Perplexity Can Miss SAE Feature Damage Under Quantization

Anonymous authors
Paper under double-blind review

Abstract

Quantization is a standard path to deploying large language models, and quantized models are typically judged acceptable when perplexity or downstream accuracy remains close to the full-precision original. But behavioral parity need not imply feature fidelity: the sparse-autoencoder (SAE) features used to interpret a full-precision model may change after weight rounding. We test this directly by using a frozen SAE as a fixed measurement basis, encoding full-precision and round-to-nearest (RTN) quantized activations on identical tokens, and measuring per-feature survival by Pearson correlation across bit-widths from INT8 to INT4 on Pythia-70M and Gemma-2-2B. Our central finding is that perplexity can miss feature damage: on Gemma-2-2B, INT7 improves perplexity while degrading 18.7% of active SAE features, and under sliding-window evaluation INT6 also improves perplexity while only 51.3% of active features survive. Feature survival is graded rather than cliff-like, with 62.4% of active Pythia features and 51.3% of active Gemma features surviving at INT6; most non-surviving features are blurred rather than fully damaged. Survival is also predictable from full-precision feature statistics alone, with cross-validated AUC 0.92–0.97 and peak activation as the strongest marginal predictor. Finally, RTN quantization and matched-perplexity magnitude pruning damage strongly overlapping feature sets, with Jaccard overlap 0.79–0.86 and damage-score Spearman correlation 0.98. These results show that behavioral metrics alone are insufficient evidence that full-precision interpretability findings transfer to quantized models, motivating feature-level audits of compression.

1 Introduction

Quantization is a standard path to deploying large language models at scale: rounding weights from 16-bit floating point to 8, 4, or fewer bits can reduce memory and latency while preserving task-level performance under standard evaluation metrics (Xiao et al., 2023; Frantar et al., 2022; Li et al., 2024). That accounting is almost always behavioral. A quantized model is typically judged acceptable if its perplexity or downstream benchmark accuracy remains close to that of the full-precision original. Whether the model still computes in the same way—whether the internal features identified by interpretability research in the full-precision model survive rounding—is rarely tested.

This question is increasingly consequential. Sparse autoencoders (SAEs) have become a standard tool for decomposing language-model activations into interpretable features (Cunningham et al., 2023), and a growing body of work builds analyses, safety audits, and steering interventions on top of features extracted from full-precision models (Chalnev et al., 2024; O'Brien et al., 2024; Bayat et al., 2025). If those models are then deployed in quantized form, the features used to reason about model behavior may no longer be the features the deployed model actually uses. The reliability of interpretability under compression is therefore a precondition for interpretability being useful in deployment, yet it remains largely uncharacterized for quantization.

Prior work has introduced SAE feature survival as an audit tool for model compression under pruning (Borobia et al., 2026). We use this instrument to ask a different deployment question: whether behavioral parity under quantization implies feature fidelity. Unlike pruning, quantization preserves all weights but lowers

their precision, making it possible for perplexity to remain stable while the feature geometry read out by a full-precision SAE changes. Across Pythia-70M and Gemma-2-2B, we sweep RTN from 8 to 4 bits, predict feature survival from full-precision statistics, compare against pruning, and relate feature fidelity to perplexity.

We make the following empirical contributions:

Perplexity can miss feature damage. On Gemma-2-2B, INT7 improves perplexity while degrading 18.7% of active SAE features. Under sliding-window evaluation, INT6 also improves perplexity while only 51.3% of active features survive, showing that task-level metrics can underestimate representational change.

Quantization and pruning damage similar features. At similar perplexity regimes, RTN quantization and magnitude pruning affect strongly overlapping feature sets, with Jaccard overlap 0.79–0.86 and damage-score Spearman correlation 0.98. This suggests a shared pattern of compression-induced feature vulnerability despite different compression mechanisms.

Feature survival is graded. As bit-width decreases, SAE features degrade systematically rather than failing all at once. At INT6, survival falls to 62.4% on Pythia-70M and 51.3% on Gemma-2-2B, with most non-surviving features degraded rather than fully damaged.

Feature survival is predictable. INT6 survival can be predicted from full-precision feature statistics with cross-validated AUC 0.92 on Pythia-70M and 0.97 on Gemma-2-2B. Peak activation is the strongest marginal predictor: high-peak features survive reliably, while weak-signal features are most vulnerable to rounding-induced perturbation.

Together, these results show that behavioral metrics alone are insufficient evidence that full-precision interpretability findings transfer to quantized deployments.

2 Related Work

2.1 Sparse autoencoders and monosemantic features

Sparse autoencoders (SAEs) decompose the dense activation vectors of a language model into a larger set of sparse, individually interpretable features, offering a practical response to the superposition hypothesis under which networks represent more features than they have dimensions (Elhage et al., 2022). Bricken et al. (2023) showed that SAEs trained on language-model activations recover monosemantic features, and Cunningham et al. (2023) demonstrated that the recovered features are highly interpretable.

Templeton et al. (2024) scaled the approach to a production model, and the Gemma Scope project (Lieberum et al., 2024) released pretrained residual-stream SAEs across the layers of the Gemma-2 family, providing the standardized dictionaries we use for our Gemma experiments. A growing body of work builds analyses, audits, and steering interventions on top of SAE features (Chalnev et al., 2024; O'Brien et al., 2024; Bayat et al., 2025), typically using features extracted from full-precision models.

The reliability of SAEs as a measurement instrument is itself an active question. Paulo & Belrose (2025) show that SAEs trained on identical data with different random seeds learn substantially different features, and Chanin & Garriga-Alonso (2025) show that feature recovery is sensitive to the sparsity hyperparameter. Recent methods aim to stabilize SAE training: Jedryszek & Crook (2026) add weight-regularization penalties that increase cross-seed feature sharedness and steering reliability, while other work encourages convergence across parallel or sequential SAE training runs (Marks et al., 2024; Martin-Linares & Ling, 2025). This line of work concerns variability introduced on the SAE-training side of the pipeline; it is complementary to our question, which concerns variability introduced on the model-compression side. In our experiments the SAE is held fixed and only the model weights change, so SAE-training variability is not the source of the feature changes we measure.

2.2 Compression of language models

Quantization and pruning are two dominant families of post-training compression. Quantization reduces numerical precision: round-to-nearest is the simplest scheme, while GPTQ and AWQ reduce quantization error more carefully (Frantar et al., 2022; Lin et al., 2024); mixed-precision and low-bit formats are now common in deployment (Dettmers et al., 2022; 2023). Pruning removes weights instead: magnitude pruning removes the smallest-magnitude weights (Han et al., 2015), while SparseGPT and Wanda provide scalable one-shot pruning methods for large language models (Frantar & Alistarh, 2023; Sun et al., 2024). Both compression families are typically evaluated by perplexity or downstream accuracy; we instead measure feature-level representational change under quantization and use pruning as a matched-perplexity baseline.

2.3 Interpretability under compression

The intersection of SAE-based interpretability and model compression remains sparsely explored. The closest prior work is Borobia et al. (2026), who study how unstructured pruning reshapes SAE features across multiple model families, pruning methods, and sparsity levels. They find that rare, low-firing features survive pruning better than frequent ones, that Wanda better preserves feature structure than magnitude pruning, and that geometric feature survival does not necessarily predict causal importance under ablation.

We build on this audit perspective but apply it to a question pruning cannot directly answer: whether a quantized model that remains behaviorally close to its full-precision original also preserves the SAE features used to interpret it. This changes the setting along three axes. First, we study quantization rather than pruning: weight rounding preserves all parameters but reduces their precision, while pruning removes weights entirely. Second, we use a frozen full-precision SAE on identical dense and compressed activations, measuring per-feature correlation directly rather than retraining and matching separate SAE dictionaries. Third, we vary bit-width from INT8 to INT4, tracing feature survival as a function of precision rather than sparsity.

3 Methodology

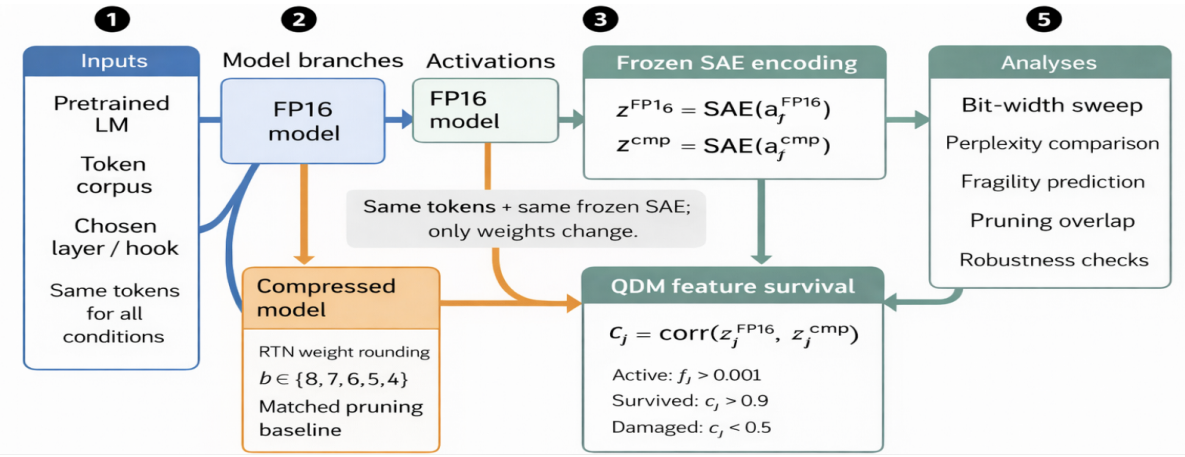


Figure 1: QDM pipeline: We encode FP16 and compressed activations with the same frozen SAE on identical tokens, measuring feature survival by per-feature correlation c_j.

We measure how fixed SAE features change when the underlying language model is compressed. Across conditions, we keep the SAE, token set, and read-out site fixed, varying only the model weights. This section describes the models and SAEs, compression operators, feature-stability metric, behavioral evaluation, pruning baseline, fragility predictor, and stability checks. A compact glossary of notation and threshold conventions is provided in Appendix J

3.1 Models, SAEs, and read-out site

We study two models spanning roughly a 30× parameter range: Pythia-70M-deduped (Biderman et al., 2023) and Gemma-2-2B (Gemma Team et al., 2024). For each model, we use a publicly released residual-stream SAE at a fixed read-out layer: the `pythia-70m-deduped-res-sm` SAE at `blocks.4.hook_resid_post` for Pythia-70M, and the Gemma Scope layer-12 residual-stream SAE with width $d_{\text{sae}} = 16384$ for Gemma-2-2B. Appendix K summarizes the model, SAE, token-budget, and evaluation configuration.

Let $h^{(l)}(t;\theta)$ denote the residual-stream activation of model f_θ at layer l and token position t. A fixed SAE encoder E maps this activation into feature space:

$$z(t;\theta) = E\left(h^{(l)}(t;\theta)\right), \tag{1}$$

where $z_j(t;\theta)$ is feature j's activation. Across all compression conditions, E, the token set, and the read-out layer are fixed; only the model parameters θ change.

3.2 Compression operators

We consider two families of weight compression. Both are applied to transformer-block linear weights: attention projections, MLP projections, and, for Gemma-2-2B's gated MLP, the gate projection. Layernorm and embedding parameters are left in full precision. Exact module-name patterns and exclusions are listed in Appendix L.

Round-to-nearest quantization (RTN). For each targeted tensor W, we apply per-output-channel RTN quantization. For bit-width b, with signed range $q_{\min} = -2^{b-1}$ and $q_{\max} = 2^{b-1} - 1$, each output channel is scaled by its maximum absolute weight and quantized as

$$\widehat{W}_{i,c} = s_c \ \text{clip}\left(\text{round}\left(\frac{W_{i,c}}{s_c}\right), q_{\min}, q_{\max}\right), \qquad s_c = \frac{\max_i |W_{i,c}|}{q_{\max}}. \tag{2}$$

We sweep $b \in \{8, 7, 6, 5, 4\}$ and dequantize weights back into floating-point tensors, simulating low-bit rounding without changing the model architecture.

Magnitude pruning. We zero the smallest-magnitude fraction p of weights in each targeted tensor and use pruning only as a matched-perplexity baseline, calibrated as described in Section 3.6.

3.3 Feature-stability metric

We quantify feature stability by comparing each SAE feature's activation pattern before and after compression. For feature j, let $x_t = z_j(t;\theta_{\text{FP16}})$ and $y_t = z_j(t;\theta_C)$ denote its full-precision and compressed activations at token position t over a shared token set of size N. We define the feature-stability score as the Pearson correlation

$$c_j = \frac{\sum_t (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_t (x_t - \bar{x})^2 \sum_t (y_t - \bar{y})^2}}, \tag{3}$$

where $\bar{x}$ and $\bar{y}$ are means over token positions. Since both activations are evaluated on identical token positions with the same frozen SAE, this comparison isolates the compression intervention from token-sampling variation. To avoid storing the full $N \times d_{\text{sae}}$ activation matrix, we compute the same correlation from streaming sufficient statistics; implementation details are provided in Appendix I.

We restrict survival statistics to features that are active in the full-precision model. Feature j is active if its FP16 firing rate

$$f_j = \frac{1}{N} \sum_t \mathbf{1}[z_j(t;\theta_{\text{FP16}}) > 0] \tag{4}$$

exceeds 0.001.

3.4 Survival taxonomy

We summarize the distribution of $\{c_j\}$ over active features with three bands, using a survival threshold $t_s = 0.9$ and a damage threshold $t_d = 0.5$:

$$\begin{array}{lll} \text{survived:} & c_j > t_s & \text{strongly aligned with the FP16 activation pattern,} \\ \text{degraded:} & t_d \le c_j \le t_s & \text{partially aligned,} \\ \text{damaged:} & c_j < t_d & \text{weakly aligned under the frozen SAE basis.} \end{array} \tag{5}$$

We use $t_s = 0.9$ and $t_d = 0.5$ as default thresholds and evaluate sensitivity to alternative cutoffs in Appendix B.

3.5 Behavioral evaluation

We report token-level perplexity on WikiText-2-raw using two protocols. In the main feature-extraction pipeline, we use a fixed chunked protocol: the token stream is split into non-overlapping blocks of length L ($L = 512$ for Pythia and $L = 256$ for Gemma), and mean autoregressive cross-entropy is computed over scored positions within each block. Because each block resets context, this protocol can inflate absolute perplexity, but it is held fixed across compression conditions and therefore supports within-protocol perplexity deltas.

For Gemma, we additionally run a sliding-window behavioral check with window size $W = 2048$ and stride $S = 512$. This evaluation uses a HuggingFace-format Gemma model with the same per-output-channel RTN scheme and is used only to test whether perplexity trends are robust to a stronger behavioral evaluation protocol. The SAE feature analysis itself uses the TransformerLens/Gemma Scope activation-extraction pipeline. Full sliding-window details are provided in Appendix D.

For any condition C, we report the relative perplexity change

$$\Delta_{\mathrm{PPL}}(C) = \frac{\mathrm{PPL}(C)}{\mathrm{PPL}(\mathrm{FP16})} - 1. \tag{6}$$

3.6 Matched-perplexity pruning baseline

To compare quantization and pruning at a similar behavioral cost, we calibrate magnitude-pruning sparsity to the RTN INT6 perplexity regime. Let $\mathrm{PPL}^\star = \mathrm{PPL}(\text{RTN INT6})$. We choose the pruning sparsity

$$p^\star = \arg\min_p \left| \mathrm{PPL}(\mathrm{prune}(\theta, p)) - \mathrm{PPL}^\star \right|. \tag{7}$$

The sparsity search and achieved perplexity matches are reported in Appendix F. The calibrated pruning condition is then analyzed with the same frozen-SAE feature pipeline as RTN INT6.

We compare RTN INT6 and pruning using two per-feature overlap measures: the Jaccard overlap of non-survived feature sets,

$$J = \frac{|A \cap B|}{|A \cup B|},$$

where A and B are the features with $c_j \le t_s$ under each method, and the Spearman correlation of damage scores $d_j = 1 - c_j$ across active features.

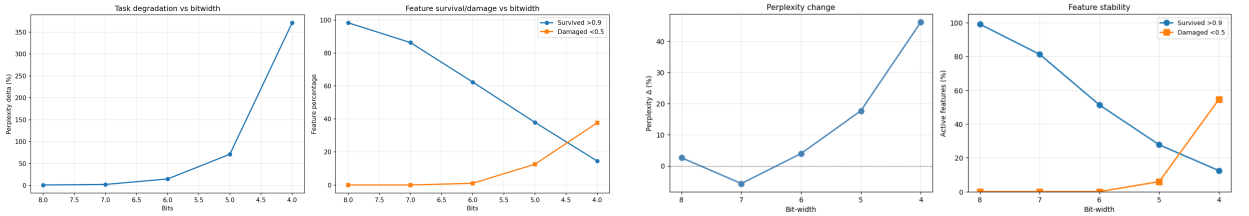
3.7 Fragility predictor

To test whether survival is predictable from full-precision statistics alone, we fit an L2-regularized logistic regression predicting the INT6 survival outcome $y_j = \mathbf{1}[c_j > t_s]$. The predictors are four FP16 feature statistics: rarity $-\log(f_j + \epsilon)$, log mean activation $\log(\mu_j + \epsilon)$, log peak activation $\log(\max_t z_j(t; \theta_{\mathrm{FP16}}) + \epsilon)$, and log concentration

$$\log\left(\frac{\max_t z_j(t; \theta_{\mathrm{FP16}}) + \epsilon}{\mu_j + \epsilon}\right),$$

Table 1: Cross-model RTN quantization sweep and pruning baseline.

Model	Condition	Bits	Prune sparsity	ΔPPL (%)	Mean corr.	Survived (%)	Degraded (%)	Damaged (%)
Pythia-70M	FP16 baseline	16	—	0.000	1.000	100.000	0.000	0.000
Pythia-70M	RTN INT8	8	—	1.020	0.981	98.312	1.688	0.000
Pythia-70M	RTN INT7	7	—	2.258	0.957	86.322	13.678	0.000
Pythia-70M	RTN INT6	6	—	14.722	0.888	62.395	36.586	1.020
Pythia-70M	RTN INT5	5	—	71.208	0.777	37.957	49.455	12.588
Pythia-70M	RTN INT4	4	—	371.112	0.583	14.504	47.785	37.711
Pythia-70M	Magnitude pruning matched INT6	16	0.175	13.488	0.891	60.970	38.731	0.299
Gemma-2-2B	FP16 baseline	16	—	0.000	1.000	100.000	0.000	0.000
Gemma-2-2B	RTN INT8	8	—	2.624	0.966	99.090	0.910	0.000
Gemma-2-2B	RTN INT7	7	—	-5.648	0.943	81.257	18.743	0.000
Gemma-2-2B	RTN INT6	6	—	3.989	0.891	51.302	48.698	0.000
Gemma-2-2B	RTN INT5	5	—	17.716	0.764	27.716	66.363	5.921
Gemma-2-2B	RTN INT4	4	—	46.081	0.494	12.402	32.895	54.703
Gemma-2-2B	Magnitude pruning matched INT6	—	0.162	6.126	0.841	38.578	61.184	0.238



(a) Pythia-70M (b) Gemma-2-2B

Figure 2: Cross-scale RTN bit-width sweep. Panel (a) shows Pythia-70M; panel (b) shows Gemma-2-2B.

where f_j is the firing rate, $\mu_j = (1/N)\sum_t z_j(t;\theta_{\text{FP16}})$, and ϵ is a small numerical-stability constant. After standardizing predictors, we fit $P(y_j = 1) = \sigma(\beta_0 + \beta^\top \phi_j)$ with inverse regularization strength $C = 1$. We report 5-fold stratified cross-validated AUC and standardized coefficients, with additional diagnostics in Appendix G. Because the predictors are correlated, we interpret coefficients jointly rather than as isolated causal effects.

3.8 Stability protocol

Unless stated otherwise, we use 200k tokens for Pythia-70M and 500k tokens for Gemma-2-2B. On Pythia INT6, we test token-budget sensitivity, random-subset stability, and an FP16-vs-FP16 null; we also repeat the sweep at a second layer to check layer sensitivity.

4 Results

We report results for Pythia-70M and Gemma-2-2B using the fixed-SAE protocol from Section 3. Survival, degradation, and damage are computed over FP16-active features using the thresholds in Section 3.4; all correlations use identical token positions, isolating the effect of weight compression.

4.1 Feature survival under quantization is graded and consistent across scale

Table 1 reports the round-to-nearest (RTN) bit-width sweep for both models together with a magnitude-pruning baseline; Figure 2 plots survival and damage against bit-width.

On Pythia-70M, feature survival declines monotonically as precision decreases: 98.3% of active features survive at INT8, 86.3% at INT7, 62.4% at INT6, 38.0% at INT5, and 14.5% at INT4. The fraction of damaged features remains negligible through INT7 (0.0%), reaches 1.0% at INT6, and then rises sharply to 12.6% at INT5 and 37.7% at INT4. Mean per-feature correlation falls correspondingly from 0.981 (INT8) to 0.583 (INT4).

Gemma-2-2B exhibits the same qualitative trajectory at a 30× larger parameter count. Survival declines from 99.1% (INT8) to 81.3% (INT7), 51.3% (INT6), 27.7% (INT5), and 12.4% (INT4). Damage is again

negligible until low bit-widths, remaining at 0.0% through INT6, rising to 5.9% at INT5, and reaching 54.7% at INT4. The two models differ in detail—Gemma retains marginally more features at INT8 and fewer at INT6—but the shape of the curve, a slow decline at high precision followed by an accelerating collapse below INT6, is shared (Figure 2).

Two features of the sweep are notable for the analysis that follows. First, in both models the transition from "mostly intact" to "mostly degraded" occurs over a narrow band between INT7 and INT5, rather than as a smooth linear decline. Second, the degraded band (correlation in $[0.5, 0.9]$) is consistently larger than the damaged band at intermediate bit-widths: at Gemma INT6, 48.7% of features are degraded but 0.0% are damaged, indicating that intermediate quantization predominantly blurs features rather than destroying them.

4.2 Feature survival is predictable from full-precision statistics

We test whether INT6 feature survival can be predicted from FP16 statistics alone. For each active feature, we compute rarity, log mean activation, log peak activation, and log activation concentration, then fit an L2-regularized logistic regression to the survival label $c_j > 0.9$. Table 2 reports coefficients and 5-fold cross-validated AUC; Figure 3 shows survival by predictor quartile.

Table 2: Predicting INT6 feature survival from FP16 feature statistics. AUC is the 5-fold cross-validated mean ± standard deviation; coefficients are standardized logistic-regression coefficients.

Model	Active feats.	Survived (%)	AUC	Rarity	Log mean	Log peak	Log concentration
Pythia-70M	5,688	62.4	0.924 ± 0.007	4.52	2.72	1.66	-1.67
Gemma-2-2B	7,144	51.3	0.971 ± 0.002	17.84	9.79	4.14	-8.16

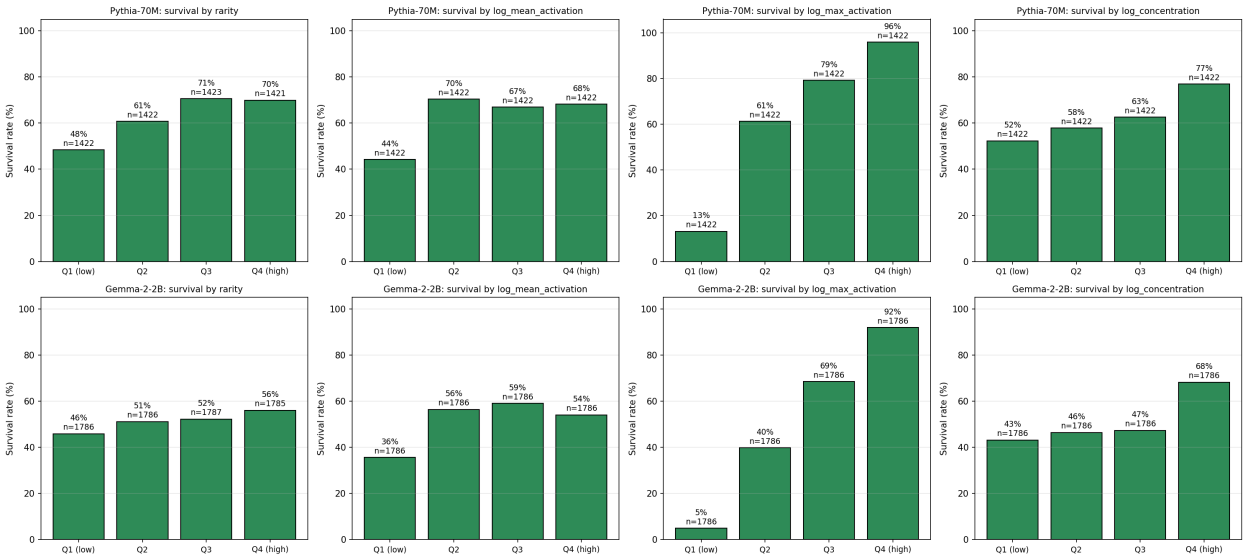


Figure 3: Feature survival by quartile of FP16 feature statistics under RTN INT6. Quartiles are computed within each model and statistic; bars show the percentage of active features with correlation $c_j > 0.9$ after quantization.

The predictor is highly accurate in both models: cross-validated AUC is 0.924 ± 0.007 on Pythia-70M ($n = 5{,}688$) and 0.971 ± 0.002 on Gemma-2-2B ($n = 7{,}144$). All four predictors agree in sign across models, suggesting that the relationship between FP16 feature statistics and quantization survival generalizes across scale.

Peak activation is the strongest marginal predictor. Survival rises from 13% to 96% across peak-activation quartiles on Pythia-70M and from 5% to 92% on Gemma-2-2B (Figure 3). Rarity has the largest multivariate coefficient (4.52 on Pythia-70M, 17.84 on Gemma-2-2B), but because rarity and the activation-magnitude predictors are correlated, the multivariate coefficient ranking is not a reliable guide to which predictor matters most marginally. The robust conclusion is that survival is highly predictable from simple FP16 statistics, peak activation is the strongest single marginal predictor, and rarer features are not more fragile after controlling for activation statistics.

4.3 Feature damage can diverge from task-level perplexity

Table 1 shows a dissociation between perplexity and feature survival on Gemma-2-2B. INT7 improves perplexity by 5.65% while survival falls to 81.3%, meaning 18.7% of active features degrade despite improved task-level performance. INT6 further reduces feature survival to 51.3%, even though its perplexity increase remains modest at 3.99%.

Table 3: Gemma behavioral protocol check. Chunked and sliding-window perplexity for FP16 and RTN INT8/INT7/INT6, with SAE survival from the main QDM run.

Condition	Chunked PPL	Chunked ΔPPL (%)	Sliding PPL	Sliding ΔPPL (%)	SAE survival (%)
FP16 baseline	458.53	0.00	46.39	0.00	100.00
RTN INT8	470.56	+2.62	46.78	+0.82	99.09
RTN INT7	432.64	-5.65	43.38	-6.49	81.26
RTN INT6	476.82	+3.99	44.79	-3.45	51.30

To check that this dissociation is not an artifact of chunked perplexity, we re-evaluate FP16, INT8, INT7, and INT6 with a sliding-window protocol (window 2048, stride 512; Table 3, Appendix D). Sliding-window evaluation lowers absolute perplexity but preserves the dissociation: INT7 improves perplexity under both protocols, and INT6 improves perplexity under sliding-window while only 51.3% of active features survive. The INT7 diagnostic suite in Appendix C confirms that the effect is reproducible and not a quantization failure.

4.4 Quantization and pruning damage overlapping feature sets

Table 4 compares RTN INT6 against magnitude pruning calibrated to match INT6 perplexity (Pythia sparsity 0.175, perplexity 76.99 vs. INT6 77.83; Gemma sparsity 0.1625, perplexity 486.6 vs. INT6 476.8; calibration in Appendix F). On Pythia-70M the two methods produce similar aggregate damage (37.6% non-survived under RTN, 39.0% under pruning); on Gemma-2-2B pruning is more aggressive at the matched operating point (48.7% vs. 61.4% non-survived).

Table 4: RTN INT6 versus pruning overlap. Summary of non-survived feature-set overlap and per-feature damage-score correlation between RTN INT6 and calibrated magnitude pruning.

Model	RTN non-survived (%)	Pruning non-survived (%)	Jaccard overlap	Pearson damage corr.	Spearman damage corr.
Pythia-70M	37.61	39.03	0.860	0.951	0.976
Gemma-2-2B	48.70	61.42	0.792	0.959	0.978

At the per-feature level the two methods are strongly concordant. The Jaccard overlap of non-survived features is 0.86 (Pythia) and 0.79 (Gemma), and the Spearman correlation of per-feature damage scores is 0.98 in both models (Table 4; scatter and decile plots in Appendix E). The methods differ in the tail: RTN produces no damaged ($c_j < 0.5$) features on Gemma while pruning produces 17, and the firing-rate decile analysis (Appendix E) shows pruning damaging more features than RTN in every decile on Gemma, with the gap largest at high firing rates. The decile analysis also shows, for both methods and both models, that non-survival rises with firing rate—high-firing features are more vulnerable—consistent with Section 4.2.

4.5 Methodology ablations show stable QDM estimates

Table 5 summarizes methodology ablations for Pythia-70M INT6. QDM estimates are stable across token budgets and random subsets: survival is 60.7–62.8% across 50k–200k tokens, and three independent 100k-token subsets vary by only 0.71 percentage points. The FP16-vs-FP16 null gives mean correlation 1.000000 and 0.0% damaged features, confirming that the pipeline does not manufacture drift. Threshold sensitivity checks preserve the qualitative pattern (Appendix B), and the bit-width sweep also holds at a second layer (Figure 4), indicating that the result is not specific to the primary read-out site.

Table 5: Methodology ablations for Pythia-70M INT6. Summary of token-budget, random-subset, FP16-null, and threshold-sensitivity checks used to assess QDM measurement stability.

Survival threshold	Damage threshold	Survived	Degraded	Damaged
0.80	0.50	80.13%	18.85%	1.02%
0.90	0.50	62.39%	36.59%	1.02%
0.95	0.50	44.41%	54.57%	1.02%

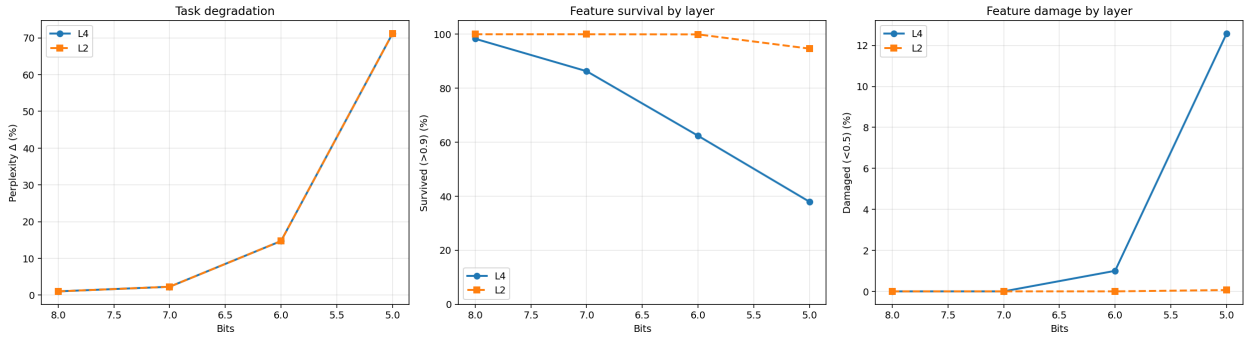


Figure 4: Layer-sensitivity check for Pythia-70M. RTN bit-width sweep repeated at two residual-stream layers, comparing feature survival across layers.

5 Analysis and Discussion

5.1 Task metrics are insufficient indicators of feature-level fidelity

The most consequential dissociation is between perplexity and feature survival (Section 4.3). At Gemma INT7, perplexity improves while nearly one in five features degrades; under sliding-window evaluation INT6 also improves perplexity while roughly half of features fall below the survival threshold. The INT6 sign change between the chunked and sliding-window protocols underscores the point: the task metric is sensitive enough to the evaluation protocol that its sign can flip, while the feature-survival measurement—computed on fixed tokens—does not depend on the perplexity protocol at all.

One plausible interpretation is that INT7 rounding acts as mild regularization under this corpus and protocol, but we do not establish the mechanism at a bit-width where the perturbation is large enough to be beneficial but not yet destructive; the logit-drift diagnostics (Appendix C) show INT7 perturbs the model's outputs more than INT8 and less than INT6, yet reduces loss where INT8 and INT6 do not. The mechanism is secondary to the implication: a practitioner selecting a quantization configuration by perplexity alone could choose a setting that appears lossless, or even beneficial, while having substantially altered the model's internal feature structure. For interpretability work that is developed on a full-precision model and then deployed in quantized form, perplexity parity is therefore not sufficient evidence that the analyzed features remain intact.

5.2 Vulnerability is shared across compression methods

The matched-perplexity comparison (Section 4.4) shows that quantization and pruning, at equal task degradation, damage strongly overlapping sets of features—Jaccard 0.79 to 0.86 and damage-score Spearman 0.98. The two methods are therefore better described as agreeing on which features are vulnerable than as targeting different feature classes. They differ in severity: on Gemma, matched pruning is somewhat more aggressive overall (and was calibrated to a slightly higher perplexity than INT6, so part of this gap reflects residual mismatch), and pruning produces a small tail of fully damaged features that quantization does not.

The shared vulnerability ranking connects directly to the signal-strength account in Section 5.4: if feature survival depends on whether a feature's signal exceeds the perturbation scale, then both rounding and removal should endanger many of the same low-signal features. The high cross-method damage-score correlation matches this prediction and extends the pruning-based picture of Borobia et al. (2026) to quantization: feature vulnerability also appears under weight rounding, where it can be traced across bit-widths, compared to perplexity, and measured on the same frozen SAE basis.

5.3 Quantization induces graded mechanistic change, not a precision cliff

Feature survival declines systematically with bit-width rather than remaining intact until a sharp precision floor. Because correlations are computed on identical tokens with a fixed SAE, this isolates the effect of weight rounding on the residual-stream geometry read out by the SAE. The same pattern across Pythia-70M and Gemma-2-2B suggests that quantization fragility is not specific to one model or scale.

At intermediate bit-widths, degradation dominates damage (e.g., Gemma INT6: 48.7% degraded, 0% damaged): features are often blurred rather than destroyed. Their activations remain positively correlated with the full-precision feature, but no longer strongly enough to be treated as the same interpretability unit. This matters because SAE analyses typically assume features are stable referents; a feature at correlation 0.7 is neither the original feature nor pure noise, and analyses transferred from FP16 to a quantized model can silently inherit this blurring.

5.4 Fragility is structured and predictable

The logistic-regression results (Section 4.2) show that feature survival is highly predictable from simple FP16 statistics. Mechanistically, quantization perturbations compete with each feature's activation signal: high-signal features remain above the perturbation floor, while weak-signal features are more easily blurred. This explains why peak activation is the strongest marginal predictor and why the predictors generalize across model scales.

The role of rarity requires care. Rarity has the largest standardized coefficient in the multivariate model, and its sign is positive—rarer features survive better. This is initially counter-intuitive if one expects specialized, rarely firing features to be fragile. However, the marginal effect of rarity is modest relative to peak activation, and the two predictors are correlated, so the large multivariate coefficient should not be read as rarity being the dominant causal factor. The defensible statement is that survival is jointly predictable from these statistics with high accuracy, that peak activation is the strongest single marginal predictor, and that, controlling for the other variables, rarer features are not more fragile and if anything survive better.

Borobia et al. (2026) report the same pattern under unstructured pruning. Our results show the pattern is not specific to weight removal: it also holds under weight rounding. The agreement across two mechanically distinct compression operations strengthens the interpretation that vulnerability is governed by a feature's signal strength rather than by the specific way weights are altered.

6 Conclusion

We asked whether behavioral parity under quantization implies SAE feature fidelity. Holding the SAE, token set, and read-out site fixed while varying only model weights, we measured per-feature survival across RTN bit-widths on Pythia-70M and Gemma-2-2B. Three findings stand out. First, task-level metrics can

miss feature damage: perplexity can remain stable or improve while many SAE features degrade. Second, quantization and matched-perplexity pruning damage many of the same features, suggesting a shared compression-induced vulnerability. Third, feature survival under quantization is graded and predictable: features are often blurred rather than destroyed, and survival can be forecast from full-precision statistics alone, with peak activation as the strongest marginal predictor.

These results show that behavioral parity is not sufficient evidence that full-precision interpretability transfers to quantized deployment. An analysis and a quantization setting can each appear sound in isolation while failing to compose. We release our pipeline and per-feature results, and view causal validation as the next step: testing whether these feature changes alter steering, auditing, or other safety-relevant interventions in deployed quantized models.

7 Limitations

Our study is limited to two model families, publicly available residual-stream SAEs, and simulated round-to-nearest weight quantization. We evaluate fixed token budgets, read-out layers, and compression settings, so the quantitative survival rates should not be interpreted as universal across architectures, datasets, SAE training recipes, activation sites, or deployment quantizers. We also focus on feature-level correlation and perplexity rather than downstream tasks or causal interventions; future work should test whether the same feature changes alter steering, auditing, or safety-relevant behavior in deployed quantized models.

References

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Hector Borobia, Elies Seguí-Mas, and Guillermina Tormo-Carbó. How pruning reshapes features: Sparse autoencoder analysis of weight-pruned language models. *arXiv preprint arXiv:2603.25325*, 2026.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL `https://transformer-circuits.pub/2023/monosemantic-features`.

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

David Chanin and Adrià Garriga-Alonso. Sparse but wrong: Incorrect l0 leads to incorrect features in sparse autoencoders. *arXiv preprint arXiv:2508.16560*, 2025.

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.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36:10088–10115, 2023.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, 28, 2015.

Piotr Jedryszek and Oliver M. Crook. Stable and steerable sparse autoencoders with weight regularization. *arXiv preprint arXiv:2603.04198*, 2026.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Evaluating quantized large language models. *arXiv preprint arXiv:2402.18158*, 2024.

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*, pp. 278–300, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

Luke Marks, Alasdair Paren, David Krueger, and Fazl Barez. Enhancing neural network interpretability with feature-aligned sparse autoencoders. *arXiv preprint arXiv:2411.01220*, 2024.

Cristina P. Martin-Linares and Jonathan P. Ling. Attribution-guided distillation of matryoshka sparse autoencoders. *arXiv preprint arXiv:2512.24975*, 2025.

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

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

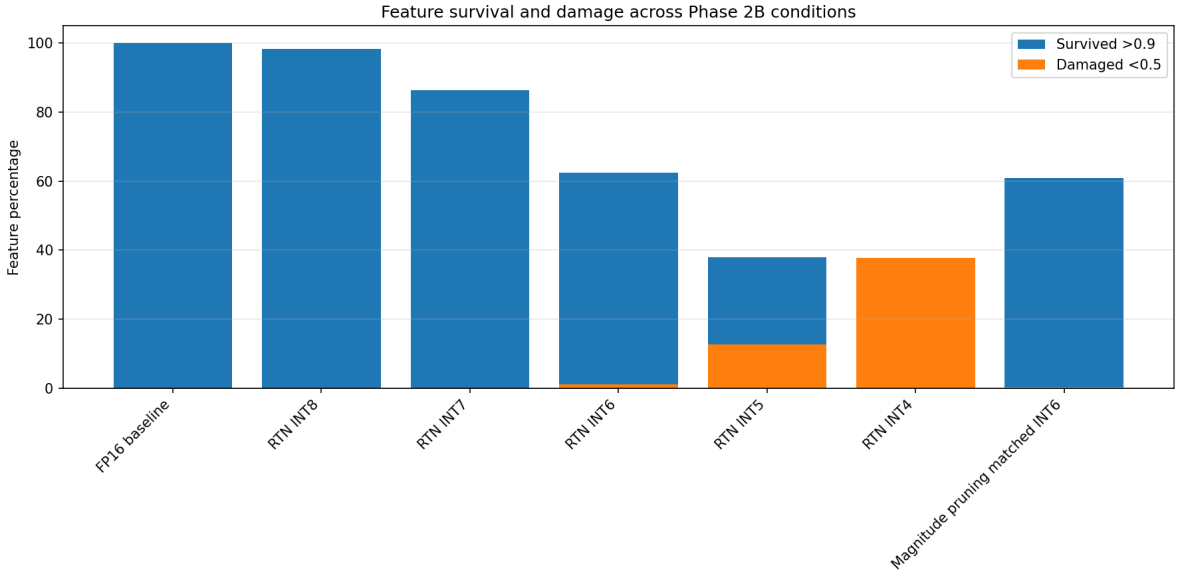
Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, 2024.

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`.

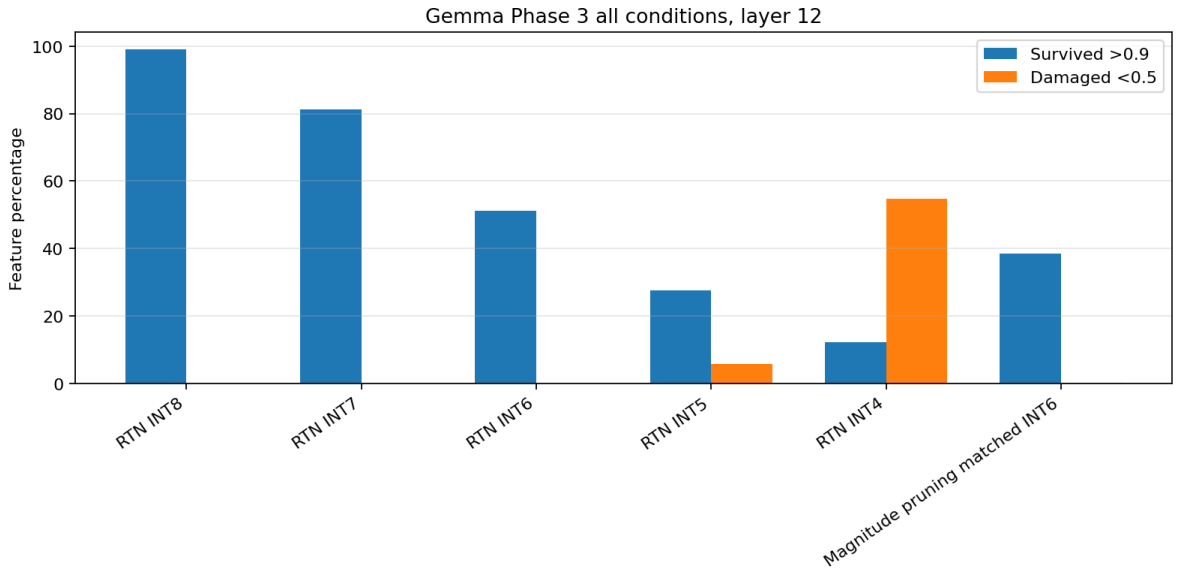
Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Appendix A Full all-condition compression plots

Appendix A reports the full all-condition bar plots for Pythia-70M and Gemma-2-2B. These figures supplement the main RTN bit-width sweep by showing all evaluated compression conditions, including FP16, RTN INT8–INT4, and the pruning baseline.

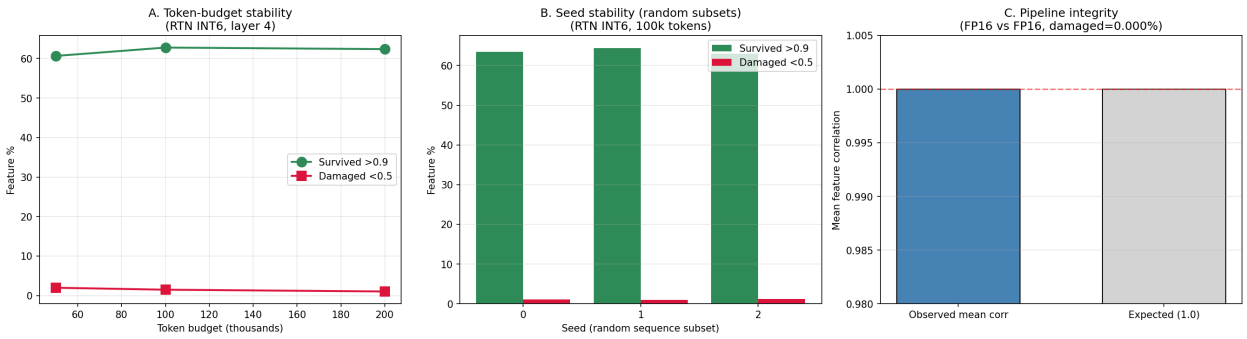


Appendix Figure A.1: Full Pythia-70M compression comparison across FP16, RTN bit-widths, and matched pruning.



Appendix Figure A.2: Full Gemma-2-2B compression comparison across FP16, RTN bit-widths, and approximately matched pruning.

Appendix B Stability ablation plots and threshold sensitivity



Appendix Figure B.1: Stability ablation plots for Phase 4.

Appendix Table B.1: Threshold sensitivity for ablation D. Percentages are reported over all evaluated conditions.

Survival threshold	Damage threshold	Survived (%)	Degraded (%)	Damaged (%)
0.80	0.30	80.13	19.83	0.04
0.80	0.40	80.13	19.64	0.23
0.80	0.50	80.13	18.85	1.02
0.80	0.60	80.13	15.84	4.03
0.80	0.70	80.13	10.36	9.51
0.85	0.30	73.12	26.85	0.04
0.85	0.40	73.12	26.65	0.23
0.85	0.50	73.12	25.86	1.02
0.85	0.60	73.12	22.86	4.03
0.85	0.70	73.12	17.37	9.51
0.90	0.30	62.39	37.57	0.04
0.90	0.40	62.39	37.38	0.23
0.90	0.50	62.39	36.59	1.02
0.90	0.60	62.39	33.58	4.03
0.90	0.70	62.39	28.09	9.51
0.95	0.30	44.41	55.56	0.04
0.95	0.40	44.41	55.36	0.23
0.95	0.50	44.41	54.57	1.02
0.95	0.60	44.41	51.56	4.03
0.95	0.70	44.41	46.08	9.51

Appendix Table B.2: Token-budget stability ablation. This table reports QDM feature-survival statistics for RTN INT6 using increasing token budgets. It tests whether estimated survival and damage rates are sensitive to the number of evaluated tokens.

Token budget	Evaluated tokens	Active features	Mean corr.	Survived (%)	Damaged (%)
50k	49,664	5,609	0.8796	60.67	1.96
100k	99,840	5,719	0.8858	62.79	1.47
200k	199,680	5,688	0.8879	62.39	1.02

Appendix Table B.3: Random-subset seed stability ablation. This table reports QDM statistics across independently sampled 100k-token subsets. It tests whether the measured feature-survival rate depends on the random token subset used for activation comparison.

Seed	Evaluated tokens	Active features	Mean corr.	Survived (%)	Damaged (%)
0	99,840	5,721	0.8905	63.52	1.01
1	99,840	5,788	0.8926	64.39	0.93
2	99,840	5,726	0.8883	62.98	1.24

Appendix Table B.4: FP16-vs-FP16 pipeline null check. This table compares FP16 activations against the FP16 reference under the same QDM pipeline. The expected result is mean correlation 1.0, 100% survival, and 0% damage, confirming that the pipeline does not create artificial feature drift.

Test	Active features	Mean corr.	Survived (%)	Damaged (%)
Strict null (same model, same tokens)	5,719	1.0000	100.00	0.00

Appendix C INT7 investigation checks

Appendix Table C.1: Corrected 50k-token perplexity sweep verifying the INT7 decrease persists under shifted-loss evaluation.

Condition	Bits	Loss	Perplexity	PPL delta (%)	Prediction tokens
FP16	16	6.2134	499.40	0.00	49,725
RTN INT8	8	6.2335	509.52	2.03	49,725
RTN INT7	7	6.1520	469.66	-5.96	49,725
RTN INT6	6	6.2481	517.04	3.53	49,725

Appendix Table C.2: Five-seed subset check showing the INT7 perplexity decrease is reproducible across token samples.

Seed	Prediction tokens	FP16 PPL	INT8 PPL	INT7 PPL	INT6 PPL	INT8 delta (%)	INT7 delta (%)	INT6 delta (%)
0	76,500	495.06	505.05	463.97	515.05	2.02	-6.28	4.04
1	76,500	458.03	467.96	427.55	473.73	2.17	-6.65	3.43
2	76,500	448.90	458.81	419.73	470.33	2.21	-6.50	4.77
3	76,500	493.42	504.91	460.44	510.32	2.33	-6.69	3.42
4	76,500	468.28	479.36	437.37	484.99	2.37	-6.60	3.57

Appendix Table C.3: Weight-error diagnostics confirming INT7 quantization behaves normally between INT8 and INT6.

Bits	Mean MSE	Mean MAE	Mean relative MAE	Mean cosine	Min q seen	Max q seen
4	4.51×10^{-6}	0.001698	0.2141	0.9805	-7	7
5	9.92×10^{-7}	0.000794	0.1002	0.9956	-15	15
6	2.34×10^{-7}	0.000385	0.0485	0.9990	-31	31
7	5.74×10^{-8}	0.000190	0.0239	0.9997	-63	63
8	1.50×10^{-8}	0.000095	0.0120	0.9999	-128	127

Appendix Table C.4: Logit-drift check showing INT7 output perturbation lies between INT8 and INT6.

Bits	MSE mean	MSE std.	MAE mean	MAE std.	Cosine mean	Cosine std.	Loss delta mean	Loss delta std.
6	1.7596	0.4365	1.0002	0.1072	0.9779	0.0068	-0.0012	0.0831
7	0.6222	0.1735	0.5827	0.0560	0.9922	0.0021	-0.0543	0.0391
8	0.1758	0.0397	0.3104	0.0289	0.9978	0.0006	0.0252	0.0310

Appendix D Gemma sliding-window behavioral evaluation

To check whether the high absolute Gemma perplexity under the chunked Phase 3 protocol was caused by context resets, we reran behavioral evaluation with a sliding-window protocol. This check was applied to FP16, RTN INT8, RTN INT7, and RTN INT6 Gemma-2-2B on WikiText-2-raw.

The evaluation used a HuggingFace-format Gemma model in bfloat16 with per-output-channel RTN quantization. The token stream contained 288,894 tokens, of which 288,893 were scored. We used window size $W = 2048$ and stride $S = 512$. For each window, previous tokens served as context and only newly introduced target positions were included in the loss. The aggregate perplexity was computed as a token-weighted mean negative log-likelihood:

$$\mathrm{PPL} = \exp\left(\frac{\sum_w m_w \mathcal{L}_w}{\sum_w m_w}\right), \tag{8}$$

where m_w is the number of scored tokens in window w and $\mathcal{L}_w$ is the mean loss over those scored tokens.

For the quantized conditions, the HuggingFace RTN implementation quantized 182 tensors and 2,024,275,968 parameters. This behavioral check is distinct from the main SAE feature-survival pipeline, which uses TransformerLens/Gemma Scope activations. We therefore use the sliding-window results only as a robustness check on the perplexity trend, not as a replacement for the main feature-correlation results or as a benchmark-comparable Gemma perplexity evaluation.

Appendix Table D.1: Full Gemma sliding-window perplexity results using window size 2048 and stride 512.

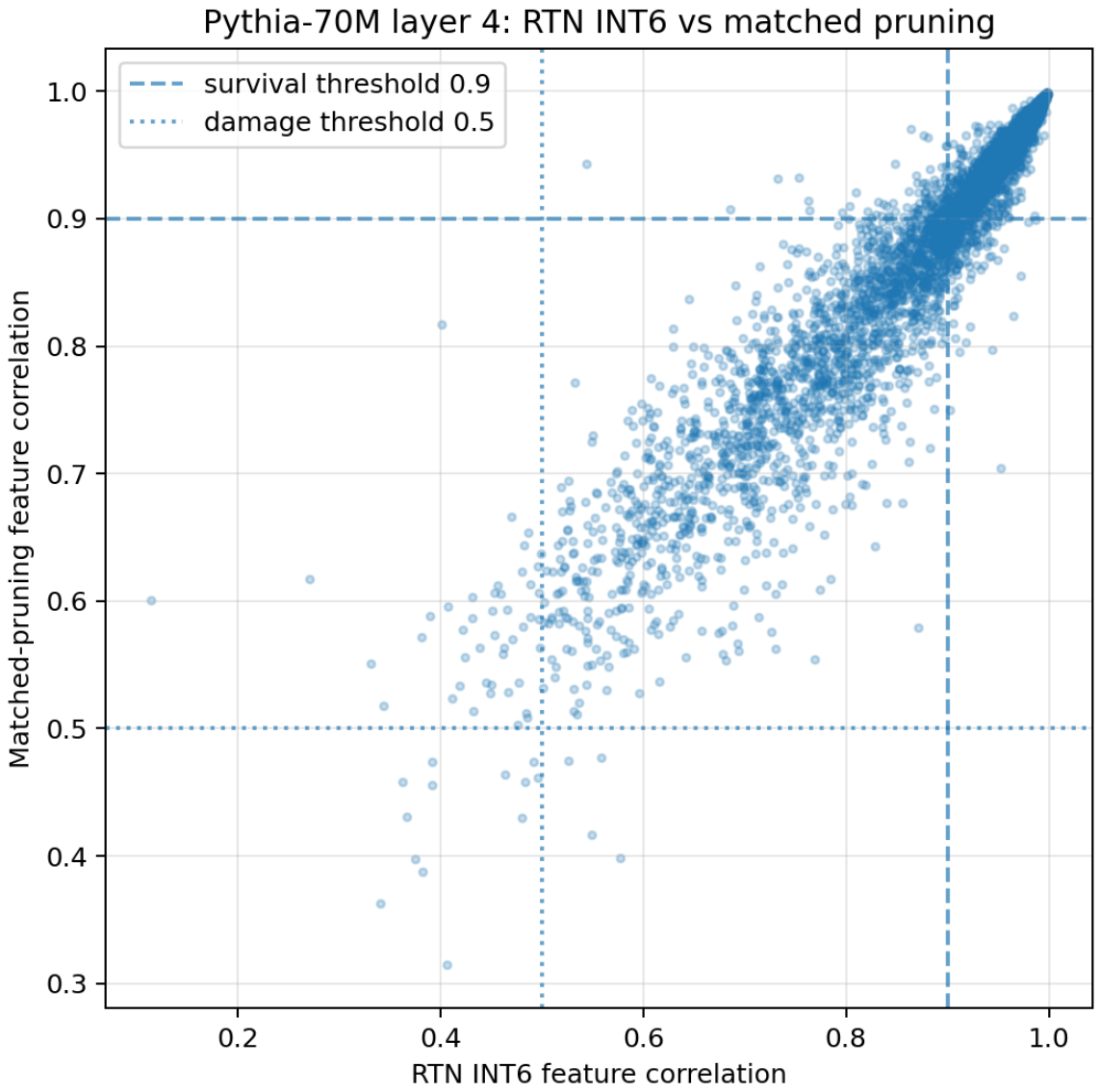
Condition	Bits	Loss	PPL	Total tokens	Loss tokens	Window	Stride	Quantized tensors	Quantized params	PPL delta (%)
FP16 baseline	16	3.8372	46.39	288,894	288,893	2048	512	0	0	0.00
RTN INT8	8	3.8454	46.78	288,894	288,893	2048	512	182	2,024,275,968	0.82
RTN INT7	7	3.7700	43.38	288,894	288,893	2048	512	182	2,024,275,968	-6.49
RTN INT6	6	3.8020	44.79	288,894	288,893	2048	512	182	2,024,275,968	-3.45

Appendix Table D.2: Comparison of chunked and sliding-window Gemma perplexity with corresponding SAE feature-survival rates.

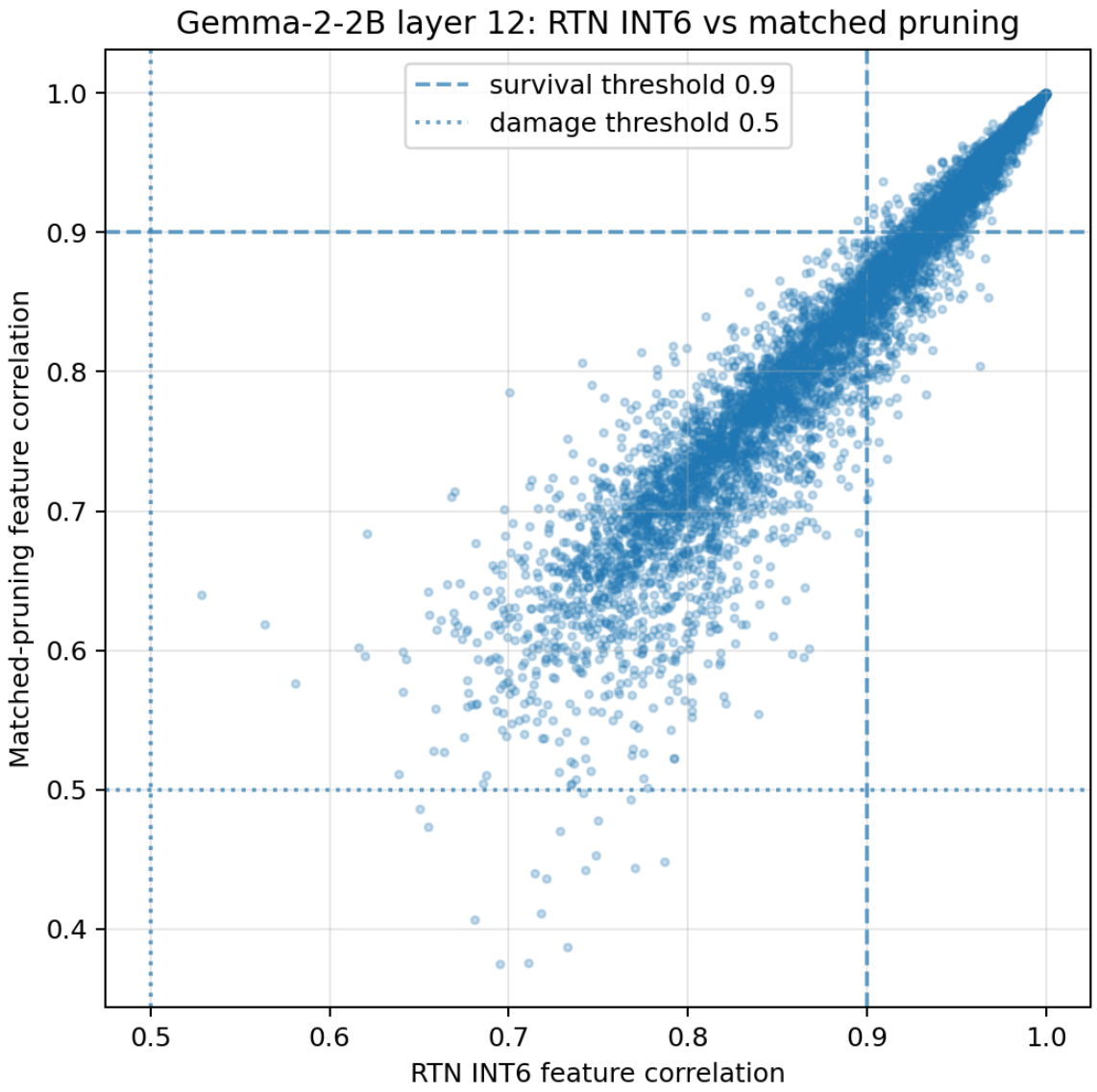
Condition	Chunked PPL	Chunked delta (%)	Sliding PPL	Sliding delta (%)	Survived (%)	Degraded (%)	Damaged (%)	PPL ratio	Median corr.
FP16 baseline	458.53	0.00	46.39	0.00	100.00	0.00	0.00	9.88	1.0000
RTN INT8	470.56	2.62	46.78	0.82	99.09	0.91	0.00	10.06	0.9707
RTN INT7	432.64	-5.65	43.38	-6.49	81.26	18.74	0.00	9.97	0.9499
RTN INT6	476.82	3.99	44.79	-3.45	51.30	48.70	0.00	10.65	0.9027

Appendix E RTN-versus-pruning overlap visualizations

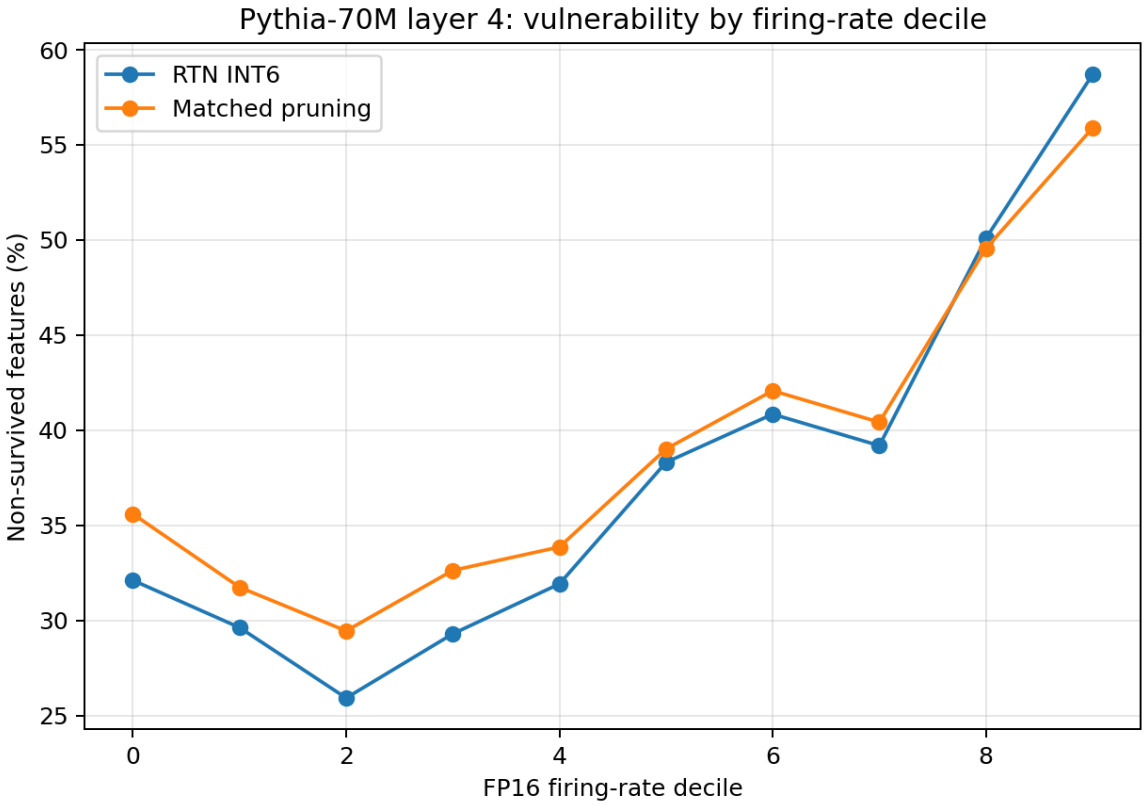
Appendix E visualizes the RTN-versus-pruning overlap analysis. The scatter plots compare per-feature survival correlations under RTN INT6 and magnitude pruning, while the decile plots show how non-survival varies across FP16 firing-rate deciles. These figures support the main-text result that quantization and pruning largely affect overlapping vulnerable feature sets rather than unrelated feature classes.



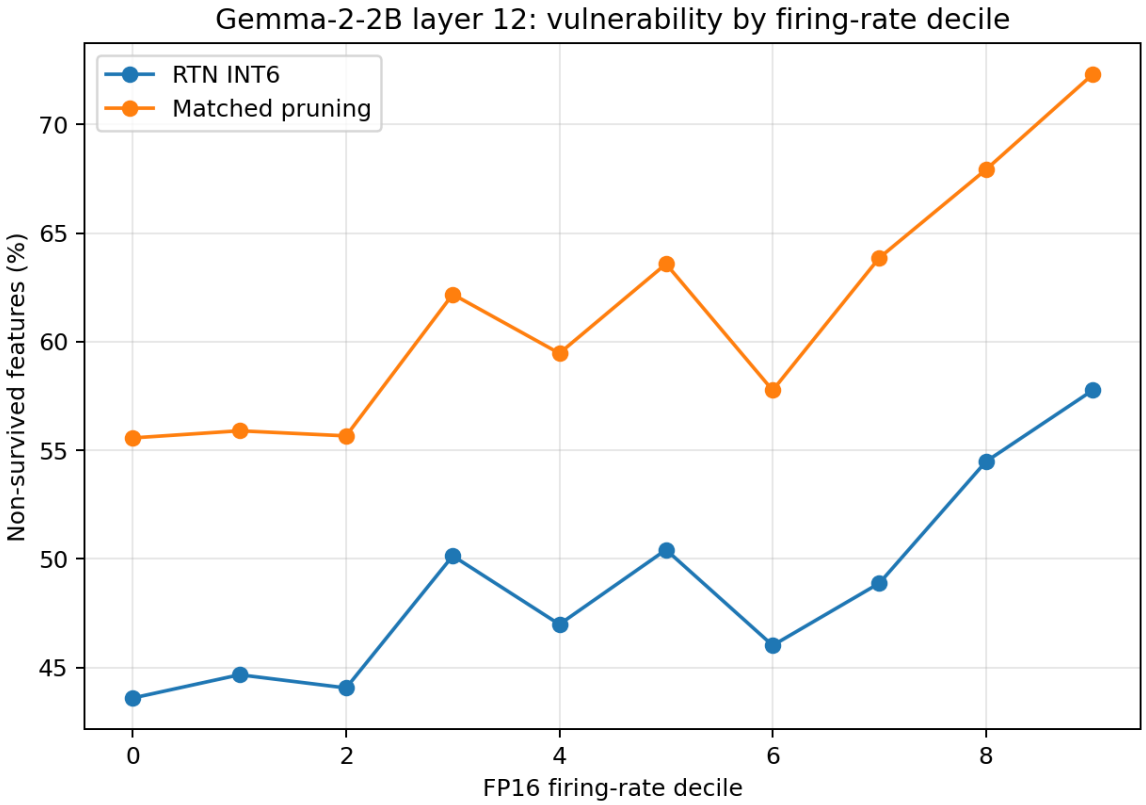
Appendix Figure E.1: Pythia-70M per-feature correlation scatter comparing RTN INT6 and matched magnitude pruning.



Appendix Figure E.2: Gemma-2-2B per-feature correlation scatter comparing RTN INT6 and approximately matched magnitude pruning.



Appendix Figure E.3: Pythia-70M non-survival rates by FP16 firing-rate decile for RTN INT6 and matched pruning.



Appendix Figure E.4: Gemma-2-2B non-survival rates by FP16 firing-rate decile for RTN INT6 and approximately matched pruning.

Appendix F Pruning calibration

We calibrate magnitude pruning by searching over sparsity values $p \in [0, 0.8]$ to match the RTN INT6 perplexity regime. The search is performed with a fixed-step bisection procedure over six steps on a held-out token budget. For Pythia-70M, the selected pruning sparsity is $p = 0.175$, giving perplexity 76.99 and ΔPPL $= +13.49\%$, close to RTN INT6 perplexity 77.83 and ΔPPL $= +14.72\%$. For Gemma-2-2B, the selected pruning sparsity is $p = 0.1625$, giving perplexity 486.63 and ΔPPL $= +6.13\%$, compared with RTN INT6 perplexity 476.82 and ΔPPL $= +3.99\%$. Thus, the Pythia pruning baseline is closely matched, while the Gemma pruning baseline is approximately matched and slightly harsher.

Appendix Table F.1: Pythia-70M pruning calibration search for matching the RTN INT6 perplexity regime.

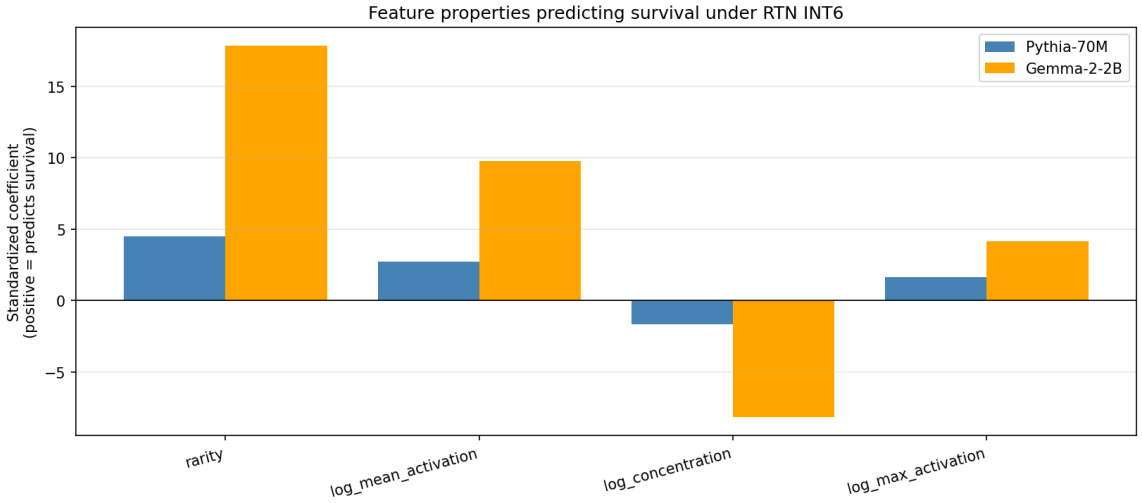
Step	Sparsity	Perplexity	Delta (%)
0	0.4000	282.41	316.29
1	0.2000	82.77	22.01
2	0.1000	69.90	3.04
3	0.1500	74.25	9.45
4	0.1750	76.99	13.49
5	0.1875	78.75	16.08

Appendix Table F.2: Gemma-2-2B pruning calibration search for approximately matching the RTN INT6 perplexity regime.

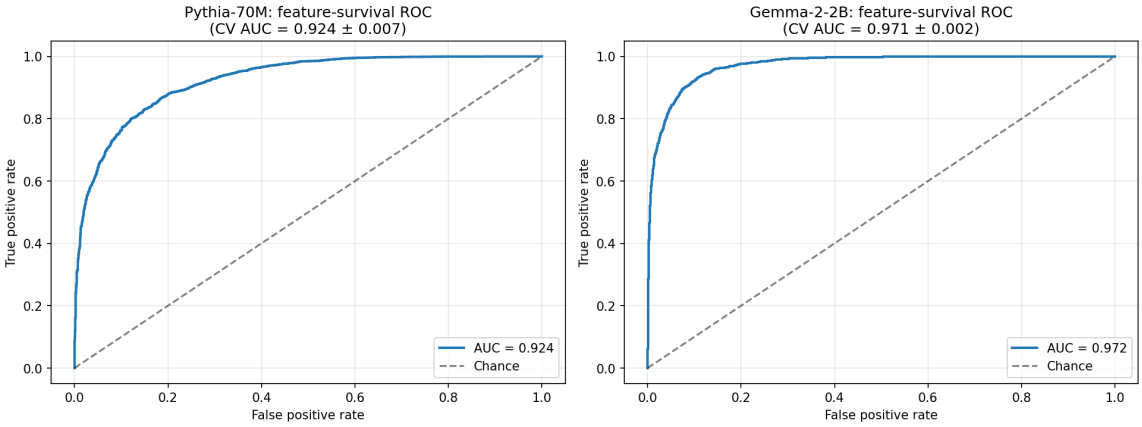
Step	Sparsity	Perplexity	Loss	Target PPL	Absolute error	Tensors	Params	Zeroed params
0	0.4000	8,140.18	9.0046	497.99	7,642.19	182	2,024,275,968	811,793,027
1	0.2000	585.27	6.3721	497.99	87.28	182	2,024,275,968	405,794,737
2	0.1000	460.93	6.1333	497.99	37.06	182	2,024,275,968	202,895,355
3	0.1500	484.27	6.1827	497.99	13.71	182	2,024,275,968	304,557,827
4	0.1750	530.74	6.2743	497.99	32.76	182	2,024,275,968	355,143,150
5	0.1625	508.64	6.2317	497.99	10.65	182	2,024,275,968	329,809,692

Appendix G Fragility predictor details

Appendix G reports additional details for the fragility-prediction analysis: multivariate logistic-regression coefficients, cross-validated ROC curves, and a cross-model coefficient comparison. Because the predictors are correlated, the coefficients should be interpreted jointly rather than as isolated causal effects.



Appendix Figure G.1: Standardized logistic-regression coefficients for predicting INT6 feature survival from FP16 feature statistics.



Appendix Figure G.2: Cross-validated ROC curves for INT6 feature-survival prediction on Pythia-70M and Gemma-2-2B.

Appendix Table G.1: Cross-model comparison of standardized fragility-predictor coefficients.

Feature statistic	Pythia coef.	Gemma coef.	Sign agreement	Mean |coef.|
Rarity	4.52	17.84	Yes	11.18
Log mean activation	2.72	9.79	Yes	6.26
Log concentration	-1.67	-8.16	Yes	4.92
Log peak activation	1.66	4.14	Yes	2.90

Appendix H Condition-level verification summaries

Appendix H lists condition-level verification summaries for the main RTN INT6, pruning, and layer-sensitivity results. These files support the aggregate values reported in the main tables and figures.

Appendix Table H.1: Pythia-70M layer 4 summary statistics for the Phase 2A RTN quantization sweep.

Feature-survival statistics

Condition	Bits	Layer	Tokens	Total feats.	Active feats.	Mean corr.	Median corr.	Survived (%)	Damaged (%)
FP16 baseline	16	4	199,680	32,768	5,686	1.0000	1.0000	100.00	0.00
per-channel INT8	8	4	199,680	32,768	5,686	0.9813	0.9920	98.33	0.00
per-channel INT7	7	4	199,680	32,768	5,686	0.9574	0.9804	86.32	0.00
per-channel INT6	6	4	199,680	32,768	5,686	0.8880	0.9395	62.40	1.00
per-channel INT5	5	4	199,680	32,768	5,686	0.7773	0.8497	37.95	12.57

Behavioral and degraded-feature statistics

Condition	Degraded (%)	PPL	Loss	PPL delta (%)
FP16 baseline	0.00	67.80	4.2166	0.00
per-channel INT8	1.67	68.49	4.2267	1.02
per-channel INT7	13.68	69.34	4.2390	2.27
per-channel INT6	36.60	77.78	4.3538	14.71
per-channel INT5	49.47	116.05	4.7540	71.16

Appendix Table H.2: Pythia-70M layer 2 summary statistics for the Phase 2A.5 layer-sensitivity check.

Feature-survival statistics

Condition	Bits	Layer	Tokens	Total feats.	Active feats.	Mean corr.	Median corr.	Survived (%)	Damaged (%)
FP16 baseline	16	2	199,680	32,768	6,139	1.0000	1.0000	100.00	0.00
per-channel INT8	8	2	199,680	32,768	6,139	0.9994	0.9996	100.00	0.00
per-channel INT7	7	2	199,680	32,768	6,139	0.9978	0.9985	100.00	0.00
per-channel INT6	6	2	199,680	32,768	6,139	0.9915	0.9942	99.95	0.00
per-channel INT5	5	2	199,680	32,768	6,139	0.9644	0.9753	94.67	0.07

Behavioral and degraded-feature statistics

Condition	Degraded (%)	PPL	Loss	PPL delta (%)
FP16 baseline	0.00	67.80	4.2166	0.00
per-channel INT8	0.00	68.49	4.2267	1.02
per-channel INT7	0.00	69.34	4.2390	2.27
per-channel INT6	0.05	77.78	4.3538	14.71
per-channel INT5	5.26	116.05	4.7540	71.16

Appendix Table H.3: Pythia-70M RTN INT6 per-condition summary used to verify the main INT6 results.

Condition: RTN INT6

Bits	Layer	Tokens	Active feats.	Mean corr.	Median corr.	Survived (%)	Damaged (%)
6	4	199,680	5,688	0.8879	0.9395	62.39	1.02

PPL	Loss	Method	Prune sparsity
77.83	4.3545	TL	—

Appendix Table H.4: Pythia-70M matched-pruning per-condition summary used to verify the pruning baseline.

Condition: Magnitude pruning matched INT6

Bits	Layer	Tokens	Active feats.	Mean corr.	Median corr.	Survived (%)	Damaged (%)
16	4	199,680	5,688	0.8905	0.9329	60.97	0.30

PPL	Loss	Method	Prune sparsity
76.99	4.3437	TL	0.1750

Appendix Table H.5: Gemma-2-2B RTN INT6 per-condition summary used to verify the main Gemma INT6 results.

Condition: RTN INT6

Bits	Layer	Tokens	Total feats.	Active feats.	Mean corr.	Median corr.	Survived (%)	Degraded (%)	Damaged (%)
6	12	499,968	16,384	7,144	0.8911	0.9027	51.30	48.70	0.00

PPL	Loss	PPL delta (%)	Method
476.82	6.1671	3.99	RTN

Appendix Table H.6: Gemma-2-2B approximately matched-pruning per-condition summary used to verify the Gemma pruning baseline.

Condition: Magnitude pruning matched INT6

Layer	Tokens	Total feats.	Active feats.	Mean corr.	Median corr.	Survived (%)	Degraded (%)	Damaged (%)
12	499,968	16,384	7,144	0.8407	0.8551	38.58	61.18	0.24

PPL	Loss	PPL delta (%)	Method	Matched bits	Sparsity	Zeroed params
486.63	6.1875	6.13	pruning	6	0.1625	329,809,692

Appendix Table H.7: Full combined cross-model RTN quantization and pruning summary.

Model	Condition	Layer	Bits	Sparsity	Tokens	Active feats.	PPL	ΔPPL (%)	Mean corr.	Median corr.	Survived (%)	Degraded (%)	Damaged (%)
Pythia-70M	FP16 baseline	4	16	—	199,680	5,688	67.840	0.000	1.000	1.000	100.000	0.000	0.000
Pythia-70M	RTN INT8	4	8	—	199,680	5,688	68.532	1.020	0.981	0.992	98.312	1.688	0.000
Pythia-70M	RTN INT7	4	7	—	199,680	5,688	69.372	2.258	0.957	0.980	86.322	13.678	0.000
Pythia-70M	RTN INT6	4	6	—	199,680	5,688	77.828	14.722	0.888	0.940	62.395	36.586	1.020
Pythia-70M	RTN INT5	4	5	—	199,680	5,688	116.148	71.208	0.777	0.850	37.957	49.455	12.588
Pythia-70M	RTN INT4	4	4	—	199,680	5,688	319.602	371.112	0.583	0.636	14.504	47.785	37.711
Pythia-70M	Magnitude pruning matched INT6	4	16	0.175	199,680	5,688	76.990	13.488	0.891	0.933	60.970	38.731	0.299
Gemma-2-2B	FP16 baseline	12	16	—	499,968	7,144	458.535	0.000	1.000	1.000	100.000	0.000	0.000
Gemma-2-2B	RTN INT8	12	8	—	499,968	7,144	470.565	2.624	0.966	0.971	99.090	0.910	0.000
Gemma-2-2B	RTN INT7	12	7	—	499,968	7,144	432.636	-5.648	0.943	0.950	81.257	18.743	0.000
Gemma-2-2B	RTN INT6	12	6	—	499,968	7,144	476.824	3.989	0.891	0.903	51.302	48.698	0.000
Gemma-2-2B	RTN INT5	12	5	—	499,968	7,144	539.768	17.716	0.764	0.778	27.716	66.363	5.921
Gemma-2-2B	RTN INT4	12	4	—	499,968	7,144	669.833	46.081	0.494	0.451	12.402	32.895	54.703
Gemma-2-2B	Magnitude pruning matched INT6	12	—	0.162	499,968	7,144	486.626	6.126	0.841	0.855	38.578	61.184	0.238

Appendix I Streaming estimator for per-feature correlations

Computing the feature-stability score in Eq. 3 naively requires storing the full activation matrices $Z^{\mathrm{FP16}}, Z^{C} \in \mathbb{R}^{N \times d_{\mathrm{sae}}}$. This is memory-intensive at the token budgets used in our experiments, especially for Gemma-2-2B where $N \approx 500{,}000$ and $d_{\mathrm{sae}} = 16{,}384$. We therefore compute the same Pearson correlations in a single streaming pass over batches of token positions.

For each feature j, define

$$x_t = z_j(t; \theta_{\mathrm{FP16}}), \qquad y_t = z_j(t; \theta_C).$$

During streaming evaluation, we accumulate the following sufficient statistics:

$$\begin{aligned} S_x &= \sum_t x_t, & S_y &= \sum_t y_t, \\ S_{x^2} &= \sum_t x_t^2, & S_{y^2} &= \sum_t y_t^2, \\ S_{xy} &= \sum_t x_t y_t. \end{aligned} \tag{I.1}$$

The Pearson correlation for feature j is then recovered as

$$c_j = \frac{S_{xy} - S_x S_y / N}{\sqrt{\left(S_{x^2} - S_x^2/N\right)\left(S_{y^2} - S_y^2/N\right)}}. \tag{I.2}$$

This expression is algebraically equivalent to Eq. 3, but reduces memory from $O(N d_{\mathrm{sae}})$ to $O(d_{\mathrm{sae}})$.

We accumulate all sufficient statistics in float64 to reduce numerical error in the sum-of-squares computation. In addition to the correlation statistics, we stream the FP16 firing count

$$\sum_t \mathbf{1}[x_t > 0]$$

and the FP16 running maximum

$$\max_t x_t$$

used for the active-feature filter and feature-property analyses. As a pipeline null check, comparing FP16 activations against themselves yields $c_j = 1$ for all active features to six decimal places.

Appendix J Notation and Glossary

This appendix summarizes the notation and threshold conventions used throughout the paper.

Appendix Table J.1: Notation and threshold conventions used in QDM feature-survival analysis.

Term / notation	**Meaning**
QDM	Our feature-level audit for measuring how SAE features change under model compression. QDM compares FP16 and compressed SAE activations on identical tokens using a fixed SAE.
FP16 model	The full-precision reference model. All feature-survival scores are measured relative to this model.
Compressed model	A model obtained by applying RTN quantization or magnitude pruning to the FP16 model weights.
RTN quantization	Round-to-nearest weight quantization. In our experiments, weights are rounded to a simulated b-bit grid for $b \in \{8, 7, 6, 5, 4\}$.
Matched pruning	Magnitude pruning calibrated to approximately match the RTN INT6 perplexity regime. It is used as a comparison baseline for quantization.
Read-out site	The residual-stream layer or hook where model activations are extracted and passed through the SAE.
Frozen SAE	The same pretrained SAE encoder is used for FP16 and compressed activations. The SAE is not retrained for compressed models.
$z_j(t;\theta)$	Activation of SAE feature j at token position t for model parameters θ.
f_j	FP16 firing rate of feature j, defined as the fraction of evaluated token positions where $z_j(t;\theta_{\mathrm{FP16}}) > 0$.
Active feature	A feature with FP16 firing rate $f_j > 0.001$. Survival, degradation, and damage percentages are reported over active features unless stated otherwise.
c_j	Feature-stability score for feature j, defined as the Pearson correlation between FP16 and compressed activations over identical token positions.
Survived feature	An active feature with $c_j > 0.9$, meaning its compressed activation pattern remains strongly aligned with its FP16 activation pattern.
Degraded feature	An active feature with $0.5 \leq c_j \leq 0.9$, meaning its compressed activation pattern remains partially but not strongly aligned with FP16.
Damaged feature	An active feature with $c_j < 0.5$, meaning its compressed activation pattern is weakly aligned with FP16 under the frozen SAE basis.
Feature fidelity	The degree to which SAE activation patterns are preserved after compression, measured primarily by c_j and the survived/degraded/damaged taxonomy.
Behavioral metric	A task-level metric such as perplexity or downstream accuracy. Behavioral metrics are distinct from feature-level fidelity metrics.
ΔPPL	Relative perplexity change compared with FP16, defined as $\mathrm{PPL}(C)/\mathrm{PPL}(\mathrm{FP16}) - 1$ for condition C.
Damage score	The per-feature quantity $d_j = 1 - c_j$, used when comparing how similarly RTN quantization and pruning affect individual features.
Jaccard overlap	The overlap between non-survived feature sets under two compression methods, defined as $\lvert A \cap B \rvert / \lvert A \cup B \rvert$.

Appendix K Model, SAE, and Evaluation Configuration

This appendix summarizes the model, SAE, and evaluation settings used in the main QDM experiments.

Appendix Table K.1: Model, SAE, and token-budget configuration for the main QDM experiments. The SAE, token set, and read-out site are fixed within each model; only the model weights vary across compression conditions.

Model	SAE source	Read-out site	SAE width	Token budget	Block length
Pythia-70M	`pythia-70m-deduped-res-sm`	Layer 4 residual stream, `blocks.4.hook_resid_post`	—	200k	512
Gemma-2-2B	Gemma Scope canonical residual-stream SAE	Layer 12 residual stream	16,384	500k	256

Appendix Table K.2: Shared experimental conventions for QDM feature-survival evaluation.

Setting	**Value / description**
Dataset	WikiText-2-raw. Non-empty lines are concatenated and tokenized before evaluation.
Compression methods	Round-to-nearest quantization and magnitude pruning.
RTN bit-widths	INT8, INT7, INT6, INT5, and INT4.
Quantized modules	Attention projections and MLP projections; for Gemma-2-2B, the gated MLP gate projection is also quantized. Layer-norm and embedding parameters are left in full precision.
Quantization granularity	Per-output-channel RTN. Quantized weights are dequantized back into floating-point tensors for simulated low-bit evaluation.
Pruning baseline	Magnitude pruning calibrated to approximately match the RTN INT6 perplexity regime.
Active-feature threshold	A feature is active if its FP16 firing rate satisfies $f_j > 0.001$.
Survival threshold	A feature survives if its FP16-vs-compressed activation correlation satisfies $c_j > 0.9$.
Damage threshold	A feature is damaged if its FP16-vs-compressed activation correlation satisfies $c_j < 0.5$.
Default survival taxonomy	Survived: $c_j > 0.9$; degraded: $0.5 \leq c_j \leq 0.9$; damaged: $c_j < 0.5$.
Feature-stability metric	Pearson correlation c_j between FP16 and compressed SAE activations over identical token positions.
Behavioral metric	Token-level perplexity, reported as relative change ΔPPL from FP16.
Sliding-window check	Gemma-2-2B behavioral robustness check with window size $W = 2048$ and stride $S = 512$.
Fragility predictor	L2-regularized logistic regression predicting INT6 survival from FP16 feature statistics.
Stability checks	Token-budget sensitivity, random-subset stability, FP16-vs-FP16 null, threshold sensitivity, and second-layer sensitivity.

Appendix L Quantized Module Implementation Details

This appendix specifies which weight tensors are modified by the compression operators. For both RTN quantization and magnitude pruning, we apply compression only to transformer block linear weights. Embeddings, unembeddings, layer-normalization parameters, biases, and SAE parameters are left unchanged.

For each transformer block, the targeted attention weights are the query, key, value, and output projections. The targeted MLP weights are the input/up projection and output/down projection. For Gemma-2-2B, which uses a gated MLP, we additionally compress the gate projection. Thus, the compressed module families are:

Appendix Table L.1: Transformer block module families included in, and excluded from, compression.

Component	**Targeted weights**
Attention	Query projection, key projection, value projection, output projection.
MLP	Input/up projection, output/down projection.
Gated MLP, Gemma only	Gate projection.
Excluded	Embeddings, unembeddings, layer norms, biases, SAE weights.

In TransformerLens-style notation, the Pythia targeted modules correspond to the attention projection weights and MLP projection weights in each block, including names such as `W_Q`, `W_K`, `W_V`, `W_O`, `W_in`, and `W_out`. In HuggingFace/Gemma-style notation, the targeted modules correspond to names such as `q_proj`, `k_proj`, `v_proj`, `o_proj`, `up_proj`, `down_proj`, and `gate_proj`.

All targeted tensors are compressed with the same per-output-channel procedure described in Section 3.2. For RTN quantization, weights are rounded to the target integer grid and then dequantized back into floating-point tensors, so the experiment simulates low-bit weights without changing the model architecture. For pruning, the smallest-magnitude weights within each targeted tensor are set to zero according to the calibrated sparsity. The SAE encoder is never quantized, pruned, or retrained.

The output-channel convention follows the tensor layout used by the loaded model implementation. For each targeted matrix, the scale is computed over all dimensions except the output-channel dimension, so each output channel receives its own RTN scale. This convention is applied consistently across Pythia and Gemma experiments; where module layouts differ between TransformerLens and HuggingFace implementations, the corresponding output-channel axis is selected before applying the same per-output-channel rule.

Appendix M Reproducibility and Hardware

We provide an anonymized code and artifact repository at `https://anonymous.4open.science/r/sae-feature-survival-quantization-0141/`. The repository includes the experiment scripts, configuration files, lightweight utility code, generated summary outputs, and figures used to reproduce the main analyses. The scripts cover the RTN bit-width sweeps, streaming feature-correlation pipeline, Gemma sliding-window perplexity check, stability ablations, fragility-predictor analysis, and table/figure generation. Large model weights, datasets, and third-party SAE checkpoints are not redistributed; they should be downloaded from their original sources subject to their respective licenses. The repository is anonymized for review and does not include author-identifying metadata.

Experiments were run on CUDA-enabled A100 GPU hardware. Approximately 24 hours of total computation time including debugging. The Pythia-70M experiments are lightweight and can be reproduced on a single consumer GPU with sufficient memory, while the Gemma-2-2B streaming and sliding-window evaluations require a larger GPU or careful batching/offloading. To reduce memory usage, our implementation computes feature correlations from streaming sufficient statistics rather than storing the full token-by-feature activation matrix. The main experiments use 200k tokens for Pythia-70M and 500k tokens for Gemma-2-2B, with smaller-token-budget runs used for stability checks.