\sigma_1, ..., \sigma_d)$ is the diagonal matrix of singular values sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq ...\sigma_d$. We define the "live" space as directions corresponding to the largest r singular values $\sigma_i, i \leq r$, collected in the diagonal matrix $\Sigma_{live} = diag(\sigma_1, ..., \sigma_r)$. The corresponding left and

Figure 2: The MF-SAE. The input x is mapped to a latent z by the outer encoder parametrized by W_e and b_e. The latent z is correlated with unit-norm atoms, and the k largest rectified correlations are kept. These are used to decode the latent. The decoded latent is mapped to the reconstruction $\widehat{x}$ by the outer decoder parametrized by W_d and b_d.

right eigenmodes are collected into $d \times r$ matrices U_{live} and V_{live}, respectively. The value of r used to partition into live and quiet spaces is a hyperparameter that we experiment with.

Shaping via the live space: Under the intuition that the live modes are more semantically informative given their higher downstream gain, we shape the activations by projecting them onto the live space. That is, for a residual-stream activation h, we obtain a shaped activation x by projecting h onto the live space as follows.

$$h_{live} = V_{live} V_{live}^T h \tag{2}$$

We may view discarding the projection in the quiet space as a denoising operation. As we have mentioned, quiet directions are not necessarily devoid of semantic information, but given their lower average downstream gain, we hope to gain in semantic selectivity when operating on a tight feature budget.

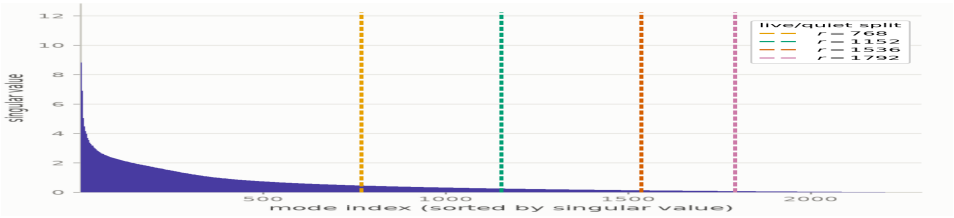


Figure 1: **linear projection ablation (SAEBench, Gemma-2-2B, layer 12, 16K dictionary, $k = 40$).** SB-Acc against k_{probe} for the raw and live MF-SAEs with and without the linear projections. The projections significantly improve both.

In our benchmarking with SAE Bench and SAE Probes, we will compare dense activations h and shaped dense activations h_{live} against the SAEs that we train, as well as against SOTA SAEs.

4 The Matched Filter SAE (MF-SAE)

Let $x \in \mathbb{R}^d$ denote the input to the SAE. In our setting, we may have $x = h$, the original dense activations, or $x = h_{live}$, the projection onto the live J-space.

The MF-SAE consists of an outer affine encoder, a dictionary with n elements, and an outer affine decoder (Figure 2). The norms of the dictionary elements are not constrained (thus simplifying training), but the atoms used as matched filters for encoding and decoding are unit-norm versions of the dictionary elements.

Outer encoder The input is first mapped to a latent

$$z = W_e x + b_e, \tag{3}$$

with $W_e \in \mathbb{R}^{d \times d}$ and $b_e \in \mathbb{R}^d$ learned.

Dictionary elements and normalized atoms The dictionary is a matrix $D \in \mathbb{R}^{n \times d}$ with rows d_j, and the atoms are their normalized versions $\hat{d}_j = d_j / \|d_j\|_2$. The atoms have no biases, and the same unit vector $\hat{d}_j$ is used both to detect atom j in the latent and to reconstruct it. The encoder (W_e, b_e) therefore shapes the space, and hence the metric, in which matching is performed.

Sparse coding via TopK matched filter selection We select the top K atoms with the largest rectified matched filter outputs. Specifically, let

$$w_j = max(\hat{d}_j^\top z, 0) = \text{ReLU}(\hat{d}_j^\top z),\ j = 1, ..., n \tag{4}$$

denote the rectified projections along the atoms.

We keep the k atoms with the largest projections

$$\mathcal{S} = \arg\operatorname*{top\text{-}k}_{j} \; w_j. \tag{5}$$

(If fewer than k rectified projections are positive, the number of selected atoms falls below k.)

The sparse code $c \in \mathbb{R}^n$ output by the MF-SAE is therefore $c_j = w_j$, $j \in \mathcal{S}$, $c_j = 0$ otherwise.

Decoding. The latent is reconstructed using the selected atoms, and then mapped back to a reconstruction of the input through the outer affine decoder.

$$\hat{z} = \sum_{j=1}^{n} c_j \hat{d}_j = \sum_{j \in \mathcal{S}} w_j \hat{d}_j, \qquad \hat{h} = W_d \hat{z} + b_d, \tag{6}$$

with $W_d \in \mathbb{R}^{d \times d}$ and $b_d \in \mathbb{R}^d$ learned.

Gradients and cost. Selection uses detached scores, and the coefficients of the selected atoms are recomputed differentiably, so W_e, W_d and the atoms receive gradients only through $\mathcal{S}$. Scoring all atoms is a single product $Z\hat{D}^\top$ and decoding is a single dense product, so the MF-SAE has the same cost as a Top-K SAE of the same width.

Relation to Top-K SAEs. Consider a Top-K SAE Gao et al. (2025) wrapped in the same linear projections, $z = W_{\text{in}} h + b_{\text{in}}$ and $\hat{h} = W_{\text{out}} \hat{z}$. Atom j has pre-activation

$$p_j = e_j^\top (z - b_{\text{pre}}) + b_j = \|e_j\| \, \|z - b_{\text{pre}}\| \cos(z - b_{\text{pre}}, e_j) + b_j, \tag{7}$$

where e_j is a free encoder row and b_j a free bias. The k largest pre-activations are kept with coefficients $\mathrm{ReLU}(p_j)$ and decoded with free columns $W_{\text{dec}}[:, j]$. The MF-SAE differs in three ways. (i) *Scale.* In (7) the norm $\|e_j\|$ and the bias b_j can make an atom win the selection independently of how well it matches, and the coefficient is the pre-activation itself. In the MF-SAE the score depends only on the alignment and the coefficient is the least-squares fit of a unit atom. (ii) *Bias.* The only per-atom offset in the MF-SAE is $\hat{d}_j^\top b_e$, which is fixed by the direction of the atom and one shared vector. Top-K Gao et al. (2025), BatchTopK Bussmann et al. (2024), Matryoshka Bussmann et al. (2025) and JumpReLU Rajamanoharan et al. (2024b) SAEs all learn a free bias per latent, and JumpReLU additionally learns a per-latent threshold. (iii) *Tying.* The effective encoder direction of atom j in the activation space is $W_e^\top \hat{d}_j$ and its decoder atom is $W_d \hat{d}_j$. When W_d is invertible,

$$W_e^\top \hat{d}_j = G\,(W_d \hat{d}_j), \qquad G = W_e^\top W_d^{-1}, \tag{8}$$

with G shared by all atoms, so encoder and decoder are tied up to one learned linear map. In a Top-K SAE, e_j and $W_{\text{dec}}[:, j]$ are independent. Consequently the linear projections are redundant for a Top-K SAE, since W_{in} and W_{out} can be absorbed into e_j and W_{dec}, but not for the MF-SAE, where they are the only mechanism that lets the encoder differ from the decoder.

Relation to matching pursuit SAEs: Subject to the non-negativity constraint in (4), $w_j \hat{d}_j^\top$ is the least squares estimate of z when we use only the jth atom. Thus, we may view this as the first step of non-negative matching pursuit Mallat & Zhang (1993), but applied to all atoms at once: every atom is scored against the same latent, with no residual update between selections. This is unlike MP-SAE Costa et al. (2026), which selects atoms sequentially against a residual. Sequential selection improves reconstruction on LLM activations but has not improved sparse probing or feature absorption Chanin & Garriga-Alonso (2026), and it requires k serial encoder passes.

4.1 Training objective

For a batch of B tokens with MF-SAE inputs x_b and reconstructions $\hat{x}_b$, the training loss is

$$\mathcal{L} = \frac{1}{B\,m} \sum_{b=1}^{B} \left\| \hat{x}_b - x_b \right\|_2^2 + \lambda\, \bar{c}, \qquad \bar{c} = \frac{1}{B\,k} \sum_{b=1}^{B} \sum_{j \in \mathcal{S}_b} c_{bj}, \tag{9}$$

where $\mathcal{S}_b$ and c_{bj} are the selected atoms and coefficients of token b from (5) and (6), and m is the dimension of the space in which the input lives. On raw activations, $x_b = h_b$ and $m = d$. On the live space, $x_b = h_{\text{live},b}$, the reconstruction is projected back onto the live space as in (**??**), and $m = r$, so the error is measured only on the live directions and normalized by their number. Since the coefficients are non-negative, $\bar{c}$ is the mean coefficient magnitude of the active atoms. Sparsity is enforced by the top-k selection, so the auxiliary term only regularizes the scale of the coefficients and must be kept small: we use $\lambda = 0.01$ in all experiments, as larger values collapse the dictionary to a small number of alive atoms (Appendix **??**).

5 Experimental Setup

5.1 Models, data and training

Models. We study Gemma-2-2B Gemma Team (2024) ($d = 2304$, 26 layers) at the residual stream after blocks 12 and 20. Results on GPT-2 Large Radford et al. (2019) are reported in Appendix E.

Jacobian. For each layer, the averaged Jacobian (1) is estimated once by reverse-mode automatic differentiation over 1000 prompts truncated to 64 tokens, with the final block as the target, and is shared by every SAE trained at that layer. Unless stated otherwise, the live space has $r = 1536$ directions (768 quiet).

Training. SAEs are trained on text from the Pile Gao et al. (2020) (uncopyrighted subset) in sequences of 128 tokens. Activations from 2048 sequences are pooled and shuffled, and each step uses a minibatch of 4096 tokens drawn from the pool, so that a minibatch mixes tokens from many contexts. We train for 7000 steps (about 28.7M tokens) with AdamW Loshchilov & Hutter (2019), learning rate 2×10^{-4}, no weight decay, and $\lambda = 0.01$. Evaluation uses a held-out partition of the same corpus.

Dictionary and sparsity. At each layer we train both the raw and the live MF-SAE with $n \in \{16384, 32768\}$ atoms (16K and 32K) and $k \in \{40, 80, 160, 320, 640\}$ active atoms per token.

5.2 Baselines

Dense activations. As SAE-free references, we probe the coordinates of the raw activation h and of the shaped activation h_{live} directly. The latter isolates the effect of shaping from that of sparse coding.

SAEBench Top-K SAEs. SAEBench Karvonen et al. (2025) provides Top-K SAEs Gao et al. (2025) on Gemma-2-2B with 16K latents at $k \in \{40, 80, 160, 320, 640\}$, matching our sparsity grid. We use two checkpoints of each: one after about 30M training tokens, which matches the 28.7M tokens seen by our SAEs, and the final one after 300M tokens.

Gemma Scope. We compare against the residual-stream JumpReLU SAEs Rajamanoharan et al. (2024b) of Gemma Scope Lieberum et al. (2024) at widths 16K and 32K. Their sparsity is set by a learned threshold rather than a fixed k, so we pair each with the MF-SAE of nearest average L0 (Table **??**).

Matryoshka SAE. We include the Matryoshka BatchTopK SAE Bussmann et al. (2025) with 32K latents and L0 $= 40$, the only configuration available for Gemma-2-2B.

5.3 Evaluation

SAEBench sparse probing. Following SAEBench Karvonen et al. (2025), we evaluate on 36 binary classification tasks from 8 datasets (Bias in Bios, Amazon reviews and sentiment, GitHub code, AG News and Europarl), with 4000 training and 1000 test examples per task and a context length of 128 tokens. SAE features are mean-pooled over tokens, the k_{probe} features with the largest difference in class means are selected, and a logistic-regression probe is trained on them. We report test accuracy averaged over tasks for $k_{\text{probe}} \in \{1, 2, 4, 8, 16, 32\}$, and refer to it as SB-Acc.

SAE Probes. To test probing on a broader set of tasks, we use the benchmark of Kantamneni et al. (2025), with 113 binary datasets in its standard data setting, and report test accuracy averaged over datasets for probes on the top-k_{probe} SAE features.

Reconstruction and downstream loss. We report the fraction of variance explained (PVE) of the MF-SAE input x, the fraction of alive atoms (atoms that are active on at least one evaluation token), and the cross-entropy loss recovered when the activation is replaced by its reconstruction,

$$\text{CE recovered} = \frac{\text{CE}_{\text{zero}} - \text{CE}_{\text{rec}}}{\text{CE}_{\text{zero}} - \text{CE}_{\text{clean}}}, \tag{10}$$

where CE_{zero} is obtained by zero-ablating the activation and CE_{clean} is the loss of the unmodified model. For the live MF-SAE we patch the model in three ways: *live only*, replacing h with $\hat{x}$; *live+quiet*, replacing h with $\hat{x} + (h - h_{\text{live}})$, which restores the exact quiet component; and *projection only*, replacing h with h_{live} without any SAE, which measures the cost of dropping the quiet component and upper-bounds the live-only value. Since the live PVE is computed on h_{live} rather than h, it is not directly comparable to the raw PVE.

Automated interpretability. For each feature we collect its top-activating windows on 6000 sequences of 128 tokens from the Pile. Qwen2.5-7B-Instruct Qwen Team (2025) is shown 12 activating windows and writes a one-line explanation. Given only the explanation, the same model then classifies 8 held-out activating windows against 8 top windows of randomly chosen other features, and we report the balanced accuracy as the detection score. Features with fewer than 20 activating windows are not scored. Figure 3 shows example dashboards of features from layer 20. Dashboards for all features with detection score above 0.7, including layer 12 and other models, are available at `https://huggingface.co/spaces/user1378916/MF-SAE/tree/main`.



(a) Raw, feature 2283: “a little bit”

(b) Raw, feature 4548: “the reason behind”

(c) Live, feature 2703: “later”

(d) Live, feature 9055: “example”

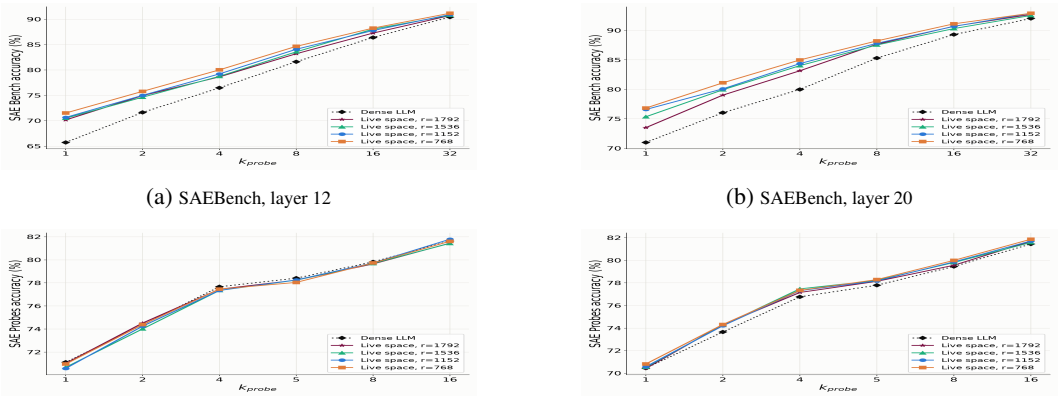
Figure 3: **Example feature dashboards (Gemma-2-2B, layer 20, 16K dictionary). (a, b)** Features of the raw MF-SAE. **(c, d)** Features of the live MF-SAE. Each dashboard shows the feature’s top-activating contexts together with its LLM-generated explanation. Dashboards for all features with detection score above 0.7, for layer 12 and other models, are available at the link in Section 5.3.

6 Results and Analysis

6.1 Shaping without an SAE

Before training any SAE, we test whether shaping preserves the information that probes rely on. We apply both probing benchmarks directly to the coordinates of the shaped activation h_{live} for $r \in \{768, 1152, 1536, 1792\}$ and compare with the raw activation h (Figure 4).

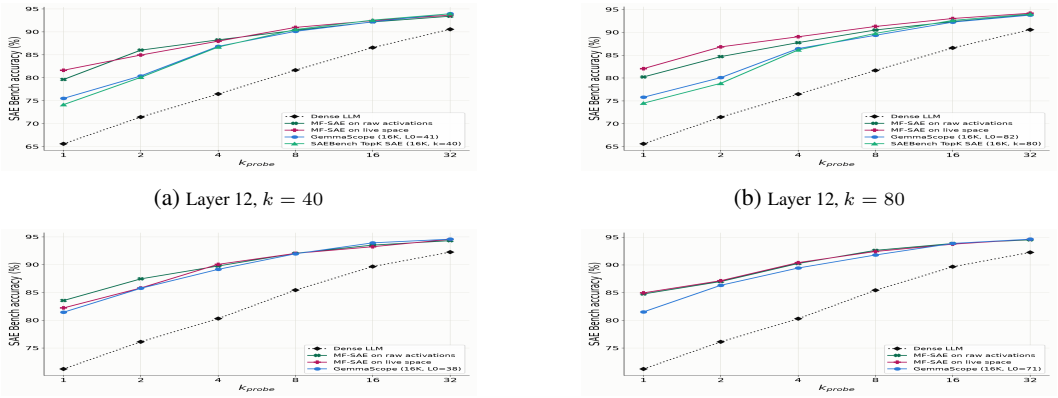
On SAEBench, h_{live} outperforms h at both layers and for every live dimension r (Figures 4a and 4b). Removing the quiet directions therefore does not discard the information these tasks need; it makes that information easier to find for a probe restricted to a few coordinates. On SAE Probes, h_{live} is comparable to h at layer 12 and slightly better at layer 20 (Figures 4c and 4d). Across both benchmarks, shaping never hurts probing, and the result is not sensitive to the choice of r. We use $r = 1536$ throughout.



(a) SAEBench, layer 12 (b) SAEBench, layer 20

(c) SAE Probes, layer 12 (d) SAE Probes, layer 20

Figure 4: **Probing shaped activations without an SAE (Gemma-2-2B).** Probe accuracy against k_{probe} for the raw activation h and for the shaped activation h_{live} with $r \in \{768, 1152, 1536, 1792\}$ live directions. **(a, b)** On SAEBench, h_{live} outperforms h at both layers and for every r. **(c, d)** On SAE Probes, h_{live} is comparable to h at layer 12 and slightly better at layer 20.



(a) Layer 12, $k = 40$ (b) Layer 12, $k = 80$

(c) Layer 20, $k = 40$ (d) Layer 20, $k = 80$

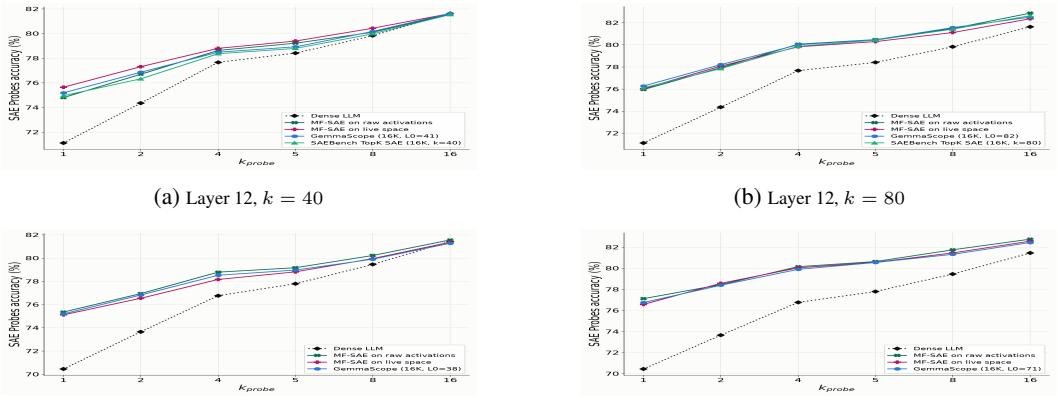
Figure 5: **SAEBench sparse probing (Gemma-2-2B, 16K dictionary).** SB-Acc against k_{probe} for the raw and live MF-SAEs and the available baseline SAEs. Rows are layers 12 and 20; columns are sparsity $k = 40$ and $k = 80$. Gemma Scope has no fixed sparsity, and the SAE with the closest average L0 is shown. Both MF-SAEs outperform the baselines in every panel. Raw and live are similar except in (b), where the live MF-SAE is consistently better.

6.2 Sparse probing

We compare the raw and live MF-SAEs with the baseline SAEs of Section 5.2 at sparsity $k = 40$ and $k = 80$ with a 16K dictionary, at layers 12 and 20. For Gemma Scope, sparsity is not fixed; we use the SAE whose average L0 is closest to k, so this comparison is approximate.

SAEBench. Both MF-SAEs outperform the available baselines at both layers and both sparsity levels (Figure 5). The raw and live MF-SAEs perform similarly in most settings. The exception is layer 12 at $k = 80$, where the live MF-SAE is consistently better than the raw one across k_{probe} (Figure 5b). Most of the improvement over the baselines therefore comes from the MF-SAE itself, with shaping adding a further gain in some settings. The benefit of shaping without an SAE (Section 6.1) thus carries over only partially once sparse coding is applied.

SAE Probes. On the 113 datasets of Kantamneni et al. (2025), the MF-SAEs improve over the baselines only marginally, and the raw and live MF-SAEs are again similar (Figure 6). This is consistent with the finding of Kantamneni et al. (2025) that SAE features offer little advantage on this broader benchmark, and indicates that the gains on SAEBench do not transfer fully to it.



(a) Layer 12, $k = 40$

(b) Layer 12, $k = 80$

(c) Layer 20, $k = 40$

(d) Layer 20, $k = 80$

Figure 6: **SAE Probes (Gemma-2-2B, 16K dictionary).** Mean test accuracy over the 113 datasets of Kantamneni et al. against k_{probe}, with the same layout and methods as Figure 5. The MF-SAEs improve over the baselines only marginally, and raw and live perform similarly.

7 Discussion and Limitations

Our results illustrate the potential gains from shaping activations to enhance semantic selectivity under tight feature budgets. We have chosen to illustrate our ideas through transformations for which we can offer intuitive interpretations. Projecting onto the live J-space selects directions with larger downstream gain, while the matched filter SAE is inspired by concepts of whitening and matched filtering from communication theory. Apart from performing well on benchmarks, the MF-SAE formulation results in a large fraction of useful atoms without requiring additional mechanisms to address dead atoms. We hope our work stimulates further research into the concept of shaping via linear or nonlinear transformations.

Limitations. *Shaping mechanisms* Our current work shows that, on the SAE Bench, semantic shaping performs better than dense activations, and MF-SAE performs better than SOTA SAEs. However, combining semantic shaping with MF-SAE yields gains only at tight feature budgets. Thus, enhancing performance by combining shaping mechanisms remains an important topic for future work. *Jacobian analysis* While we have empirically shown the benefits of projecting onto the live J-space, a more detailed analysis of the live and quiet components is an important topic for future work, which is complementary to the directions pursued in Gurnee et al. (2026). The quiet space is of particular interest: the Jacobian is a linearization averaged over contexts and target positions, so a quiet direction with small average output layer gain may matter in specific contexts or at intermediate layers. *Scope.* Our main results are on Gemma-2-2B at two layers, with GPT-2 Large in the appendix; we have not tested larger models. *Baselines.* Gemma Scope and the Matryoshka SAE were trained on different data and token budgets, and their sparsity is only approximately matched to ours. SAE Probes results are available for these baselines at a single sparsity level. *Automated interpretability.* Explanations and detection scores come from a single 7B model, and negatives are drawn from other features' activating windows rather than from non-activating text.

AI USE STATEMENT

We used generative AI tools to draft and edit parts of the manuscript from the authors' descriptions of the methods and results, to suggest its organization, to help locate and verify references, and to assist with code. The research ideas, methods, experiments and results are the authors' own. All AI-assisted text was reviewed and revised by the authors, and AI-assisted code was inspected and tested by the authors. We take full responsibility for the content of this paper.

REFERENCES

Joseph Isaac Bloom. Open source sparse autoencoders for all residual stream layers of GPT2-Small. Alignment Forum, February 2024. URL https://www.alignmentforum.org/posts/f9EgfLSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, et al. Towards monosemanticity: Decomposing language models with dictionary learning, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=d4dpOCqybL.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=m25T5rAy43.

David Chanin and Adrià Garriga-Alonso. Synthsaebench: Evaluating sparse autoencoders on scalable realistic synthetic data. *arXiv preprint arXiv:2602.14687*, 2026.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Isaac Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=R73ybUciQF.

Valérie Costa, Thomas Fel, Ekdeep S Lubana, Bahareh Tolooshams, and Demba Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. *Advances in Neural Information Processing Systems*, 38:34381–34424, 2026.

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. URL https://arxiv.org/abs/2309.08600.

Lucy Farnik, Tim Lawson, Conor Houghton, and Laurence Aitchison. Jacobian sparse autoencoders: Sparsify computations, not just activations. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=TPuFRuNano.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, T. Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, July 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rNfzT8YkgO.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qrU3yNfX0d.

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. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=sSrOwve6vb.

Kenny Peng, Rajiv Movva, Jon Kleinberg, Emma Pierson, and Nikhil Garg. Position: Use sparse autoencoders to discover unknowns. In *Forty-third International Conference on Machine Learning Position Paper Track*, 2026. URL https://openreview.net/forum?id=x1Px74tbvs.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

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

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=zLBlin2zvW.

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

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. Transformer Circuits Thread, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=K2CckZjNy0.

A AVAILABLE BASELINE SAEs

Tables 1 and 2 list the open-source SAEs we compare against at each model and layer. JumpReLU SAEs have no fixed sparsity; their L0 is the average number of active latents per token, and we pair each MF-SAE with the JumpReLU SAE of closest average L0. The Matryoshka SAE exists only with a 32K dictionary at L0 $= 40$, so it appears only in the 32K comparisons of Appendix C and not in the main text.

Table 1: **Baseline SAEs on Gemma-2-2B.** For JumpReLU SAEs, L0 is the average number of active latents per token.

Layer	SAE	Architecture	Width	Sparsity
12	SAEBench Karvonen et al. (2025)	Top-K	16K	$k = 40, 80, 160, 320$
	Gemma Scope Lieberum et al. (2024)	JumpReLU	16K	L0 $= 22, 41, 82, 176, 445$
	Gemma Scope Lieberum et al. (2024)	JumpReLU	32K	L0 $= 12, 22, 40, 76, 155, 360$
	Matryoshka Bussmann et al. (2025)	BatchTopK	32K	L0 $= 40$
20	Gemma Scope Lieberum et al. (2024)	JumpReLU	16K	L0 $= 22, 38, 71, 139, 294$
	Matryoshka Bussmann et al. (2025)	BatchTopK	32K	L0 $= 40$

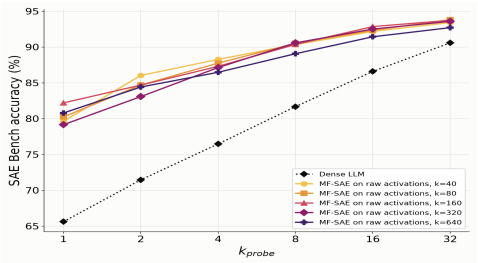
Table 2: **Baseline SAEs on GPT-2 Small.** Our MF-SAEs are trained at the same hook point, with the same dictionary size and matched sparsity.

Layer	SAE	Hook	Width	Sparsity
7	ReLU, L1 Bloom (2024)	`resid_pre`	24,576	L0 ≈ 54
8	Top-K Gao et al. (2025)	`resid_post`	32,768	$k = 32$

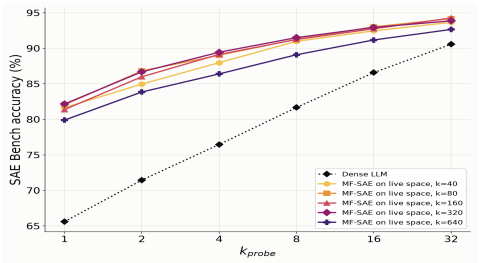
B SPARSITY LEVEL ABLATION, 16K DICTIONARY

This section extends Section 6.2 to all sparsity levels $k \in \{40, 80, 160, 320\}$ with a 16K dictionary. Across sparsity levels, $k = 320$ gives the best MF-SAE on SAE Probes in every setting, while on SAEBench it is among the best sparsity levels but not always the best (Figures 7 and 8). At layer 12 the baselines are the SAEBench Top-K and Gemma Scope 16K SAEs; at layer 20, only Gemma Scope 16K is available (Table 1).

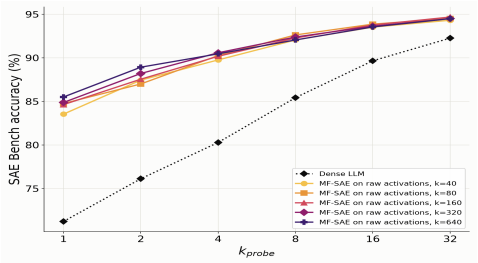
B.1 MF-SAE ACROSS SPARSITY LEVELS



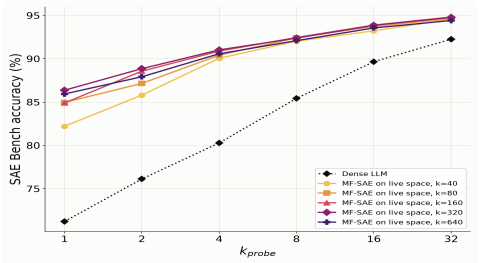
(a) Raw, layer 12



(b) Live, layer 12

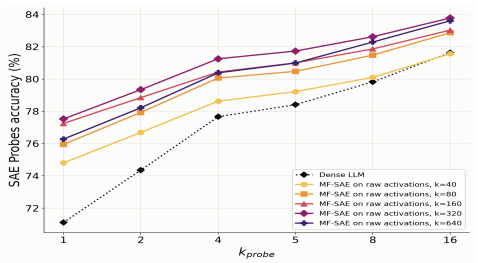


(c) Raw, layer 20

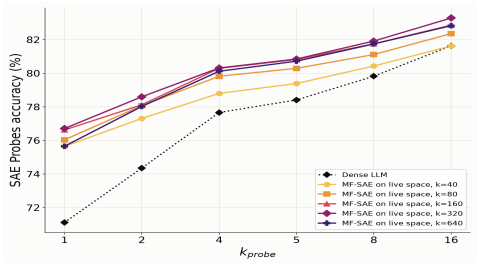


(d) Live, layer 20

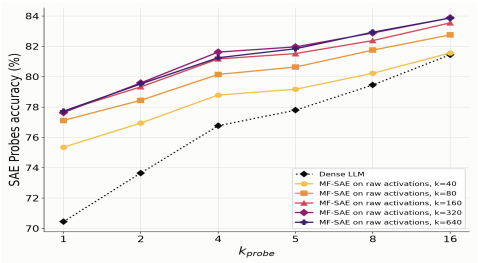
Figure 7: **SAEBench, MF-SAE across sparsity levels (Gemma-2-2B, 16K dictionary).** SB-Acc against k_{probe} for the MF-SAE at $k \in \{40, 80, 160, 320\}$. Columns are the raw and live MF-SAE; rows are layers 12 and 20. $k = 320$ is among the best sparsity levels but not the best in every panel.



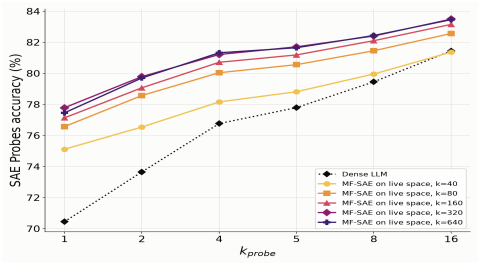
(a) Raw, layer 12



(b) Live, layer 12



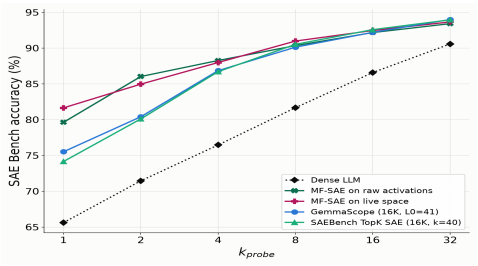
(c) Raw, layer 20



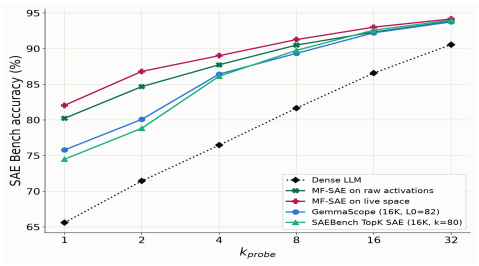
(d) Live, layer 20

Figure 8: **SAE Probes, MF-SAE across sparsity levels (Gemma-2-2B, 16K dictionary).** Mean test accuracy against k_{probe} for the MF-SAE at $k \in \{40, 80, 160, 320\}$, with the same layout as Figure 7. $k = 320$ is consistently the best sparsity level.

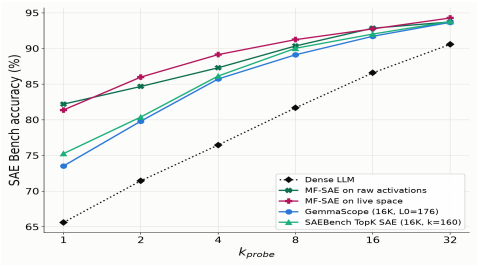
B.2 COMPARISON WITH BASELINES, LAYER 12



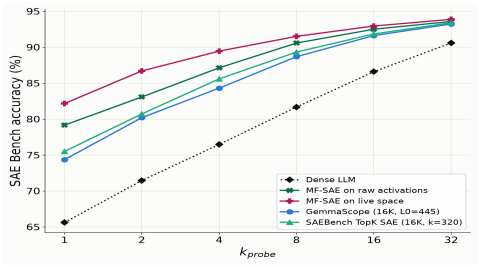
(a) $k = 40$



(b) $k = 80$

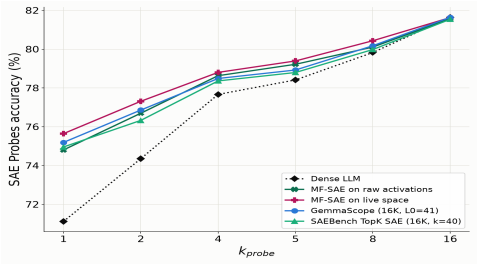


(c) $k = 160$

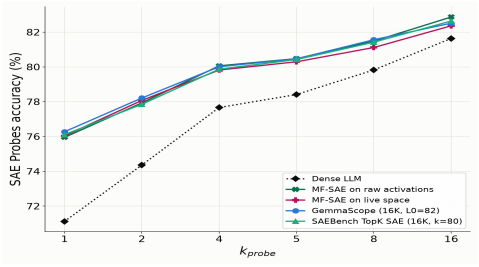


(d) $k = 320$

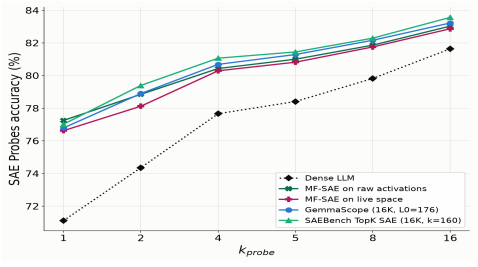
Figure 9: **SAEBench, layer 12, 16K dictionary.** SB-Acc against k_{probe} for the raw and live MF-SAEs, the SAEBench Top-K SAE at the same k, and the Gemma Scope 16K SAE of closest average L0.



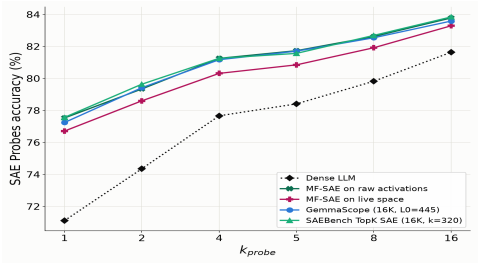
(a) $k = 40$



(b) $k = 80$



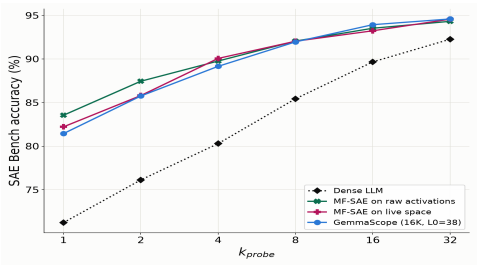
(c) $k = 160$



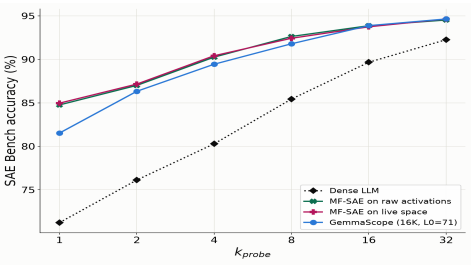
(d) $k = 320$

Figure 10: **SAE Probes, layer 12, 16K dictionary.** Mean test accuracy against k_{probe}, with the same methods and layout as Figure 9.

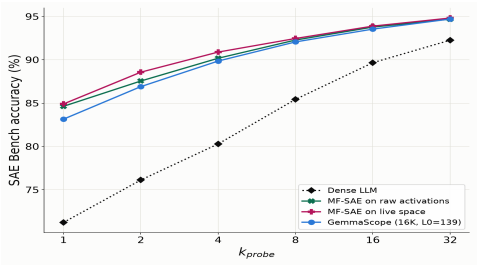
B.3 COMPARISON WITH BASELINES, LAYER 20



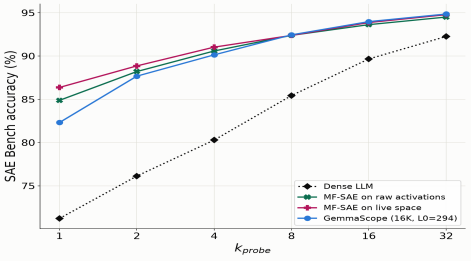
(a) $k = 40$



(b) $k = 80$

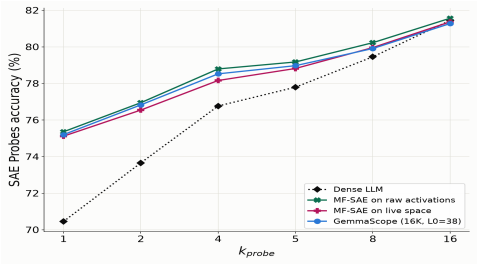


(c) $k = 160$

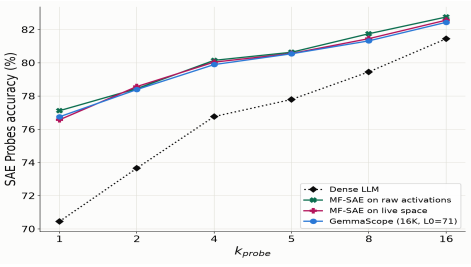


(d) $k = 320$

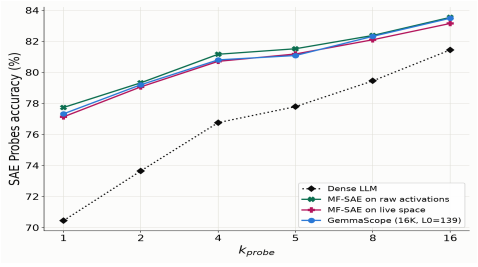
Figure 11: **SAEBench, layer 20, 16K dictionary.** SB-Acc against k_{probe} for the raw and live MF-SAEs and the Gemma Scope 16K SAE of closest average L0 (38, 71, 139 and 294 for $k = 40, 80, 160, 320$). No SAEBench SAEs are available at this layer.



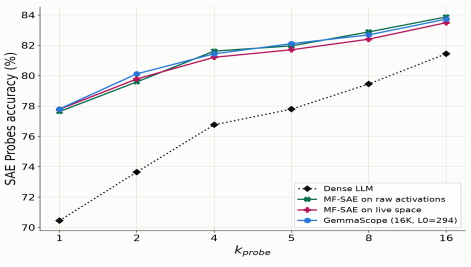
(a) $k = 40$



(b) $k = 80$



(c) $k = 160$



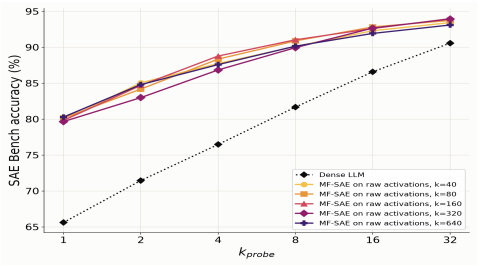
(d) $k = 320$

Figure 12: **SAE Probes, layer 20, 16K dictionary.** Mean test accuracy against k_{probe}, with the same methods and layout as Figure 11.

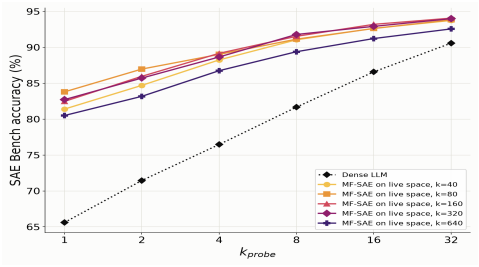
C DICTIONARY SIZE ABLATION, 32K DICTIONARY

This section repeats Appendix B with a 32K dictionary. As with 16K, $k = 320$ gives the best MF-SAE on SAE Probes, while on SAEBench it is among the best sparsity levels but not always the best. SAEBench does not release 32K SAEs. At layer 12 the baselines are Gemma Scope 32K and the Matryoshka SAE at L0 $= 40$; at layer 20, Gemma Scope has no 32K SAEs and only the Matryoshka SAE is available. Since the Matryoshka SAE exists only with a 32K dictionary, it appears only in this section. On SAE Probes at $k = 40$, the Matryoshka SAE is comparable to the MF-SAEs at layer 12 and outperforms them at layer 20.

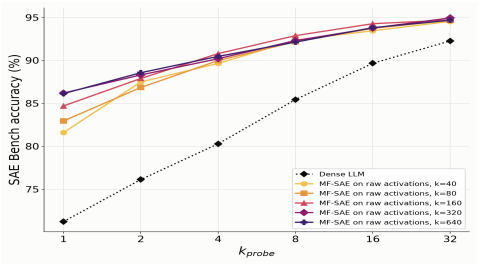
C.1 MF-SAE ACROSS SPARSITY LEVELS



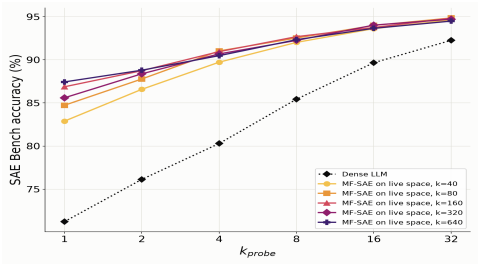
(a) Raw, layer 12



(b) Live, layer 12

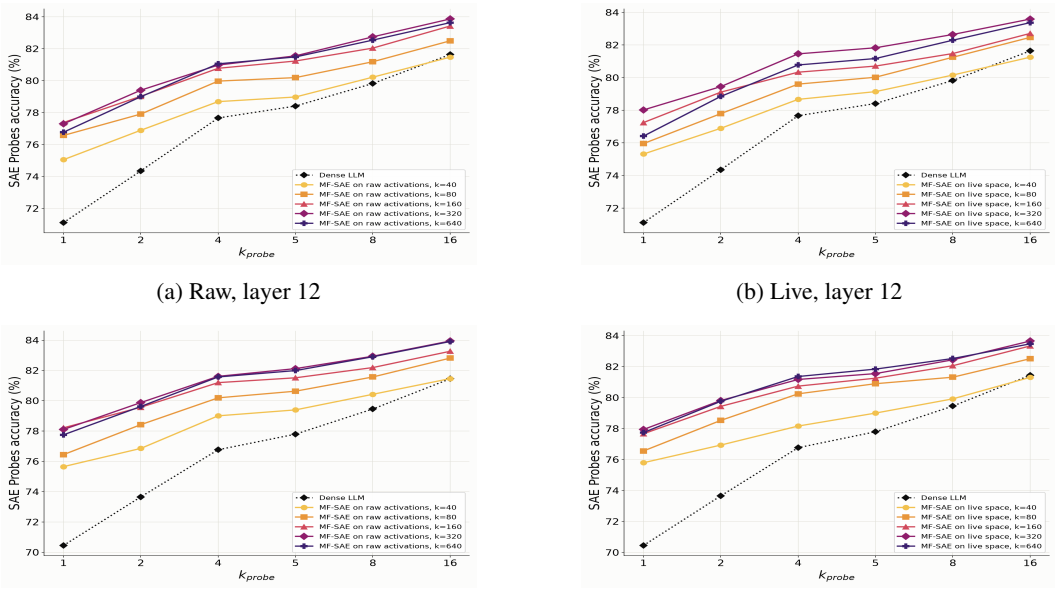


(c) Raw, layer 20



(d) Live, layer 20

Figure 13: **SAEBench, MF-SAE across sparsity levels (Gemma-2-2B, 32K dictionary).** SB-Acc against k_{probe} for the MF-SAE at $k \in \{40, 80, 160, 320\}$. Columns are the raw and live MF-SAE; rows are layers 12 and 20. $k = 320$ is among the best sparsity levels but not the best in every panel.

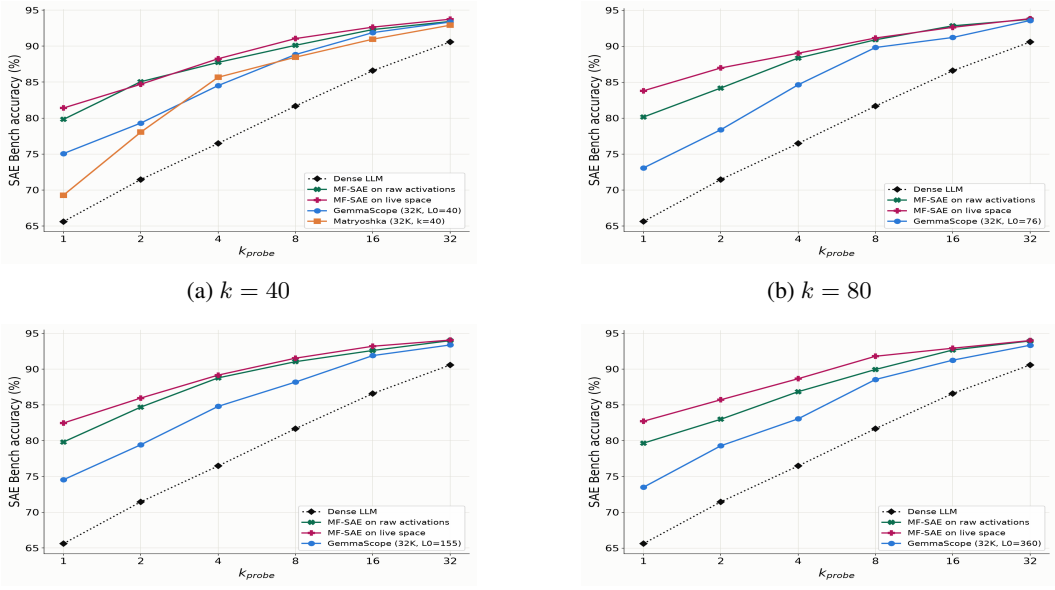


(a) Raw, layer 12 (b) Live, layer 12

(c) Raw, layer 20 (d) Live, layer 20

Figure 14: **SAE Probes, MF-SAE across sparsity levels (Gemma-2-2B, 32K dictionary).** Mean test accuracy against k_{probe} for the MF-SAE at $k \in \{40, 80, 160, 320\}$, with the same layout as Figure 13. $k = 320$ is consistently the best sparsity level.

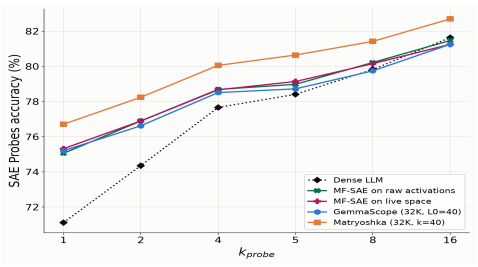
C.2 Comparison with baselines, layer 12



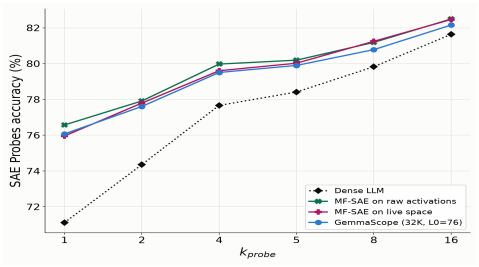
(a) $k = 40$ (b) $k = 80$

(c) $k = 160$ (d) $k = 320$

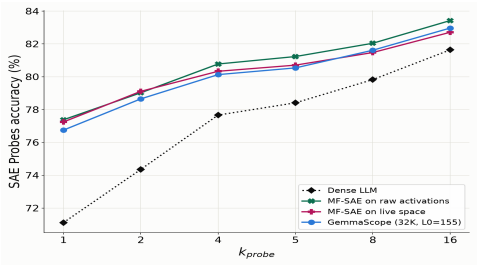
Figure 15: **SAEBench, layer 12, 32K dictionary.** SB-Acc against k_{probe} for the raw and live MF-SAEs, the Gemma Scope 32K SAE of closest average L0, and, in (a), the Matryoshka SAE.



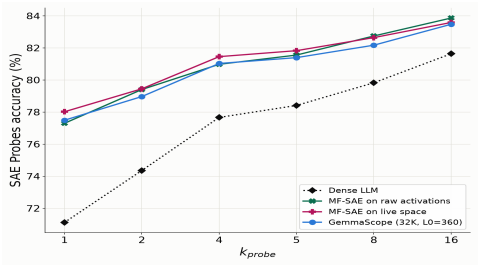
(a) $k = 40$



(b) $k = 80$



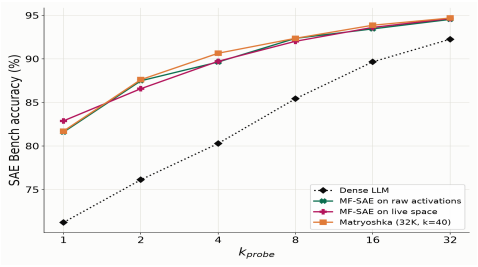
(c) $k = 160$



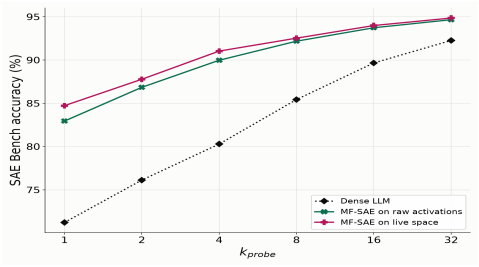
(d) $k = 320$

Figure 16: **SAE Probes, layer 12, 32K dictionary.** Mean test accuracy against k_{probe}, with the same methods and layout as Figure 15. In (a), the Matryoshka SAE is comparable to the MF-SAEs.

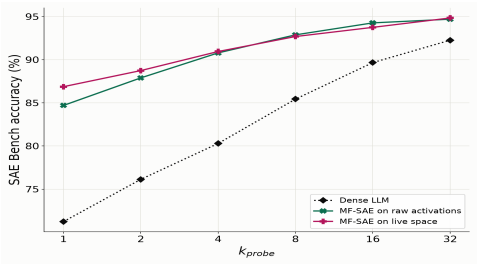
C.3 Comparison with baselines, layer 20



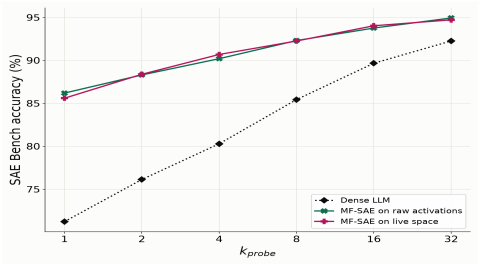
(a) $k = 40$



(b) $k = 80$

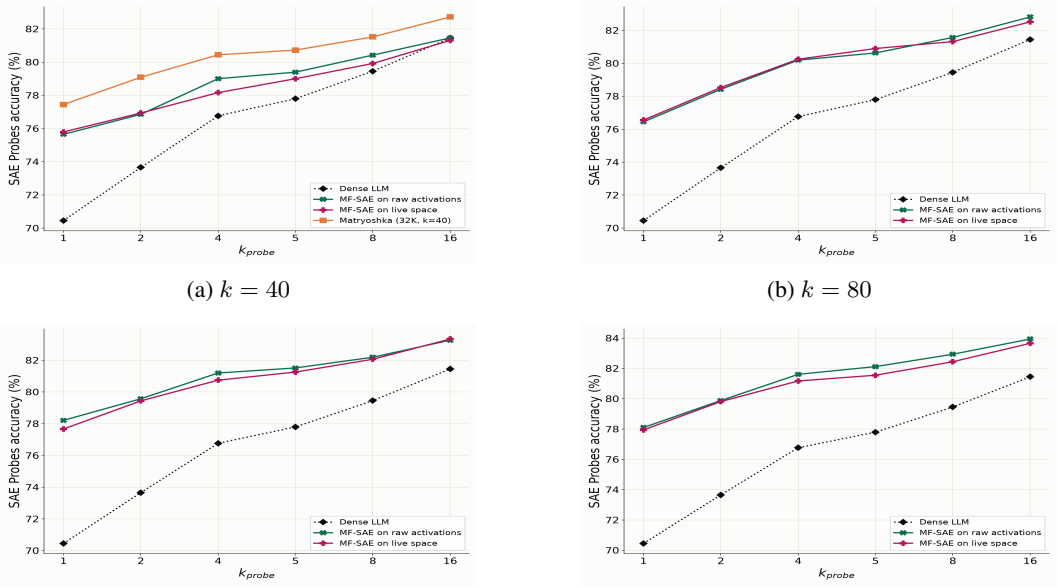


(c) $k = 160$



(d) $k = 320$

Figure 17: **SAEBench, layer 20, 32K dictionary.** SB-Acc against k_{probe} for the raw and live MF-SAEs and, in (a), the Matryoshka SAE. No SAEBench or Gemma Scope 32K SAEs are available at this layer.



(a) $k = 40$ (b) $k = 80$

(c) $k = 160$ (d) $k = 320$

Figure 18: **SAE Probes, layer 20, 32K dictionary.** Mean test accuracy against k_{probe}, with the same methods and layout as Figure 17. In (a), the Matryoshka SAE outperforms the MF-SAEs.

D LINEAR PROJECTION ABLATION

We compare the MF-SAE with and without the linear projections W_e and W_d (Section 4) at layer 12, 16K dictionary, $k = 40$. The linear linear encoder and decoder significantly improve probing accuracy on both the raw and the live space (Figure 19), consistent with Section 4: they are the only mechanism that lets the encoder of the MF-SAE differ from its decoder.

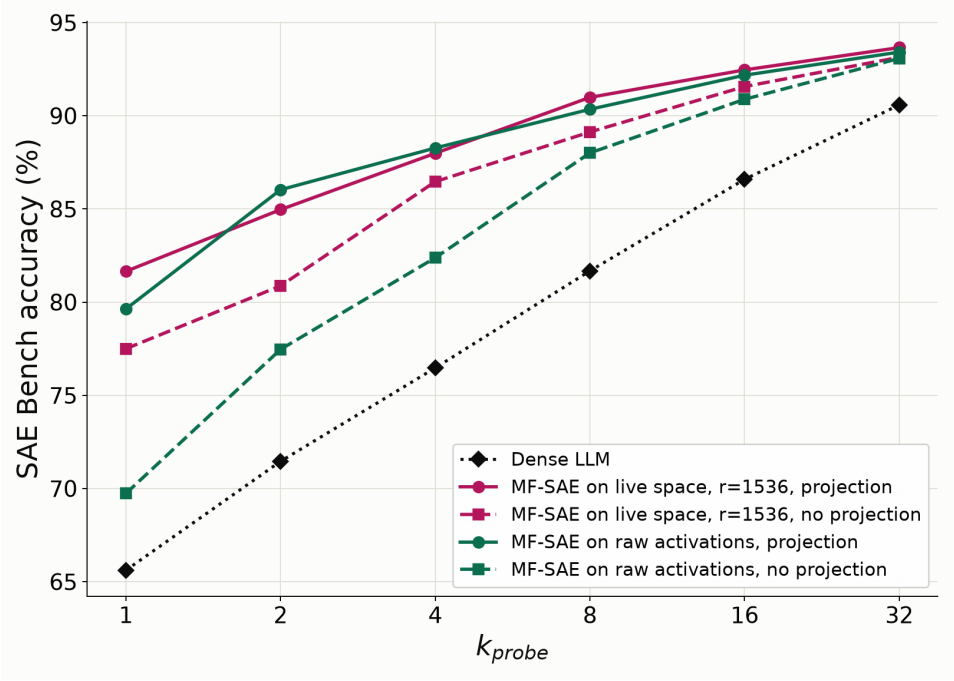


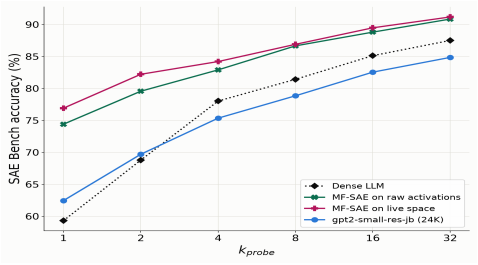
Figure 19: **linear projection ablation (SAEBench, Gemma-2-2B, layer 12, 16K dictionary,** $k = 40$**).** SB-Acc against k_{probe} for the raw and live MF-SAEs with and without the linear projections. The projections significantly improve both.

E GPT-2 RESULTS

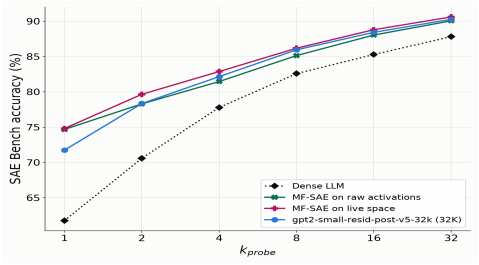
We train MF-SAEs on GPT-2 Small ($d = 768$) at layers 7 and 8, matching the hook point, dictionary size and sparsity of an available open-source SAE at each layer (Table 2). At layer 7 and 8 we train both the raw and the live MF-SAE, with the Jacobian estimated as in Section 5.1 and $r = 512$ live directions.

Layer 7. We compare with the residual-stream SAE of Bloom (2024) at `blocks.7.hook_resid_pre`, a ReLU SAE trained with an L1 penalty on 300M tokens of OpenWebText, with 24,576 latents and average L0 ≈ 62. Our MF-SAEs use $n = 24{,}576$ and $k = 64$.

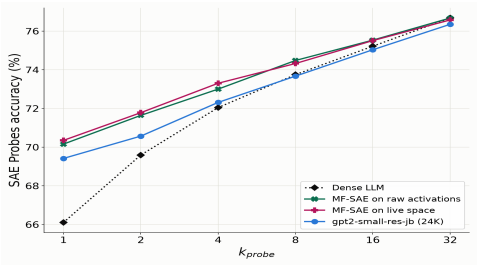
Layer 8. We compare with the Top-K SAE of Gao et al. (2025) at `blocks.8.hook_resid_post`, with 32,768 latents and $k = 32$. Our MF-SAE uses $n = 32{,}768$ and $k = 32$. At this layer we report only the raw MF-SAE; the live MF-SAE was not trained.



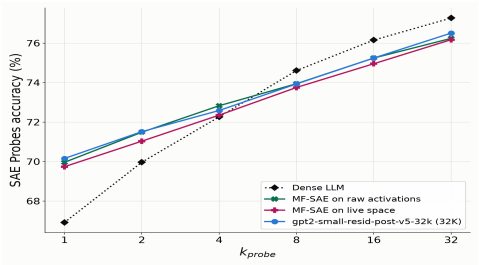
(a) SAEBench, layer 7



(b) SAEBench, layer 8



(c) SAE Probes, layer 7



(d) SAE Probes, layer 8

Figure 20: **GPT-2 Small, matched baselines.** Probe accuracy against k_{probe}. Rows are SAEBench and SAE Probes; columns are layer 7 (raw and live MF-SAEs, $n = 24{,}576$, $k = 64$, against the SAE of Bloom (2024)) and layer 8 (raw MF-SAE, $n = 32{,}768$, $k = 32$, against the Top-K SAE of Gao et al. (2025)).