Scaling Laws for SAE Training Data

Nikita Koriagin[*] Yandex

Nikita Balagansky T-Tech

Daniil Gavrilov T-Tech

Abstract

Activation storage is often a bottleneck in sparse autoencoder (SAE) training. We study how many unique activations an SAE needs when its total training budget is fixed. Quality improves rapidly as the buffer grows, then reaches a plateau where replaying the buffer is nearly as effective as using fresh activations throughout training. Beyond this point, additional unique activations bring little benefit to reconstruction or interpretability, so the buffer need not grow with the training budget. We describe this tradeoff between diversity and repetition with a scaling law for limited data. We validate it across dictionary sizes and token budgets, early, middle, and late layers of Llama-3.1, Qwen3 model sizes, and BatchTopK, TopK, and JumpReLU SAEs. Sweeps using mathematical text and Python code test the effect of changing the activation source. Evaluation covers explained variance (EV), reconstruction MSE, feature absorption, automated interpretability, and sparse probing. These results give a practical training rule: choose the smallest buffer that reaches the quality plateau and replay it for the remaining budget. In our experiments, this reduces activation storage by 8–64$\times$ with negligible quality loss.

1 Introduction

Sparse autoencoders (SAEs) decompose neural representations into interpretable features [Cunningham et al., 2024, Gao et al., 2025, Lieberum et al., 2024]. Training them at scale can require several terabytes of stored activations. Generating activations once lets different dictionary sizes, sparsities, and architectures share the cost of forward passes through the base model. Storage still grows with the training budget if every SAE update uses fresh activations. We separate the total number of activation presentations to the optimizer from the number of *unique* activations in the buffer, and ask how large that buffer must be before replay can replace further fresh samples.

To learn a feature's direction and threshold, the autoencoder must encounter that feature in its training activations. If samples were independent, a feature with occurrence probability p would be absent from a buffer of size U with probability $(1-p)^U$. Replay gives the optimizer more opportunities to fit features already represented in the buffer. It cannot introduce an absent feature. The number of unique activations measures distributional coverage, while total presentations measure optimization effort. Contextual activations are correlated, so this argument motivates our controlled sweep without providing a guarantee on sample complexity. Figure 1 shows the two training schedules.

We vary buffer diversity across dictionary sizes and budgets while holding total optimization tokens fixed within each sweep. Reconstruction improves rapidly as small buffers grow and then plateaus. A scaling law using effective activation counts, adapted from Muennighoff et al. [2023], fits the 96-run grid with RMSE 4.8×10^{-3} EV. The Akaike Information Criterion (AIC) favors a version whose repetition scale depends on dictionary size. The fitted law also interpolates held-out budgets and predicts held-out widths when the dictionaries are sufficiently trained.

Saturation also appears across Llama layers and Qwen model sizes, with different SAE gates and with activations from mathematical text and Python code. SAEBench evaluations show strong AutoInterp

[*]Work done while at T-Tech.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

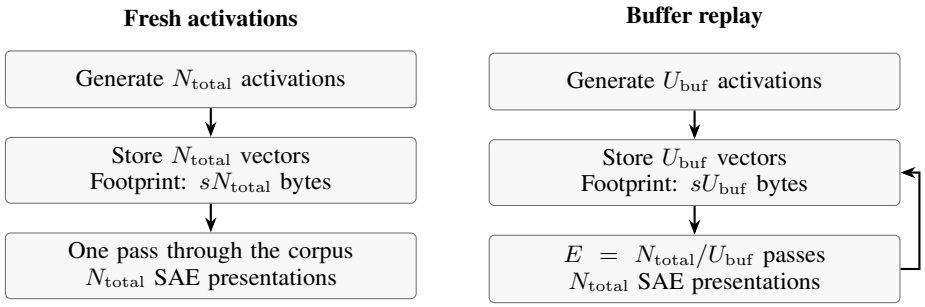

Figure 1: Fresh activations and replay at a fixed SAE optimization budget. The replay pipeline generates and stores U_{buf} activations once and uses $E = N_{\text{total}}/U_{\text{buf}}$ passes. s denotes bytes per activation. Generation counts refer to token activations. Batching and sequence packing determine the number of calls to the base model.

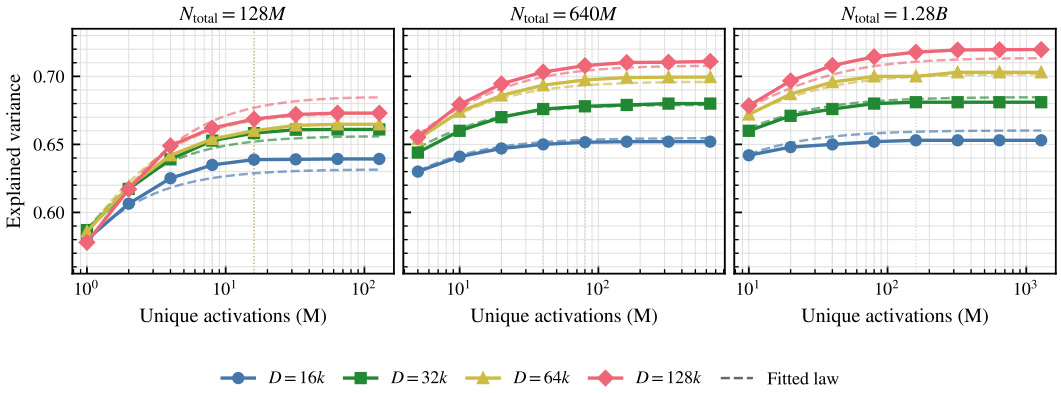

Figure 2: Explained variance as a function of unique activation count under fixed total SAE training budgets. With total optimization tokens fixed in each panel, moving right increases activation diversity and decreases the number of epochs over the buffer. EV improves sharply when activation diversity is low and then saturates.

and reconstruction performance with the buffer chosen from the EV plateau. Feature absorption and sparse probing provide further checks on the consequences of replay. We use the EV plateau to select a buffer and replay it for the remaining training budget. The resulting buffer sizes reduce activation storage by 8–64$\times$ in our grid. Tolerance tests quantify how these storage savings change with the target EV gap.

2 Related Work

Sparse autoencoders for mechanistic interpretability. Cunningham et al. [2024] trained sparse autoencoders to decompose language model activations into sparse, interpretable features. Gao et al. [2025] studied SAE scaling across sparsities, data budgets, and layers. Templeton et al. [2024] trained SAEs on Claude 3 Sonnet. Lieberum et al. [2024] released Gemma Scope, a suite of open SAEs spanning layers and dictionary widths in Gemma 2. Bills et al. [2023] and Paulo et al. [2025] study automated feature interpretation, while SAEBench [Karvonen et al., 2025] evaluates SAE quality on downstream tasks relevant to interpretability. We study unique activations and total training presentations as separate variables. BatchTopK [Bussmann et al., 2024] and JumpReLU [Rajamanoharan et al., 2024] change the tradeoff between sparsity and reconstruction. Chanin and Garriga-Alonso [2026] discuss how inappropriate L_0 can mix features. Our law therefore holds sparsity fixed, and the JumpReLU comparison tests saturation without establishing optimal sparsity or an architecture ranking at matched L_0.

Table 1: Saturation intervals for each dictionary size and total training budget ($\epsilon = 0.005$).

Budget	Dict. size	Plateau EV	Saturation interval	Storage reduction
128M	16k	0.6393	8M–16M	8–16×
128M	32k	0.6610	16M–32M	4–8×
128M	64k	0.6648	16M–32M	4–8×
128M	128k	0.6730	16M–32M	4–8×
640M	16k	0.6522	20M–40M	16–32×
640M	32k	0.6803	40M–80M	8–16×
640M	64k	0.6995	80M–160M	4–8×
640M	128k	0.7110	80M–160M	4–8×
1.28B	16k	0.6530	20M–40M	32–64×
1.28B	32k	0.6810	40M–80M	16–32×
1.28B	64k	0.7030	80M–160M	8–16×
1.28B	128k	0.7197	160M–320M	4–8×

Scaling laws for limited data and compute. Hoffmann et al. [2022] derived how to allocate compute between language model size and training tokens (Chinchilla). Muennighoff et al. [2023] extended this framework to finite text corpora, where training must reuse data. Their effective token count describes the diminishing returns from repetition and supports scaling laws when data are limited. Other scaling studies distinguish the capacity of routed models from compute [Clark et al., 2022] and examine data quality and training strategy [Chen et al., 2025]. We adapt this framework of effective tokens to SAE training to model the joint effects of activation diversity and dictionary width.

3 Setup

Base models and activations. The main experiments use residual stream activations from layer 15 of Llama-3.1 8B. We sample sequences from FineWeb [Penedo et al., 2024] and construct activation replay buffers at a fixed layer. We also validate on layers 5 and 25 of Llama-3.1 8B and on comparable layers near the middle of Qwen3-1.7B, Qwen3-4B, and Qwen3-8B.

SAE architecture. The main grid trains BatchTopK sparse autoencoders [Bussmann et al., 2024] with target average sparsity $k = 32$ and dictionary sizes $D \in \{16\text{k}, 32\text{k}, 64\text{k}, 128\text{k}\}$. We test whether the observed saturation pattern transfers beyond this setting using BatchTopK for validation across layers, standard TopK [Gao et al., 2025] for Qwen validation, and comparisons with JumpReLU [Rajamanoharan et al., 2024] at $D = 32\text{k}$.

Diversity–repetition grid. We train with total token budgets $N_{\text{total}} \in \{128\text{M}, 640\text{M}, 1.28\text{B}\}$. For each budget, we vary the number of unique activations U by sweeping

$$U/N_{\text{total}} \in \{1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1\}.$$

Thus each curve holds total optimization tokens fixed while moving from training with fresh activations throughout ($U = N_{\text{total}}$) to 128 replay epochs over a small buffer ($U = N_{\text{total}}/128$). This isolates activation diversity from total optimization compute.

Evaluation. We evaluate each SAE on held-out activations excluded from every training buffer and use explained variance (EV) as the primary quality metric. The lower endpoint of a saturation interval is the smallest tested buffer whose EV is within $\epsilon = 0.005$ of the maximum observed EV for that configuration. The upper endpoint is the next larger grid point. These two tested sizes describe the plateau range without estimating uncertainty or locating the threshold between measurements. We also evaluate downstream SAEBench metrics in Section 4.6.

Training hyperparameters. Unless otherwise stated, all SAEs use the same optimizer, schedule, batch size, and activation normalization. The held-out EV set contains 10^6 FineWeb tokens with no overlap with any training buffer. Full training hyperparameters are given in Appendix A.

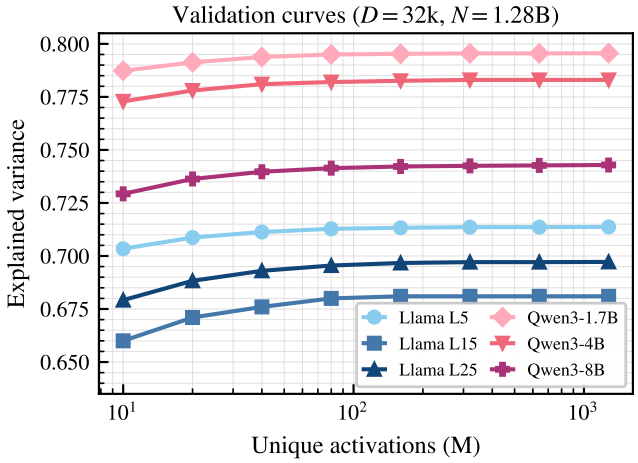

Figure 3: Validation across layers and models ($D = 32$k, $N = 1.28$B). Llama-3.1 8B layer sweeps use BatchTopK SAEs, and sweeps across Qwen3 model sizes use TopK SAEs. All curves show the same saturation pattern as diversity increases despite changes in layer, model size, and SAE activation function.

4 Results: The Diversity–Repetition Tradeoff

We first study how explained variance depends on the number of unique activations when the total SAE training budget is held fixed. For each total budget N_{total}, we vary the unique activation count N_{unique} over a logarithmic grid and train for

$$E = \frac{N_{\text{total}}}{N_{\text{unique}}} \tag{1}$$

epochs. Thus, all points within a curve use the same number of total SAE optimization tokens, but differ in how much activation diversity is available.

4.1 Unique Activations Improve EV Under a Fixed Token Budget

At a fixed total token budget, reconstruction improves substantially as the number of unique activations increases from the smallest buffers (Figure 2). Training for many epochs over a very small buffer therefore gives worse reconstruction than spending the same budget on fewer epochs over a larger buffer.

For the budget of 128M tokens, the effect is large across all dictionary sizes. For example, increasing the unique activation count from 1M to the full 128M budget improves EV from 0.5801 to 0.6393 for the 16k dictionary, and from 0.5870 to 0.6610 for the 32k dictionary. The 64k dictionary shows the same pattern, improving from 0.5860 at 1M unique activations to 0.6648 at 128M unique activations.

The curves at 1.28B tokens have higher absolute EV, but small buffers still limit reconstruction. Even at this larger budget, additional replay cannot fully compensate for insufficient activation coverage.

4.2 Explained Variance Saturates in Unique Activation Count

Across all dictionary sizes, EV rises rapidly as small buffers grow and then reaches a plateau. Once the buffer reaches this plateau, additional unique activations provide negligible reconstruction benefit. Training can therefore reuse a sufficiently diverse buffer without storing a fresh activation for every optimization token.

We define saturation as reaching EV within $\epsilon = 0.005$ of the maximum observed EV on the corresponding curve. Because the tested buffer sizes form a logarithmic grid, we report the saturation threshold as an interval between adjacent grid points: the smallest observed buffer size on the plateau and the next larger grid point.

Table 2: Transfer across activation functions at $D = 32$k, $N_{\text{total}} = 640$M, Llama-3.1 8B layer 15. BatchTopK and TopK are matched at $k = 32$. JumpReLU converges to $L_0 \approx 100$ and is therefore less sparse. All three variants saturate at 40M unique activations, showing that the diversity–repetition tradeoff is not specific to BatchTopK.

Activation	Sparsity	Plateau EV	u_{sat}	Storage reduction
BatchTopK	average $k = 32$	0.6803	40M	16×
TopK	$k = 32$	0.6807	40M	16×
JumpReLU	$L_0 \approx 100$	0.7634	40M	16×

Two patterns emerge from Table 1. First, at the largest budget the saturation interval scales approximately linearly with dictionary size: 20M–40M for 16k, 40M–80M for 32k, 80M–160M for 64k, and 160M–320M for 128k. Second, once the total budget comfortably exceeds the saturation interval for that dictionary, the interval depends only weakly on N_{total}. For example, the 16k, 32k, and 64k dictionaries have the same saturation intervals at 640M and 1.28B. The implied storage reduction therefore grows with total budget, reaching $\mathbf{64}\times$ for the 16k dictionary at 1.28B tokens.

4.3 Larger Dictionaries Reach Higher Plateaus but Require More Diversity

Larger dictionaries reach higher EV but require more unique activations to saturate. At a budget of 1.28B tokens, doubling D raises the best observed EV by roughly 2–3 EV points and moves the saturation interval upward by roughly $2\times$ (Table 1). The gain is smaller at lower budgets: the 128k dictionary reaches 0.711 EV at 640M tokens and 0.720 at 1.28B. This suggests that 0.711 reflects incomplete training rather than the dictionary's asymptotic performance. Total token count alone cannot distinguish dictionary capacity from the diversity needed to train it. Since $E = N_{\text{total}}/U$, the smallest buffers in Figure 2 also receive the most replay. Those extra passes do not make up for the missing diversity.

4.4 Validation across layers and models

We test other layers and models at $D = 32$k and $N_{\text{total}} = 1.28$B, using the same fractions of unique activations as the main grid. These sweeps test transfer of the saturation pattern at one width and budget. They do not repeat the full scaling grid.

Layer generalization. We isolate layer effects by training BatchTopK SAEs on Llama-3.1 8B residual stream activations from layers 5, 15, and 25. Each layer has a different plateau EV: 0.713 at layer 5, 0.681 at layer 15, and 0.697 at layer 25. Despite these differences in representation structure, EV improves with activation diversity and then plateaus at all three layers.

Generalization across model sizes. We train SAEs on Qwen3-1.7B, Qwen3-4B, and Qwen3-8B activations from comparable layers near the middle of each model. These runs use standard TopK to reduce the cost of the validation sweep. Both the base model and SAE activation function therefore differ from the main grid, so this comparison tests their combined effect. Plateau EV ranges from 0.743 for Qwen3-8B to 0.796 for Qwen3-1.7B. All three model sizes show saturation as activation diversity increases.

Figure 3 shows all six curves: early, middle, and late Llama layers with BatchTopK, and three Qwen model sizes with TopK.

4.5 Transfer across activation functions

We compare activation functions on Llama-3.1 8B layer 15 at $D = 32$k and $N_{\text{total}} = 640$M. BatchTopK, standard TopK [Gao et al., 2025], and JumpReLU [Rajamanoharan et al., 2024] use the same grid of buffer sizes, $u \in \{5, 10, 20, 40, 80, 160, 320, 640\}$M. BatchTopK targets $k = 32$ on average, while TopK uses $k = 32$ active features per token. JumpReLU learns a threshold and converges to an average $L_0 \approx 100$. Its lower sparsity prevents a direct comparison of absolute EV with the two TopK variants.

Saturation pattern. All three activation functions improve rapidly with the number of unique activations and reach saturation at 40M unique activations (Table 2). Their different gates and plateau

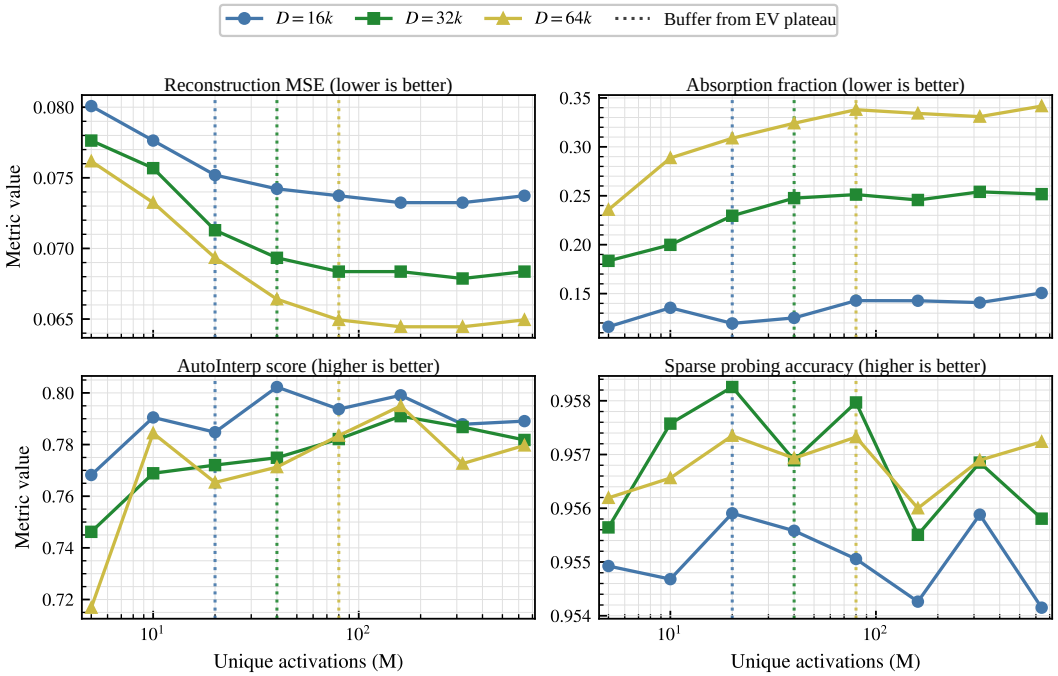

Figure 4: SAEBench metrics versus the number of unique activations for the BatchTopK sweep at layer 15 with a budget of 640M tokens. Dotted vertical lines mark the buffer chosen from the EV plateau for each dictionary. Reconstruction MSE follows the EV plateau, and AutoInterp remains high beyond this buffer scale. Feature absorption measures disentanglement. Sparse probing changes little with buffer size. None of the evaluated downstream metrics drops sharply at the recommended buffer size. Lower MSE and absorption fraction are better. Higher AutoInterp and sparse probing accuracy are better.

EV values leave the saturation pattern unchanged at $D = 32$k. The tradeoff between diversity and repetition therefore extends beyond BatchTopK in this setting.

Plateau levels. At matched $k = 32$, TopK reaches 0.681 EV and BatchTopK reaches 0.680. JumpReLU reaches 0.763 with many more active latents per token ($L_0 \approx 100$). That difference in sparsity prevents attributing the higher EV to the gate alone. The experiment establishes transfer of the saturation pattern without ranking the gates at a common sparsity.

4.6 Downstream Interpretability Tracks EV Saturation

To check the downstream effects of choosing a buffer by EV, we evaluate 24 checkpoints from the BatchTopK sweep at layer 15 with a budget of 640M tokens on SAEBench. There are eight buffer sizes for each of $D \in \{16\text{k}, 32\text{k}, 64\text{k}\}$. Dotted vertical lines in Figure 4 mark the buffer chosen using EV for each dictionary, allowing comparison with the reconstruction and interpretability metrics.

Comparison to the EV buffer scale. Reconstruction MSE improves rapidly at low diversity and is near its best value at the buffer chosen using EV. EV rescales reconstruction error on a fixed evaluation set, so MSE checks consistency with the reconstruction result. Automated interpretability remains high across the sweep, with only modest further changes as the buffer grows beyond the EV recommendation. Table 7 reports thresholds for each metric, including AutoInterp at 40M, 160M, and 160M for the three widths. Sparse probing accuracy is nearly flat. Absorption fraction, for which lower values are better, does not improve monotonically with diversity: the 32k and 64k curves rise. Reconstruction and disentanglement therefore require separate evaluation.

Implications. AutoInterp and the other downstream metrics support using EV saturation to choose the buffer, even when their optima occur elsewhere on the grid. Both absolute and relative changes with buffers larger than the recommendation are small. At this scale, reconstruction quality is preserved,

Table 3: Additional domain sweeps: held-out EV at each tested buffer size, with 640M total presentations. The reference is each domain's 640M run without replay.

Domain	10M	20M	40M	80M	640M	$U_{0.005}$ / reduction
FineMath-3plus	0.6312	0.6523	0.6608	0.6649	0.6673	80M / 8×
Stack-Edu Python	0.7013	0.7081	0.7119	0.7143	0.7155	40M / 16×

Table 4: First qualifying tested buffer / storage reduction at three absolute EV tolerances, relative to the domain's 640M run.

Domain	$\epsilon = 0.0025$	$\epsilon = 0.005$	$\epsilon = 0.010$
FineMath-3plus	80M / 8×	80M / 8×	40M / 16×
Stack-Edu Python	80M / 8×	40M / 16×	20M / 32×

feature absorption and automated interpretability stay in the same narrow performance band, and sparse probing is largely unaffected.

4.7 Math and code

We change the activation source while keeping Llama-3.1 8B layer 15, BatchTopK, $D = 32$k, $k = 32$, and 640M total presentations fixed. Evaluation uses held-out activations from the same domain. FineMath first meets $\epsilon = 0.005$ at 80M unique activations, relative to its run without replay. Stack-Edu Python meets the same tolerance at 40M (Table 3). The corresponding storage reductions are $8\times$ and $16\times$. Saturation thus persists in both domains, although the tested counts cannot locate a continuous threshold or establish transfer of the FineWeb coefficients.

4.8 Tolerance sensitivity

Table 4 reports the first qualifying buffers at half and twice the default EV tolerance. Tightening ϵ to 0.0025 doubles the Python buffer and halves its storage savings. Loosening it to 0.010 halves the buffer and doubles the savings. FineMath selects the same buffer under the stricter tolerance because of the spacing between tested counts. The 640M sweeps confirm the default recommendations and test sensitivity across widths (Appendix E). A buffer recommendation therefore depends on its reference EV, tested grid, and tolerance.

4.9 Effective Activation Count

When a replay buffer contains U unique activations and is used for N_{total} training tokens, each activation is replayed on average N_{total}/U times. We define the repetition count as

$$R = \frac{N_{\text{total}}}{U} - 1. \tag{2}$$

Following Muennighoff et al. [2023], we convert (U, N_{total}) into an effective activation count

$$N_{\text{eff}}(U, N_{\text{total}}, R^*) = U \left[1 + R^* \left(1 - e^{-R/R^*}\right)\right], \tag{3}$$

where $R^* > 0$ is the fitted characteristic repetition scale. We obtain this form by assuming that the marginal contribution of replay decays as e^{-r/R^*} and integrating from $r = 0$ to R. The effective count is a phenomenological discount on repeated data and does not measure the number of features. When $R \ll R^*$, replay is nearly as useful as fresh data and $N_{\text{eff}} \approx N_{\text{total}}$. When $R \gg R^*$, further replay adds little, and N_{eff} is limited to a constant multiple of U.

4.10 Joint Scaling Law

We model explained variance as

$$\text{EV}(D, N_{\text{eff}}) = E_\infty - AD^{-\alpha} - BN_{\text{eff}}^{-\beta}. \tag{4}$$

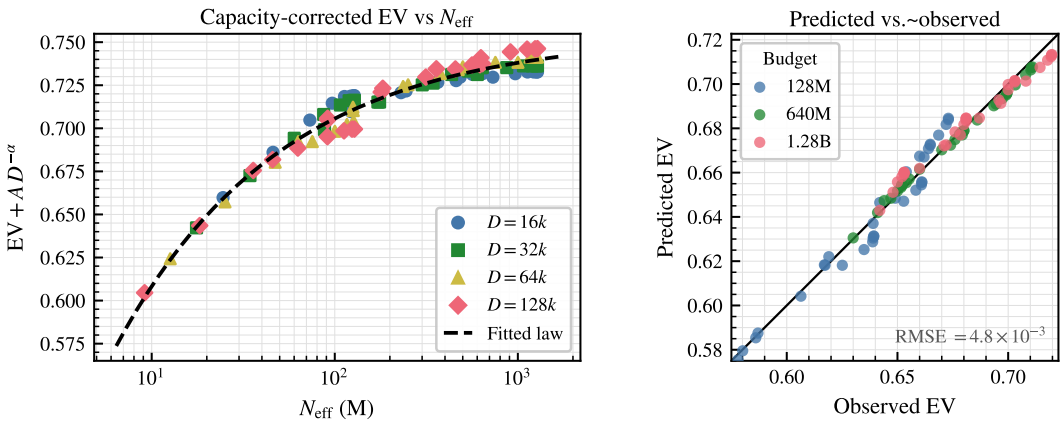

Figure 5: Capacity-corrected EV ($\text{EV} + AD^{-\alpha}$) versus effective activation count N_{eff} under the extended $R^*(D)$ model. Data points from all four dictionary sizes and three total budgets collapse onto a single power-law curve, supporting the functional form within the fitted regime.

The extrapolated ceiling E_∞ depends on the model, layer, normalization, objective, and sparsity. Its fitted value of 0.760 exceeds the best EV observed on the main grid, 0.7197. The first penalty describes finite dictionary capacity and decreases as the dictionary grows. The second describes limited effective data. At a fixed training budget, larger buffers require fewer repetitions and reduce this penalty.

For larger dictionaries, activation diversity appears to become limiting after fewer replay passes. We model this with a repetition scale R^* that depends on dictionary size:

$$R^*(D) = cD^{-\gamma}. \tag{5}$$

The coefficient c sets the repetition scale, and γ determines how it changes with width. With the fitted $\gamma = 0.526$, doubling the width multiplies R^* by $2^{-0.526} \approx 0.69$. Larger dictionaries therefore tolerate fewer replay passes before diversity becomes limiting.

4.11 Fitting and Model Comparison

We fit to the 96 observations in the main grid: three total budgets, four dictionary sizes, and eight fractions of unique activations for Llama-3.1 8B layer 15 BatchTopK SAEs. We compare the models with constant R^* and with $R^*(D)$ using the Akaike Information Criterion (AIC). Appendix B describes the fitting procedure.

Allowing the repetition scale to depend on dictionary size reduces the fit error to $\text{RMSE} = 4.8 \times 10^{-3}$ EV and improves AIC. This comparison supports a dependence between dictionary size and the benefit of activation diversity. Appendix B gives the full fitted parameters.

After removing the fitted capacity term, the 96 data points follow a common power-law curve: Figure 5 (left) plots $\text{EV} + AD^{-\alpha}$ against $N_{\text{eff}}(D)$. The right panel compares predicted and observed EV, with $\text{RMSE} = 4.8 \times 10^{-3}$.

4.12 Validation

The common curve in Figure 5 describes the fitted grid as a function of N_{eff}. We examine the residuals within individual panels and test predictions for budgets, widths, layers, and models excluded from the fit.

Panel residuals and buffer decisions. Fit errors are largest at the smallest budget and at the extreme widths of the largest budget (Table 5). Undertraining and the smooth law's approximation to steep curves at low diversity may contribute to these residuals. Fitted and observed plateau intervals coincide in six of the twelve configurations. In the remaining six, the fitted interval is one grid step larger, leading to a larger buffer recommendation. This conservative direction of error on the measured grid need not hold in new settings.

Table 5: RMSE for each panel used in the fit ($\times 10^{-3}$ EV), with eight buffer sizes per cell. Residuals concentrate at the smallest budget and extreme widths at the largest budget.

Budget	16k	32k	64k	128k
128M	7.7	5.0	6.1	8.1
640M	1.8	1.6	2.9	2.4
1.28B	5.9	2.5	1.6	5.2

Held-out training budget. Fitting the law on the two extreme budgets, 128M and 1.28B tokens, gives predictions for the intermediate 640M budget with test RMSE 3.1×10^{-3} EV. Predicted saturation intervals agree with the observed intervals to within one grid step.

Held-out dictionary size. We fit on the three smaller dictionaries and predict EV for the held-out 128k dictionary. Extrapolation succeeds at 1.28B tokens, where that dictionary is sufficiently trained, but becomes less accurate at smaller budgets. In the undertrained regime, its best observed EV has not reached the asymptotic plateau. A fit to smaller dictionaries that have received sufficient training does not capture this regime.

Shape transfer. For the curves from other layers and models, we keep the nonlinear shape fixed and refit the level parameters. The validation fits have RMSE at most 0.9×10^{-3} EV, supporting reuse of the curve shape after this recalibration. Because all these checks use one dictionary width, they cannot establish transfer of width exponents or unchanged coefficients to new domains.

5 Practical Implications: Activation Buffer Sizing

The saturation intervals determine how much activation storage a training run needs. A buffer that reaches the quality plateau can be replayed for the remaining budget, avoiding the storage of one fresh activation per training token.

For dictionary size D and total SAE training budget N_{total}, choose U_{buf} from the corresponding saturation interval. Table 8 uses each interval's lower endpoint: the smallest tested buffer whose EV is within $\epsilon = 0.005$ of the best value on the curve. The number of replay epochs is

$$E = \frac{N_{\text{total}}}{U_{\text{buf}}}, \tag{6}$$

and the storage reduction relative to storing all activations is

$$\text{StorageReduction} = \frac{N_{\text{total}}}{U_{\text{buf}}}. \tag{7}$$

Every configuration in our grid permits at least an $8\times$ reduction in activation storage. Smaller dictionaries at the largest budget reach $64\times$ (Appendix D), reducing several terabytes of activations to a few hundred gigabytes. The 128k dictionary at 1.28B tokens still needs a 160M buffer (1.19 TiB), an $8\times$ reduction from the 9.54 TiB full corpus. A Llama-3.1 8B residual stream in bf16 takes 8 KiB per activation. At $N_{\text{total}} = 1.28\text{B}$ and $D = 16\text{k}$, the full corpus requires $9.54\,\text{TiB}$ of storage, compared with $152.6\,\text{GiB}$ for the recommended buffer.

Storage savings grow with total budget once the dictionary has enough activation diversity to saturate. Smaller dictionaries reach this point early, so further increases in N_{total} mainly add replay. Larger dictionaries at small budgets can remain limited by capacity or optimization. Their savings grow more slowly until the budget is sufficient to reach the full plateau.

Training recipe. For a chosen dictionary size and total budget, estimate the saturation interval with a pilot sweep or the fitted law. Set U_{buf} to the smallest buffer on the plateau and train for $N_{\text{total}}/U_{\text{buf}}$ replay epochs.

Stored buffers and fresh streaming. When storage permits caching, generating U_{buf} once and replaying it reduces the work of generating activations relative to generating fresh activations on the fly. SAE optimization still uses N_{total} presentations. For generation and SAE costs per token C_{gen} and C_{SAE}, the idealized compute ratio is

$$\frac{N_{\mathrm{total}}(C_{\mathrm{gen}} + C_{\mathrm{SAE}})}{U_{\mathrm{buf}}C_{\mathrm{gen}} + N_{\mathrm{total}}C_{\mathrm{SAE}}}. \tag{8}$$

This ratio approaches the storage ratio when activation generation dominates the cost. We do not measure end-to-end runtime. Recomputing activations on every replay pass preserves the diversity schedule but requires the same number of generation forwards. Fresh streaming remains useful when storage is constrained or the distribution changes. The storage ratio equals the number of passes: both are $N_{\mathrm{total}}/U_{\mathrm{buf}}$.

6 Conclusion

At a fixed SAE training budget, activation diversity limits reconstruction until the buffer reaches a quality plateau. Replay beyond that point preserves reconstruction and downstream interpretability performance. We observe this behavior in the 96-run scaling grid and validate it on held-out settings, other layers and models, and different SAE gates. The experiments on mathematical text and Python code extend the domain coverage. Choosing the smallest tested buffer within the EV tolerance yields 8–$64\times$ savings in activation storage.

Limitations and future work. The full grid uses FineWeb and Llama-3.1 8B at fixed sparsity. The domain sweeps test qualitative transfer, while the layer and model sweeps test curve shape after recalibration. These checks leave the generality of the coefficients unresolved. Extrapolation is less accurate for undertrained wider dictionaries, and stricter EV tolerances can require larger buffers. Gate comparisons cover one width. SAEBench covers the BatchTopK sweep at layer 15. We have not tested objectives across tokens, broader modalities, or end-to-end runtime. Future work could examine adaptive buffer refreshing and fits that account for sparsity.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Vaibhav Srivastav, Joshua Lochner, Clémentine Fourrier, Hugo Larcher, Ben Burtenshaw, Haojun Zhao, Caleb Fahlgren, Mathieu Morlon, Agustín Piqueres Lajarín, Xuan-Son Nguyen, Cyril Zakka, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big – data-centric training of a fully open small language model. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=3JiCl2A14H.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024. URL https://arxiv.org/abs/2412.06410.

David Chanin and Adrià Garriga-Alonso. Sparse but wrong: Incorrect L_0 leads to incorrect features in sparse autoencoders. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 12889–12915. PMLR, 2026. URL https://proceedings.mlr.press/v306/chanin26a.html.

Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. Revisiting scaling laws for language models: The role of data quality and training strategies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23881–23899. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1163. URL https://aclanthology.org/2025.acl-long.1163/.

Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George van den Driessche, Eliza Rutherford, Tom Hennigan, Matthew Johnson, Katie Millican, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 4057–4086. PMLR, 2022. URL https://proceedings.mlr.press/v162/clark22a.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/42ef3308c230942d223c411adf182c88-Paper-Conference.pdf.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 29223–29264. PMLR, 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

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. URL https://arxiv.org/abs/2408.05147.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9d89448b63ce1e2e8dc7af72c984c196-Paper-Conference.pdf.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 48393–48421. PMLR, 2025. URL https://proceedings.mlr.press/v267/paulo25a.html.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

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*, 2024. URL https://arxiv.org/abs/2407.14435.

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, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, 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.

A Additional Experimental Details

Training hyperparameters. Unless otherwise stated, SAEs are trained with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay 0.05), peak learning rate 3×10^{-4}, cosine decay to 10^{-6}, and a 10% linear warmup. Batch size is 4096 activations and gradients are clipped to unit norm. For TopK variants, k is linearly annealed from 128 to the target value over the first 10% of training. BatchTopK runs use an auxiliary penalty for dead latents of $1/32$ and straight-through estimator bandwidth 0.001. We normalize all activations to unit norm before passing them to the SAE.

B Scaling Law Fitting Details

With the nonlinear parameters fixed, Eq. 4 is linear in (E_∞, A, B). At each candidate set of nonlinear parameters, we solve for the linear parameters by ordinary least squares, then refine the best solution with local coordinate descent. The nonlinear parameters are (α, β, R^*) for the baseline with constant R^* and $(\alpha, \beta, c, \gamma)$ for the model whose repetition scale varies with dictionary size. We fit both to the 96-point main grid and compare them using the Akaike Information Criterion (AIC).

Table 6: Fitted parameters for the model with constant R^* and the extended $R^*(D)$ model, fit on all 96 data points from the main grid. The extended model is decisively preferred by AIC.

Model	E_∞	A	α	B	β	R^* / c	γ	ΔAIC
Const. R^*	0.738	0.387	0.688	0.432	0.466	13.7	—	0
Extended $R^*(D)$	0.760	0.328	0.486	0.440	0.477	103.3	0.526	-77.3

C SAEBench Threshold Table

For each SAEBench metric, Table 7 gives the smallest buffer whose score is within 5% of the metric's observed range from the best value on that curve. These thresholds describe the individual metrics. The main rule for sizing the buffer uses the saturation intervals determined by EV.

Table 7: SAEBench saturation thresholds at $N_{\text{total}} = 640\text{M}$, Llama-3.1 8B layer 15, BatchTopK. The EV row is reproduced from Table 1.

Quantity	$D = 16\text{k}$	$D = 32\text{k}$	$D = 64\text{k}$
Explained variance (main)	20M	40M	80M
Reconstruction MSE	80M	80M	80M
Feature absorption fraction	80M	40M	80M
Automated interpretability score	40M	160M	160M
SAE sparse probing accuracy	20M	20M	20M

D Storage Reduction Recommendations

Table 8: Recommended activation buffer sizes and implied storage reductions ($\epsilon = 0.005$). Epochs $= N_{\text{total}}/u_{\text{sat}}$.

Budget	Dict. size	u_{sat}	Epochs	Storage reduction
128M	16k	8M	16	16×
128M	32k	16M	8	8×
128M	64k	16M	8	8×
128M	128k	16M	8	8×
640M	16k	20M	32	32×
640M	32k	40M	16	16×
640M	64k	80M	8	8×
640M	128k	80M	8	8×
1.28B	16k	20M	64	64×
1.28B	32k	40M	32	32×
1.28B	64k	80M	16	16×
1.28B	128k	160M	8	8×

E Additional Results on Threshold Sensitivity

The domain sweeps use `HuggingFaceTB/finemath` with configuration `finemath-3plus`, and `HuggingFaceTB/stack-edu` with configuration `Python`. The datasets are described by Ben Allal et al. [2025]. The sweeps use Llama-3.1 8B layer 15, BatchTopK, $D = 32$k, $k = 32$, and 640M total presentations. Held-out evaluation uses activations from the same domain, and the tolerance reference is that domain's 640M run without replay.

The comparison across dictionary widths uses a budget of 640M total presentations and the best final EV on each sweep as its reference. Stricter tolerances can require larger buffers and reduce the storage savings. The default recommendations agree with those in the main saturation table.

Table 9: Buffer recommendations at 640M total presentations for three EV tolerances. The reference is the best final EV on each sweep over eight buffer sizes. Each cell gives the smallest qualifying buffer and the resulting storage reduction.

Width	$\epsilon = 0.0025$	$\epsilon = 0.005$	$\epsilon = 0.010$
16k	40M / 16×	20M / 32×	20M / 32×
32k	80M / 8×	40M / 16×	20M / 32×
64k	80M / 8×	80M / 8×	40M / 16×
128k	160M / 4×	80M / 8×	40M / 16×

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the observed saturation, the settings tested, the fitted law, and the storage savings. The limitations paragraph states where these findings may not generalize.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations paragraph discussing the main grid, targeted domain transfer, fixed sparsity, prediction limits, tolerance sensitivity, and limited downstream scope.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper proposes and fits a phenomenological scaling law but does not present formal theoretical results, theorems, or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The setup and appendices describe the experimental design, hyperparameters, held-out FineWeb size, domain sweeps, and threshold sensitivity. Complete training, evaluation, and fitting code and full split and run metadata are not provided. Appendix E reports additional threshold sensitivity results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The supplemental files include reported results, EV measurements, and scripts for checking summary calculations. A complete implementation for training, evaluation, and fitting is not included.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: The models, datasets, layer-15 setting, budgets, grid, optimizer, normalization, and the FineWeb evaluation set of one million tokens are described. The sizes of domain evaluation sets, detailed split construction, model/dataset revisions, and evaluator settings are not fully specified. Appendix E reports additional threshold sensitivity results.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The main grid uses single training runs. We do not report error bars or confidence intervals across random seeds.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper describes the training scale and requirements for activation storage, but does not provide a complete accounting of GPU type, memory, wall-clock time, or total compute for each experiment.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is an empirical study of SAE training efficiency using publicly available models and text/code datasets, and we are not aware of any deviation from the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is foundational research on reducing costs of activation storage for sparse autoencoder training and does not introduce a deployed system or direct application pathway. Its main expected impact is improved accessibility of interpretability experiments through lower storage costs.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-risk pretrained model, image generator, or scraped dataset requiring special safeguards.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the main existing assets used, including the named base models, FineWeb, FineMath, Stack-Edu, SAEBench, and prior SAE methods, but it does not explicitly enumerate all asset licenses and terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The supplemental files document the reported results and scripts for summary calculations and figure annotations. They contain no new training dataset, benchmark, or model.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The evaluation of automated interpretability uses an existing benchmark method that relies on an LLM. It does not introduce an original LLM component.