informatics

MDPI

Article

When Calibration Data Become Training Data: Specialization and Self-Generated Text in Learned Low-Rank Compression of Large Language Models

Van Tien Pham

Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam; phamvantien@tdtu.edu.vn

Abstract

Low-rank compression shrinks large language models by factorizing their weights, guided by a small calibration set. When the factorization is learned by distillation, as in learned subspace projections (LSP), the calibration set becomes training data, yet is still chosen by convention. We compressed OPT-1.3B, Llama-3.2-1B, and Qwen3-1.7B by 30% and 50% with LSP, calibrated on six sources at a matched budget (WikiText-2, C4, SlimPajama, Alpaca, and two kinds of self-generated text) over three seeds, and repeated the comparison on Llama-3.1-8B and with a second compressor. Training removed most of the up to 600-fold perplexity differences between untrained models but turned them into specialization: each source best preserved its own text, and WikiText-2, the customary choice, gave the least general models. Zero-shot accuracy slightly favored instruction data, through content rather than template. Self-generated documents gave the most uniformly faithful Llama models without external data, whereas exact samples of the model, whose expected training objective is the sequence-level divergence, failed on Qwen3-1.7B. The initialization's domain survived training, and compressed Llama models collapsed without their calibration beginning-of-sequence token. With every source, compression lost most instruction following and multi-step reasoning. These results indicate that calibration sources should be reported with every perplexity.

Keywords: large language models; model compression; low-rank factorization; knowledge distillation; calibration data; self-generated data

1. Introduction

Large language models are costly to serve, as generating each token reads all their weights from memory and their size determines the hardware that they can fit on. Post-training compression reduces this cost without retraining the model [1]. Low-rank factorization, one of its main families, replaces each weight matrix with the product of two thin matrices; unlike unstructured sparsity, the result is a dense model with smaller matrices, which stores fewer weights and needs no sparse kernels [2,3]. Whether it also runs faster depends on the model and batch size (Section 4.11).

Pretrained weight matrices are not low-rank, but the activations they act on are [4], so practical low-rank methods depend on calibration data consisting of a few hundred sequences. The activation statistics of these sequences then decide which directions may be discarded at each layer [5–7]. These criteria are local to each layer, and their errors accumulate through depth [8,9]. Learned subspace projections (LSP) replace them with a

Check for updates

Academic Editor: Francisco José García-Peñalvo

Received: 4 October 2026
Revised: 13 November 2026
Accepted: 16 November 2026
Published: 19 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

global objective [10], so that each layer removes a subspace with an orthogonal projector; the projectors are trained with the pretrained weights frozen, so that the compressed model reproduces the output distribution of the dense model on the calibration data. Learning makes a large difference at high compression: on Llama-2-7B compressed by 70%, it lowers WikiText-2 perplexity to 10.9, compared to 222.8 for the untrained initialization [10].

Learning also changes the role of the calibration data. A training-free method sees these data only through second-order statistics. SVD-LLM, a truncation method based on singular value decomposition (SVD), reports that their size, seed, and source change its results by at most 3% [6]. A learned method fits them; the compressed model is optimized to agree with the dense model on the calibration windows and nowhere else. Therefore, which data to fit becomes a modeling decision that is made by convention: WikiText-2 for perplexity tables and Alpaca or C4 for zero-shot accuracy [6,7,10]. It is also constrained in practice, as the pretraining corpus of an open-weight model is rarely released and domain or user data may be proprietary, sensitive, or licensed for non-commercial use only.

The dense model can supply its own calibration data. For the distillation objective of LSP, text sampled from the dense model is not merely a substitute for real text; with exact ancestral sampling, the expected training objective becomes the divergence between the distributions that the two models assign to whole sequences, which is the quantity that a drop-in replacement should minimize (Section 3.3). This is a direct consequence of the chain rule of the Kullback–Leibler (KL) divergence, which we use here to identify the objective. Whether this principled objective also preserves the abilities measured by benchmarks, which score the model on human-written text, is an empirical question.

These observations leave three gaps. First, the effect of calibration data has been measured for pruning, quantization [11–13], and training-free low-rank truncation [6], where the data enter only through statistics; learned low-rank compressors [6,10,14,15] train on the calibration set, but each fixes it by convention and reports results for a single source, so it is unknown how much of a reported perplexity measures compression quality and how much measures fit to the calibration domain. Second, text generated by the model itself has been proposed as calibration data for quantization and pruning [13,16,17], but has not been studied as training data for a learned compressor, where its relation to the distillation objective is exact and its failure modes can be traced. Third, the choices that accompany the data (window format and beginning-of-sequence tokens, document length and boundaries, number of distinct windows, statistics of the initialization, and optimization settings) have not been separated from the content of the data.

This paper is an empirical study of these questions; it does not propose a new compression method. We compress OPT-1.3B, Llama-3.2-1B, and Qwen3-1.7B by 30% and 50% with LSP, calibrated on six sources at a matched budget: two common corpora, a pretraining-like mixture, an instruction dataset, and two kinds of text sampled from the dense model. Each compressed model is evaluated by perplexity, by its divergence from the dense model on held-out text of every source and of six further domains chosen independently of the sources, and by zero-shot accuracy, with three seeds and paired comparisons over evaluation items. The main comparisons are repeated on Llama-3.1-8B, with another learned compressor (SVD-LLM with low-rank adaptation), and on an instruction-tuned model. We address the three research gaps described above by providing the following contributions:

- *A controlled comparison of calibration sources as training data* (first gap). Learning removes most of the dependence of the training-free truncation on the calibration source, which reaches two orders of magnitude in perplexity and persists under the oblique truncation of SVD-LLM, but turns it into specialization. Each source best preserves its own text: WikiText-2 calibration gives the least general compressed models while zero-

shot accuracy favors instruction data rather than any domain, by at most two points and through their content rather than their template (Sections 4.1–4.3, 4.9, and 4.10).

- *The objective of self-generated calibration and its failure mode* (second gap). Calibrating on exact samples of the dense model makes the expected distillation objective the sequence-level divergence between the two models (Proposition 1); such samples are best drawn at temperature one, but can fail, as on Qwen3-1.7B in every seed, for reasons having to do with neither the length nor the composition of the samples. On the other hand, independent self-generated documents give the most uniformly faithful compressed models on the two Llama models without any external data, rank among the best sources elsewhere, and when mixed half-and-half with SlimPajama, come close to the most uniform source in every setting (Sections 3.3, 4.3, 4.5, and 4.6).
- *Separating the content of the data from its presentation and optimization* (third gap). The calibration data act through the initialization, for which the domain survives training (training only slightly rotates the removed subspaces), as well as through the format of the windows. On the Llama models, the beginning-of-sequence token of the calibration windows decides which input formats the compressed model handles, and a compressed model collapses when fed text without it. Under per-source learning rates, longer schedules and fixed checkpoints, and with another learned compressor, the specialization persists when every source is packed with the same boundaries and segment lengths. Neither a data-dependent rank allocation nor free low-rank factors changes the ranking of sources, and free factors specialize more (Sections 4.4, 4.7, and 4.8; Appendix D).
- More distinct calibration windows continue to improve the compressed model at a fixed number of steps, where they also mean fewer passes over each window; calibration budgets that suffice for training-free methods do not saturate learned ones (Section 4.5).

Section 2 reviews related work; Section 3 describes the method, calibration sources, and protocol; Section 4 reports the results; and Section 5 discusses their practical consequences and limitations.

2. Related Work

2.1. Low-Rank Compression of Language Models

The weights of pretrained transformers are not low-rank, while the activations flowing between layers are [4]; thus, practical low-rank methods choose the subspace to be kept from calibration activations. Activation-aware SVD (ASVD) scales weights by activation magnitudes before truncation [5], Fisher-weighted SVD (FWSVD) weights the factorization by Fisher information [18], and SVD-LLM whitens the weight with the Cholesky factor of the input Gram matrix so that truncating singular values minimizes the error of the layer output [6]. Later work refines this layer-wise optimum with different ranks [7,19], jointly decomposes modules [20], or removes rows and columns of the residual stream after a rotation [21]. Loss-aware methods bring the network objective in through a curvature model [22] or a learned truncation position [23], with recent analyses showing that layer-wise errors accumulate through depth [8,9]. Training-based methods instead optimize against the network output: SVD-LLM recovers accuracy with low-rank adaptation (LoRA) [24] on the factors. A three-level scheme refines factors per matrix, per block, and end to end [15]. FlexRank distills nested factors for elastic deployment [14], while LSP trains orthogonal projectors with the pretrained weights frozen [10]. Tying the projection of the keys and values also reduces the key–value (KV) cache [10,25]. Surveys cover these families together with pruning, quantization, and distillation [1]. All of these methods fix

their calibration data by convention; how the choice of data changes what the compressed model preserves is yet to be studied.

2.2. Feature Selection, Subspace Learning, and Representation Learning

At every layer, low-rank compression decides which part of an activation the next layer keeps; in this sense, it selects features, as the kept subspace should preserve information that is needed downstream and discard the remaining redundant information. Feature selection pursues the same aim on input variables. Recent work has made the selection robust to noise and outliers through sparse optimization [26,27], captured nonlinear dependencies with multiple kernels [28], balanced the information retained from several views [29,30], and adapted neighborhood structures to the data [31,32]. Two differences separate learned subspace projections from these methods: first, they select directions of an activation space rather than coordinates; second, their criterion is the fidelity of the model's output distribution rather than a supervised or reconstruction loss. One question carries over directly, namely, that a selection fitted to one distribution preserves what that distribution exercises, so the data that the selection is fitted on decide what the reduced representation retains elsewhere. This is the question the present study asks of calibration data.

2.3. Calibration Data in Post-Training Compression

For pruning and quantization, the choice of calibration data changes downstream accuracy [11]; C4 is not always the best source [12], data close to the pretraining distribution help most [13], and text generated by the model itself is competitive with real data [13,17]. For training-free low-rank methods, SVD-LLM reports that the size, seed, and source of its calibration data produce at most a 3% change in the performance of LLaMA-7B compressed by 20% [6]. For learned subspace projections on a vision transformer, the composition of the calibration pool changes how well the compressed features transfer [10]. The role of calibration data in learned low-rank compression of language models, where the calibration set is also the training set, has yet to be studied systematically. The above studies vary the data of methods that do not train on them, while the learned methods do not vary their data.

2.4. Distillation and Self-Generated Data

Knowledge distillation trains a student on the teacher's output distribution [33]; for sequence models, sequence-level distillation trains on sequences generated by the teacher [34], while on-policy distillation trains on sequences generated by the student [35]. LLM-QAT distills a quantized model on text generated by the full-precision model, thereby removing the need for training data [16]. The composition of the distillation data shapes what the student retains: under class imbalance, distilling in stages over resampled subsets and then refining on the original distribution improves retention of the minority classes [36]. Calibration data pose a related problem for compressed language models, with uneven representation of domains of unlabeled text in place of imbalanced classes. In learned low-rank compression, the student is a constrained copy of the teacher, and the per-token divergence of Equation (2) on teacher samples coincides in expectation with the sequence-level divergence (Section 3.3). However, whether this makes self-generated text a good training set for the compressed model has not been tested.

3. Materials and Methods

3.1. Learned Subspace Projections

We compress with learned subspace projections (LSP) [10], which we reimplemented; this section states the parts that are important for the role of the calibration data. Consider

a linear layer $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$ with $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. To remove k directions from its input, LSP applies the orthogonal projector $\mathbf{P} = \mathbf{I} - \mathbf{U}\mathbf{U}^\top$, where $\mathbf{U} = \text{qf}(\mathbf{V})$ is the orthonormal factor of the thin QR decomposition of an unconstrained $\mathbf{V} \in \mathbb{R}^{d_{\text{in}} \times k}$. The projected layer $\mathbf{WP}$ has rank at most $d_{\text{in}} - k$ and factors exactly as

$$\mathbf{WP} = (\mathbf{W}\mathbf{U}_\perp)\,\mathbf{U}_\perp^\top = \mathbf{B}\mathbf{A}, \qquad \mathbf{A} = \mathbf{U}_\perp^\top \in \mathbb{R}^{r \times d_{\text{in}}}, \quad \mathbf{B} = \mathbf{W}\mathbf{U}_\perp \in \mathbb{R}^{d_{\text{out}} \times r}, \tag{1}$$

with $r = d_{\text{in}} - k$ and $\mathbf{U}_\perp$ an orthonormal basis of $\text{span}(\mathbf{U})^\perp$; the output-side form $\mathbf{PW} = \mathbf{U}_\perp(\mathbf{U}_\perp^\top \mathbf{W})$ is analogous. Layers that read the same activation (the query, key, and value projections; the gate and up-projections) form a *tied group* that shares one input-side projector, and hence one factor $\mathbf{A}$ after merging. Every other layer is projected on its smaller side, and the square attention output projections on their input side; Appendix B shows that this choice, which the description of LSP leaves open, explains why our untrained initialization is worse than the published one at high ratios. A unit (a tied group or a single layer) is left dense when its factors would store more weights than its dense matrices.

The projectors are initialized from the whitened truncation of SVD-LLM [6]. With the calibration Gram matrix $\mathbf{G} = \frac{1}{n}\mathbf{X}^\top\mathbf{X} = \mathbf{S}\mathbf{S}^\top$ of the unit's input, the truncation keeps the leading right-singular directions of $\mathbf{WS}$ (row-stacked over a tied group); on the input side, LSP replaces this oblique map by the orthogonal projector that keeps $\text{span}(\mathbf{S}\mathbf{V}_r)$ and removes $\text{span}(\mathbf{S}^{-\top}\mathbf{V}_{>r})$, a step that we call the *orthogonal recast*; on the output side, the whitened truncation is already an orthogonal projector ([10] Proposition 1). We refer to the initialized model before any training as the *whitened initialization*; the oblique truncation itself serves as an additional training-free baseline (Section 3.4).

All projectors are then trained jointly to minimize the token-averaged divergence from the dense model, with every pretrained weight frozen:

$$\mathcal{L}(\{\mathbf{V}\}) \;=\; \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\,\frac{1}{T}\sum_{t=1}^{T}\text{KL}\big(p(\cdot \mid x_{\le t}) \,\big\|\, q_{\{\mathbf{V}\}}(\cdot \mid x_{\le t})\big) \tag{2}$$

where p is the dense model, q is the compressed one, $\mathbf{x} = (x_1, \ldots, x_T)$ is a calibration window, and $\mathcal{D}$ is the calibration distribution. The teacher is the same network with its projectors disabled, so no second copy of the weights is stored. After training, every projector merges into the factors of Equation (1), and the deployed model contains no LSP-specific operation.

The calibration set $\mathcal{D}$ enters this pipeline at three points: through the Gram matrices of the initialization, through the windows of Equation (2), and through the held-out windows used for early stopping. Learned methods fit the calibration set, while training-free methods use it only through second-order statistics.

3.2. Calibration Sources

We compare six calibration sources at a matched budget of 512 training windows of $T = 2048$ tokens, about one million tokens, plus 32 validation windows for early stopping (Table 1):

- **WikiText-2** [37]: Wikipedia articles; the usual calibration set of perplexity-oriented compression work [6,10,21].
- **C4** [38]: Filtered web text; the default calibration set of pruning and quantization [11,39,40].
- **SlimPajama** [41]: A deduplicated version of RedPajama [42], mixing web crawl, C4, code, books, arXiv, Wikipedia, and StackExchange, used here through its six billion token sample; it approximates a pretraining mixture.

- **Alpaca** [43]: Instruction–response pairs in the Alpaca prompt template; the calibration set of the zero-shot experiments of SVD-LLM and LSP.
- **Self-generated sequences**, abbreviated Self (seq.): Consists of 2048-token sequences sampled from the dense model (Section 3.3).
- **Self-generated documents**, abbreviated Self (doc.): Independent documents sampled from the dense model, packed like a corpus (Section 3.3).

Windows are built identically for every corpus. Documents are drawn without replacement, tokenized without special tokens, concatenated with the end-of-sequence (EOS) token as separator, and cut into windows of $T-1$ tokens preceded by the beginning-of-sequence (BOS) token; for Qwen3, for which the tokenizer has no BOS token, windows are T tokens without it. WikiText-2 windows are random non-overlapping chunks of the training split, and its validation split provides the validation windows; for every other source, training and validation windows come from disjoint documents. The perplexity and zero-shot evaluation sets are disjoint from every calibration set (Section 3.5).

Table 1. Calibration sources, showing the mean number of documents per 2048-token training window for each model and the dense model's mean next-token negative log-likelihood on the source's validation windows. On exact self-samples, the NLL equals the model's own entropy in expectation; long samples drift to high entropy (Section 3.3).

Source	Content	OPT-1.3B	Llama-3.2-1B	Qwen3-1.7B
		Documents per Window/Dense NLL (Nats per Token)		
WikiText-2	Wikipedia articles	1.0/2.70	1.0/2.29	1.0/2.29
C4	filtered web text	5.2/2.80	5.4/2.60	5.3/2.66
SlimPajama	web, code, books, arXiv, Wikipedia, Q&A	3.1/2.38	3.2/2.44	3.2/2.45
Alpaca	instruction–response pairs	19.1/1.91	20.7/1.43	20.6/1.33
Self (seq.)	2048-token samples of the dense model	8.7/2.88	7.8/4.65	3.5/3.90
Self (doc.)	independent samples, $\leq$512 tokens	11.9/2.91	8.4/3.01	6.2/3.18

3.3. Self-Generated Calibration and Sequence-Level Divergence

Let P_T and Q_T denote the distributions that the dense and compressed models assign to sequences $(x_2, \ldots, x_{T+1})$ following a fixed start token x_1. By the chain rule of the Kullback–Leibler divergence [44],

$$\mathrm{KL}(P_T \,\|\, Q_T) \;=\; \sum_{t=1}^{T} \mathbb{E}_{x_{\leq t} \sim P}\, \mathrm{KL}\big(p(\cdot \mid x_{\leq t}) \,\big\|\, q(\cdot \mid x_{\leq t})\big), \tag{3}$$

where $x_{\leq t} \sim P$ denotes the prefix distribution of the dense model. Comparing with Equation (2) gives the following statement.

Proposition 1. *If the calibration windows are drawn from the dense model by ancestral sampling at temperature one, starting from the start token, then the expected training objective (the population objective) equals the sequence-level divergence per token, $\mathbb{E}_{\mathbf{x}\sim P}\,\mathcal{L} = \mathrm{KL}(P_T\|Q_T)/T$, and, at any fixed projectors, the gradient of the objective on a window drawn independently of them is unbiased for $\nabla\mathrm{KL}(P_T\|Q_T)/T$. For calibration windows drawn from any other distribution $\mathcal{D}$, the objective weights the same per-context divergences by the prefix distribution of $\mathcal{D}$ instead of that of P.*

Proof. For windows sampled from p, each prefix $x_{\leq t}$ has the prefix marginal of P, so the expectation of Equation (2) is the right-hand side of Equation (3) divided by T. The sampling distribution does not depend on the projectors, so at fixed projectors the gradient of the objective on an independent window is an unbiased estimate of the gradient of its expectation. The last statement is Equation (2) with $x_{\leq t}$ drawn from $\mathcal{D}$. □

Proposition 1 identifies what each calibration source optimizes. On self-generated sequences, the population training objective is the divergence between the two models' distributions over whole sequences, which is the natural target when the compressed model will replace the dense one for generation. On an external corpus, the objective measures the same per-context divergences, but only on contexts typical of that corpus; how well the result transfers to other contexts is covered in Section 4. The proposition does not imply that self-generated data maximize downstream accuracy; benchmarks score the model on human-written contexts, which the model's own samples cover only as well as the model itself covers human text.

The proposition concerns the population objective, not its optimization on a finite calibration set. In practice, the windows form a pool of n samples that is reused over several epochs, and the projectors are fitted to the pool. A minibatch gradient is then an unbiased estimate of the gradient of the empirical objective on the pool at the current projectors; it is not an unbiased estimate of the population gradient at projectors that already depend on the pool, and the minimizer of the empirical objective can fit the pool rather than the model's distribution. The population statement holds at every step only when each step sees fresh samples, as in the experiment of Section 4.5, in which 4096 windows are each seen once; for a reused pool, it identifies the target that the empirical objective estimates. The identity is the chain rule of Equation (3) applied to the calibration distribution. We do not claim it as a theoretical contribution; its role is to show which divergence the different calibration sources optimize.

Two practical properties of self-generated sequences are important here. First, sampling at a temperature other than one draws prefixes from a tempered distribution, which reweights the divergence toward high-likelihood contexts (below one) or high-entropy contexts (above one); the targets stay the untempered distributions of the dense model. Second, long samples of small models drift [45]: the entropy of the next-token distribution of Llama-3.2-1B on its own samples grows from 2.6 nats over the first 128 positions to 5.7 nats over the last 128 of a 2048-token sequence, while that of Qwen3-1.7B grows from 3.2 to 4.7 nats as the text degenerates into high-entropy strings (Figure 1); OPT-1.3B does not drift. For the models that drift, the exact objective of Proposition 1 spends much of its weight on contexts that are far from natural text.

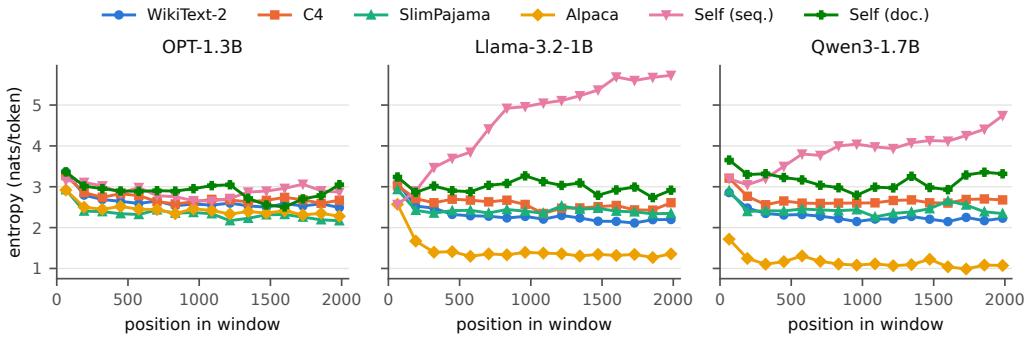


Figure 1. Entropy of the dense model's next-token distribution along the calibration windows of each source, averaged over 32 validation windows in buckets of 128 positions. Exact self-samples of Llama-3.2-1B and Qwen3-1.7B drift to high entropy, while self-generated documents restarted at least every 512 tokens do not.

The *document* variant bounds the drift: each document is sampled independently from a fresh start token, ends at the model's first EOS or after 512 tokens, and the documents are packed into windows as in any corpus. Each document remains an exact sample of the model's distribution truncated at 512 tokens, but a window concatenates independent

documents, so the objective on it is the same per-context divergence as for a corpus, with the dense model as the corpus. To test whether bounding the drift is what matters, Section 4.6 varies the length limit from 128 to 2048 tokens and restarts exact samples every 512 tokens.

3.4. Controls

The following controls separate where the calibration data act and how much of their effect comes from their content, rather than from how they are presented to the optimizer.

3.4.1. Training-Free Truncations

Evaluating the whitened initialization prior to training isolates the effect of the data on second-order statistics. Because the orthogonal recast of an input-side truncation has a larger layer-wise error than the truncation ([10] Proposition 1), we also evaluate the *oblique* whitened truncation, that is, the criterion of SVD-LLM [6] with the same ties and ranks. The shared factor is $\mathbf{A} = \mathbf{V}_r^\top \mathbf{S}^{-1}$ and each member keeps $\mathbf{B}_m = \mathbf{W}_m \mathbf{S} \mathbf{V}_r$, so it stores exactly as many weights as the merged projector.

3.4.2. Data-Independent Allocation

Every unit receives its rank from the uniform rule of LSP [10]: each matrix keeps the same fraction ρ of its dense parameters, $r = \lfloor \rho \, d_{\text{in}} d_{\text{out}} / (d_{\text{in}} + d_{\text{out}}) \rfloor$, a tied group keeps the largest rank of its members, and ρ is set by bisection so that the merged model meets the target ratio. The rule depends only on the shapes of the weights, so all calibration sources compress every unit to the same rank, and the comparison isolates the two stages where the data act, namely, initialization and training. Section 4.4 repeats part of the comparison with the measured-KL allocation of LSP, which depends on the data.

3.4.3. Free Factors

The projector parameterization keeps the pretrained weights, and learns only which subspace each unit removes; to test whether this restriction matters when the calibration data differ from the evaluation data, we also train the factors $\mathbf{A}$ and $\mathbf{B}$ of Equation (1) freely using the exact merge of the same initialization, so that both parameterizations start from the same function with the same objective, data, and schedule.

3.4.4. Another Learned Compressor

To test whether the behavior is specific to LSP, we compress using the truncation of SVD-LLM followed by its recovery stage [6]. The oblique whitened truncation with the same ranks is fine-tuned with LoRA of rank 8 ($\alpha = 16$) on both factors of every compressed matrix. Updating is performed together rather than in SVD-LLM's alternating order, and is done by next-token cross-entropy on the calibration windows without using a teacher; the epoch is selected on validation cross-entropy, and the LoRA updates are merged into the factors prior to evaluation. The step budget, learning rate schedule, and data are those of LSP, with a learning rate of 10^{-4}.

3.4.5. Packing and Format

The sources differ in how their text fills a window, specifically in document length, the number of document boundaries per window, and for the exact self-samples, in the number of BOS tokens (Table 1). In the *matched-segment format*, every window of every source holds 16 segments of 127 tokens separated by EOS, after a single BOS for the models that have one. Each document contributes its consecutive 127-token spans from its start (at most 32), documents shorter than 127 tokens contribute none, and the spans are pooled and shuffled, with training and validation spans from disjoint documents. The number of boundaries per window are then identical for all sources, as are the segment length and BOS density;

only the content differs. WikiText-2 is split into its articles for this purpose, while Alpaca contributes only its examples of at least 127 tokens, 25 to 30% of which depend on the tokenizer. To test whether Alpaca's advantage in accuracy comes from its template, we also calibrate on the same examples without the template (*Alpaca, plain*: instruction, input, and response separated by blank lines) and on the responses alone.

3.4.6. Mixtures and Self-Generated Text

A matched-budget mixture calibrates on 256 windows of SlimPajama and 256 of self-generated documents. To trace the failure of exact self-samples, self-generated documents are also sampled with length limits of 128, 1024, and 2048 tokens instead of 512, and exact samples are restarted every 512 tokens (four exact 512-token samples per window, each from a fresh start token). The repetition, script, and kind of content of the self-generated text are measured along the position in the sequence (Section 4.6).

3.4.7. Optimization Settings

The learning rate was selected once on C4 (Section 3.5), and each source keeps the epoch with its own lowest validation KL. Three controls test whether the comparison depends on these choices. Every source is also trained at learning rates of 3×10^{-5} and 3×10^{-4}, which are the neighbors of the selected rate in the bracket used for C4 (10^{-3} was the worst rate on every model, cf. Appendix C), and keeps the rate with its own lowest validation KL. Instead of eight epochs, every source is trained for 24 (1536 steps); every model is also evaluated at its last epoch, a checkpoint that is fixed in advance and identical across sources.

3.5. Experimental Protocol

3.5.1. Models and Compression Ratios

We compress three decoder-only models from three families and generations: OPT-1.3B [46], Llama-3.2-1B [47], and Qwen3-1.7B-Base [48]. OPT uses multi-head attention and a ReLU multilayer perceptron (MLP); Llama-3.2 and Qwen3 use grouped-query attention and gated MLPs, and Qwen3 normalizes queries and keys. A compression ratio is the fraction of the weights of all attention and MLP projections that is removed, counted on the merged model with shared factors counted once; embeddings, the language-model head, and normalizations are not compressed, as in SVD-LLM and LSP [6,10]. We report -30% and -50%; the uniform rule meets every target within 0.1 percentage points. A scale check compresses Llama-3.1-8B [47] by 30% with all six sources and three seeds, and by 50% with one seed (Section 4.10). Instruction-following is measured on Llama-3.2-1B-Instruct [47] compressed by 30% (Section 4.11). For a reproduction check against published numbers, we also compress OPT-125M and OPT-1.3B at $-30/50/70\%$ with the WikiText-2 calibration of LSP (Appendix B).

3.5.2. Training

Projectors are trained with Adam without weight decay, a learning rate of 10^{-4} (3×10^{-5} for Llama-3.1-8B), linear warm-up over 5% of the steps followed by cosine decay, the projector strength ramped from 0 to 1 over the first epoch, direction dropout $p = 0.05$, and an orthogonality penalty of weight 0.05, as in LSP [10]. Each step processes eight windows; training runs eight epochs of 512 windows (512 steps, 8.4 M tokens) and keeps the epoch with the lowest validation KL. The learning rate was selected once per model on the validation KL of C4 calibration at -50% (Appendix C), or for Llama-3.1-8B at -30% and again at -50% (Section 4.10), and is shared by all sources to ensure that no source is tuned on evaluation data. Weights are kept in bfloat16 and frozen; projector

parameters are in float32. Gram matrices, Cholesky factors, and SVDs are computed in float64. Every run uses one NVIDIA A100 (40 or 80 GB), H100, or H200 GPU.

3.5.3. Implementation

We reimplemented LSP in PyTorch 2.6 on Hugging Face Transformers 4.57. The projection is applied in activation space, $\mathbf{x} - \mathbf{U}(\mathbf{U}^\top \mathbf{x})$, so no projected weight is ever formed, and a tied group projects its shared input once. The basis $\mathbf{U} = \mathrm{qf}(\mathbf{V})$ is computed once per optimizer step; the micro-batches accumulate the gradient with respect to a detached copy of $\mathbf{U}$, which is then propagated through the QR decomposition once. This gives the same gradient as one decomposition per micro-batch and lowers the time per epoch of OPT-1.3B from 411 s to 175 s on an A100 GPU. The gradient of the divergence with respect to the student logits, $\mathrm{softmax}(s) - p$, is formed in chunks of positions, so the vocabulary-sized softmax is never kept for backpropagation. Every reported number comes from the merged model of Equation (1), which differs from the projected model by a KL of at most 2.4×10^{-3} nats per token in bfloat16 over all runs with projectors. CPU tests on small random models check that merging is exact, that the initialization satisfies the error identities of LSP's Proposition 1, and that the single-decomposition gradient matches the per-microbatch one.

3.5.4. Ablations and Replication

Unless stated otherwise, the following ablations use Llama-3.2-1B at -50%. They cross the sources of the initialization statistics and of the training windows; change the window format, with a BOS at every document start or no BOS at all; sample the exact self-generated sequences at temperatures 0.7 and 1.3; calibrate on single SlimPajama domains (GitHub, arXiv, Books); vary the number of distinct C4 and self-generated calibration windows from 64 to 4096 at the fixed budget of 512 steps, where 4096 windows are each seen once; replace the uniform allocation by the measured-KL allocation of LSP; and train free factors on Llama-3.2-1B and OPT-1.3B. Training-free baselines and the analysis of the removed subspaces cover all three models at both ratios. The controls of Section 3.4 cover the three models at -50% (matched-segment format), at both ratios (Alpaca templates, mixture), Qwen3-1.7B and Llama-3.2-1B at -30% (length limits and restarts of self-generated text), Llama-3.2-1B and OPT-1.3B at -30% (SVD-LLM with LoRA), and Llama-3.2-1B at -50% (optimization settings, data size with learning curves). The full grid of three models, two ratios, and six sources was also repeated with three seeds (Section 4.9). The main grid and the ablations above are single runs with 32 held-out windows per source. The three-seed repetition, the controls of Section 3.4, and the experiments on Llama-3.1-8B and Llama-3.2-1B-Instruct use an extended evaluation: 128 held-out windows per source, the six independent domains described below, and a record of the result of every evaluation item.

3.5.5. Evaluation

We report perplexity on the WikiText-2 test split and on the first 1100 documents of the C4 validation shard, with non-overlapping 2048-token windows that each start with BOS (the window format of calibration and of zero-shot inputs); Appendix D reports the convention of GPTQ and SparseGPT [39,40], where only the first window starts with BOS, and also discusses the difference. Domain-level perplexity uses 32 windows from the SlimPajama validation split for each of its seven RedPajama subsets. Zero-shot accuracy is measured with the LM Evaluation Harness, version 0.4.9.2 [49], on ARC-Easy and ARC-Challenge [50], PIQA [51], HellaSwag [52], WinoGrande [53], OpenBookQA [54], and BoolQ [55]; we report plain accuracy and its mean over the seven tasks. Finally, the divergence of the compressed model from the dense one, $\mathrm{KL}(p\|q)$ averaged over tokens, is measured on 32 held-out windows of each source (128 in the extended evalua-

tion): WikiText-2 test chunks, C4 validation documents after the 1100 used for perplexity, SlimPajama validation documents, 2000 Alpaca examples excluded from calibration, and a separate pool of exact samples of the dense model drawn with a different seed. By Proposition 1, the last of these estimates the sequence-level divergence per token.

3.5.6. Independent Domains, Behavior, and Cost

The held-out sets above are drawn from the calibration sources themselves. The extended evaluation also measures the divergence and perplexity on 64 windows of each of six domains chosen independently of every source: human-written instruction–response pairs (Dolly-15k [56]), grade-school mathematics with worked solutions (GSM8K [57]), biomedical abstracts (PubMedQA [58]), Python programs with their tests (MBPP [59]), sentences in eight languages other than English (OPUS-100 [60]), and nineteenth-century books (PG-19 [61], one window per book). Beyond perplexity and multiple choice, the models of the first seed are evaluated on open-ended generation (128-token continuations of 200 held-out C4 prompts sampled at temperature one and greedily, scored by the dense model's negative log-likelihood of the continuation, the rate of repeated 4-grams, and distinct-n [62]), on long contexts (perplexity on 8192-token windows of sixteen PG-19 books, which is four times the calibration length, and retrieval of a passkey hidden in about 2000, 4000, and 8000 tokens of filler text [63]), and on generative tasks at the harness's default few-shot settings and limit of 256 generated tokens (GSM8K and NQ-Open [64], for Qwen3-1.7B and Llama-3.1-8B, the models that solve them). Instruction following is measured with IFEval [65] on Llama-3.2-1B-Instruct in its chat format compressed by 30% after calibration on C4, SlimPajama, Alpaca in the model's chat format (instruction as the user turn, response as the assistant turn), self-generated documents, and the model's own responses to Alpaca instructions outside the held-out set, sampled at temperature one; every calibration document of this model starts with BOS, as does its chat format. Memory and latency are measured in bfloat16 on one A100 GPU for the dense model and the merged factors at each ratio. The ranks, and hence the cost, do not depend on the calibration source.

3.5.7. Statistical Analysis

The main grid and its ablations are single runs, and the three-seed repetition reports the mean and the standard deviation over seeds. The seed selects and orders the calibration windows and draws the direction dropout; for the self-generated sources, each seed also samples its own pool, so their variation includes the draw of the data. For differences between two calibration sources, the extended evaluation records the result of every evaluation item (each window of perplexity and divergence, each benchmark question); a difference is reported with a 95% confidence interval from a paired bootstrap [66] that resamples items within each task for accuracy after averaging every item over the three seeds. Accuracy is reported as the mean over the seven tasks, and because the compressed models score close to the majority-class rate of BoolQ (62.2%), also as the mean over the six other tasks and as the mean of the chance-normalized accuracies $(a - c)/(1 - c)$, with c the chance rate of each task.

4. Results

4.1. Training-Free Truncation

Before any training, the whitened initialization depends heavily on the calibration source (Figure 2, hollow markers). At -30%, its perplexity varies across the six sources by a factor of 2.7 to 7.3 on OPT-1.3B, 18 to 108 on Llama-3.2-1B, and 4 to 32 on Qwen3-1.7B, depending on the evaluation corpus. At -50%, the untrained Llama-3.2-1B and Qwen3-1.7B have perplexities in the thousands from every source, with one exception discussed

below. Zero-shot accuracy is close to chance for every untrained model (32 to 41%), so perplexity is the informative measure at this stage.

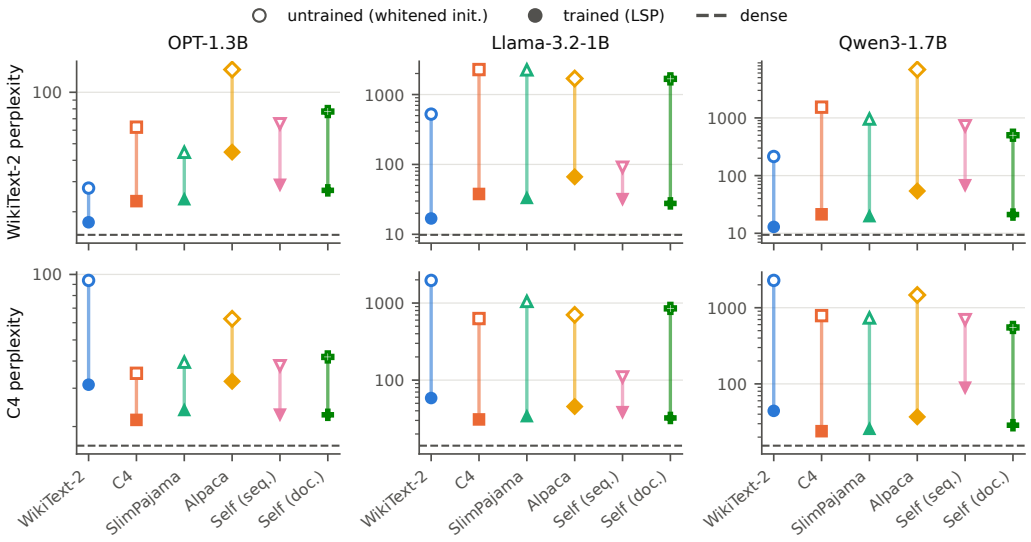


Figure 2. Perplexity of the models compressed by 30% before training (hollow markers, the whitened initialization) and after training (filled markers) for each calibration source. Dashed line indicates a dense model. Logarithmic scale. Training lowers perplexity by one to two orders of magnitude and shrinks the spread across sources, but the ranking of sources becomes domain-specific. The same plot at −50% is Figure A1.

Two regularities hold. First, a corpus helps itself: among the real sources, WikiText-2 calibration gives the lowest untrained WikiText-2 perplexity on all three models at −30% and C4 calibration the lowest C4 perplexity on OPT-1.3B and Llama-3.2-1B; on Qwen3-1.7B, SlimPajama (which contains C4 text) does slightly better on C4. Second, self-generated text gives competitive training-free truncations. On Qwen3-1.7B at −30%, self-generated documents give the lowest untrained C4 perplexity of all sources (551, against 734 to 2285 for the real corpora), and on Llama-3.2-1B exact self-samples give by far the best untrained model at both ratios (WikiText-2 perplexity 91 against 525 to 2277 at −30%; 454 against 4716 to 12,529 at −50%). The Llama exception comes from the density of BOS tokens in the calibration windows (Section 4.4 and Appendix D); its exact self-samples contain 7.8 BOS tokens per window. Every other source contains exactly one, and self-generated documents packed with a single BOS per window do not share the advantage.

The orthogonal recast that LSP needs as its starting point is not the cause of this dependence (Table 2). The oblique truncation of SVD-LLM has lower perplexity than the recast in 70 of the 72 cells of the table, by up to a factor of 68, as the layer-wise analysis of LSP predicts ([10] Proposition 1); the two exceptions are the exact self-samples of Llama-3.2-1B at −30%. Its dependence on the source is smaller but of the same kind. At −30%, its perplexity varies across the six sources by a factor of 1.6 to 2.3 on OPT-1.3B, 9 to 47 on Llama-3.2-1B, and 2.7 to 20 on Qwen3-1.7B. C4 calibration again gives the lowest C4 perplexity on OPT-1.3B and Qwen3-1.7B, and self-generated documents the lowest SlimPajama perplexity on Qwen3-1.7B; on Llama-3.2-1B, the exact self-samples are best on both corpora. In zero-shot accuracy, the oblique truncation of OPT-1.3B at −30% is within 1.3 points of the trained model for every source, though not in perplexity. In the other settings, training adds 1.5 to 11.2 points for every source except the exact self-samples of Qwen3-1.7B at −30% (Section 4.2). Both truncations vary across sources far more than the 3% at most that SVD-LLM reports for LLaMA-7B compressed by 20% [6], a setting with an older model and a lower ratio.

Table 2. Perplexity of the two training-free truncations on C4 and SlimPajama (SP, geometric mean over its seven domains), by calibration source: the orthogonal recast that initializes LSP (rec.) and the oblique whitened truncation of SVD-LLM (obl.), with the same ranks and the same number of stored weights.

Calibration	OPT-1.3B				Llama-3.2-1B				Qwen3-1.7B			
	C4		SP		C4		SP		C4		SP	
	rec.	obl.	rec.	obl.	rec.	obl.	rec.	obl.	rec.	obl.	rec.	obl.
−30%												
WikiText-2	93.9	40.3	134	31.3	1967	1176	7556	3555	2285	251	17,321	1257
C4	35.1	24.5	59.6	27.3	629	460	4185	1955	788	94.3	8752	504
SlimPajama	39.5	28.1	18.3	13.7	1050	628	1428	569	734	118	1871	116
Alpaca	62.5	30.6	57.9	24.9	702	687	2531	1347	1469	144	8565	501
Self (seq.)	38.1	25.5	41.1	18.2	110	131	70.3	75.5	698	145	944	84.3
Self (doc.)	41.8	26.0	55.4	20.3	853	591	1218	540	551	106	575	61.5
−50%												
WikiText-2	1608	272	3491	312	10,903	5185	27,904	13,558	11,878	2070	40,036	10,608
C4	265	66.9	1701	142	3423	1885	18,807	7696	11,444	405	59,037	4068
SlimPajama	342	95.6	97.3	32.7	3003	1448	6314	2748	15,062	528	44,925	1403
Alpaca	534	133	548	111	8336	2321	37,759	5915	11,799	2705	34,501	20,846
Self (seq.)	355	83.3	875	100	473	410	393	323	12,314	461	32,059	468
Self (doc.)	379	96.1	1230	139	4810	1948	7490	2683	6693	485	24,384	460

4.2. Trained Projections

Training changes the picture in two ways (Figure 2, Table 3). First, it lowers perplexity by one to two orders of magnitude and compresses the spread across sources: on Llama-3.2-1B at −30%, the WikiText-2 perplexities of the six sources span 16.9 to 66.5 after training, against 91 to 2277 before. The dependence on the source does not vanish, however, instead turning into a dependence on domain. Each model reaches its lowest WikiText-2 perplexity when calibrated on WikiText-2 and its lowest C4 perplexity when calibrated on C4, at both ratios and in each of the three seeds of the repetition (Section 4.9). Calibrating on WikiText-2 roughly doubles C4 perplexity relative to calibrating on C4 (Llama-3.2-1B, −30%: 58.5 against 30.9; OPT-1.3B, −50%: 50.3 against 27.7). The broadest evaluation consists of the seven SlimPajama domains, and is best served by SlimPajama calibration on OPT-1.3B and Qwen3-1.7B and by self-generated documents on Llama-3.2-1B, at both ratios; on Qwen3-1.7B, the three-seed means of the two sources are within 4%. Therefore, a WikiText-2 perplexity measured after WikiText-2 calibration (the most common number in the low-rank compression literature) measures fit to the calibration domain as much as compression quality.

Zero-shot accuracy behaves differently, varying much less across sources, by 1.7 to 3.5 points on OPT-1.3B and Llama-3.2-1B and 5.0 to 5.4 points on Qwen3-1.7B, excluding at −30% the exact self-samples discussed below, against a seed standard deviation of 0.3 points in the median and at most 1.1 points, excluding the exact self-samples of Qwen3-1.7B (Section 4.9). The most accurate source is Alpaca, on Llama-3.2-1B and Qwen3-1.7B; on OPT-1.3B, SlimPajama leads only through BoolQ, a yes/no task on which the compressed models score between 42 and 66%, around its majority-class rate of 62% (Table A1). Without BoolQ, Alpaca is the most accurate source on every model and ratio. However, its perplexities are among the worst. WikiText-2 calibration gives the lowest accuracy of all real sources on every model and ratio.

Table 3. Models compressed by learned subspace projections, by calibration source: WikiText-2 (Wiki), C4 and SlimPajama (SP, geometric mean over its seven domains), perplexity, and mean zero-shot accuracy over seven tasks (Acc, %). Bold: best per column and ratio. Per-task accuracies are in Table A1.

Calibration	OPT-1.3B				Llama-3.2-1B				Qwen3-1.7B			
	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc
Dense	14.7	16.3	9.5	47.6	9.8	14.1	8.0	53.3	9.4	15.4	7.7	59.0
−30%												
WikiText-2	**17.4**	31.1	25.5	42.3	**16.9**	58.5	132	41.4	**12.9**	44.2	86.0	44.6
C4	23.1	**21.5**	17.2	43.4	37.8	**30.9**	63.5	43.1	21.3	**23.9**	37.4	47.7
SlimPajama	23.7	23.8	**12.4**	**43.4**	33.2	34.1	21.3	42.0	19.8	25.7	**14.7**	46.7
Alpaca	44.7	32.3	25.5	43.1	66.5	45.4	51.3	**43.3**	54.1	36.9	42.8	**49.8**
Self (seq.)	28.8	22.7	16.5	42.5	31.9	38.4	25.2	41.4	68.1	89.2	64.5	37.0
Self (doc.)	26.7	22.6	16.4	41.7	27.7	32.3	**20.8**	42.0	21.1	28.7	16.3	44.4
−50%												
WikiText-2	**20.8**	50.3	55.3	38.7	**22.8**	102	262	36.3	**16.3**	74.7	170	39.9
C4	39.3	**27.7**	27.5	40.1	62.7	**45.3**	109	39.3	33.7	**32.6**	63.3	43.1
SlimPajama	38.8	32.8	**15.6**	**41.4**	59.6	51.5	33.7	38.1	32.3	37.1	**23.3**	41.5
Alpaca	120	58.9	53.5	41.2	137	76.7	99.3	**39.8**	104	60.6	91.8	**44.9**
Self (seq.)	47.2	30.1	25.3	39.5	53.2	59.3	39.0	38.4	90.2	123	73.7	39.9
Self (doc.)	46.5	31.3	27.9	38.3	45.2	48.2	**32.2**	39.2	36.2	43.7	25.4	40.5

The three-seed repetition with paired comparisons over benchmark items qualifies these rankings (Section 4.9).

Without BoolQ, Alpaca leads the next most accurate source by 0.9 to 1.9 points, with 95% intervals that exclude zero, in five of the six settings, and ties with C4 on OPT-1.3B at −30% (difference −0.0 points, interval −0.6 to 0.6). WikiText-2 calibration remains the least accurate real source in every setting, significantly so in all but one (Llama-3.2-1B at −50%: −0.3 points, interval −1.0 to 0.3). BoolQ needs separate care (Table A2), as 32 of the 36 three-seed means of the compressed models lie below its majority-class rate of 62.2%. This is the accuracy of always answering yes, so the differences on BoolQ reflect shifts in the answer bias more than reading comprehension. Which source ranks first depends on the summary; Alpaca is first by the six-task mean in five of the six settings, but only in three by the mean of chance-normalized accuracies, which rescales each task by its distance from chance and hence weights the two-choice tasks (BoolQ among them) 1.5 times as heavily as the four-choice ones; C4 or SlimPajama are first in the others. On Llama-3.1-8B (last two rows of Table A2; Section 4.9), for which the compressed models all exceed the majority-class rate of BoolQ at −30% (67.3 to 73.2%) but with one exception fall below it at −50% (53.6 to 64.1%), Alpaca's lead over C4 is within the noise at −30% (0.6 points, interval −0.1 to 1.2) and significant at −50% (2.1 points, interval 1.2 to 3.0, one seed); WikiText-2 is again significantly the least accurate real source at both ratios.

The advantage of Alpaca does not come from its template (Table 4). Without the template, the same examples give the same or higher accuracy without BoolQ on every model and ratio, by 0.1 to 0.8 points. The responses alone without instructions remain within 0.7 points of the full examples, above C4 at −50%. With seed standard deviations of 0.1 to 0.7 points for Alpaca, these single-run differences are small; however, none of them favors the template, as the advantage comes from what the examples say rather than from how they are formatted. Removing the template also lowers the perplexity of Alpaca calibration in four of the six settings.

Table 4. Alpaca with its template, without it (plain), and its responses alone, against C4 and SlimPajama; seed 0. Acc (6) and Acc (7): mean zero-shot accuracy (%) without and with BoolQ.

Model	Ratio	Calibration	Wiki	C4	SP	Acc (6)	Acc (7)
OPT-1.3B	−30%	Alpaca	44.2	31.1	25.8	42.1	43.1
OPT-1.3B	−30%	Alpaca, plain	44.6	31.9	28.5	42.7	43.9
OPT-1.3B	−30%	C4	23.7	21.7	17.3	42.3	43.5
OPT-1.3B	−30%	SlimPajama	24.0	23.5	12.3	40.8	43.2
OPT-1.3B	−50%	Alpaca	101	52.1	43.4	39.6	41.1
OPT-1.3B	−50%	Alpaca, plain	88.9	50.0	40.7	40.4	41.6
OPT-1.3B	−50%	Alpaca, responses	88.9	46.6	39.2	39.9	41.8
OPT-1.3B	−50%	C4	40.7	28.0	28.0	39.4	40.6
OPT-1.3B	−50%	SlimPajama	39.2	33.3	15.9	37.7	41.0
Llama-3.2-1B	−30%	Alpaca	66.2	45.3	50.8	42.0	43.4
Llama-3.2-1B	−30%	Alpaca, plain	71.5	44.7	56.6	42.0	42.7
Llama-3.2-1B	−30%	C4	38.0	30.8	62.5	41.0	43.2
Llama-3.2-1B	−30%	SlimPajama	33.3	34.1	21.3	39.0	42.0
Llama-3.2-1B	−50%	Alpaca	135	76.4	99.2	37.8	39.5
Llama-3.2-1B	−50%	Alpaca, plain	121	72.2	97.1	38.4	40.7
Llama-3.2-1B	−50%	Alpaca, responses	133	70.6	101	38.4	39.9
Llama-3.2-1B	−50%	C4	64.5	45.3	108	36.6	39.2
Llama-3.2-1B	−50%	SlimPajama	59.7	52.0	34.3	34.8	37.9
Qwen3-1.7B	−30%	Alpaca	51.0	38.0	53.1	47.0	49.7
Qwen3-1.7B	−30%	Alpaca, plain	43.1	34.9	38.4	47.5	50.1
Qwen3-1.7B	−30%	C4	21.3	23.9	37.4	44.8	47.8
Qwen3-1.7B	−30%	SlimPajama	19.8	25.7	14.7	43.4	46.7
Qwen3-1.7B	−50%	Alpaca	95.9	59.4	87.6	42.1	45.0
Qwen3-1.7B	−50%	Alpaca, plain	83.8	55.3	75.9	42.3	45.1
Qwen3-1.7B	−50%	Alpaca, responses	85.4	52.0	70.7	41.8	44.3
Qwen3-1.7B	−50%	C4	33.8	32.6	63.4	39.9	42.9
Qwen3-1.7B	−50%	SlimPajama	31.9	36.4	22.9	38.5	41.7

Exact self-samples are the one source that fails, which occurs on one model. On Qwen3-1.7B, the compressed model trained on self-samples loses on real text, with 37.0% accuracy at −30% against 44.4 to 49.8% for every other source along with a C4 perplexity of 89.2. In the three seeds of the repetition, its C4 perplexity is 30.5 to 53.5 at −30%, against three-seed means of 23.9 to 43.4 for the other sources, and 155 to 227 at −50% against 32.6 to 74.5 (Section 4.6). Self-generated documents, which end at the model's first end-of-text token, are in the top group on every model: on Qwen3-1.7B at −30%, they give 44.4% accuracy and C4 perplexity 28.7, while on Llama-3.2-1B they provide the best SlimPajama perplexity at both ratios.

4.3. What Each Source Preserves

The divergence from the dense model on held-out text of every source makes the specialization clear (Figure 3). On the held-out set of each real corpus, the divergence is lowest when calibrating on that corpus for every model and ratio. There is one near-tie: on Llama-3.2-1B at −50%, self-generated documents match SlimPajama on SlimPajama (1.258 against 1.262 nats per token). In the three-seed repetition, with 128 held-out windows per set, the matching source has the lowest divergence in 86 of the 90 comparisons (five held-out sets in each of the 18 combinations of model, ratio and seed); the four exceptions are SlimPajama on Llama-3.2-1B, where self-generated documents and SlimPajama calibration tie (ratio of the three-seed divergences 0.99, 95% paired-bootstrap interval 0.98 to 1.01 at −30% and 0.97 to 1.00 at −50%). On held-out self-samples, which estimate the sequence-level divergence of Proposition 1, self-generated calibration is always best: the exact

sequences on Llama-3.2-1B, and the documents on the two other models. Away from the diagonal, narrow sources are costly; calibrating on WikiText-2 multiplies the divergence on another set by up to 9.6 relative to the best source for that set, and Alpaca does so by up to 5.4 (Table 5). Across all five held-out sets, self-generated documents are the most uniformly faithful calibration by this summary. They have the lowest geometric-mean relative divergence in five of the six settings, 1.4% above C4 in the sixth (OPT-1.3B at −50%), and the lowest maximum on Llama-3.2-1B and Qwen3-1.7B at both ratios, while C4 has the lowest maximum on OPT-1.3B; they also reach this without being the best source for any real corpus. Exact self-samples generalize well on OPT-1.3B and Llama-3.2-1B but poorly on Qwen3-1.7B, where they are the worst of all sources at −30% and the second-worst at −50%. Drift alone does not explain this difference, since the exact samples of Llama-3.2-1B reach a higher entropy (Figure 1; Section 4.6).

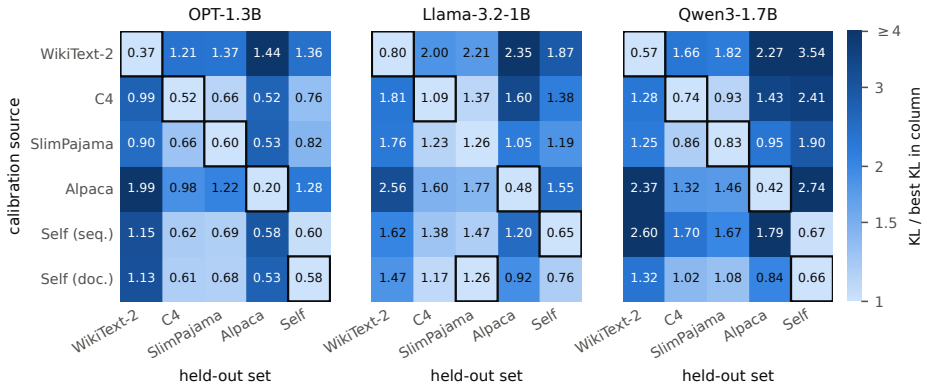


Figure 3. Divergence $\mathrm{KL}(p\|q)$ between the dense and the compressed model (nats per token) on held-out windows of each source (columns) for each calibration source (rows), at −50%. Colors give the ratio to the best calibration source for that held-out set, while black outlines mark the best source for each held-out set. Self denotes exact samples of the dense model, meaning that this column estimates the sequence-level divergence (Proposition 1). The same matrix at −30% is shown in Figure A2.

Table 5. How uniformly each calibration source preserves the dense model: for every held-out set of Figure 3, the divergence is shown relative to the best source for that set, summarized over the five sets by its geometric mean (geo) and its maximum (max). Bold: lowest per column.

Calibration	−30%						−50%					
	OPT		Llama		Qwen3		OPT		Llama		Qwen3	
	Geo	Max	Geo	Max	Geo	Max	Geo	Max	Geo	Max	Geo	Max
WikiText-2	2.79	9.57	2.31	5.21	3.22	7.35	2.47	7.31	2.14	4.87	2.70	5.45
C4	1.64	**2.84**	1.92	3.53	2.58	8.92	**1.60**	**2.69**	1.77	3.32	1.99	3.63
SlimPajama	1.69	2.90	1.61	2.27	1.84	3.52	1.64	2.70	1.58	2.21	1.76	2.88
Alpaca	2.18	5.27	1.82	3.50	2.47	4.98	2.16	5.43	1.74	3.21	2.22	4.15
Self (seq.)	1.74	3.33	1.53	2.46	3.75	7.38	1.67	3.14	1.49	2.49	2.47	4.56
Self (doc.)	**1.61**	2.96	**1.37**	**1.91**	**1.55**	**2.26**	1.62	3.10	**1.34**	**1.90**	**1.53**	**2.30**

This summary depends on the evaluation sets (Table 6, three-seed means). Over the five held-out sets, self-generated documents have the lowest geometric-mean relative divergence in all six settings (tied with C4 on OPT-1.3B at −50%), and remain the most common best source when the five sets are weighted at random, with weights drawn uniformly from the simplex (34–77% of 5000 draws, depending on the setting). Without the held-out samples of the dense model, which favor self-generated calibration, they remain the most uniform only on Llama-3.2-1B; C4 is more uniform on OPT-1.3B and SlimPajama on Qwen3-1.7B. On the six domains chosen independently of every source, they are the

most uniform on Llama-3.2-1B, where they are the best source for every domain, and on Qwen3-1.7B, where they come within 16% of the best source on every domain, while SlimPajama is clearly the most uniform on OPT-1.3B (1.03 against 1.35). On Llama-3.1-8B (last two rows), they are the most uniform by all three summaries at both ratios. Therefore, self-generated documents are the most uniformly faithful calibration of the Llama models and of Qwen3-1.7B on independent domains, but not of OPT-1.3B.

A matched-budget mixture of SlimPajama and self-generated documents combines the two (Table 7). It is the most uniform of the seven calibrations on the four real held-out sets in five of the six settings and tied in the sixth, the most uniform on all three summaries on Qwen3-1.7B, and within 11% of the most uniform in every setting and summary. Its perplexities are within 6% of the better of its two components on every corpus, and lower than both in 10 of the 18 cases, at an accuracy within a single point of both.

Table 6. How the uniform-fidelity summary depends on the evaluation sets (three-seed means; one seed for Llama-3.1-8B at −50%): geometric-mean divergence relative to the best source for each set for self-generated documents and the best other source (named) over the five held-out sets of Table 5, the four real ones (without the model's own samples), and the six independent domains. Bold: the lower of the two.

Model	Ratio	Five Sets		Four Real Sets		Six Independent Domains	
		Self (Doc.)	Best Other	Self (Doc.)	Best Other	Self (Doc.)	Best Other
OPT-1.3B	−30%	**1.59**	1.64 (C4)	1.79	**1.70** (C4)	1.36	**1.03** (SlimPajama)
OPT-1.3B	−50%	**1.59**	1.59 (C4)	1.79	**1.67** (C4)	1.35	**1.03** (SlimPajama)
Llama-3.2-1B	−30%	**1.35**	1.47 (Self (seq.))	**1.36**	1.47 (SlimPajama)	**1.00**	1.14 (Self (seq.))
Llama-3.2-1B	−50%	**1.32**	1.44 (Self (seq.))	**1.35**	1.47 (SlimPajama)	**1.00**	1.14 (Self (seq.))
Qwen3-1.7B	−30%	**1.53**	1.90 (SlimPajama)	1.70	**1.55** (SlimPajama)	**1.06**	1.41 (SlimPajama)
Qwen3-1.7B	−50%	**1.48**	1.98 (SlimPajama)	1.63	**1.51** (SlimPajama)	**1.05**	1.36 (SlimPajama)
Llama-3.1-8B	−30%	**1.37**	1.58 (Self (seq.))	**1.35**	1.54 (SlimPajama)	**1.02**	1.31 (Self (seq.))
Llama-3.1-8B	−50%	**1.33**	1.51 (Self (seq.))	**1.32**	1.50 (SlimPajama)	**1.00**	1.28 (Self (seq.))

Table 7. A matched-budget mixture of SlimPajama and self-generated documents (256 windows each), seed 0, showing perplexity, accuracy without BoolQ, and geometric-mean divergence relative to the best of the seven calibrations over the five held-out sets and the six independent domains.

Model	Ratio	Calibration	Wiki	C4	SP	Acc	Rel. KL, 5 Sets	Rel. KL, 6 Domains
OPT-1.3B	−30%	SlimPajama	24.0	23.5	12.3	40.8	1.65	1.02
OPT-1.3B	−30%	Self (doc.)	26.9	22.6	16.4	41.1	1.57	1.41
OPT-1.3B	−30%	SlimPajama + Self (doc.)	25.5	22.1	12.9	41.0	1.47	1.10
OPT-1.3B	−50%	SlimPajama	39.2	33.3	15.9	37.7	1.68	1.06
OPT-1.3B	−50%	Self (doc.)	47.6	30.9	27.5	38.0	1.60	1.42
OPT-1.3B	−50%	SlimPajama + Self (doc.)	39.6	29.0	16.8	37.7	1.45	1.08
Llama-3.2-1B	−30%	SlimPajama	33.3	34.1	21.3	39.0	1.62	1.43
Llama-3.2-1B	−30%	Self (doc.)	27.6	32.3	20.8	39.1	1.36	1.01
Llama-3.2-1B	−30%	SlimPajama + Self (doc.)	28.0	32.1	20.2	39.1	1.38	1.12
Llama-3.2-1B	−50%	SlimPajama	59.7	52.0	34.3	34.8	1.57	1.41
Llama-3.2-1B	−50%	Self (doc.)	45.4	48.4	32.5	35.6	1.32	1.00
Llama-3.2-1B	−50%	SlimPajama + Self (doc.)	45.2	47.9	31.0	35.5	1.35	1.11
Qwen3-1.7B	−30%	SlimPajama	19.8	25.7	14.7	43.4	1.91	1.42
Qwen3-1.7B	−30%	Self (doc.)	21.1	28.7	16.3	43.5	1.55	1.07
Qwen3-1.7B	−30%	SlimPajama + Self (doc.)	19.2	25.6	14.8	44.4	1.40	1.06
Qwen3-1.7B	−50%	SlimPajama	31.9	36.4	22.9	38.5	1.96	1.33
Qwen3-1.7B	−50%	Self (doc.)	36.2	43.7	25.4	37.9	1.51	1.07
Qwen3-1.7B	−50%	SlimPajama + Self (doc.)	30.9	37.6	23.0	38.7	1.42	1.07

The domain profile shows where specialization costs most (Figure 4). Code and the multilingual Wikipedia subset of SlimPajama, which inherits the 20-language Wikipedia dumps of RedPajama [42], suffer the most from narrow English calibration. At −30%, WikiText-2 calibration multiplies the code perplexity of Llama-3.2-1B by 144 relative to the dense model, while Alpaca multiplies its multilingual Wikipedia perplexity by 97. SlimPajama, which contains both, and self-generated documents, which the models write in several languages and formats, keep every domain within a factor of 7 of the dense model at that ratio.

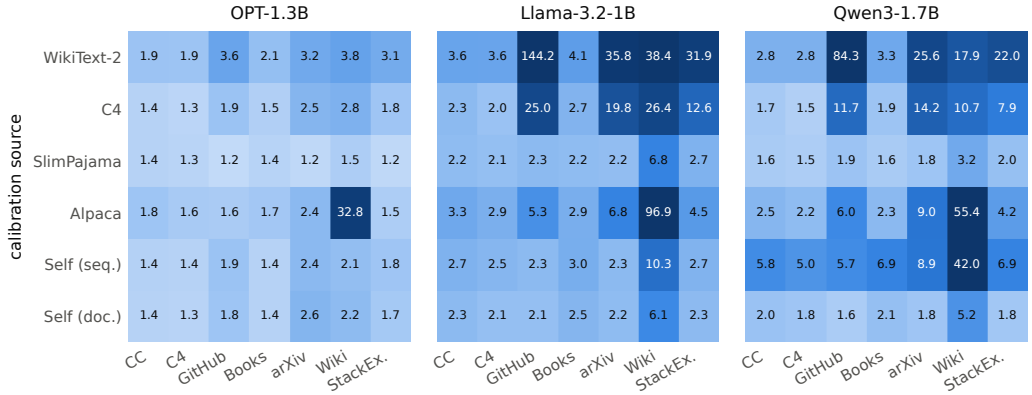


Figure 4. Perplexity on the seven SlimPajama domains relative to the dense model, after training, at −30% (CC: CommonCrawl; Wiki: multilingual Wikipedia; StackEx.: StackExchange). Each cell gives the ratio of the compressed model's perplexity to the dense model's, and its color encodes the same ratio on a logarithmic scale shared by the three panels, from light (close to the dense model) to dark blue (far from it). The same profile at −50% is shown in Figure A3.

Calibrating on a single domain takes the specialization further (Table 8). On Llama-3.2-1B at −50%, calibrating on the GitHub, arXiv, or Books subset of SlimPajama instead of the whole mixture lowers the perplexity on that domain from 3.56 to 1.60 times that of the dense model for code, from 3.21 to 1.65 for arXiv and from 3.40 to 2.37 for books. However, the other domains pay for this; the code-calibrated model has 9 to 49 times the dense perplexity on web, book, and Wikipedia text and the arXiv-calibrated model 185 times on Wikipedia, against at most 13.5 times for the mixture. Related domains benefit together; code calibration also gives the lowest StackExchange perplexity of the four calibrations. Zero-shot accuracy changes by at most 1.0 points.

Table 8. Single-domain calibration of Llama-3.2-1B compressed by 50%, showing perplexity on each SlimPajama domain relative to the dense model after calibrating on the whole SlimPajama mixture or on one of its subsets along with mean zero-shot accuracy (Acc, %). Bold: best per column.

Calibration	CC	C4	GitHub	Books	arXiv	Wiki	StackEx.	Acc
SlimPajama (all)	**3.39**	**3.11**	3.56	3.40	3.21	**13.54**	4.26	38.1
GitHub	10.96	9.39	**1.60**	11.90	4.46	49.41	**3.51**	37.1
arXiv	8.49	8.20	6.52	11.90	**1.65**	185	6.73	**38.4**
Books	4.51	4.57	38.19	**2.37**	39.59	121	16.10	37.8

4.4. Where the Data Act

The calibration data act at two stages: the Gram statistics of the initialization, and the training windows. Crossing them separates the two (Table 9, top). Both matter, and the domain of the initialization survives training. Training on C4 from the statistics of self-generated documents instead of those of C4 lowers the SlimPajama perplexity of the trained model from 109 to 45.8 and its WikiText-2 perplexity from 62.7 to 49.4, at a cost of 1.6

in C4 perplexity. Training on WikiText-2 from SlimPajama statistics instead of WikiText-2 statistics lowers the C4 perplexity of the trained model from 102 to 76.2 and its SlimPajama perplexity from 262 to 71.3, while its WikiText-2 perplexity rises only from 22.8 to 25.5. Conversely, WikiText-2 statistics carry their domain into a model trained on SlimPajama, lowering its WikiText-2 perplexity from 59.6 to 35.8.

Table 9. Ablations on Llama-3.2-1B compressed by 50%, showing initialization statistics and training windows from different sources, calibration window format, and sampling temperature of the exact self-samples. Columns as in Table 3.

Variant	Wiki	C4	SP	Acc
Initialization statistics vs. training windows				
C4 init, C4 training	62.7	45.3	109	39.3
Self (doc.) init, C4 training	49.4	46.9	45.8	38.3
Self (doc.) init, Self (doc.) training	45.2	48.2	32.2	39.2
C4 init, Self (doc.) training	51.0	46.7	40.1	38.4
WikiText-2 init, WikiText-2 training	22.8	102	262	36.3
SlimPajama init, WikiText-2 training	25.5	76.2	71.3	38.1
SlimPajama init, SlimPajama training	59.6	51.5	33.7	38.1
WikiText-2 init, SlimPajama training	35.8	55.8	41.5	39.1
Window format (C4)				
BOS at window start	62.7	45.3	109	39.3
BOS at every document start	61.5	45.2	109	40.3
Plain chunks, no BOS	68.5	46.7	119	39.2
Self-generated sequences: temperature				
$T = 0.7$	123	124	78.7	38.1
$T = 1.0$	53.2	59.3	39.0	38.4
$T = 1.3$	120	131	83.5	36.9

The removed subspaces show why the initialization persists (Figure A4, Appendix A). Training moves them little; the overlap $\|\mathbf{U}_0^\top \mathbf{U}\|_F^2/k$ between the subspaces that a unit removes before ($\mathbf{U}_0$) and after ($\mathbf{U}$) training averages 0.996 to 0.997 for every model and ratio, where 1 means identical subspaces and k/d (here, 0.50 to 0.65) is expected for unrelated instances. Subspaces initialized from different sources overlap less, 0.85 to 0.90 on average, and training does not bring them closer; for every pair of sources, model, and ratio, the overlap is slightly lower after training than before. Thus, within the budget of 512 steps, training refines the subspaces it starts from rather than replacing them, and the two stages can be calibrated on different data; broad statistics for the initialization and domain data for training keep most of the domain fit at a much smaller cost elsewhere.

The format of the calibration windows also matters on Llama-3.2-1B (Table 9, middle; Appendix D). Every compressed Llama-3.2-1B model collapses on inputs that do not start with the BOS token, unless it was calibrated on such inputs. A model calibrated on C4 windows without BOS has nearly the same perplexity on BOS-less text as on BOS-prefixed text, at a small cost on the latter. Conversely, a BOS at every document start, as in the model's own samples, improves the untrained truncation severalfold but not the trained model.

Equalizing the presentation of the data leaves their content effects in place (Table 10). In the matched-segment format, where every window of every source holds 16 segments of 127 tokens after a single BOS and the sources differ only in content, each real corpus keeps the lowest divergence on its own held-out text in 11 of the 12 cases (three models, four corpora); the exception is SlimPajama on OPT-1.3B, where self-generated documents are 2% lower. Each corpus also gives the lowest perplexity on itself in 7 of the 9 cases; in the other two, self-generated documents are better: on C4 for OPT-1.3B, and as with default packing, on SlimPajama for Llama-3.2-1B. Alpaca remains the most accurate source without BoolQ

on all three models, by 0.2 to 0.4 points. The format itself matters little on Llama-3.2-1B and Qwen3-1.7B, for which the WikiText-2 perplexities change by at most 21% between the two formats; however, on OPT-1.3B the segments raise the WikiText-2 perplexity of every source by a factor of 1.4 to 2.3. The specialization to the calibration domain survives both formats.

Table 10. Default packing (packed) against the matched-segment format (segments: 16 segments of 127 tokens per window, a single BOS), all models at −50%, seed 0: perplexities, accuracy without BoolQ, and divergence on the held-out sets of the four real sources. In the segment format, WikiText-2 is split into articles.

Model	Calibration	Format	Wiki	C4	SP	Acc	KL Wiki	KL C4	KL SP	KL Alpaca
OPT-1.3B	WikiText-2	packed	20.8	49.8	56.6	36.5	0.411	1.175	1.389	1.475
OPT-1.3B	WikiText-2	segments	48.8	63.0	96.4	36.9	1.273	1.217	1.621	1.125
OPT-1.3B	C4	packed	40.7	28.0	28.0	39.4	1.045	0.524	0.688	0.528
OPT-1.3B	C4	segments	80.5	38.3	46.8	39.4	1.701	0.623	0.981	0.533
OPT-1.3B	SlimPajama	packed	39.2	33.3	15.9	37.7	0.926	0.698	0.642	0.538
OPT-1.3B	SlimPajama	segments	62.3	53.0	21.0	39.0	1.405	0.761	0.871	0.455
OPT-1.3B	Alpaca	packed	101	52.1	43.4	39.6	1.854	0.931	1.179	0.190
OPT-1.3B	Alpaca	segments	209	61.5	49.4	39.8	2.601	0.984	1.337	0.191
OPT-1.3B	Self (doc.)	packed	47.6	30.9	27.5	38.0	1.123	0.595	0.711	0.550
OPT-1.3B	Self (doc.)	segments	65.1	36.6	33.5	38.3	1.440	0.664	0.852	0.570
Llama-3.2-1B	WikiText-2	packed	22.8	102	261	35.1	0.890	1.970	2.151	2.367
Llama-3.2-1B	WikiText-2	segments	27.5	104	350	35.5	1.051	1.921	2.173	2.507
Llama-3.2-1B	C4	packed	64.5	45.3	108	36.6	1.831	1.098	1.352	1.646
Llama-3.2-1B	C4	segments	69.3	47.0	136	37.4	1.911	1.120	1.400	1.713
Llama-3.2-1B	SlimPajama	packed	59.7	52.0	34.3	34.8	1.760	1.237	1.261	1.073
Llama-3.2-1B	SlimPajama	segments	60.6	50.4	38.1	35.9	1.784	1.194	1.231	1.083
Llama-3.2-1B	Alpaca	packed	135	76.4	99.2	37.8	2.576	1.603	1.800	0.493
Llama-3.2-1B	Alpaca	segments	135	74.4	102	37.5	2.580	1.575	1.780	0.515
Llama-3.2-1B	Self (doc.)	packed	45.4	48.4	32.5	35.6	1.492	1.177	1.220	0.927
Llama-3.2-1B	Self (doc.)	segments	48.3	49.0	33.8	36.2	1.555	1.194	1.240	0.990
Qwen3-1.7B	WikiText-2	packed	16.3	74.6	169	37.8	0.629	1.597	1.808	2.308
Qwen3-1.7B	WikiText-2	segments	18.2	70.5	301	37.5	0.732	1.501	1.802	2.366
Qwen3-1.7B	C4	packed	33.8	32.6	63.4	39.9	1.297	0.718	0.937	1.481
Qwen3-1.7B	C4	segments	39.2	32.4	80.4	41.0	1.455	0.713	0.950	1.461
Qwen3-1.7B	SlimPajama	packed	31.9	36.4	22.9	38.5	1.232	0.830	0.832	0.988
Qwen3-1.7B	SlimPajama	segments	35.1	36.1	25.5	39.2	1.327	0.807	0.837	1.018
Qwen3-1.7B	Alpaca	packed	95.9	59.4	87.6	42.1	2.321	1.295	1.519	0.424
Qwen3-1.7B	Alpaca	segments	101	60.6	93.2	41.4	2.375	1.305	1.513	0.458
Qwen3-1.7B	Self (doc.)	packed	36.2	43.7	25.4	37.9	1.367	1.020	1.063	0.852
Qwen3-1.7B	Self (doc.)	segments	38.2	43.7	26.1	38.4	1.423	1.034	1.072	0.882

With the measured-KL allocation of LSP, the data also decide the ranks (Table 11). Computed on each source's windows, this allocation keeps 13 to 20 of the 64 units of Llama-3.2-1B dense (mostly tied query, key, and value groups) and compensates mainly in the gate and up projections. The allocations computed on the four corpora and on self-generated documents are close to one another, differing by 89 to 314 removed directions per unit on average, about one step of the measurement grid of 256, and are far from the uniform rule (503 to 597); the exact self-samples, which compress the attention groups of most layers, are the outlier. Measured allocation improves 21 of the 24 cells of Table 11, by up to 17% in perplexity and 1.8 points in accuracy, and leaves the ranking of the sources unchanged. WikiText-2 calibration remains best on WikiText-2 and worst on C4 and SlimPajama, C4 best on C4, self-generated documents best on SlimPajama, and Alpaca best in accuracy.

Table 11. Uniform against measured-KL rank allocation, Llama-3.2-1B compressed by 50%; the measured-KL allocation is computed on the windows of each calibration source. Columns as in Table 3.

Calibration	Uniform				Measured KL			
	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc
WikiText-2	22.8	102	262	36.3	22.5	91.7	224	37.6
C4	62.7	45.3	109	39.3	63.2	43.6	100	40.8
SlimPajama	59.6	51.5	33.7	38.1	63.5	51.2	33.4	38.0
Alpaca	137	76.7	99.3	39.8	114	69.9	84.4	41.6
Self (seq.)	53.2	59.3	39.0	38.4	49.2	55.2	35.6	39.2
Self (doc.)	45.2	48.2	32.2	39.2	44.4	46.4	30.4	39.9

4.5. Sampling Temperature and Amount of Data

Two choices remain open with self-generated calibration data: the sampling temperature, and how much data to generate. The amount of data matters for any source, so we vary it for C4 as well.

Temperature one is best (Table 9, bottom). Sampling the exact sequences of Llama-3.2-1B at $T = 0.7$ or $T = 1.3$ instead of $T = 1$ more than doubles the WikiText-2 and C4 perplexities of the model compressed by 50% (WikiText-2: 123 and 120 against 53.2; C4: 124 and 131 against 59.3) and raises its divergence on held-out samples at $T = 1$—which estimates the sequence-level divergence—from 0.65 to 1.38 and 1.00 nats per token. Only $T = 1$ makes the training objective that divergence (Proposition 1); colder samples concentrate on the most likely continuations and hotter ones on noisier text, and both plausibly cover the contexts of real text less well.

At the fixed budget of 512 steps, more distinct windows help both C4 and self-generated documents (Figure 5). From 64 to 4096 windows (at 4096, each window is seen once), the WikiText-2 perplexity falls from 114 to 48.9 with C4 and from 64.3 to 42.9 with self-generated documents, while the SlimPajama perplexity falls from 208 to 74.7 and from 47.2 to 29.8. The 512 windows of the main experiments do not saturate learned compression; for its training-free truncation, SVD-LLM reports changes of at most 3% with the size of the calibration set [6]. Our untrained truncations at -50%, with perplexities in the thousands, vary across sizes without a trend. The gains level off sooner for self-generated documents, on which SlimPajama perplexity changes by 2% between 1024 and 4096 windows against 19% for C4, but the ranking of the two sources barely depends on size. Self-generated documents are better on WikiText-2 and SlimPajama at every size, and C4 is better on C4 from 128 windows on, after a tie at 64 (61.8 against 61.6). With only 64 windows, self-generated documents even give a lower SlimPajama perplexity than C4 with 4096. Accuracy rises from 36.3 and 36.6% with 64 windows to 39.3 and 39.2% with 512, and does not improve beyond.

At a fixed number of steps, more distinct windows also means fewer passes over each, from 64 passes with 64 windows to one with 4096, so this design cannot attribute the gain to diversity rather than to less repetition. The learning curves show what the repetition accomplishes (Figure 6). With 64 windows, the divergence on the training windows falls to 0.21 nats per token for C4 and 0.23 for self-generated documents by the last epoch, while that on validation windows of the same source stays at 1.39 and 1.28; the projectors fit the windows that they see far more closely than other text of the same source. The gap narrows as the windows multiply and closes at 4096, where the final training and validation divergences coincide (1.01 and 1.01 for C4; 0.88 and 0.89 for self-generated documents). The validation divergence levels off without rising, and every run selects its last epoch, so the repetition does not overfit in the usual sense; rather, it limits how well the fit transfers. On the five held-out sets common to all runs, the geometric-mean divergence

falls monotonically with the number of distinct windows, from 1.86 to 1.26 for C4 and from 1.36 to 1.03 for self-generated documents; from 1024 to 4096 windows, it still falls by 7% for C4, against 2% for self-generated documents.

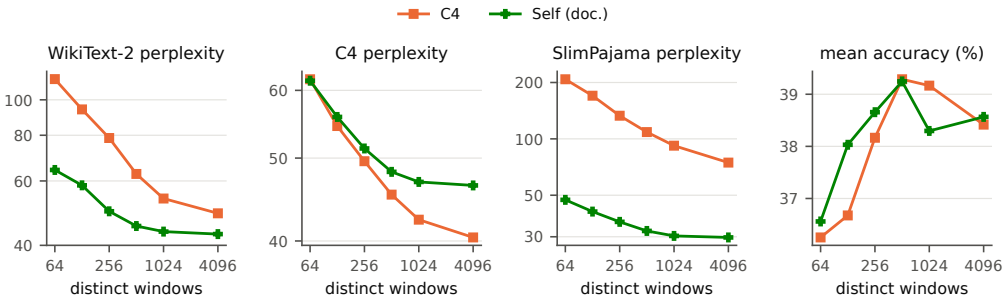


Figure 5. Number of distinct calibration windows at a fixed budget of 512 training steps (4096 window presentations), Llama-3.2-1B compressed by 50%, C4 against self-generated documents. With 4096 windows, each window is seen once; the self-generated ones come from a larger pool of documents. Logarithmic axes except accuracy (mean over seven tasks), for which standard deviation over three seeds at 512 windows is 0.6 points for C4 and 1.1 points for self-generated documents (Section 4.9).

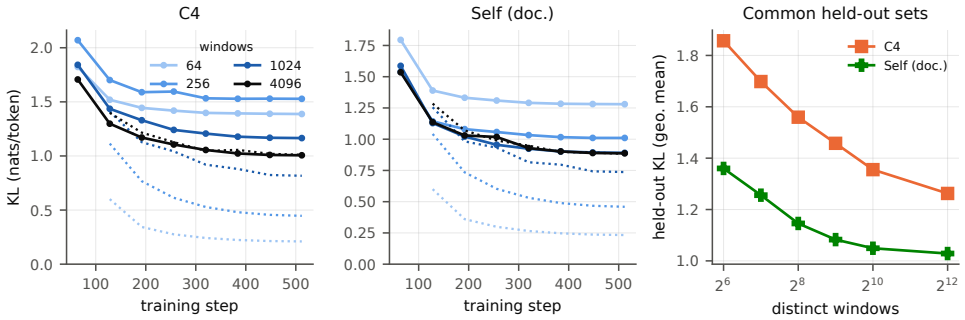


Figure 6. Learning curves at the fixed budget of 512 steps, Llama-3.2-1B compressed by 50%. Left and center: divergence on the training windows (dotted, from the second epoch, after the projectors are ramped in) and on the source's own validation windows (solid) for 64 to 4096 distinct windows; validation windows differ between runs. Right: geometric-mean divergence over the five held-out sets common to all runs, against the number of distinct windows.

4.6. Exact Self-Samples and Failure on Qwen3

The failure of exact self-samples on Qwen3-1.7B recurs in every seed, with a severity that varies from run to run (Tables A3 and A4). Each seed samples its own pool; calibrated on the three pools, the model compressed by 30% reaches C4 perplexities of 53.5, 30.5, and 39.5 and the model compressed by 50% 227, 164, and 155, against three-seed means of 23.9 to 43.4 and 32.6 to 74.5 for the other sources. Its accuracy without BoolQ is the lowest of all sources in five of the six runs. On OPT-1.3B and Llama-3.2-1B, the C4 perplexity after calibration on exact self-samples varies by at most 6% across seeds. Every run fits the held-out samples of the dense model about as well as self-generated documents do, with divergences of 0.42 to 0.45 nats per token at −30% and 0.65 to 0.67 at −50%, against 0.41 and 0.65 for the documents. The objective of Proposition 1 is met in every run, but on Qwen3-1.7B it leaves the behavior of the compressed model on real text loosely constrained, and that behavior changes from run to run.

The length of the samples is not the cause (Table 12). Self-generated documents work with every length limit we tried, from 128 tokens to 2048 (the length of the exact samples); their C4 perplexities range from 28.7 to 31.0 and their accuracies from 42.2 to 43.8%. Restarting the exact samples every 512 tokens so that each window holds four exact

samples from a fresh start recovers only part of the gap (C4 perplexity 44.0, against 53.5 without restarts and 28.7 for the documents). On Llama-3.2-1B, restarts and documents both help somewhat, and documents of at most 512 or 1024 tokens are best.

The composition of the samples does not explain the failure either (Table 13). The exact samples of Qwen3-1.7B repeat themselves more than those of the other models (7 to 9% of the 4-grams of each block repeat an earlier one, against at most 5% for Llama-3.2-1B), about a fifth of the letters are Chinese or Japanese, and a tenth of the lines are code; however, the self-generated documents of the same model, which do work, have the same composition. Across the three pools of exact samples, the composition does not track the outcome; the pool that provided the best compressed model has the fewest Chinese and Japanese letters (15%, against 19–21%) but also the most blocks that are mostly repetition (2–4% of its blocks, against 0.4–2.3%).

Table 12. Self-generated calibration of Qwen3-1.7B and Llama-3.2-1B compressed by 30%, seed 0: exact samples of 2048 tokens, exact samples restarted every 512 tokens, and self-generated documents with four length limits. Columns as in Table 3, accuracy without BoolQ; KL Self and KL C4: divergence on held-out exact samples and on C4.

Model	Self-Generated Text	Wiki	C4	SP	Acc	KL Self	KL C4
Qwen3-1.7B	Exact, 2048 tokens	39.4	53.5	34.5	39.2	0.419	1.200
Qwen3-1.7B	Exact, restarted every 512	37.9	44.0	33.4	42.3	0.371	0.963
Qwen3-1.7B	Documents, at most 128	21.6	28.7	16.3	43.8	0.424	0.618
Qwen3-1.7B	Documents, at most 512	21.1	28.7	16.3	43.5	0.406	0.616
Qwen3-1.7B	Documents, at most 1024	22.2	31.0	17.6	42.2	0.464	0.683
Qwen3-1.7B	Documents, at most 2048	21.3	29.8	16.9	42.2	0.461	0.648
Llama-3.2-1B	Exact, 2048 tokens	31.2	37.8	24.1	37.5	0.423	0.951
Llama-3.2-1B	Exact, restarted every 512	29.5	34.3	22.5	39.1	0.404	0.861
Llama-3.2-1B	Documents, at most 128	30.3	33.5	22.2	39.8	0.570	0.834
Llama-3.2-1B	Documents, at most 512	27.6	32.3	20.8	39.1	0.556	0.792
Llama-3.2-1B	Documents, at most 1024	27.5	32.6	20.6	38.9	0.539	0.799
Llama-3.2-1B	Documents, at most 2048	28.7	33.1	21.0	38.6	0.533	0.819

Table 13. What the dense models write: repetition (share of 4-gram occurrences that repeat an earlier one, %), compressed size over raw size (gzip), share of letters in CJK and Cyrillic scripts (%), share of Latin script words that are among 50 common English words (%), share of lines that look like code (%), and document start tokens per 512-token block of the exact samples (averaged over 256 samples) and over the self-generated documents.

Model	Text	Rep. 4-Grams	Gzip Ratio	CJK	Cyrillic	English Words	Code Lines	Doc. Starts
OPT-1.3B	exact, tokens 0–512	1.7	0.54	0.0	0.0	34.2	0.1	3.5
OPT-1.3B	exact, tokens 512–1024	2.2	0.53	0.0	0.0	33.9	0.0	1.7
OPT-1.3B	exact, tokens 1024–1536	2.4	0.54	0.0	0.0	34.0	0.1	1.7
OPT-1.3B	exact, tokens 1536–2048	2.8	0.54	0.0	0.0	33.4	0.2	1.8
OPT-1.3B	documents	1.4	0.72	0.0	0.0	34.1	0.1	0.0
Llama-3.2-1B	exact, tokens 0–512	3.6	0.55	1.2	0.5	28.4	5.7	6.0
Llama-3.2-1B	exact, tokens 512–1024	2.6	0.60	1.9	0.9	25.2	4.6	3.4
Llama-3.2-1B	exact, tokens 1024–1536	1.8	0.63	2.4	1.3	23.3	4.2	2.5
Llama-3.2-1B	exact, tokens 1536–2048	1.7	0.65	2.7	1.7	21.9	4.5	2.0
Llama-3.2-1B	documents	3.7	0.66	1.5	0.6	26.5	5.5	0.0
Qwen3-1.7B	exact, tokens 0–512	8.0	0.51	19.2	0.3	20.1	11.0	1.8
Qwen3-1.7B	exact, tokens 512–1024	7.8	0.53	20.5	0.6	19.3	11.2	0.7
Qwen3-1.7B	exact, tokens 1024–1536	7.5	0.54	19.2	0.6	18.6	10.8	0.7
Qwen3-1.7B	exact, tokens 1536–2048	6.7	0.55	18.9	0.4	18.3	11.3	0.5
Qwen3-1.7B	documents	8.0	0.56	15.6	0.4	19.8	13.6	0.0

What distinguishes the exact samples from the documents is what follows a document boundary inside a sample. A document ends at the model's first end-of-text token, whereas an exact sample carries on and the model writes the next document, conditioned on the previous one. On Qwen3-1.7B, 58% of the tokens of the exact samples follow such a boundary, 23% when the samples are restarted every 512 tokens, and none in the documents, in the order of their C4 perplexities (53.5, 44.0 and 28.7). The same continuation does not harm Llama-3.2-1B, where 76% of the tokens of the exact samples follow a boundary, so the controls locate the failure without explaining why it affects one model only. Independent documents avoid it on every model.

4.7. Projectors Versus Free Factors

Learning the factors **A** and **B** freely does not improve on the projectors (Table 14). Free factors train 2.3 to 2.8 times as many parameters, yet in the calibration domain they are no better (WikiText-2 perplexity after WikiText-2 calibration: 23.0 against 22.8 on Llama-3.2-1B, 21.2 against 20.8 on OPT-1.3B), and away from it they are worse. After WikiText-2 calibration, the C4 perplexity rises from 102 to 113 on Llama-3.2-1B and from 50.3 to 55.4 on OPT-1.3B, and the SlimPajama perplexity from 262 to 301 and from 55.3 to 66.2; after C4 calibration, the WikiText-2 and SlimPajama perplexities rise by 8 to 20%. With exact self-samples, the two parameterizations agree within 4% on every corpus, and zero-shot accuracy differs by less than one point in every row. The ranking of the three sources is the same for both parameterizations on every corpus and in accuracy. Thus, keeping the pretrained weights and learning only which subspaces to remove limits the specialization to the calibration data at no cost in fit to them.

Table 14. Learned projectors against free low-rank factors trained from the same initialization, with the same objective, data and schedule, at −50%; free factors use the better of two learning rates (10^{-5}, 10^{-4}) on validation KL. Trainable: number of trained parameters. Columns as in Table 3.

Model	Calibration	Learned	Trainable	Wiki	C4	SP	Acc
Llama-3.2-1B	WikiText-2	projectors	174 M	22.8	102	262	36.3
		free factors	486 M	23.0	113	301	36.5
	C4	projectors	174 M	62.7	45.3	109	39.3
		free factors	486 M	69.0	47.2	130	39.5
	Self (seq.)	projectors	174 M	53.2	59.3	39.0	38.4
		free factors	486 M	52.5	59.5	38.7	38.5
OPT-1.3B	WikiText-2	projectors	260 M	20.8	50.3	55.3	38.7
		free factors	604 M	21.2	55.4	66.2	38.1
	C4	projectors	260 M	39.3	27.7	27.5	40.1
		free factors	604 M	42.5	29.1	30.6	40.8
	Self (seq.)	projectors	260 M	47.2	30.1	25.3	39.5
		free factors	604 M	49.1	31.3	25.8	39.2

The dependence on the source is not specific to LSP (Table 15). After the truncation of SVD-LLM and its recovery with LoRA, each of the three corpora again gives the lowest perplexity on its own text, on both models, and Alpaca the highest accuracy without BoolQ (42.1% on Llama-3.2-1B and 43.5% on OPT-1.3B). In the calibration domain, the two compressors are close on Llama-3.2-1B (WikiText-2 perplexity after WikiText-2 calibration: 17.0 against 16.8 for LSP), and LoRA recovery is better on OPT-1.3B, by 5–24% on the matching corpus. Away from it, the narrow sources pay more with LoRA recovery: on Llama-3.2-1B, WikiText-2 calibration gives 2.8 times the C4 perplexity of C4 calibration (95.0 against 33.4), against 1.9 times with LSP, and on OPT-1.3B 1.8 times against 1.4. The two objectives explain part of the difference. Cross-entropy on the calibration windows

fits the text itself, not the dense model's predictions on it: after Alpaca calibration, the divergence from the dense model on held-out Alpaca examples is 0.87 nats per token on Llama-3.2-1B and 1.97 on OPT-1.3B with LoRA recovery, against 0.30 and 0.10 with LSP. On exact self-samples, the expected cross-entropy is the dense model's entropy plus the sequence-level divergence of Proposition 1, so the two objectives share their population target, and the two compressors reach similar divergences on held-out self-samples (0.46 and 0.42 nats per token on Llama-3.2-1B, 0.29 and 0.32 on OPT-1.3B).

Table 15. Another learned compressor: the oblique truncation of SVD-LLM recovered with LoRA (rank 8, cross-entropy on the calibration windows), against LSP, by calibration source, −30%, seed 0. Accuracy without BoolQ.

Calibration	Llama-3.2-1B				OPT-1.3B			
	Wiki	**C4**	**SP**	**Acc**	**Wiki**	**C4**	**SP**	**Acc**
SVD-LLM + LoRA								
WikiText-2	17.0	95.0	216	37.7	13.2	36.8	28.5	40.9
C4	40.8	33.4	69.6	40.9	21.4	20.5	17.0	42.5
SlimPajama	36.1	37.4	23.0	38.7	19.7	22.3	11.6	42.3
Alpaca	89.5	58.8	70.9	42.1	34.0	28.6	19.7	43.5
Self (seq.)	31.8	38.8	24.7	38.2	22.4	21.4	14.0	41.9
Self (doc.)	31.5	35.7	23.2	38.3	23.6	21.6	14.6	41.6
LSP								
WikiText-2	16.8	58.3	134	38.2	17.3	31.3	26.2	38.9
C4	38.0	30.8	62.5	41.0	23.7	21.7	17.3	42.3
SlimPajama	33.3	34.1	21.3	39.0	24.0	23.5	12.3	40.8
Alpaca	66.2	45.3	50.8	42.0	44.2	31.1	25.8	42.1
Self (seq.)	31.2	37.8	24.1	37.5	27.4	22.6	16.1	41.0
Self (doc.)	27.6	32.3	20.8	39.1	26.9	22.6	16.4	41.1

4.8. Optimization Settings

The comparison does not depend on the optimization settings (Table 16). Tuned separately on its own validation windows, every source reaches its lowest validation KL at the shared learning rate of 10^{-4}; for the exact self-samples, 3×10^{-5} comes within 0.1% (0.6696 against 0.6691 nats per token) and changes the perplexities by at most 1.5%. Equally budgeted tuning selects the setting of the main experiments for every source. Training three times as long changes the perplexities more than the comparison: after 24 epochs (1536 steps), every perplexity is 1 to 21% lower and every accuracy 0.4 to 1.1 points higher, but the six sources keep their order on WikiText-2, C4, and SlimPajama. Alpaca remains the most accurate, and WikiText-2 calibration still gives 2.1 times the C4 perplexity of C4 calibration (2.2 times after eight epochs). The initialization keeps its imprint as well. After 24 epochs, the removed subspaces overlap their initialization by 0.992 on average (0.997 after eight epochs), and the subspaces of different sources stay as far apart as before training (0.88 against 0.89, slightly lower after training for each of the fifteen pairs). In the crossover, the subspaces trained on SlimPajama windows from the WikiText-2 statistics overlap those of WikiText-2 calibration by 0.99 and those of SlimPajama calibration by 0.89, and conversely. The perplexities follow the statistics: SlimPajama windows trained from the WikiText-2 statistics reach a WikiText-2 perplexity of 34.7, against 51.3 from SlimPajama's own statistics (35.8 and 59.7 after eight epochs), and WikiText-2 windows trained from the SlimPajama statistics a SlimPajama perplexity of 68.4, against 206 (71.3 and 261). The selected checkpoint does not matter either; in 92 of the 108 runs of the three-seed grid, the epoch with the lowest validation KL is the last, and in the other sixteen the last epoch changes the WikiText-2 and C4 perplexities by at most 0.8%.

Table 16. Optimization controls on Llama-3.2-1B compressed by 50%, seed 0. Validation KL: the lowest validation KL (nats per token) of each source on its own validation windows, at three learning rates; bold: the lowest of the three rates for each source; 10^{-4} is the rate of the main experiments. With 8 and 24 epochs: the trained model at 10^{-4} after the 512 steps of the main experiments and after three times as many, with accuracy without BoolQ. Last two rows: training windows of one source from the initialization statistics of the other (at 8 epochs, the runs of Table 9).

Calibration	Validation KL			8 Epochs				24 Epochs			
	3×10^{-5}	10^{-4}	3×10^{-4}	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc
WikiText-2	0.967	**0.888**	0.909	22.8	102	261	35.1	21.2	86.3	206	35.5
C4	1.165	**1.082**	1.112	64.5	45.3	108	36.6	54.3	41.6	97.3	37.6
SlimPajama	1.302	**1.215**	1.230	59.7	52.0	34.3	34.8	51.3	47.6	31.4	35.9
Alpaca	0.524	**0.456**	0.477	135	76.4	99.2	37.8	134	71.4	93.9	38.4
Self (seq.)	0.670	**0.669**	0.761	54.2	59.8	39.3	34.5	47.1	54.0	35.3	35.4
Self (doc.)	1.074	**0.969**	0.981	45.4	48.4	32.5	35.6	40.2	43.7	29.3	36.4
WikiText-2, SlimPajama statistics	–	–	–	25.5	76.2	71.3	35.5	23.1	71.5	68.4	36.0
SlimPajama, WikiText-2 statistics	–	–	–	35.8	55.8	41.5	35.4	34.7	50.3	37.0	36.0

4.9. Variability and Statistical Reliability

The full grid of three models, two ratios, and six sources was repeated with three seeds (Tables A3 and A4). The first seed repeats the configuration of the main runs on different hardware: over its 36 runs, the WikiText-2, C4, and SlimPajama perplexities differ from those of the main runs by a median of 0.9% (90th percentile 6.3%) and the mean accuracy by a median of 0.16 points. The larger differences come from the configurations that also vary most across seeds: the exact self-samples of Qwen3-1.7B (Section 4.6) and Alpaca on Qwen3-1.7B at −30% and on OPT-1.3B at −50%, for which the SlimPajama perplexities differ by 24% and 19%. Across seeds, the perplexities vary by a median of 2.1% of their mean, and the mean accuracy has a median standard deviation of 0.3 points and at most 1.1 points, again except for the exact self-samples of Qwen3-1.7B.

The comparisons of perplexity and divergence hold seed-by-seed. In all 18 combinations of model, ratio, and seed, WikiText-2 calibration gives the lowest WikiText-2 perplexity and C4 calibration the lowest C4 perplexity; averaged over the seeds, the margin over the closest other source is at least 4.5%, with 95% paired-bootstrap intervals over the evaluation windows that exclude equality. On the held-out set of each real corpus, the seed-averaged divergence of the matching source is lower than that of the closest other source by a factor of 1.08 to 2.94, with intervals that exclude one, in 22 of the 24 combinations of model, ratio, and corpus; the two exceptions are the ties on SlimPajama for Llama-3.2-1B (Section 4.3). Resampling the held-out windows also gives the probability that a source ranks first in uniform fidelity over the five held-out sets: 1.00 for self-generated documents in five settings and 0.54 against 0.46 for C4 on OPT-1.3B at −50%.

Accuracy differences are of the order of the seed variation, and the paired comparisons settle most but not all (Table 17). Alpaca's lead without BoolQ is significant in five of the six settings and its lead with BoolQ in four; on OPT-1.3B at −30%, C4 is more accurate with BoolQ (0.7 points, interval 0.1 to 1.2). WikiText-2 calibration is less accurate than every other real source, with intervals that exclude zero, in all six settings with BoolQ (although only barely on Llama-3.2-1B, at −30%) and in five without it. On Llama-3.1-8B, Alpaca's lead over C4 is not significant with or without BoolQ at −30%, but is significant in both at −50%, and WikiText-2 is significantly less accurate than every other real source in both, at both ratios.

Table 17. Paired comparisons of accuracy, three seeds (one for Llama-3.1-8B at −50%): Alpaca minus the next most accurate source and WikiText-2 minus the next least accurate real source (named), as differences of the mean accuracy over six tasks (without BoolQ) and over seven (in points), with 95% paired-bootstrap intervals over benchmark items averaged over the seeds.

Model	Ratio	Alpaca Minus Runner-Up		WikiText-2 Minus Next Real Source	
		6 Tasks	7 Tasks	6 Tasks	7 Tasks
OPT-1.3B	−30%	−0.0 [−0.6, +0.6] (C4)	−0.7 [−1.2, −0.1]	−1.9 [−2.5, −1.3] (SlimPajama)	−1.3 [−1.9, −0.7]
OPT-1.3B	−50%	+0.9 [+0.3, +1.5] (C4)	+0.7 [+0.1, +1.2]	−1.6 [−2.3, −1.0] (SlimPajama)	−3.7 [−4.4, −3.1]
Llama-3.2-1B	−30%	+1.3 [+0.6, +1.9] (C4)	+0.0 [−0.6, +0.6]	−0.9 [−1.6, −0.2] (SlimPajama)	−0.6 [−1.2, −0.0]
Llama-3.2-1B	−50%	+1.2 [+0.5, +1.9] (C4)	+1.3 [+0.7, +1.9]	−0.3 [−1.0, +0.3] (SlimPajama)	−1.6 [−2.2, −1.0]
Qwen3-1.7B	−30%	+1.9 [+1.2, +2.6] (C4)	+1.2 [+0.6, +1.9]	−2.0 [−2.7, −1.4] (SlimPajama)	−2.6 [−3.2, −2.0]
Qwen3-1.7B	−50%	+1.9 [+1.1, +2.5] (C4)	+1.8 [+1.2, +2.4]	−1.2 [−1.9, −0.6] (SlimPajama)	−3.1 [−3.7, −2.5]
Llama-3.1-8B	−30%	+0.6 [−0.1, +1.2] (C4)	+0.3 [−0.3, +0.9]	−1.6 [−2.2, −0.9] (SlimPajama)	−2.2 [−2.7, −1.6]
Llama-3.1-8B	−50%	+2.1 [+1.2, +3.0] (C4)	+1.5 [+0.7, +2.3]	−1.6 [−2.5, −0.7] (SlimPajama)	−1.4 [−2.3, −0.6]

4.10. Scale

A larger model behaves the same way (Table 18). We compressed Llama-3.1-8B [47] with all six sources, by 30% with three seeds and by 50% with one, following the protocol of the main experiments with a learning rate of 3×10^{-5}, which reached the lowest validation KL of C4 calibration at −30% among four rates from 3×10^{-6} to 10^{-4} (0.575, against 0.587 to 0.614), and again at −50% (0.931, against 0.940 at 10^{-4}). Before training, the perplexity of the whitened initialization varies across sources by a factor of 12 to 15 on WikiText-2 and 13 to 16 on C4, depending on the seed, and the exact self-samples give by far the lowest (38.7 and 46.4 in the first seed, against at least 130 and 154 for the other sources); training narrows them to 3.0 to 3.1 on WikiText-2 and 2.2 to 2.4 on C4, and the exact self-samples lose their lead. After training, in every seed, WikiText-2 calibration gives the lowest WikiText-2 perplexity (10.0 on average, against 13.9 to 30.5) and C4 calibration the lowest C4 perplexity (18.2, against 18.6 for self-generated documents, the closest, a margin of 2.2% with a 95% paired-bootstrap interval of 0.4 to 3.8%), and on every held-out set the divergence from the dense model is lowest when calibrating on that set's source, with the exception already seen on Llama-3.2-1B: on SlimPajama, self-generated documents and SlimPajama tie (ratio of divergences 1.01, interval 0.99 to 1.03). Narrow sources again pay elsewhere: WikiText-2 calibration more than doubles the C4 perplexity (41.5 against 18.2) and leaves the SlimPajama perplexity at 18 times that of the dense model, with code at 220 times (Figure 7). Self-generated documents are again the most uniformly faithful source: they give the lowest SlimPajama perplexity (11.3), the lowest geometric-mean and maximum relative divergence over the five held-out sets (1.37 and 1.90) and over the six independent domains (1.02 and 1.10), and keep every SlimPajama domain within a factor of 3.5 of the dense model, which SlimPajama calibration does not (4.8 on Wikipedia). The exact self-samples, which failed on Qwen3-1.7B, work on Llama-3.1-8B as on Llama-3.2-1B: they are the most faithful source on held-out exact samples and the second over the five held-out sets, but give the least accurate models. Zero-shot accuracy without BoolQ varies by 5.7 points, from 46.5% with exact self-samples to 52.1% with Alpaca. Its lead over C4, the next most accurate source, is within the noise (0.6 points, interval −0.1 to 1.2; Table 17). Across seeds, the perplexities vary by a median of 1.6% of their mean and the accuracy by a standard deviation of at most 0.4 points.

Table 18. Llama-3.1-8B compressed by 30% (mean over three seeds) and by 50% (one seed), by calibration source: WikiText-2 and C4 perplexity, SlimPajama perplexity (SP, geometric mean over its seven domains), mean zero-shot accuracy without BoolQ (Acc, %), and the divergence from the dense model relative to the best source, summarized over the five held-out sets as in Table 5 (Rel. KL). Bold: best per column. The dense model has perplexities of 6.31 (WikiText-2), 9.64 (C4) and 5.74 (SP), and an accuracy of 63.7% without BoolQ.

Calibration	−30%, Three Seeds					−50%, One Seed				
	Wiki	**C4**	**SP**	**Acc**	**Rel. KL**	**Wiki**	**C4**	**SP**	**Acc**	**Rel. KL**
WikiText-2	**10.0**	41.5	101	49.1	3.32	**13.7**	73.4	194	41.7	2.73
C4	19.3	**18.2**	31.4	51.6	2.26	40.8	**26.5**	65.3	44.2	2.01
SlimPajama	17.6	19.6	12.4	50.7	1.76	32.4	29.4	19.7	43.3	1.70
Alpaca	30.5	25.8	26.5	**52.1**	2.04	68.8	42.0	51.9	**46.3**	1.87
Self (seq.)	16.8	21.8	13.6	46.5	1.58	28.3	34.0	22.4	39.1	1.51
Self (doc.)	13.9	18.6	**11.3**	50.9	**1.37**	22.9	27.0	**17.7**	43.7	**1.33**

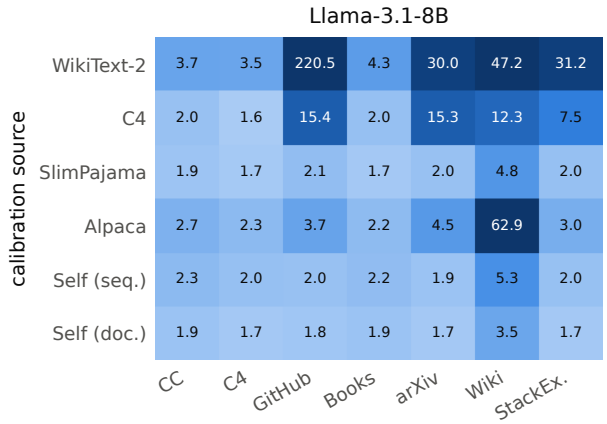


Figure 7. Perplexity on the seven SlimPajama domains relative to the dense model, Llama-3.1-8B compressed by 30% after training, geometric mean over three seeds; each cell gives the ratio to the dense model, and its color encodes the same ratio on the logarithmic scale of Figure 4, from light (close to the dense model) to dark blue (far from it).

At −50%, with one seed, the same pattern holds with wider gaps. Before training, the sources differ by a factor of 127 on WikiText-2, 302 on C4 and 594 on SlimPajama, the exact self-samples again giving by far the lowest WikiText-2 and C4 perplexity (159 and 152, against at least 775 and 512); training narrows these factors to 5.0, 2.8 and 11. WikiText-2 calibration gives the lowest WikiText-2 perplexity (13.7, against 22.9 to 68.8), and C4 calibration the lowest C4 perplexity, though within the noise of self-generated documents (26.5 against 27.0, a margin of 1.9%, interval −0.2 to 3.7%); the divergence on each held-out set is again lowest with that set's source, except on SlimPajama, where self-generated documents now lead SlimPajama itself (ratio of divergences 0.96, interval 0.95 to 0.98). WikiText-2 calibration nearly triples the C4 perplexity (73.4) and leaves the SlimPajama perplexity at 34 times that of the dense model, with code at 413 times. Self-generated documents remain the most uniformly faithful source by every summary: they give the lowest SlimPajama perplexity (17.7, against 19.7 after SlimPajama calibration), the lowest geometric-mean and maximum relative divergence over the five held-out sets (1.33 and 1.82), over the four real ones, and over the six independent domains (1.00 and 1.01: the best source on every domain, or within 1% of it), and the smallest worst-domain degradation (8.2 times the dense perplexity, on Wikipedia, against 11.5 after SlimPajama calibration). The exact self-samples are again the most faithful source on held-out exact samples, the second over the five held-out sets, and the least accurate. Zero-shot accuracy

without BoolQ ranges from 39.1% with exact self-samples to 46.3% with Alpaca, against 63.7% for the dense model; Alpaca leads C4, the next most accurate source, by 2.1 points (interval 1.2 to 3.0), and five of the six compressed models fall below the majority-class rate of BoolQ.

4.11. Behavior Beyond Perplexity, and Cost

Beyond perplexity and multiple choice, compression costs more, and the calibration source matters as much (Table 19, models compressed by 30%, seed 0). Continuations sampled from the compressed models are less plausible to the dense model than its own: their negative log-likelihood under the dense model rises from about 3.1 nats per token to 3.8 to 5.0 on OPT-1.3B, 4.8 to 5.4 on Llama-3.2-1B and 3.6 to 4.9 on Qwen3-1.7B, and from 2.8 to 4.0 to 4.9 on Llama-3.1-8B. Greedy continuations repeat more: with every source but Alpaca, 63 to 82% of their 4-grams are repeated, against 48 to 56% for the dense models, and 55 to 68% on Llama-3.1-8B, against 34%; Alpaca calibration gives the least repetitive continuations on the Llama and Qwen models, near the dense level on Llama-3.2-1B and Qwen3-1.7B (58 and 46%). The likelihood alone can mislead: the Qwen3-1.7B model calibrated on exact self-samples at −50% writes the most likely continuations of all (1.57 nats per token, below the dense model's own 3.13) because they are loops: 94% of their 4-grams repeat.

On long contexts, no compressed model breaks down beyond the 2048 tokens of its calibration windows: on positions 4096 to 8192 of PG-19 books, the perplexity of every Llama-3.2-1B and Qwen3-1.7B model is within 13% of its perplexity on the first 2048 positions, and that of every Llama-3.1-8B model within 7%, except after WikiText-2 calibration (25% higher); it follows the domain ranking, lowest after SlimPajama calibration (OPT-1.3B, with 2048 positions, cannot be evaluated). Passkey retrieval separates the sources more sharply. The dense models retrieve a key hidden in about 8000 tokens of filler every time; compressed by 30%, the models calibrated on C4, SlimPajama or self-generated text still do so 88 to 100% of the time, while WikiText-2 calibration falls to 8% on Llama-3.2-1B, and Alpaca calibration to 0% on Llama-3.2-1B and Qwen3-1.7B, on Llama-3.2-1B already at 2000 tokens, and to 53% on Llama-3.1-8B. The Alpaca-calibrated Llama-3.2-1B does not fail through the format of its answer: given more tokens, it continues the prompt with generic text ("the key to the door") instead of copying the key, which the C4-calibrated model does at once. Calibration on short instruction–response pairs, for which packed examples never depend on text far back in the window, plausibly removes the directions that such copying uses. Generative tasks that need several reasoning steps or recalled facts do not survive compression, whatever the source. Qwen3-1.7B solves 70.1% of the GSM8K problems and answers 8.7% of the NQ-Open questions; compressed by 30%, it solves 1.8 to 4.7% and answers 0.4 to 1.8%, the most after calibration on self-generated documents. Llama-3.1-8B falls from 49.8 to 1.8 to 6.6% on GSM8K and from 19.5 to 3.0 to 4.7% on NQ-Open, again with the most GSM8K problems solved after calibration on self-generated documents.

Instruction-following degrades far more than perplexity or multiple-choice accuracy (Table 20). Compressed by 30%, Llama-3.2-1B-Instruct follows 9.8 to 14.0% of the IFEval prompts strictly, against 49.7% for the dense model, and 18.6 to 24.1% of the individual instructions, against 63.1%, while its accuracy without BoolQ falls from 52.0% to between 39.5 and 42.3%. The model's own responses to Alpaca instructions are the best source on both measures and the most faithful on held-out Alpaca examples (0.55 nats per token). Alpaca in the model's chat format, where pairs have the format of the evaluation but were written by another model, is the worst source on both measures and has by far the highest WikiText-2 perplexity (102). The difference between the two is beyond sampling

noise (4.3 points at the prompt level, 95% paired-bootstrap interval 1.3 to 7.2, over the 541 prompts), while the model's own responses are not distinguishable at the prompt level from C4, SlimPajama or self-generated documents (differences of 1.5 to 2.8 points, intervals including zero). Every source loses most of the dense model's instruction following at this ratio.

Compression reduces memory, but in our implementation, does not reduce latency (Table 21). The merged models need 24 to 46% less parameter memory than the dense ones, less than the compression ratio because the embeddings and the language-model head are not compressed, and 12 to 36% less peak memory when decoding a batch of 16. They are not faster: each compressed matrix runs as two consecutive matrix products in eager PyTorch, decoding at batch size one is 8 to 15% slower than with the dense model, and the prefill time of a 2048-token prompt ranges from 7% shorter to 1.9 times longer. Turning the smaller number of operations into speed would require kernels designed for the factored form, which we did not attempt; the costs do not depend on the calibration source.

Table 19. Behavior of the models compressed by 30% (seed 0) and of the dense models. Gen. NLL: the dense model's negative log-likelihood (nats per token) of 128-token continuations sampled from the model at temperature one, from 200 held-out C4 prompts; Rep.: share of repeated 4-grams in greedy continuations; PPL 4k–8k: perplexity on positions 4096 to 8192 of 16 PG-19 books; Passkey 8k: retrieval accuracy (%) at about 8000 tokens; GSM8K and NQ: exact match (%) of the last number of the answer and of the answer, at the harness's default few-shot settings, for the two models that solve them; n/a: not evaluated (OPT-1.3B has 2048 positions).

Model	Calibration	Gen. NLL	Rep. (Greedy)	PPL 4k–8k	Passkey 8k	GSM8K	NQ
OPT-1.3B	Dense	3.09	0.56	n/a	n/a	n/a	n/a
OPT-1.3B	WikiText-2	5.00	0.66	n/a	n/a	n/a	n/a
OPT-1.3B	C4	3.96	0.80	n/a	n/a	n/a	n/a
OPT-1.3B	SlimPajama	3.92	0.64	n/a	n/a	n/a	n/a
OPT-1.3B	Alpaca	4.14	0.69	n/a	n/a	n/a	n/a
OPT-1.3B	Self (seq.)	3.84	0.66	n/a	n/a	n/a	n/a
OPT-1.3B	Self (doc.)	3.91	0.63	n/a	n/a	n/a	n/a
Llama-3.2-1B	Dense	3.07	0.51	12.3	100	n/a	n/a
Llama-3.2-1B	WikiText-2	5.37	0.69	96.3	8	n/a	n/a
Llama-3.2-1B	C4	4.82	0.68	64.5	98	n/a	n/a
Llama-3.2-1B	SlimPajama	4.80	0.68	33.7	98	n/a	n/a
Llama-3.2-1B	Alpaca	5.21	0.58	70.4	0	n/a	n/a
Llama-3.2-1B	Self (seq.)	5.11	0.78	39.5	93	n/a	n/a
Llama-3.2-1B	Self (doc.)	5.09	0.75	35.5	98	n/a	n/a
Qwen3-1.7B	Dense	3.13	0.48	12.4	100	70.1	8.7
Qwen3-1.7B	WikiText-2	4.90	0.71	83.7	98	1.8	1.0
Qwen3-1.7B	C4	4.34	0.68	45.7	100	2.0	0.4
Qwen3-1.7B	SlimPajama	4.35	0.63	24.5	88	2.5	1.1
Qwen3-1.7B	Alpaca	4.87	0.46	52.1	0	3.1	1.0
Qwen3-1.7B	Self (seq.)	3.58	0.82	42.1	100	2.0	0.5
Qwen3-1.7B	Self (doc.)	4.94	0.72	28.5	100	4.7	1.8
Llama-3.1-8B	Dense	2.81	0.34	7.9	100	49.8	19.5
Llama-3.1-8B	WikiText-2	4.86	0.60	99.9	100	1.9	3.2
Llama-3.1-8B	C4	3.99	0.55	35.2	100	1.8	3.3
Llama-3.1-8B	SlimPajama	4.21	0.59	17.8	100	2.2	3.0
Llama-3.1-8B	Alpaca	4.63	0.49	31.3	53	3.5	4.7
Llama-3.1-8B	Self (seq.)	4.22	0.68	20.3	100	2.8	3.1
Llama-3.1-8B	Self (doc.)	4.42	0.59	18.4	100	6.6	4.7

Table 20. Instruction-following of Llama-3.2-1B-Instruct compressed by 30%, by calibration source, seed 0: IFEval prompt- and instruction-level strict accuracy (%), mean accuracy without BoolQ, WikiText-2 perplexity, and divergence on held-out Alpaca and exact samples. Bold: highest IFEval accuracy.

Calibration	IFEval, Prompt	IFEval, Instruction	Acc	Wiki	KL Alpaca	KL Self
Dense	49.7	63.1	52.0	13.30	0	0
C4	11.3	22.4	42.0	37.8	0.974	0.855
SlimPajama	11.8	21.9	40.2	35.2	0.720	0.551
Alpaca, chat format	9.8	18.6	42.3	102	0.692	1.063
Self (doc.)	12.6	21.2	39.5	40.0	0.723	0.178
Self (responses)	**14.0**	**24.1**	42.0	64.3	0.549	0.351

Table 21. Memory and latency of the dense and the merged compressed models, bfloat16 on one A100 GPU (median of repeated runs): parameter memory, prefill time of a 2048-token prompt (batch 1 and 8), decoding throughput after a 512-token prompt (batch 1 and 16), and peak memory of the batch-16 decoding (weights, KV cache and activations). The ranks, and hence these costs, do not depend on the calibration source.

Model	Version	Weights (GB)	Prefill, b1 (ms)	Prefill, b8 (ms)	Decode, b1 (tok/s)	Decode, b16 (tok/s)	Peak, b16 (GB)
OPT-1.3B	Dense	2.45	35.0	213	99	1308	4.75
OPT-1.3B	−30%	1.78	62.8	424	85	1003	4.08
OPT-1.3B	−50%	1.33	33.8	211	85	1238	3.63
Llama-3.2-1B	Dense	2.30	45.8	308	91	1418	4.54
Llama-3.2-1B	−30%	1.76	58.0	388	83	1248	4.00
Llama-3.2-1B	−50%	1.40	58.7	405	84	1272	3.64
Qwen3-1.7B	Dense	3.20	70.6	486	41	644	6.43
Qwen3-1.7B	−30%	2.42	65.9	445	38	588	5.66
Qwen3-1.7B	−50%	1.89	101	679	38	578	5.12
Llama-3.1-8B	Dense	14.96	188	1349	46	722	17.98
Llama-3.1-8B	−30%	11.06	360	2653	42	525	14.07
Llama-3.1-8B	−50%	8.46	277	2037	42	626	11.48

5. Discussion

5.1. Calibration Data Are Training Data

In learned low-rank compression, the calibration set is no longer a sample from which to estimate activation statistics; it is the data the compressed model is fitted on, and it behaves like training data. The fit specializes: the in-domain advantage reaches a factor of 6.4 in WikiText-2 perplexity against other sources after training (Table 3), it is visible in the divergence from the dense model on every held-out set, and a single-domain calibration set brings the perplexity on its domain from 3.2 to 3.6 times that of the dense model down to 1.6 to 2.4 times, while multiplying it elsewhere (Table 8). More data help: at a fixed number of steps, every perplexity improves from 512 to 1024 distinct windows, and C4 calibration keeps improving up to 4096 windows, each seen once (Figure 5); more distinct windows also means fewer passes over each, and the learning curves show the repeated windows being fitted far more closely than unseen ones (Figure 6), whereas the training-free truncation is reported to be insensitive to the size of its calibration set [6]. The parameterization also acts as a regularizer: free factors, with more than twice the trainable parameters, fit the calibration domain no better and specialize more (Table 14). Two consequences follow for reporting. First, a perplexity measured on the corpus the model was calibrated on overstates the quality of the compressed model elsewhere, so the

calibration source should accompany every reported perplexity, and comparisons between methods should hold it fixed. Second, no single number summarizes the compressed model: perplexity rewards the matching domain, zero-shot accuracy rewards instruction data, and the two disagree on which source is best. Measuring the divergence from the dense model on several held-out sources, as in Figure 3, exposes the trade-off directly and needs no labels.

5.2. Which Data to Use

For a general-purpose compressed model, the results favor broad data: a pretraining-like mixture (SlimPajama) is among the best sources on every model, and self-generated documents are the most uniformly faithful source on the Llama models, by every summary of the divergence we computed, with the best SlimPajama perplexity on Llama-3.2-1B and Llama-3.1-8B; on OPT-1.3B, C4 and SlimPajama are more uniform once the model's own samples are left out of the evaluation. A half-and-half mixture of SlimPajama and self-generated documents is the most robust choice we found: it is never more than 11% less uniform than the best source, and is the most uniform on the real held-out sets in five of the six settings (Table 7). Narrow corpora specialize: WikiText-2 is the worst general-purpose choice on every model, and Alpaca buys up to two points of zero-shot accuracy with perplexity. For deployment on a known domain, calibrating on that domain is the right choice and the specialization is the intended effect (Table 8). Because the initialization and the training windows act separately, the two can also be combined: whitening statistics from broad data and training windows from the domain keep most of the domain fit at a much smaller cost elsewhere (Table 9).

5.3. Self-Generated Calibration

Text sampled from the dense model removes the need for any external data, which matters when the pretraining corpus is unavailable or when domain data must remain with the owner. With exact sampling at temperature one, the expected training objective is the sequence-level divergence from the dense model (Proposition 1), a principled target for a drop-in replacement, and on Llama-3.2-1B other temperatures are clearly worse. The experiments qualify the principle: on Qwen3-1.7B, the model that best matches the dense model on its own long samples is a poor model of real text, while on the other two models the exact samples work well. On Qwen3-1.7B the failure recurs in every seed and worsens at -50%; every run matches the dense model on its held-out samples about equally well, but the behavior on real text varies widely between runs, so the exact objective, met on the model's own sequences, does not pin down the compressed model elsewhere (Section 4.6). Sampling independent documents removes the failure, whatever their length limit up to the 2048 tokens of a window, and gives the most uniformly faithful compressed models on the Llama models; the price is that the objective becomes a corpus-style divergence on the model's own documents rather than the exact sequence-level one. Self-generated data are also unlimited and cheap: sampling a pool of 2.3 million tokens of documents took 5 to 26 min on one GPU, the same order as training (14 to 43 min), and drawing fresh documents for every step gave the self-generated model with the lowest perplexities in Figure 5, although the gains from more self-generated data level off sooner than those from C4.

5.4. Fidelity Is Not Utility

The comparisons of this paper rank calibration sources for a given compressor, ratio, and budget; they do not show that any compressed model can replace its dense model. A lower divergence from the dense model means a more faithful copy, not a copy that keeps the dense model's abilities: compressed by 30%, the Llama-3.1-8B models calibrated

on self-generated documents are the most uniformly faithful by divergence, yet reach a mean accuracy of 53.9%, against 66.4% for the dense model (50.9 and 63.7% without BoolQ; Section 4.10), and every compressed model loses part of its open-ended generation quality and of its long-context perplexity, while Qwen3-1.7B solves at most 4.7% of the GSM8K problems, against 70.1% for the dense model, and Llama-3.1-8B at most 6.6%, against 49.8% (Section 4.11). Instruction following suffers most: compressed by 30%, Llama-3.2-1B-Instruct follows at most 14% of the IFEval prompts, against 50% for the dense model, whichever source it is calibrated on. Abilities that the calibration windows do not exercise can disappear: after Alpaca calibration, the compressed Llama-3.2-1B no longer retrieves a key from its context, although its divergence from the dense model is not the largest of the six sources. Nor is a smaller model faster by itself: in our implementation, the merged models save 24 to 46% of the parameter memory but decode 8 to 15% more slowly at batch size one (Table 21). The recommendations below concern the choice of calibration data under these conditions, measured by the fidelity and accuracy measures we report, not the readiness of compressed models for deployment.

5.5. The First Token

For the Llama models, the calibration format matters as much as its content: compressed models are reliable only on inputs that start with the BOS token, and the density of BOS tokens in the calibration windows changes the training-free truncation by up to an order of magnitude (Appendix D). The failure is severe and general within the family. On WikiText-2 windows that do not start with BOS, the trained Llama-3.2-1B models have 112 to 1058 times the perplexity they reach on BOS-prefixed windows, and the trained Llama-3.1-8B models 1700 to 37,000 times, for every calibration source; the dense models are insensitive to the difference (Table 22). OPT-1.3B, for which the BOS token doubles as the document separator, is unaffected (at most 3%), and Qwen3, which has no BOS token, cannot fail in this way. A compressed Llama model served without its BOS token, as in batched or streaming inference that splits a document, would be unusable, although every standard benchmark, which adds BOS, would rate it normally. Quantization methods already protect the tokens that attention concentrates on [67,68]; low-rank compression needs the same care, starting with calibration windows in the format the model will be used in: calibrating without BOS removes the collapse at a small cost on BOS-prefixed inputs (Appendix D).

Table 22. Sensitivity to the first token: WikiText-2 perplexity of the trained models on windows that do not start with BOS (GPTQ-style stream windows), divided by that on BOS-prefixed windows, over the six calibration sources (five for Llama-3.1-8B); the dense models are given for reference. Qwen3-1.7B has no BOS token, so both formats coincide.

Model	Ratio	Dense	Trained, Range Over Sources
OPT-1.3B	−30%	1.00	1.00 to 1.01
OPT-1.3B	−50%	1.00	1.00 to 1.03
Llama-3.2-1B	−30%	0.99	178 to 1058
Llama-3.2-1B	−50%	0.99	112 to 232
Qwen3-1.7B	−30%	1.00	1.00 (no BOS token)
Qwen3-1.7B	−50%	1.00	1.00 (no BOS token)
Llama-3.1-8B	−30%	0.99	1722 to 36,834

5.6. Limitations

The main experiments compress three models of 1.2 to 1.7 billion parameters; Section 4.10 repeats the main comparison on an eight billion parameter model, with all six sources, three seeds at −30%, and one seed at −50%. Larger models were out of reach;

training the projectors of the 8B model already needs about 70 GB of memory on one GPU, and a larger model would need a multi-GPU implementation. All corpora are English, and the zero-shot benchmarks are multiple-choice; the behavioral evaluations of Section 4.11 use one seed and one ratio, and instruction-following is measured on one instruction-tuned model of 1.2 billion parameters. We perform an in-depth study of one learned method, LSP, and check the comparison of sources with a second one, SVD-LLM with LoRA recovery, on two models at one ratio and with a fixed budget of 512 steps; except in Table 11, we use a data-independent allocation. The per-source learning rates and the longer schedule of Section 4.8 cover one model and ratio. The controls of Section 3.4 are single runs; three seeds resolve the perplexity comparisons and most but not all accuracy comparisons (Section 4.9). Our reimplementation reproduces the trained results of LSP within 6% at the ratios studied here, but not its untrained initialization at high ratios; the difference comes from the side on which the square attention output projections are projected, and the output side, which brings the initialization close to the published one, also lowers the trained perplexities by 3–18% without changing the ranking of the sources (Appendix B). Finally, the role of the first token is supported by the density of BOS tokens and by a calibration-format intervention, not by an analysis of the attention circuits themselves.

6. Conclusions

Learned low-rank compression fits its calibration data, and this study measures what that fit preserves. Across three model families and two compression ratios with three seeds, learning removes most of the dependence of the training-free truncation on the calibration source, which reaches two orders of magnitude in perplexity, but turns it into specialization: each source best preserves its own text, perplexity favors the calibration domain, and zero-shot accuracy slightly favors instruction data, though through the content rather than the template. This specialization survives a format matched across sources, per-source learning rates, longer training, and a second learned compressor. WikiText-2, the customary calibration set, gives the least general compressed models, and a model with eight billion parameters shows the same specialization. The calibration set behaves like a training set: more distinct windows keep improving the compressed model at a fixed number of steps, and learning only the removed subspaces rather than free factors limits how far the model specializes. Text sampled from the dense model is a practical alternative that needs no external data. With exact samples at temperature one, the expected distillation objective is the sequence-level divergence between the two models, but exact samples fail on Qwen3-1.7B in every seed, for reasons other than their length or composition. Independent documents give the most uniformly faithful compressed Llama models, and a half-and-half mixture of them with SlimPajama comes close to the most faithful source on every model. The calibration data also act through the initialization, with the domain surviving the slight rotation that training applies to the removed subspaces (also over a schedule three times as long) and through their format. Compressed Llama models need the beginning-of-sequence token they were calibrated with, and collapse without it. Fidelity to the dense model is not preserved utility; when compressed by 30%, the instruction-tuned model loses most of its instruction following whatever the calibration source, some sources remove long-range retrieval, and the compressed models save memory without running faster in our implementation. We recommend reporting the calibration source with every compressed model perplexity, calibrating general-purpose models on broad or self-generated data (or a mixture of the two) and domain models on domain data, and calibrating in the input format that will be used for deployment.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code, configuration, and evaluation results are available at https://github.com/calibrank/calibrank (accessed on 15 November 2026). The calibration and evaluation corpora are public (WikiText-2, C4, SlimPajama-6B, Alpaca; Dolly-15k, GSM8K, PubMedQA, MBPP, OPUS-100, PG-19, NQ-Open, and IFEval); the self-generated calibration sets are regenerated by the released scripts from the public model weights and the recorded seeds.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ARC	AI2 Reasoning Challenge
ASVD	Activation-aware singular value decomposition
BOS	Beginning-of-sequence (token)
CC	CommonCrawl
EOS	End-of-sequence (token)
FWSVD	Fisher-weighted singular value decomposition
GPU	Graphics processing unit
GSM8K	Grade School Math 8K
IFEval	Instruction-Following Evaluation
KL	Kullback–Leibler (divergence)
KV	Key–value (cache)
LLM	Large language model
LoRA	Low-rank adaptation
LSP	Learned subspace projections
MBPP	Mostly Basic Python Problems
MLP	Multilayer perceptron
NLL	Negative log-likelihood
NQ	Natural Questions
OBQA	OpenBookQA
QR	Orthogonal–triangular (decomposition)
ReLU	Rectified linear unit
SP	SlimPajama
SVD	Singular value decomposition
SVD-LLM	Truncation-aware SVD for large language models

Appendix A. Additional Results

Figures A1–A3 repeat Figures 2–4 at the other ratio. Figure A4 gives the subspace overlaps of Section 4.4 by depth. Table A1 gives the per-task accuracy values behind Table 3, Table A2 the analysis of BoolQ (Section 4.2), Tables A3 and A4 the three-seed repetition of the full grid (Section 4.9), and Table A5 the divergence on the six independent domains (Section 4.3).

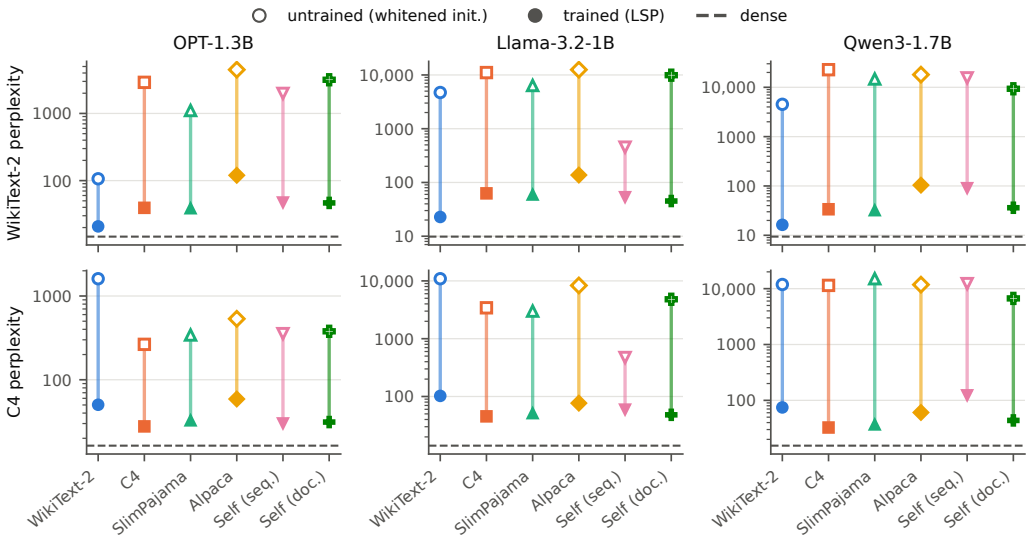


Figure A1. As in Figure 2, for the models compressed by 50%.

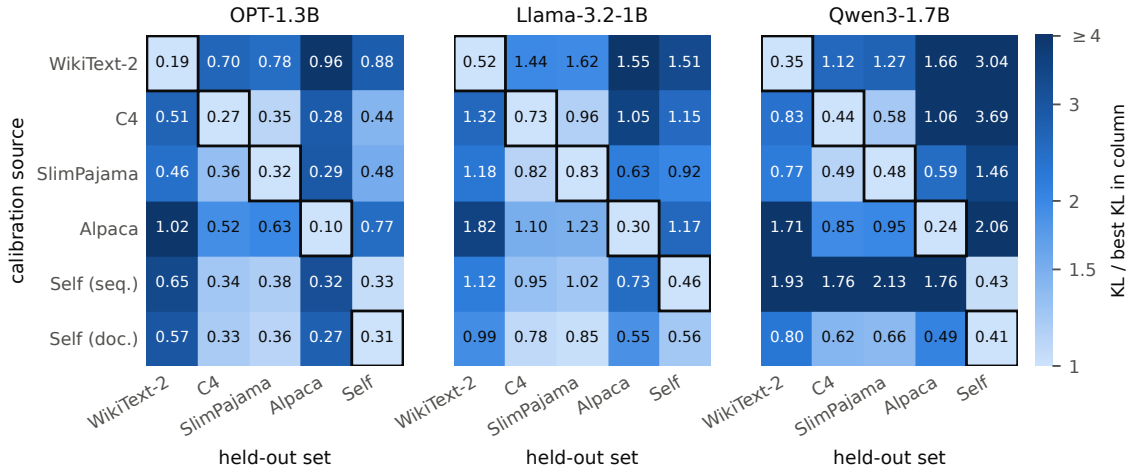


Figure A2. As in Figure 3, for the models compressed by 30%; black outlines mark the best calibration source for each held-out set.

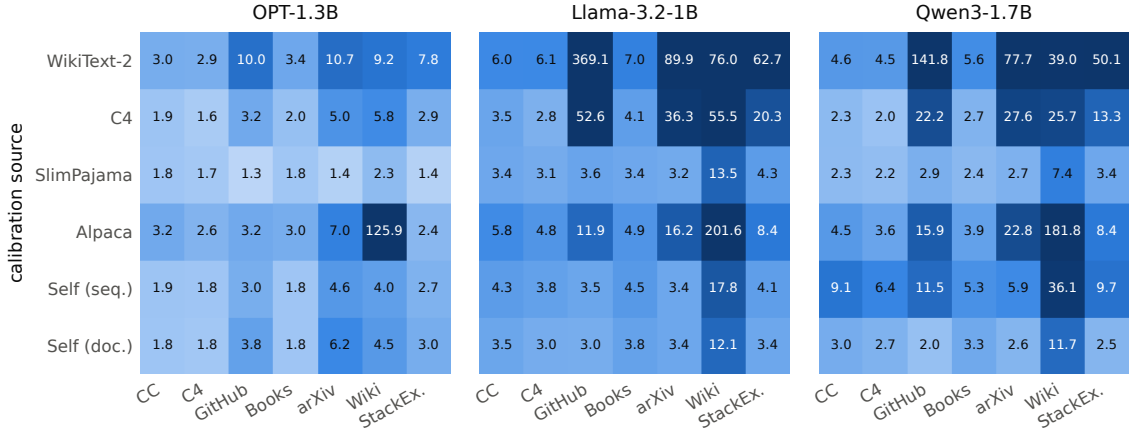


Figure A3. As in Figure 4, for the models compressed by 50%, on the same color scale (ratio to the dense model, from light to dark blue).

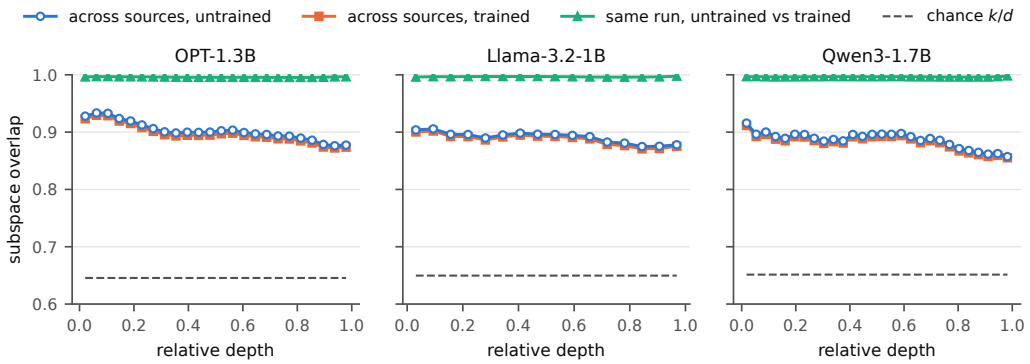


Figure A4. Overlap $\|\mathbf{U}_a^\top \mathbf{U}_b\|_F^2/k$ of the subspaces removed by each unit, models compressed by 50%, averaged per block over its units and over the 15 pairs of calibration sources (across sources, before and after training) or over the six sources (same run, before against after training). The two across-source curves nearly coincide. Dashed: the overlap k/d expected for unrelated subspaces.

Table A1. Zero-shot accuracy (%) per task of the models compressed by learned subspace projections, by calibration source. ARC-e, ARC-c: ARC-Easy, ARC-Challenge; HellaS.: HellaSwag; WinoG.: WinoGrande; OBQA: OpenBookQA.

Model	Ratio	Calibration	ARC-e	ARC-c	PIQA	HellaS.	WinoG.	OBQA	BoolQ	Mean
OPT-1.3B	0%	none (dense)	56.5	24.3	71.2	41.4	58.5	24.0	57.2	47.6
	−30%	WikiText-2	46.7	20.4	64.1	34.2	52.1	18.0	60.9	42.3
		C4	51.2	22.5	67.3	36.5	54.7	20.6	50.6	43.4
		SlimPajama	51.1	21.2	66.3	35.4	53.4	18.8	57.6	43.4
		Alpaca	53.2	23.4	67.5	35.8	54.5	19.2	47.8	43.1
		Self (seq.)	49.3	20.8	66.7	35.3	54.4	18.4	52.3	42.5
		Self (doc.)	50.5	20.7	66.3	35.7	54.7	18.0	45.9	41.7
	−50%	WikiText-2	40.9	20.3	60.4	30.9	51.5	17.0	50.0	38.7
		C4	45.5	21.4	65.5	33.5	52.6	16.4	46.1	40.1
		SlimPajama	44.8	21.2	61.9	32.0	53.0	16.0	60.9	41.4
		Alpaca	48.4	21.5	64.6	32.3	51.7	17.8	52.4	41.2
		Self (seq.)	43.3	20.6	62.2	31.9	52.9	16.8	49.1	39.5
		Self (doc.)	43.4	19.5	62.2	32.2	52.0	16.6	41.9	38.3
Llama-3.2-1B	0%	none (dense)	66.3	31.7	75.1	48.1	60.5	27.8	63.7	53.3
	−30%	WikiText-2	43.9	20.2	60.7	32.2	53.5	17.6	61.4	41.4
		C4	47.1	23.2	65.9	34.5	54.0	19.6	57.5	43.1
		SlimPajama	45.6	19.1	63.4	33.3	53.2	19.0	60.4	42.0
		Alpaca	52.6	26.3	66.4	34.3	53.4	18.8	51.3	43.3
		Self (seq.)	45.5	20.3	60.8	31.1	51.6	18.8	61.9	41.4
		Self (doc.)	47.2	20.9	64.5	32.5	50.7	18.2	59.7	42.0
	−50%	WikiText-2	38.2	18.3	57.2	28.7	53.5	16.2	42.2	36.3
		C4	39.5	21.5	61.0	29.9	50.6	16.6	55.8	39.3
		SlimPajama	37.6	19.1	58.3	29.1	50.7	14.4	57.4	38.1
		Alpaca	46.0	21.2	61.2	30.0	52.5	16.8	51.2	39.8
		Self (seq.)	37.0	18.7	57.8	28.2	52.1	13.8	61.2	38.4
		Self (doc.)	39.7	20.9	59.9	29.0	50.6	14.6	60.0	39.2
Qwen3-1.7B	0%	none (dense)	73.2	41.3	75.9	49.3	64.4	29.8	79.3	59.0
	−30%	WikiText-2	50.7	24.6	62.9	35.4	55.8	21.6	61.3	44.6
		C4	58.8	26.7	68.0	38.9	54.3	22.0	65.5	47.7
		SlimPajama	56.2	25.1	65.8	37.4	54.2	22.0	66.4	46.7
		Alpaca	65.0	32.6	67.2	37.7	56.5	24.8	64.9	49.8
		Self (seq.)	29.9	20.1	55.3	28.0	50.7	14.6	60.2	37.0
		Self (doc.)	59.6	29.4	63.9	35.1	53.2	20.4	49.6	44.4
	−50%	WikiText-2	43.4	22.2	59.5	31.0	53.0	17.6	52.8	39.9
		C4	48.5	21.3	66.1	33.8	53.8	17.6	60.8	43.1
		SlimPajama	46.4	21.2	61.4	32.0	53.6	15.0	60.5	41.5
		Alpaca	54.8	28.2	63.7	32.7	53.5	19.6	62.0	44.9
		Self (seq.)	47.2	21.3	58.4	28.9	51.1	16.8	55.6	39.9
		Self (doc.)	49.2	21.7	59.3	29.9	50.6	16.8	56.1	40.5

Table A2. BoolQ and accuracy summaries, three-seed means (one seed for Llama-3.1-8B at −50%): the range of BoolQ accuracy over the calibration sources (%), number of sources below its majority-class rate (62.2%), and source ranked first by mean accuracy over seven tasks, over six (without BoolQ), and by the mean chance-normalized accuracy.

Model	Ratio	BoolQ Range	Below 62.2%	Best, 7 Tasks	Best, 6 Tasks	Best, Chance-Norm.
OPT-1.3B	−30%	48.5–59.8	6 of 6	C4	C4	C4
OPT-1.3B	−50%	42.2–59.2	6 of 6	Alpaca	Alpaca	SlimPajama
Llama-3.2-1B	−30%	50.1–61.7	6 of 6	Alpaca	Alpaca	C4
Llama-3.2-1B	−50%	41.8–60.2	6 of 6	Alpaca	Alpaca	Alpaca
Qwen3-1.7B	−30%	53.7–67.6	3 of 6	Alpaca	Alpaca	Alpaca
Qwen3-1.7B	−50%	47.1–62.7	5 of 6	Alpaca	Alpaca	Alpaca
Llama-3.1-8B	−30%	67.3–73.2	0 of 6	Alpaca	Alpaca	C4
Llama-3.1-8B	−50%	53.6–64.1	5 of 6	Alpaca	Alpaca	Alpaca

Table A3. Three seeds of the full grid, models compressed by 30%: mean (standard deviation) of WikiText-2, C4, and SlimPajama perplexity and of the mean accuracy without BoolQ (%). Bold: best mean per column.

Calibration	OPT-1.3B				Llama-3.2-1B				Qwen3-1.7B			
	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc
WikiText-2	**17.4 (0.1)**	32.2 (1.4)	28.2 (3.7)	39.4 (0.4)	**16.9 (0.1)**	58.8 (0.8)	136 (3.3)	38.1 (0.0)	**12.8 (0.1)**	43.4 (1.5)	87.2 (4.0)	42.0 (0.2)
C4	23.6 (0.4)	**21.6 (0.2)**	17.0 (0.3)	**42.2 (0.1)**	36.5 (1.5)	**30.9 (0.3)**	57.9 (4.0)	40.3 (0.7)	21.7 (0.4)	**23.9 (0.2)**	36.1 (1.5)	44.9 (0.2)
SlimPajama	24.0 (0.4)	23.7 (0.4)	**12.6 (0.3)**	41.3 (0.4)	32.4 (1.7)	33.7 (0.3)	22.8 (1.3)	39.1 (0.1)	19.9 (0.4)	25.3 (0.4)	**15.4 (0.7)**	44.0 (0.7)
Alpaca	44.5 (0.9)	30.1 (1.1)	23.9 (2.0)	42.2 (0.1)	66.6 (0.7)	45.2 (0.2)	51.4 (0.8)	**41.6 (0.4)**	48.0 (2.6)	36.1 (1.6)	41.6 (10.0)	**46.8 (0.6)**
Self (seq.)	27.1 (0.8)	22.5 (0.1)	16.4 (0.4)	41.0 (0.1)	31.3 (0.5)	37.5 (0.8)	24.2 (0.5)	37.3 (0.2)	31.6 (8.5)	41.2 (11.6)	26.7 (8.0)	40.6 (1.7)
Self (doc.)	26.9 (0.3)	22.7 (0.1)	16.2 (0.3)	41.1 (0.3)	27.7 (0.2)	32.3 (0.1)	**20.5 (0.3)**	39.3 (0.2)	20.7 (0.4)	28.2 (0.4)	16.0 (0.3)	43.7 (0.3)

Table A4. As in Table A3, for the models compressed by 50%. Bold: best mean per column.

Calibration	OPT-1.3B				Llama-3.2-1B				Qwen3-1.7B			
	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc	Wiki	C4	SP	Acc
WikiText-2	**20.8 (0.0)**	50.2 (0.4)	54.6 (1.8)	36.6 (0.1)	**22.9 (0.1)**	102 (1.6)	262 (5.1)	35.2 (0.2)	**16.3 (0.1)**	74.5 (0.7)	177 (6.6)	37.4 (0.5)
C4	39.9 (1.5)	**27.9 (0.3)**	27.1 (0.8)	39.0 (0.4)	62.2 (2.5)	**45.3 (0.3)**	103 (5.0)	36.5 (0.2)	34.3 (0.4)	**32.6 (0.5)**	61.6 (2.0)	39.8 (0.3)
SlimPajama	38.4 (0.8)	32.9 (0.8)	**16.6 (0.6)**	38.2 (0.5)	57.3 (2.7)	51.2 (0.8)	36.3 (1.9)	35.5 (0.6)	32.9 (0.9)	35.6 (0.7)	**24.4 (1.4)**	38.6 (0.3)
Alpaca	94.1 (6.7)	49.3 (2.5)	40.5 (3.5)	**39.9 (0.2)**	139 (2.9)	77.0 (0.6)	102 (2.2)	**37.7 (0.1)**	95.8 (2.8)	59.6 (0.9)	81.1 (6.5)	**41.7 (0.7)**
Self (seq.)	48.4 (0.5)	30.3 (0.1)	27.0 (1.1)	37.6 (0.3)	53.1 (1.0)	58.4 (1.6)	38.8 (0.5)	34.6 (0.1)	205 (12.8)	182 (39.3)	225 (114)	33.8 (2.0)
Self (doc.)	45.9 (2.1)	30.8 (0.2)	26.3 (1.0)	38.1 (0.1)	45.9 (0.5)	48.7 (0.4)	**32.1 (0.4)**	35.9 (0.4)	34.8 (1.2)	42.3 (1.2)	24.5 (0.8)	38.5 (0.8)

Table A5. Divergence on the six independent domains relative to the best source for each domain, three-seed means, models compressed by 50%: geometric mean/maximum over the domains. Bold: lowest geometric mean.

Calibration	OPT-1.3B	Llama-3.2-1B	Qwen3-1.7B
WikiText-2	2.34/5.64	2.33/6.64	2.68/7.25
C4	1.34/2.61	1.72/4.48	1.74/4.41
SlimPajama	**1.03**/1.17	1.38/2.06	1.36/2.04
Alpaca	1.59/3.54	1.68/2.85	1.91/3.16
Self (seq.)	1.44/3.13	1.14/1.22	2.69/3.81
Self (doc.)	1.35/2.71	**1.00**/1.00	**1.05**/1.14

Appendix B. Reproduction Check

To check the implementation, we repeated the WikiText-2 experiments of LSP on OPT-125M and OPT-1.3B with the uniform allocation, for which LSP reports both the untrained and the trained perplexity [10]. The setting follows LSP: 1024 calibration windows of 2048 tokens from the WikiText-2 training split, batches of 32 windows, and WikiText-2 test

perplexity on GPTQ-style windows; we select the epoch on 64 validation windows. On OPT-125M, we train for 40 epochs at a learning rate of 10^{-3}, which reached the lowest validation KL among four rates from 10^{-4} to 3×10^{-3} in 10-epoch pilots at every ratio; on OPT-1.3B, for 20 epochs at 10^{-4}, the rate of the main experiments (Appendix C). Table A6 compares the results. After training, our perplexities are 3–4% above the published ones on OPT-125M, 3 and 6% above on OPT-1.3B at −30% and −50%, and 15% above at −70%. Our whitened initialization matches the published one on OPT-125M at −30% but is worse at the higher ratios, by up to a factor of 5.3 on OPT-1.3B at −70%. Untrained truncations are fragile at these ratios: with the same ranks, the perplexity of our recast is 1.4 to 7.9 times that of the oblique truncation.

Table A6. Reproduction of LSP with uniform allocation and WikiText-2 calibration: WikiText-2 test perplexity on GPTQ-style windows of the untrained whitened initialization (Init; also calibrated on windows without BOS), of the oblique truncation with the same ranks, and of the trained model, against the published numbers for the same setting [10] consisting of the whitened initialization at uniform allocation for both models, and the trained model from the initialization ablation (OPT-125M) and from the uniform allocation of the allocation ablation (OPT-1.3B).

Model	Ratio	Init	Init, No BOS	Oblique	Init (LSP)	Trained	Trained (LSP)
OPT-125M	dense			27.7 (ours), 27.6 (LSP)			
	−30%	568	580	186	582	32.6	31.7
	−50%	1126	1108	558	664	37.2	35.9
	−70%	2521	2482	1147	1538	46.7	44.8
OPT-1.3B	dense			14.6 (ours), 14.6 (LSP)			
	−30%	28.3	27.5	19.9	26.1	17.0	16.5
	−50%	114	108	40.8	74.4	19.8	18.7
	−70%	4330	3726	548	814	27.8	24.2

No implementation of LSP has been released, so we compared our initialization component by component with the choices that the LSP paper documents (Table A7; these runs use the hardware of the three-seed repetition, on which the initialization of Table A6 is reproduced within 2%). A ridge of 0.01 times the mean diagonal on the output-side Gram matrices, statistics and factorizations in fp32 instead of fp64, and calibration windows without BOS, as SliceGPT uses, each change the initialization by at most 14%. One choice that the description of LSP leaves open accounts for the gap. LSP projects every untied layer on its smaller side, which does not decide square layers, here the attention output projections. Our implementation projects them on their input side, where the orthogonal recast adds an excess to the error of the whitened optimum ([10] Proposition 1); on their output side, the whitened truncation is itself an orthogonal projector and has no excess. With the output side, our initialization comes within 1 to 10% of the published one on OPT-125M (568, 657, and 1383 against 582, 664 and 1538 at −30/50/70%) and within 4, 8, and 14% on OPT-1.3B (27.0, 80.0, and 931 against 26.1, 74.4, and 814), where it was up to 5.2 times worse. In the setting of the main experiments, the output side also gives better trained models (Table A8): on Llama-3.2-1B and OPT-1.3B compressed by 50% after calibration on WikiText-2, C4, or self-generated documents, it lowers the untrained WikiText-2 perplexity by a factor of 1.3 to 6.6 and the trained perplexities by 3 to 18%, raises the accuracy without BoolQ by 0.1 to 0.8 points, and leaves the ranking of the three sources unchanged on every corpus. In the reproduction setting as well, the output side brings the trained perplexities closer to the published ones: 32.3, 36.8, and 45.7 on OPT-125M, 1.9 to 2.4% above LSP's 31.7, 35.9, and 44.8, and 19.3 and 26.3 on OPT-1.3B at −50% and −70%, 3 and 9% above 18.7 and 24.2, against up to 4% on OPT-125M and 6 and 15% on OPT-1.3B with the input

side (Table A6). The main experiments keep the input side, so every result of the paper comes from one implementation; the comparisons between calibration sources use the same implementation for every source.

Table A7. Untrained whitened initialization of the reproduction setting under the implementation choices documented by LSP: WikiText-2 test perplexity on GPTQ-style windows, uniform allocation, 1024 WikiText-2 calibration windows. Each row changes one choice of the first; $\bar{g}$ is the mean diagonal of the Gram matrix.

Initialization	OPT-125M			OPT-1.3B		
	−30%	−50%	−70%	−30%	−50%	−70%
As in this study (square layers: input side)	568	1126	2521	28.3	114	4264
Ridge $0.01\bar{g}$ on output Gram matrices	568	1126	2521	28.3	114	4263
Statistics and factorizations in fp32	567	1126	2518	28.3	113	4291
Calibration windows without BOS	580	1109	2475	27.5	107	3726
Square layers on their output side	568	657	1383	27.0	80.0	931
Output side and ridge	568	657	1383	27.0	80.1	928
Published (LSP)	582	664	1538	26.1	74.4	814

Table A8. Square attention-output projections on their input side (the main experiments) or on their output side, Llama-3.2-1B and OPT-1.3B compressed by 50%, seed 0: WikiText-2 perplexity of the untrained whitened initialization, and the trained model as in Table 3, with accuracy without BoolQ.

Model	Calibration	Side	Untrained Wiki	Wiki	C4	SP	Acc
Llama-3.2-1B	WikiText-2	input	4716	22.8	102	261	35.1
Llama-3.2-1B	WikiText-2	output	716	21.6	94.8	220	35.9
Llama-3.2-1B	C4	input	11,075	64.5	45.3	108	36.6
Llama-3.2-1B	C4	output	2228	53.0	41.8	91.8	36.8
Llama-3.2-1B	Self (doc.)	input	9889	45.4	48.4	32.5	35.6
Llama-3.2-1B	Self (doc.)	output	2625	40.3	44.5	29.3	35.7
OPT-1.3B	WikiText-2	input	106	20.8	49.8	56.6	36.5
OPT-1.3B	WikiText-2	output	79.6	20.2	47.5	50.2	36.6
OPT-1.3B	C4	input	2905	40.7	28.0	28.0	39.4
OPT-1.3B	C4	output	538	39.3	26.8	25.7	39.7
OPT-1.3B	Self (doc.)	input	3174	47.6	30.9	27.5	38.0
OPT-1.3B	Self (doc.)	output	673	41.4	28.8	24.0	38.2

Appendix C. Learning Rate Selection

The projector learning rate was selected once per model on the validation KL of C4 calibration at −50%, with the eight epochs of the main experiments, and is shared by all calibration sources; Llama-3.1-8B is treated in the same way at −30% and checked again at −50% (Section 4.10). Table A9 lists the validation KL after each epoch for four rates. A rate of 10^{-4} reaches the lowest validation KL on all three models, 0.510 on OPT-1.3B, 1.080 on Llama-3.2-1B and 0.726 on Qwen3-1.7B, against 0.534 to 0.552, 1.093 to 1.165 and 0.765 to 0.767 for the neighboring rates 3×10^{-4} and 3×10^{-5}. The rate of 10^{-3}, the best on OPT-125M (Appendix B), is the worst on all three models. Its first epoch ends above the initial validation KL on OPT-1.3B and Llama-3.2-1B (4.79 against 2.31, and 6.17 against 5.34) and barely below it on Qwen3-1.7B (6.34 against 6.51), whereas every lower rate reduces it; the runs then recover, but end 35 to 41% above the rate of 10^{-4}. A plausible reason is the width of the projectors: the columns of $\mathbf{V}$ start orthonormal, with entries of order $d^{-1/2}$, so the same Adam step is a larger relative change at $d = 2048$ than at the $d = 768$ of OPT-125M. We use 10^{-4} for all three models.

Table A9. Validation KL (nats per token) of the learning-rate pilots after each of the eight epochs: C4 calibration, −50%. Bold: lowest value of each model.

Model	LR	1	2	3	4	5	6	7	8
OPT-1.3B	3×10^{-5}	0.748	0.624	0.587	0.570	0.559	0.554	0.552	0.553
OPT-1.3B	1×10^{-4}	0.857	0.607	0.567	0.535	0.519	0.514	**0.510**	0.511
OPT-1.3B	3×10^{-4}	1.330	0.705	0.668	0.609	0.555	0.541	0.536	0.534
OPT-1.3B	1×10^{-3}	4.785	1.196	0.957	0.842	0.772	0.736	0.721	0.718
Llama-3.2-1B	3×10^{-5}	1.584	1.330	1.251	1.210	1.185	1.173	1.165	1.165
Llama-3.2-1B	1×10^{-4}	1.597	1.310	1.205	1.139	1.109	1.091	1.081	**1.080**
Llama-3.2-1B	3×10^{-4}	2.588	1.459	1.292	1.186	1.147	1.104	1.093	1.093
Llama-3.2-1B	1×10^{-3}	6.175	3.104	2.206	1.834	1.659	1.528	1.469	1.463
Qwen3-1.7B	3×10^{-5}	1.030	0.873	0.820	0.785	0.774	0.769	0.767	0.767
Qwen3-1.7B	1×10^{-4}	1.069	0.868	0.804	0.758	0.741	0.730	**0.726**	0.727
Qwen3-1.7B	3×10^{-4}	1.617	1.006	0.934	0.856	0.807	0.772	0.765	0.765
Qwen3-1.7B	1×10^{-3}	6.338	1.581	1.301	1.149	1.095	1.037	1.020	1.018

Appendix D. Window Format and the First Token

Calibration windows and zero-shot inputs start with BOS for the models that have one, whereas perplexity windows in the convention of GPTQ and SparseGPT [39,40] do not, except for the first window of the stream. The dense models are insensitive to the difference. The compressed Llama-3.2-1B models are not: Table A10 gives their perplexity on both kinds of windows. On windows that do not start with BOS, every model calibrated with BOS collapses, whatever its calibration source and whether or not it was trained, while the trained models stay usable on BOS-prefixed windows.

Attention in large language models concentrates on the first token of a sequence, which acts as an attention sink [69] and carries massive activations that serve as implicit biases [70]. A plausible reading of Table A10 is that the dense model can create a sink on whatever token comes first, while the compressed model keeps the pathway it was calibrated on—the BOS token—and loses the others. Two interventions on the calibration format support this reading. First, the density of BOS tokens matters for the training-free initialization: exact self-samples of Llama-3.2-1B contain 7.8 BOS tokens per window on average, because the model starts new documents itself, against exactly one for every other source, and their whitened initialization is an order of magnitude better than any other (Table A10); self-generated documents, packed with a single BOS per window, do not share this advantage. Packing C4 with a BOS at the start of every document, 5.4 per window, reproduces much of the effect: it lowers the untrained WikiText-2 perplexity from 11,075 to 2602 at −50% and from 2277 to 305 at −30%, without changing the text. After training, the advantage disappears (61.5 against 62.7 at −50%), so BOS density shapes the whitened statistics, not the trained model. Second, calibrating on windows without any BOS removes the collapse: the trained model then has nearly the same perplexity on stream windows as on BOS-prefixed ones (68.1 against 68.5), at a cost of 5.8 in BOS-prefixed perplexity relative to the default format. Thus, a compressed model is reliable on the input formats it was calibrated on, and only on those.

Table A10. WikiText-2 perplexity of Llama-3.2-1B compressed by 50%, on windows that start with BOS and on GPTQ-style windows that do not (stream), before and after training. BOS/window: BOS tokens per calibration window. The last two rows calibrate on C4 with a BOS at the start of every packed document, and without any BOS. The dense model has a perplexity of 9.83 on BOS-prefixed windows and 9.76 on stream windows.

Calibration	BOS/Window	Untrained		Trained	
		BOS	Stream	BOS	Stream
WikiText-2	1.0	4716	20,034	22.8	5151
C4	1.0	11,075	66,048	62.7	12,046
SlimPajama	1.0	6364	35,785	59.6	13,815
Alpaca	1.0	12,529	36,481	137	26,346
Self (seq.)	7.8	454	7943	53.2	5961
Self (doc.)	1.0	9900	29,891	45.2	5758
C4, BOS per document	5.4	2602	14,964	61.5	15,656
C4, no BOS	0.0	10,878	13,734	68.5	68.1

The practical conclusion is simple: a compressed model of this family must receive inputs in the format it was calibrated on, here starting with BOS, and perplexities measured on BOS-less windows are not comparable across calibration formats.

References

1. Zhu, X.; Li, J.; Liu, Y.; Ma, C.; Wang, W. A Survey on Model Compression for Large Language Models. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 1556–1577. https://doi.org/10.1162/tacl_a_00704.
2. Tokcan, N.; Sofi, S.S.; Pham, V.T.; Prévost, C.; Kharbech, S.; Magnier, B.; Nguyen, T.P.; Zniyed, Y.; De Lathauwer, L. Tensor decompositions for signal processing: Theory, advances, and applications. *Signal Process.* **2026**, *238*, 110191. https://doi.org/10.1016/j.sigpro.2025.110191.
3. Pham, V.T. Basis sharing and coefficient pruning for efficient neural network compression. *Neural Netw.* **2027**, *205*, 109630. https://doi.org/10.1016/j.neunet.2026.109630.
4. Yu, H.; Wu, J. Compressing Transformers: Features Are Low-Rank, but Weights Are Not! In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11007–11015. https://doi.org/10.1609/aaai.v37i9.26304.
5. Yuan, Z.; Shang, Y.; Song, Y.; Yang, D.; Wu, Q.; Yan, Y.; Sun, G. ASVD: Activation-aware Singular Value Decomposition for Compressing Large Language Models. *arXiv* **2023**, arXiv:2312.05821.
6. Wang, X.; Zheng, Y.; Wan, Z.; Zhang, M. SVD-LLM: Truncation-aware Singular Value Decomposition for Large Language Model Compression. In Proceedings of the Thirteenth International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
7. Qi, R.; Liu, Y.; Wu, X.; Wang, X.; Li, M.; Chen, C.; Chen, J.; Chen, Y.; Weng, Q. Swift-SVD: Theoretical Optimality Meets Practical Efficiency in Low-Rank LLM Compression. In Proceedings of the 43rd International Conference on Machine Learning (ICML), Seoul, Republic of Korea, 6–11 July 2026.
8. Odema, M.; De Micheli, G.; Gou, D.; Malpeddi, N.; Vaste, P.; Song, J. Understanding Calibration and Truncation Error Propagation in Training-Free Low-Rank Compression for LLMs. In Proceedings of the Conference on Language Modeling (COLM), San Francisco, CA, USA, 6–9 October 2026.
9. Pham, V.T. Subspace-Filtered Error Propagation in Deep Low-Rank Compressed Networks. *IEEE Signal Process. Lett.* **2026**, *33*, 4416–4420. https://doi.org/10.1109/LSP.2026.3749562.
10. Bini, M.; Christensen, A.; Alaniz, S.; Goldfeder, J.; Winther, O.; LeCun, Y.; Shwartz-Ziv, R.; Akata, Z. Learning Functional Subspaces for Neural Network Compression. *arXiv* **2026**, arXiv:2609.40127.
11. Williams, M.; Aletras, N. On the Impact of Calibration Data in Post-training Quantization and Pruning. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 11–16 August 2024; pp. 10100–10118. https://doi.org/10.18653/v1/2024.acl-long.544.
12. Bandari, A.; Yin, L.; Hsieh, C.Y.; Jaiswal, A.K.; Chen, T.; Shen, L.; Krishna, R.; Liu, S. Is C4 Dataset Optimal for Pruning? An Investigation of Calibration Data for LLM Pruning. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 18089–18099.

13. Ji, Y.; Xiang, Y.; Li, J.; Xia, Q.; Li, P.; Duan, X.; Wang, Z.; Zhang, M. Beware of Calibration Data for Pruning Large Language Models. In Proceedings of the Thirteenth International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
14. Zaccone, R.; Laskaridis, S.; Ciccone, M.; Horváth, S. FlexRank: Nested Low-Rank Knowledge Decomposition for Adaptive Model Deployment. In Proceedings of the 43rd International Conference on Machine Learning (ICML), Seoul, Republic of Korea, 6–11 July 2026; Volume 306.
15. Zhang, H.; Feng, X.; Li, Z.T.; Zhu, C.; Shi, X.; Pan, X.; Liu, J.; Bai, G.; Wang, X. Per-Matrix Optimality Is Not Enough: Three-Level Optimization for Low-Rank LLM Compression. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2026, Budapest, Hungary, 24–29 October 2026; pp. 5268–5283. https://doi.org/10.18653/2026.findings-emnlp.251.
16. Liu, Z.; Oguz, B.; Zhao, C.; Chang, E.; Stock, P.; Mehdad, Y.; Shi, Y.; Krishnamoorthi, R.; Chandra, V. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024; pp. 467–484.
17. Williams, M.; Chrysostomou, G.; Aletras, N. Self-calibration for Language Model Quantization and Pruning. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Albuquerque, NM, USA, 29 April–4 May 2025; pp. 10149–10167. https://doi.org/10.18653/v1/2025.naacl-long.509.
18. Hsu, Y.C.; Hua, T.; Chang, S.; Lou, Q.; Shen, Y.; Jin, H. Language Model Compression with Weighted Low-Rank Factorization. In Proceedings of the Tenth International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
19. Wang, X.; Alam, S.; Wan, Z.; Shen, H.; Zhang, M. SVD-LLM V2: Optimizing Singular Value Truncation for Large Language Model Compression. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Albuquerque, NM, USA, 29 April–4 May 2025.
20. Lin, C.H.; Gao, S.; Smith, J.S.; Patel, A.; Tuli, S.; Shen, Y.; Jin, H.; Hsu, Y.C. MoDeGPT: Modular Decomposition for Large Language Model Compression. In Proceedings of the Thirteenth International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
21. Ashkboos, S.; Croci, M.L.; do Nascimento, M.G.; Hoefler, T.; Hensman, J. SliceGPT: Compress Large Language Models by Deleting Rows and Columns. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
22. van der Ouderaa, T.F.A.; Nagel, M.; van Baalen, M.; Asano, Y.M.; Blankevoort, T. The LLM Surgeon. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
23. Wang, Q.; Ke, J.; Tomizuka, M.; Keutzer, K.; Xu, C. Dobi-SVD: Differentiable SVD for LLM Compression and Some New Perspectives. In Proceedings of the Thirteenth International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
24. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the Tenth International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
25. Chang, C.C.; Lin, W.C.; Lin, C.Y.; Chen, C.Y.; Hu, Y.F.; Wang, P.S.; Huang, N.C.; Ceze, L.; Abdelfattah, M.; Wu, K.C. Palu: KV-Cache Compression with Low-Rank Projection. In Proceedings of the Thirteenth International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
26. Li, J.; Song, J.; Shi, J.; Xu, H.; Yu, H.; Chen, G.; Zhang, S. Robust Quantum Feature Selection With Sparse Optimization Circuit. *IEEE Trans. -Comput.-Aided Des. Integr. Circuits Syst.* **2025**, *44*, 2613–2626. https://doi.org/10.1109/TCAD.2025.3526060.
27. Li, J.; Li, Y.; Song, J.; Zhang, J.; Zhang, S. Quantum Support Vector Machine for Classifying Noisy Data. *IEEE Trans. Comput.* **2024**, *73*, 2233–2247. https://doi.org/10.1109/TC.2024.3416619.
28. Xu, H.; Li, J. Intertwine: Nonlinear Quantum Feature Selection With Multikernel Circuits. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2026**, *45*, 3346–3359. https://doi.org/10.1109/TCAD.2025.3635568.
29. Li, J.; Xie, Z.; Chen, K.; Shou, L.; Yuan, G.; Chen, G. Balanced Multi-View Feature Selection via Differential Information Loss and $\ell^2_{\ln,2}$-Norm Regularization. *IEEE Trans. Knowl. Data Eng.* **2026**, *38*, 7991–8008. https://doi.org/10.1109/TKDE.2026.3735205.
30. Li, J.; Xu, H.; Chen, G.; Zhang, S. Quantum Multiview Feature Selection With Configurable Kernel Circuits and Adaptive Fusion. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2026**, *45*, 4436–4449. https://doi.org/10.1109/TCAD.2025.3650093.
31. Li, J.; Xu, H.; Zhang, S. Adaptive kNN Graph Model. *Nat. Commun.* **2026**, *17*, 7491. https://doi.org/10.1038/s41467-026-74296-2.
32. Li, J.; Zhang, J.; Zhang, J.; Zhang, S. Quantum KNN Classification With *K* Value Selection and Neighbor Selection. *IEEE Trans. -Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 1332–1345. https://doi.org/10.1109/TCAD.2023.3345251.
33. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
34. Kim, Y.; Rush, A.M. Sequence-Level Knowledge Distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1317–1327.
35. Agarwal, R.; Vieillard, N.; Zhou, Y.; Stanczyk, P.; Ramos, S.; Geist, M.; Bachem, O. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.

36. Yousefimehr, B.; Ghatee, M.; Razavi-Far, R. Multi-Stage Self-Distillation for Class-Imbalanced Anomaly Detection. *IEEE Trans. Emerg. Top. Comput. Intell.* **2026**, *10*, 2866–2875. https://doi.org/10.1109/TETCI.2026.3671131.
37. Merity, S.; Xiong, C.; Bradbury, J.; Socher, R. Pointer Sentinel Mixture Models. In Proceedings of the Fifth International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
38. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
39. Frantar, E.; Ashkboos, S.; Hoefler, T.; Alistarh, D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
40. Frantar, E.; Alistarh, D. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In Proceedings of the 40th International Conference on Machine Learning (ICML), Honolulu, HI, USA, 23–29 July 2023; Volume 202, pp. 10323–10337.
41. Soboleva, D.; Al-Khateeb, F.; Myers, R.; Steeves, J.R.; Hestness, J.; Dey, N. *SlimPajama: A 627B Token Cleaned and Deduplicated Version of RedPajama*; Cerebras Systems blog and Hugging Face dataset cerebras/SlimPajama-627B; Cerebras: Sunnyvale, CA, USA, 2023.
42. Weber, M.; Fu, D.Y.; Anthony, Q.; Oren, Y.; Adams, S.; Alexandrov, A.; Lyu, X.; Nguyen, H.; Yao, X.; Adams, V.; Athiwaratkun, B.; et al. RedPajama: An Open Dataset for Training Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems 37, Vancouver, BC, Canada, 10–15 December 2024.
43. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Stanford Alpaca: An Instruction-following LLaMA Model. GitHub Repository, 2023. Available online: https://github.com/tatsu-lab/stanford_alpaca (accessed on 15 November 2026).
44. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; Wiley-Interscience: Hoboken, NJ, USA, 2006.
45. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The Curious Case of Neural Text Degeneration. In Proceedings of the Eighth International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
46. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X.V.; et al. OPT: Open Pre-trained Transformer Language Models. *arXiv* **2022**, arXiv:2205.01068.
47. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783.
48. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 Technical Report. *arXiv* **2025**, arXiv:2505.09388.
49. Sutawika, L.; Schoelkopf, H.; Gao, L.; Abbasi, B.; Biderman, S.; Tow, J.; Fattori, B.; Lovering, C.; Farzanehnakhaee; Phang, J.; et al. *EleutherAI/lm-Evaluation-Harness: Lm-eval v0.4.9.2*; Zenodo: Geneva, Switzerland, 2025. https://doi.org/10.5281/zenodo.17728786.
50. Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; Tafjord, O. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv* **2018**, arXiv:1803.05457.
51. Bisk, Y.; Zellers, R.; Le Bras, R.; Gao, J.; Choi, Y. PIQA: Reasoning about Physical Commonsense in Natural Language. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7432–7439. https://doi.org/10.1609/aaai.v34i05.6239.
52. Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; Choi, Y. HellaSwag: Can a Machine Really Finish Your Sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4791–4800. https://doi.org/10.18653/v1/P19-1472.
53. Sakaguchi, K.; Le Bras, R.; Bhagavatula, C.; Choi, Y. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8732–8740. https://doi.org/10.1609/aaai.v34i05.6399.
54. Mihaylov, T.; Clark, P.; Khot, T.; Sabharwal, A. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2381–2391. https://doi.org/10.18653/v1/D18-1260.
55. Clark, C.; Lee, K.; Chang, M.W.; Kwiatkowski, T.; Collins, M.; Toutanova, K. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 3–5 June 2019; pp. 2924–2936.
56. Conover, M.; Hayes, M.; Mathur, A.; Xie, J.; Wan, J.; Shah, S.; Ghodsi, A.; Wendell, P.; Zaharia, M.; Xin, R. Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM; dataset databricks/databricks-dolly-15k. *Databricks Blog*, 12 April 2023.
57. Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. Training Verifiers to Solve Math Word Problems. *arXiv* **2021**, arXiv:2110.14168.

58. Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W.; Lu, X. PubMedQA: A Dataset for Biomedical Research Question Answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 2567–2577. https://doi.org/10.18653/v1/D19-1259.
59. Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. Program Synthesis with Large Language Models. *arXiv* **2021**, arXiv:2108.07732.
60. Zhang, B.; Williams, P.; Titov, I.; Sennrich, R. Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1628–1639. https://doi.org/10.18653/v1/2020.acl-main.148.
61. Rae, J.W.; Potapenko, A.; Jayakumar, S.M.; Lillicrap, T.P. Compressive Transformers for Long-Range Sequence Modelling. In Proceedings of the Eighth International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
62. Li, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 110–119. https://doi.org/10.18653/v1/N16-1014.
63. Mohtashami, A.; Jaggi, M. Random-Access Infinite Context Length for Transformers. In Proceedings of the Advances in Neural Information Processing Systems 36, New Orleans, LA, USA, 10–16 December 2023, pp. 54567–54585. https://doi.org/10.52202/075280-2378.
64. Lee, K.; Chang, M.W.; Toutanova, K. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 6086–6096. https://doi.org/10.18653/v1/P19-1612.
65. Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; Hou, L. Instruction-Following Evaluation for Large Language Models. *arXiv* **2023**, arXiv:2311.07911.
66. Efron, B.; Tibshirani, R.J. *An Introduction to the Bootstrap*; Chapman & Hall/CRC: New York, NY, USA, 1993. https://doi.org/10.1201/9780429246593.
67. Liu, R.; Bai, H.; Lin, H.; Li, Y.; Gao, H.; Xu, Z.; Hou, L.; Yao, J.; Yuan, C. IntactKV: Improving Large Language Model Quantization by Keeping Pivot Tokens Intact. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 11–16 August 2024; pp. 7716–7741.
68. Son, S.; Park, W.; Han, W.; Kim, K.; Lee, J. Prefixing Attention Sinks can Mitigate Activation Outliers for Large Language Model Quantization. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024.
69. Xiao, G.; Tian, Y.; Chen, B.; Han, S.; Lewis, M. Efficient Streaming Language Models with Attention Sinks. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
70. Sun, M.; Chen, X.; Kolter, J.Z.; Liu, Z. Massive Activations in Large Language Models. In Proceedings of the First Conference on Language Modeling (COLM), Philadelphia, PA, USA, 7–9 October 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.