OFFLINE HIDDEN-STATE DISTILLATION FROM COMPACT TEACHER CACHES

Anonymous authors
Paper under double-blind review

ABSTRACT

For language models, logit distillation is now routinely performed offline: the teacher's next-token distributions are computed once, ahead of training, and then cached for use in subsequent training runs. Storing the full distributions would be prohibitive, but truncation or sampling compresses the cache by orders of magnitude without significantly impacting performance. Hidden-state distillation, which additionally uses the teacher's hidden states to supervise the student's intermediate computation, has no compact offline counterpart. Existing methods either perform teacher inference within the training loop, slowing down training significantly, or require enormous amounts of storage. We show that a per-token random projection of the teacher's hidden states, coarsely quantized, lets hidden-state distillation run from a cache of the same order of size as the logit cache, at the training throughput of logit distillation. With the hidden states compressed $32\times$, students distilled from four teachers of 1.5B to 8B parameters need at most about 2% more training tokens than students trained on the full hidden states to reach the same held-out loss at 1B tokens. Over training runs of up to 15B tokens, they need about 1% more tokens than the full-state students and match their downstream accuracy, while a student distilled from logits alone needs 1.4 to 1.7 times as many.

1 INTRODUCTION

Knowledge distillation (KD), which produces performant smaller models from larger ones, is now one of the primary methods for efficient LLM training (Gemma Team, 2024; Muralidharan et al., 2024; Qwen Team, 2025). KD methods use the outputs or the intermediate activations of the teacher as targets for training the student. Logit distillation (Hinton et al., 2015) minimizes a divergence between the teacher's and student's predictive distributions, typically the KL divergence (Wen et al., 2023). It is also common to add an auxiliary objective, such as a mean-squared error or a cosine loss, between the teacher's and student's hidden states at one or several layers (Romero et al., 2015; Sanh et al., 2019; Jiao et al., 2020). We refer to this as hidden-state distillation. Both require a teacher forward pass on every token of the training corpus, which is costly, since the teacher is usually several times larger than the student. This motivates precomputing the teacher's outputs once, ahead of training, and caching them, so that the cost is paid a single time and amortized over every subsequent run on the corpus (Busbridge et al., 2025).

For logits, such a cache can be compressed by orders of magnitude (Peng et al., 2025; Guigon et al., 2026). Because most of the next-token distribution's probability mass is concentrated on a small number of tokens, truncation or sampling reduces the storage cost to a few hundred bytes per token with little degradation in student quality (Peng et al., 2025; Anshumann et al., 2025). Training then runs at the step cost of a student alone, and the resident teacher's other demands on the training system, from memory to communication, disappear with it.

To our knowledge, no comparably compact cached representation of the teacher's hidden states has been proposed, which leaves hidden-state distillation significantly more costly to employ. A hidden state of Qwen3-8B has 4096 dimensions, so caching a single layer's states would cost about 8 KB per token at 16-bit precision. Caching them for 2 trillion tokens, the distillation budget of Gemma 2 2B (Gemma Team, 2024), would take 16 petabytes.

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

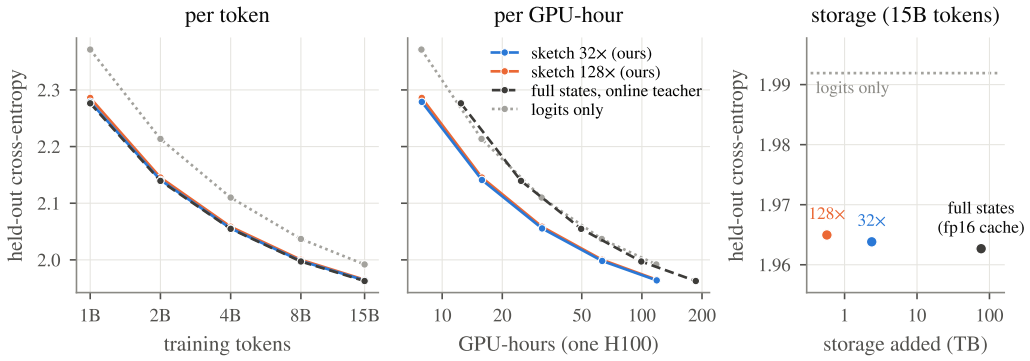


Figure 1: Students distilled from Qwen3-4B. Left: per training token, students that read a 32× or 128× sketch of the teacher's hidden states learn like the full-state student. Middle: per GPU-hour on one H100, they train at the speed of logit distillation, while the full-state student runs the teacher in the loop. Right: at 15B tokens they match full states at a few percent of the storage of an fp16 cache.

In practice, hidden-state distillation has therefore been done in one of two ways. The first computes the teacher's activations online, adding a teacher forward pass to every training step. In our experiments, with teachers two to three times the size of their students, an online teacher makes each training step 1.5 to 1.6 times as slow. The second is to store the hidden states uncompressed. Guigon et al. (2026) adopt this approach, caching the hidden states of one teacher at one layer for 168B training tokens. They cite this storage cost, which they observe cannot be reduced the way logits can, among their reasons for matching a single teacher layer.

In this paper we propose offline hidden-state distillation from a sketch of the teacher's hidden states. For each token, the sketch is a low-dimensional random projection of the teacher's state, coarsely quantized, stored together with the state's exact norm. From it, a student trains with an unbiased estimate of the uncompressed hidden-state gradient, up to clipping at the edge of the grid. The signs of the projection's columns are flipped independently for each token, so that the projection noise is independent across a batch rather than shared (Vargaftik et al., 2021). This adds no cost, since the tokens of a sequence still share one matrix multiply (Wen et al., 2018). The random signs also spread the state's few outlier channels (Dettmers et al., 2022) evenly over the coordinates, so that one coarse fixed grid serves every token without calibration (Tseng et al., 2024; Ashkboos et al., 2024). The hidden-state loss is recovered from the sketch by a variance-reduced estimator that uses the exact norms as control variates (Li et al., 2006; Kang & Hooker, 2018), so that its noise shrinks with the gap between the student's state and the teacher's. Because the projections are regenerated from a seed, nothing beyond the sketch itself is stored.

We evaluate the approach by training students from random initialization with four teachers ranging from 1.5B to 8B parameters, for up to 15B tokens. At 32× compression the students track the full-state students closely throughout, in held-out loss and in downstream accuracy, and at equal storage the sketch keeps more of the benefit of hidden-state distillation than any other compressed representation we compare.

The practical consequence is that hidden-state distillation can now be performed at a cost similar to that of logit distillation alone (Figure 1). The sketches are written once and serve every student subsequently trained on the corpus.

2 Method

Section 2.1 defines notation and the loss. Section 2.2 introduces the sketch, which Section 2.3 uses to build an unbiased estimator of the loss. Section 2.4 extends the guarantee to a batch with per-token random signs and chooses the projection. Section 2.5 quantizes the sketch to three bits, Section 2.6 collects the unbiasedness and variance results, and Section 2.7 outlines our method's cost.

2.1 PRELIMINARIES

We distill a student language model from a fixed teacher by minimizing

$$\mathcal{L} = \mathcal{L}_{\mathrm{KL}} + \lambda\, \mathcal{L}_{\mathrm{hid}}, \tag{1}$$

where $\mathcal{L}_{\mathrm{KL}}$ is the Kullback–Leibler divergence from the teacher's next-token distribution to the student's (Hinton et al., 2015), and $\mathcal{L}_{\mathrm{hid}}$ is a hidden-state term with weight λ. The logit term is served from a cache of sampled teacher logits, after Anshumann et al. (2025), with the sampler described in Appendix A.2. The question of this paper is how to serve $\mathcal{L}_{\mathrm{hid}}$ from a cache far smaller than the states themselves, without losing the benefit of the term. Let $t \in \mathbb{R}^d$ denote the teacher's hidden state at a chosen layer for a given token, let h denote the student's hidden state at a corresponding layer, and let $a = Wh \in \mathbb{R}^d$ be that state mapped into the teacher's space by a trained linear adapter W. We use the cosine loss (Sanh et al., 2019; Muralidharan et al., 2024; Guigon et al., 2026):

$$\mathcal{L}_{\mathrm{hid}} = 1 - \frac{\langle a, t\rangle}{\|a\|\,\|t\|} = \frac{1}{2}\left\| \frac{a}{\|a\|} - \frac{t}{\|t\|} \right\|^2. \tag{2}$$

The second form, a squared error between the normalized states, is the form minimized by Guigon et al. (2026), for example. The loss is summed over the tokens of a batch, weighted by λ, and differentiated through a.

2.2 SKETCHING HIDDEN STATES

The loss in equation 2 depends on t only through the scalar $\|t\|$, which we store exactly, and the inner product $\langle a, t\rangle$, which must be estimated for an a unknown at write time. Consider storing Pt for a random matrix $P \in \mathbb{R}^{k\times d}$ with $k \ll d$. At training time the inner product with the student's own projection satisfies

$$\langle Pa, Pt\rangle = a^\top P^\top P\, t, \qquad \text{so} \qquad \mathbb{E}\,\langle Pa, Pt\rangle = a^\top\, \mathbb{E}[P^\top P]\, t,$$

and therefore

$$\mathbb{E}\,\langle Pa, Pt\rangle = \langle a, t\rangle \;\text{ for all } a, t \quad\Longleftrightarrow\quad \mathbb{E}[P^\top P] = I. \tag{3}$$

Under this condition Pt is an unbiased sketch of t for inner products, and we store it together with $\|t\|$: $k+1$ values in place of d. Since this expectation is over the draw of P, we draw P afresh for each training sequence, from a seed derived from the sequence itself, and regenerate it from that seed whenever the sketch is read. P is never stored, and each sketch travels with its sequence however the data is shuffled.

2.3 LOSS ESTIMATION

The natural estimate replaces $\langle a, t\rangle$ by $\langle Pa, Pt\rangle$ and keeps the norms exact, $\|a\|$ from the student's forward pass and $\|t\|$ from the sketch:

$$1 - \frac{\langle Pa, Pt\rangle}{\|a\|\,\|t\|}. \tag{4}$$

It is unbiased, but its variance is $(1 + \mathrm{sim}(a,t)^2)/k$, where $\mathrm{sim}(a,t) = \langle a, t\rangle/(\|a\|\,\|t\|)$: largest when a and t are aligned, where the loss and its gradient vanish, so the signal is eventually overwhelmed by the noise. We instead sketch the scale-matched residual $r = a - (\|a\|/\|t\|)\, t$, for which $\mathcal{L}_{\mathrm{hid}} = \|r\|^2/(2\|a\|^2)$ by equation 2, giving the estimator

$$\hat{\mathcal{L}}_{\mathrm{hid}} = \frac{\|Pr\|^2}{2\|a\|^2}, \qquad Pr = Pa - \frac{\|a\|}{\|t\|}\, Pt. \tag{5}$$

Equivalently, equation 5 is the random-projection estimate of half the squared distance between the normalized states in equation 2, and what makes it effective is that both norms are exact. The exact norms function as control variates (Li et al., 2006; Kang & Hooker, 2018). It is unbiased in value and in gradient. Its variance is $(2/k)\,\mathcal{L}_{\mathrm{hid}}^2$ for Gaussian P and no larger for the projections we use (Appendix B), so the projection noise vanishes with the loss. The same construction serves any objective that is quadratic in a residual over exact norms, such as a normalized mean-squared error.

2.4 PER-TOKEN RANDOMIZATION

Ideally each token would have its own projection, so that the errors in its estimate are independent of every other token's, but that would be overly expensive. With a shared P, however, every token's error comes from the same matrix $M = P^\top P$, and the errors add coherently instead of averaging out. Flipping the signs of the columns of P independently for each token, as Flipout does for weight perturbations (Wen et al., 2018), keeps the efficiency of a shared P, one matrix multiply for the tokens of a sequence, while making the errors nearly independent across tokens.

Suppose that each token is projected by PD, on both the teacher and the student side, where $D = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_d)$ holds independent random signs drawn afresh for every token. The sketched inner product is then

$$\langle PDa,\, PDt\rangle = a^\top DMD\, t = \sum_{i,j} \epsilon_i \epsilon_j\, M_{ij}\, a_i t_j. \tag{6}$$

For $i \neq j$ the sign $\epsilon_i\epsilon_j$ has mean zero and is independent from one token to the next, so the off-diagonal terms decorrelate across the batch and vanish in expectation even for a fixed P. The diagonal terms are unaffected by the signs and sum to $\sum_i M_{ii}\, a_i t_i$, where $M_{ii} = \|p_i\|^2$ is the squared norm of the ith column p_i of P. Requiring every column to have unit norm makes this sum exactly $\langle a, t\rangle$. With PD in place of P, the estimator of equation 5 is therefore unbiased for every fixed P, and the errors of different tokens are uncorrelated. Consequently, redrawing P is not needed for unbiasedness. We still redraw it for each sequence, since it costs nothing and yields the variance bound of Section 2.6.

Because D only flips the signs of the coordinates of a and t before the shared P is applied, and is regenerated from the seed like P, the construction adds negligible compute and no storage. Matrices satisfying equation 3 with unit-norm columns are easy to construct. We use k random rows of a $d \times d$ Hadamard matrix scaled to $\pm 1/\sqrt{k}$. With the signs, PD is then the subsampled randomized Hadamard transform (Ailon & Chazelle, 2009; Tropp, 2011), except that its sign diagonal is drawn afresh for every token. Independent Rademacher entries $\pm 1/\sqrt{k}$ are another instance, compared in Section 2.6.

2.5 QUANTIZATION

Each coordinate of PDt is a sum of d terms of magnitude $|t_i|/\sqrt{k}$ with independent random signs. As in rotation-based quantization of model weights and activations (Tseng et al., 2024; Ashkboos et al., 2024), it therefore has mean zero and variance exactly $\|t\|^2/k$, and its tails are no heavier than those of a Gaussian with that variance, whatever outlier channels t may contain. It can therefore be quantized on a fixed uniform grid spanning $\pm 3\sigma$ with $\sigma = \|t\|/\sqrt{k}$ set by the stored norm, with no per-channel scales and no calibration; the only scale is the norm we already store. We use subtractive dither (Schuchman, 1964; Gray & Stockham, 1993), which adds a seeded uniform offset before rounding and removes it after. The quantization error is then zero-mean and independent of the value being quantized, with known expected energy $k\Delta^2/12$ for grid step Δ. In the residual Pr this error is scaled by $\|a\|/\|t\|$, so it adds $(\|a\|/\|t\|)^2 k\Delta^2/12$ to $\|PDr\|^2$ in expectation. We subtract this term, which makes the estimated loss unbiased up to clipping at the edge of the grid (Section 3.6). Because Δ is proportional to $\|t\|$, the subtracted term becomes a constant after the division by $2\|a\|^2$, so it corrects the value of the loss and leaves its gradient unchanged. For a given number of bytes per token, a coarser grid stores more coordinates, so the bit depth is a second parameter of the sketch, chosen together with k (Appendix C.1).

2.6 THEORETICAL GUARANTEES

The residual estimator of Section 2.3, the per-token signs of Section 2.4, and the dithered grid of Section 2.5 together give the following.

Proposition 1. *Let $P \in \mathbb{R}^{k\times d}$ satisfy equation 3 with unit-norm columns, let D be independent per-token random signs, and suppose that no stored coordinate falls outside the grid. Then the estimator of equation 5 with the dither energy subtracted is unbiased for $\mathcal{L}_{\mathrm{hid}}$, and its gradient with respect to a is unbiased for $\nabla_a \mathcal{L}_{\mathrm{hid}}$, for every token, every a, and every fixed P. Before quantization, its variance over the draws of P and D is at most $(2v_P/k)\,\mathcal{L}_{\mathrm{hid}}^2$, where $v_P = k\,\mathbb{E}[M_{ij}^2]$ is the off-*

diagonal variance factor of the family: $v_P = (1 - k/d)\, d/(d-1)$ *for the Hadamard rows we use, and* $v_P = 1$ *for Rademacher* P*. For Gaussian* P*, which is unbiased only in expectation over* P*, the variance is exactly* $(2/k)\, \mathcal{L}_{\text{hid}}^2$.

The proof and the exact form of the variance are in Appendix B, and Choromanski et al. (2017) derive the same factor v_P. It measures how much the columns of P overlap, which is least when the rows of P are orthogonal with equal norm (Welch, 1974; Benedetto & Fickus, 2003), so no projection with unit-norm columns has a smaller v_P than Hadamard rows. The proposition bounds the variance of the loss value. The gradient of one token's estimate is much noisier, and quantization adds gradient noise that does not shrink with the loss, but the per-token signs make both independent across tokens, so that they average out over a batch (Section 3.6).

2.7 Storage and compute cost

The sketch for one token is k three-bit values and an fp16 norm, $3k/8 + 2$ bytes. At $k \approx d/6$ this is a $32\times$ reduction from the fp16 state, 130 to 254 bytes per token for our teachers, next to about 700 bytes per token of cached logits in our experiments. A prefix of the sketch is itself a valid sketch of smaller k, so one cache serves every narrower width. Writing the cache takes one forward pass of the teacher over the corpus, which caching the logits already requires, and the projection adds 1 to 2% to it. At training time the student pays for a linear adapter and one projection of its adapted state, both negligible next to its forward and backward passes.

3 Experiments

We train students from random initialization and first ask which compact record of the teacher's hidden states best preserves the benefit of hidden-state distillation at several byte budgets (Section 3.2). We then measure how much of that benefit our sketching method keeps across four teachers (Section 3.3) and over 15B tokens of training (Section 3.4), and what it costs (Section 3.5). Finally, an ablation of the method's components shows what each one contributes (Section 3.6).

3.1 Setup

Teachers and students. We distill four teachers: Qwen3-1.7B, 4B and 8B (Qwen Team, 2025), each first adapted to the training corpus for 10k steps as in Sreenivas et al. (2024), and OLMo2-1B (OLMo Team, 2025) (1.5B parameters), used as released. Students train on the Dolmino mix (OLMo Team, 2025), a mid-training mix rich in mathematics, from random initialization, at three quarters of the teacher's width and about two thirds of its depth, 0.7B to 3.5B parameters (Appendix A.1). The hidden-state term matches the teacher layer at half depth to the student layer at the same relative depth, chosen in a preliminary comparison (Appendix A.3).

Objective and recipe. Every method trains on the same sampled teacher logits (Appendix A.2), with no next-token cross-entropy term, so the methods differ only in the hidden-state term. Replacing the sampled logits with the full-vocabulary KL changes neither the full-state nor the logits-only student by more than 0.001 nat. Following Muralidharan et al. (2024), $\lambda = H\, \mathcal{L}_{\text{KL}}/\mathcal{L}_{\text{hid}}$ is recomputed at every step and treated as a constant, with $H \in \{1, 5\}$ chosen for each record and budget by held-out cross-entropy (Appendix A.3). Every run uses 64k tokens per step, AdamW and a warmup-stable-decay schedule with a 10% cooldown, at a learning rate of 10^{-3} tuned for the full-state student (Appendix A.3).

Methods compared. *Logits only* trains on the sampled logits alone. *Full states* adds the hidden-state term computed from the teacher's uncompressed states, the reference that every compressed record approximates. *Our sketch* computes the same term from the 3-bit sketch, at $32\times$, $128\times$ and $512\times$ compression of the fp16 state. Runs with the same seed share their initialization and data order.

Other records. Section 3.2 compares our sketch with five other per-token records of the same size. *Random subspace* and *PCA subspace* store the state's coordinates in a fixed k-dimensional subspace, random or fitted to the teacher's states, as in Guo et al. (2023). *Product quantization*

codes the state with a learned codebook over coordinate groups. *Token subset* stores the full state for every Nth token and nothing for the rest. The *sparse sketch*, an ablation of our projection rather than a prior method, uses coordinate rows in place of Hadamard rows, which satisfy equation 3 but not the unit-norm columns of Section 2.4. Every record uses the same dithered grid at its own best bit depth, and records that store raw coordinates also store the two dominant channels exactly, since a coarse shared grid handles such outlier channels poorly (Appendix C.1; Dettmers et al., 2022).

Metrics. Held-out cross-entropy on a slice fixed before any run. The *benefit kept* is a method's improvement over logits only as a share of the full-state improvement. Downstream accuracy is the mean over the five tasks on which the full-state students at 300M tokens score at least three standard errors above chance: ARC-Easy, HellaSwag, PIQA and SciQ (5-shot, normalized accuracy) and LAMBADA (0-shot). At 15B tokens we add GSM8K (8-shot, chain of thought). Appendix C.6 gives every task.

Noise. Results are from one seed unless stated. Over four seeds at the Qwen3-1.7B teacher and 1B tokens, each with its own initialization, data order and sketch draw, the held-out loss of our students varies by 0.0006 to 0.0013 nat (standard deviation) when paired with the full-state run of the same seed, and by about 0.002 nat between unpaired runs. We read differences below about 0.001 nat as noise.

Table 1: Held-out cross-entropy (nats, ↓) and, in brackets, the benefit kept (↑), at three compression levels of the fp16 state (300M tokens, one seed). Bold: best in each column. The seed band on a cell is about 0.002 nat.

	Qwen3-1.7B			Qwen3-4B		
record	32×	128×	512×	32×	128×	512×
logits only	2.705 (0%)			2.658 (0%)		
rand. subspace	2.591 (80%)	2.675 (21%)	2.748 (<0%)	2.518 (85%)	2.599 (36%)	2.674 (<0%)
product quant.	2.595 (78%)	2.645 (42%)	2.724 (<0%)	2.531 (77%)	2.622 (22%)	2.713 (<0%)
PCA subspace	2.585 (85%)	2.650 (39%)	2.711 (<0%)	2.513 (88%)	2.565 (56%)	2.626 (20%)
token subset	2.596 (77%)	2.632 (51%)	2.697 (6%)	2.528 (79%)	2.562 (58%)	2.639 (12%)
sparse sketch	2.573 (93%)	2.596 (77%)	2.687 (12%)	2.505 (93%)	2.533 (76%)	2.624 (21%)
dense sketch (ours)	**2.568** (96%)	**2.586** (84%)	**2.613** (65%)	**2.497** (98%)	**2.512** (89%)	**2.545** (69%)
full states	2.563 (100%)			2.494 (100%)		

3.2 The right record for the bytes

Our sketch keeps more of the benefit than any other record, at every budget and at both teachers, and its lead grows as the budget shrinks (Table 1). At $512\times$, 8 to 10 bytes per token, it still keeps about two thirds of the benefit, while no other record keeps more than about a fifth.

At high compression, the records that fall furthest, the two subspaces and product quantization, give biased targets. Bias does not average out over a batch, so at high compression these targets hurt the student even at the smaller weight $H = 1$, and at the Qwen3-1.7B teacher all three fall below logits only. The ranking does not come from our quantizer: with fp16 values and no grid at the same bytes, our sketch still keeps the most (Appendix C.1). Downstream accuracy at 300M tokens cannot separate the records at $32\times$, but at $128\times$ and $512\times$ our sketch has the highest five-task mean at both teachers (Appendix C.6).

3.3 Across teachers

We measure how far a student falls short of the full-state student in training tokens, in the spirit of effective data (Hernandez et al., 2021; Muennighoff et al., 2023). The token ratio ρ is the number of tokens a student needs to reach the full-state student's held-out loss, divided by the number the full-state student needed (Appendix C.2). Across four teachers from two families, the $32\times$ student needs at most about 2% more tokens than the full-state student (Table 2). The shortfall does not grow with the teacher.

Table 2: Four teachers, 1B training tokens, one seed. Held-out cross-entropy (nats), the token ratio ρ of Section 3.3 where the gap is small enough to convert into tokens, and the five-task downstream mean (%). In brackets, the difference from full states, computed before rounding. Mean: over the four teachers.

	OLMo2-1B	Qwen3-1.7B	Qwen3-4B	Qwen3-8B	mean
	held-out cross-entropy, nats (difference from full states) ↓				*diff.* ↓
logits only	2.557 (+0.124)	2.428 (+0.093)	2.363 (+0.125)	2.333 (+0.144)	+0.121
sketch 512×	2.477 (+0.044)	2.360 (+0.024)	2.270 (+0.032)	2.227 (+0.039)	+0.035
sketch 128×	2.448 (+0.014)	2.346 (+0.011)	2.247 (+0.009)	2.196 (+0.008)	+0.010
sketch 32×	2.436 (+0.003)	2.337 (+0.002)	2.239 (+0.001)	2.191 (+0.002)	+0.002
full states	2.434	2.336	2.238	2.189	
	token ratio ρ ↓				*mean* ↓
sketch 128×	1.11	1.09	1.07	1.05	1.08
sketch 32×	1.02	1.01	1.01	1.01	1.01
	downstream accuracy, % (difference from full states) ↑				*diff.* ↑
logits only	51.5 (−3.8)	53.3 (−3.7)	55.4 (−4.3)	55.4 (−5.3)	−4.3
sketch 512×	53.7 (−1.5)	56.2 (−0.8)	58.2 (−1.6)	59.5 (−1.3)	−1.3
sketch 128×	54.3 (−1.0)	56.8 (−0.3)	59.3 (−0.5)	61.0 (+0.2)	−0.4
sketch 32×	55.1 (−0.2)	57.1 (+0.1)	59.9 (+0.2)	61.1 (+0.3)	+0.1
full states	55.3	57.0	59.7	60.8	

Table 3: Endpoints of the 15B-token runs, one seed: held-out cross-entropy (nats), the five-task mean and GSM8K (8-shot, chain of thought), in brackets the difference from full states, computed before rounding. The 4B run predates our switch to Hadamard rows and uses Rademacher rows, which train like them (Section 3.6).

	Qwen3-1.7B			Qwen3-4B		
	CE ↓	5-task ↑	GSM8K ↑	CE ↓	5-task ↑	GSM8K ↑
logits only	2.120 (+0.011)	64.4 (−1.1)	29.7 (−8.1)	1.992 (+0.029)	68.8 (−1.7)	44.9 (−7.4)
sketch 128×	2.112 (+0.003)	65.5 (0.0)	34.6 (−3.2)	1.965 (+0.002)	70.8 (+0.3)	51.8 (−0.5)
sketch 32×	2.110 (+0.001)	65.6 (+0.1)	38.3 (+0.5)	1.964 (+0.001)	70.5 (+0.1)	52.3 (+0.1)
full states	2.109	65.5	37.8	1.963	70.5	52.2

In downstream accuracy the 32× students match the full-state students at every teacher, within 0.3 points, while logits only trails them by 4.3 points on average.

3.4 Longer training

We follow the 1.7B and 4B teachers to 15B tokens, with cooldown branches at 1, 2, 4 and 8B tokens (Figure 1 and Table 3), and the sketch stays close to full states throughout. One token ratio fits all five branches of each run, 1.01 at 32× with both teachers and 1.04 to 1.05 at 128× (Appendix C.2). From 4B tokens on, the held-out losses of the 32× and full-state students differ by about the seed noise or less.

In held-out loss, the benefit of the hidden-state term shrinks over training, but in tokens it holds. Between 1B and 8B tokens, a logits-only student needs 1.4 to 1.7 times as many tokens as a full-state student (Appendix C.2). Downstream, the five-task mean has nearly saturated at 15B tokens, but GSM8K still separates logits only from full states, by 8.1 and 7.4 points, and the 32× students match full states on both, within 0.5 points, while the 128× student at 1.7B trails on GSM8K by 3.2 points (Table 3). These are single runs, with a GSM8K standard error of about 1.3 points.

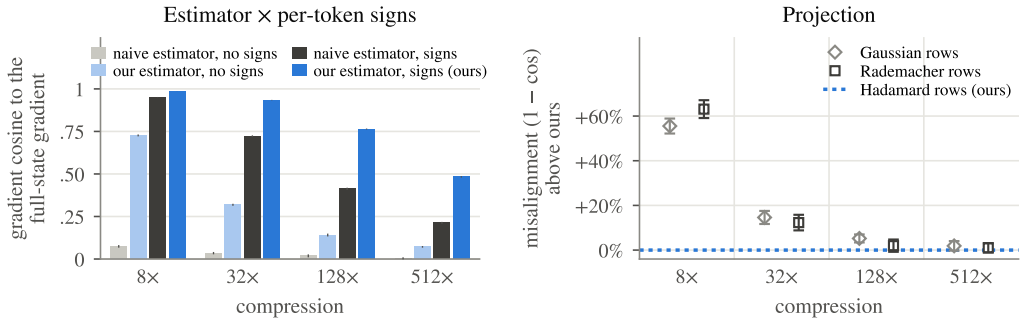


Figure 2: Each component's effect on the gradient: the cosine between the adapter gradient under a record and under full states, for a student late in the 15B-token run at the Qwen3-1.7B teacher (Appendix A.4). Left: the estimator with and without per-token signs, mean and standard error over 8 minibatches of 64k tokens. Right: the extra misalignment $(1 - \cos)$ of Gaussian and Rademacher rows over Hadamard rows (dotted, 0%), with 95% intervals.

3.5 Cost

On one H100, every cached record trains within 3% of the step time of logit distillation (Table 9 in Appendix C.3), so the sketch's advantage over a cache of fp16 states is storage. The $32\times$ sketch takes about 3% of it, and the 16 PB of states in the introduction's example become 0.5 PB. Against an online teacher, the comparison depends on the setting. Our online teacher computes its logits and states in every step and makes each minibatch 1.5 to 1.6 times as slow. With cached logits, an online teacher need only run to the deepest matched layer, which saves compute when that layer is shallow, as our half-depth layer is, and nothing when it is near the top. What an online teacher always costs is its place in the training system. Its weights and activations occupy memory beside the student, it must be served and co-located with the student, and its size is limited by the student's hardware. A cache removes all of these, and it can be written ahead of time, in independent jobs on whatever hardware is free, without the tightly connected nodes that training needs.

3.6 What each component does

To see what each component of the method contributes, we remove one at a time and train the student as in Table 1, changing nothing else (Table 10 in Appendix C.5). Figure 2 shows the same effects on the gradient itself.

The per-token signs matter most, and the estimator matters more as the compression grows. Without the signs the student keeps less than half of the benefit at $32\times$ and falls below logits only at $128\times$, because with one shared projection the errors of all tokens in a batch add up instead of averaging out (Section 2.4). The naive estimate equation 4 loses 6 points of the benefit at $32\times$ and 26 at $128\times$, because its noise does not shrink with the loss (Section 2.3). Each component matters most when the other is missing: without both, the student ends far below logits only. The gradient shows the same structure (Figure 2, left): without either component it carries almost no direction, and both together give a cosine of 0.99 at $8\times$ and 0.93 at $32\times$.

Hadamard rows align the gradient best, by a margin that vanishes with compression. Gaussian and Rademacher rows leave more than half again as much gradient misalignment as Hadamard rows at $8\times$, about 15% more at $32\times$, and at most 5% more from $128\times$ on (Figure 2, right). This is the variance factor $v_P = (1 - k/d)\, d/(d-1)$ of Hadamard rows (Section 2.6), which helps most where the sketch keeps a large share of the dimensions. In training the difference is below the noise: across the twelve cells of Table 2, runs with Rademacher rows and otherwise identical settings differ from ours by -0.0003 nat on average (standard deviation 0.0010). Hadamard rows are also the cheapest to apply.

Dither and the grid. The dither makes no measurable difference at this scale, and clipping at the edge of the grid lowers the loss value by 0.3% (Appendix C.4).

4 Related work

Hidden-state distillation. Matching a teacher's intermediate states goes back to FitNets (Romero et al., 2015) and is a common ingredient of language-model distillation (Sanh et al., 2019; Jiao et al., 2020; Muralidharan et al., 2024). How much it helps depends on the setting. Guigon et al. (2026) find a consistent gain in perplexity in LLM pretraining but not in downstream accuracy, Muralidharan et al. (2024) recommend it when the student is much shallower than the teacher, and Kim et al. (2026) find its benefit grows the further a student is pruned from its teacher. We do not take a position on when hidden-state distillation is worth using. Our contribution is that, whatever benefit it provides, it can now be delivered at about the cost of logit distillation. Guo et al. (2023) match states online through a fixed random or PCA projection, two of our baselines.

Caching teacher outputs. Caching the teacher's logits is established practice: truncation or sampling brings them to a few hundred bytes per token (Peng et al., 2025; Anshumann et al., 2025), and training from the cache avoids the teacher's cost in every later run (Busbridge et al., 2025; Ryskulov et al., 2026). Hidden states have been cached only uncompressed: Guigon et al. (2026) store one layer of one teacher for 168B tokens, DMI-Lib (Yu et al., 2026) reuses stored Qwen3-8B states across students, Shang et al. (2025) and Kim et al. (2026) precompute teacher features for vision encoders and robot policies, and draft-model training for speculative decoding writes terabytes of target states to disk (Li et al., 2026a; Park et al., 2026). The closest representation we found is that of Ma et al. (2026), who cache learned 64-dimensional codes of a video world model's features for a robot policy, with no guarantee for the loss. To our knowledge, no prior work stores a compressed per-token representation of a language model's hidden states for distillation.

Random projections and quantization. Random sign matrices (Achlioptas, 2001) and the subsampled randomized Hadamard transform (Ailon & Chazelle, 2009; Tropp, 2011) preserve inner products, Choromanski et al. (2017) give the variance factor of our estimator, and stored norms act as control variates (Li et al., 2006; Kang & Hooker, 2018). Random projections also compress activations (Shamshoum et al., 2025; Li et al., 2026b), gradients and optimizer states (Hao et al., 2024) and parameters (Wu et al., 2026), and Kalyoncuoglu & Miftachov (2025) propose one of teacher features as a distillation target. A randomized Hadamard rotation before coarse quantization spreads outlier channels, for weights and activations (Tseng et al., 2024; Ashkboos et al., 2024) and for vectors sent between workers (Suresh et al., 2017; Vargaftik et al., 2021; 2022), whose independent seeded rotations average out as our per-token signs do. Unbiased quantizers estimate vectors and inner products (Alistarh et al., 2017; Zandieh et al., 2026), with subtractive dither (Feng et al., 2026; Lin et al., 2026; Schuchman, 1964; Gray & Stockham, 1993) and at few bits (Li et al., 2014), and QJL (Zandieh et al., 2025) stores a quantized projection of each attention key with its exact norm. Our sketch instead estimates a training loss and its gradient, with noise that vanishes with the loss.

5 Conclusion

Unlike logit distillation, hidden-state distillation of language models has had no compact offline form. It has required running the teacher in every training run or storing its states at kilobytes per token. We showed that a per-token random projection of the hidden states, coarsely quantized and stored with each state's norm, lets the student estimate the hidden-state loss and its gradient without bias. At $32\times$ compression, students trained from the sketch at the speed of logit distillation match students trained on the full hidden states in held-out loss and downstream accuracy, over runs of up to 15B tokens. Hidden-state distillation can now run offline like logit distillation, at comparable cost.

Limitations. Our experiments stop at 15B tokens and students of 3.5B parameters trained from random initialization, with long runs at two teachers, mostly single seeds, and fixed rather than learned compressors as baselines. We match one layer with the cosine loss, although the estimator applies to any objective quadratic in a residual over exact norms, and at $32\times$ several sketched layers take less storage than one fp16 layer. Each run sees every token once, so repeating a token's sketch over several epochs is untested. The learning rate was tuned for the full-state student, and cost was measured on a single GPU, so the savings from removing the teacher from sharded training are argued rather than measured.

AI USE STATEMENT

We used generative AI tools to help develop and refine the method and our hypotheses, to implement methods and experiments, to give feedback on the design of experiments, to assist in formulating and proving the mathematical claims of Section 2.6 and Appendix B, and to help analyse and interpret results. We also used them to find related work and to draft and edit parts of the paper. We did not use them to generate data sets, and the remaining tasks with required disclosure do not apply to this work. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 3.1 and Appendix A describe the teachers, students, data, training recipe, logit cache, hidden-state term and gradient probe. Appendix B proves the properties of the estimator stated in Proposition 1, and Appendix C gives the operating point of every record and the per-task downstream results. We will release our code upon publication.

REFERENCES

Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 274–281. ACM, 2001. doi: 10.1145/375551.375608.

Nir Ailon and Bernard Chazelle. The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009. doi: 10.1137/060673096.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Anshumann, Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, and Joohyung Lee. Sparse Logit Sampling: Accelerating Knowledge Distillation in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18085–18108. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.885.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 100213–100240. Curran Associates, Inc., 2024. doi: 10.52202/079017-3180.

John J. Benedetto and Matthew Fickus. Finite Normalized Tight Frames. *Advances in Computational Mathematics*, 18(2-4):357–385, 2003. doi: 10.1023/a:1021323312367.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation Scaling Laws. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 5977–6045. PMLR, 2025.

Krzysztof M. Choromanski, Mark Rowland, and Adrian Weller. The Unreasonable Effectiveness of Structured Random Orthogonal Embeddings. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332. Curran Associates, Inc., 2022. doi: 10.52202/068431-2198.

Ying Feng, Piotr Indyk, Michael Kapralov, Dmitry Krachun, and Boris Prokhorov. Provable Quantization with Randomized Hadamard Transform. arXiv:2605.13810, 2026.

Gemma Team. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118, 2024.

Robert M. Gray and Thomas G. Stockham, Jr. Dithered quantizers. *IEEE Transactions on Information Theory*, 39(3):805–812, 1993. doi: 10.1109/18.256489.

Maxime Guigon, Lucas Dixon, and Michaël E. Sander. A Study on Hidden Layer Distillation for Large Language Model Pre-Training. arXiv:2605.11513, 2026.

Yi Guo, Yiqian He, Xiaoyang Li, Haotong Qin, Van Tung Pham, Yang Zhang, and Shouda Liu. RdimKD: Generic Distillation Paradigm by Dimensionality Reduction. arXiv:2312.08700, 2023.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-Rank Adapters Are Secretly Gradient Compressors. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 17554–17571. PMLR, 2024.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling Laws for Transfer. arXiv:2102.01293, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. arXiv:1503.02531, 2015.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.372.

Yusuf Kalyoncuoglu and Ratmir Miftachov. High-Dimensional Search, Low-Dimensional Solution: Decoupling Optimization from Representation. arXiv:2512.23410, 2025.

Keegan Kang and Giles Hooker. Control Variates as a Variance Reduction Technique for Random Projections. In *Pattern Recognition Applications and Methods: 6th International Conference, ICPRAM 2017, Revised Selected Papers, LNCS 10857*, pp. 1–20. Springer International Publishing, 2018. doi: 10.1007/978-3-319-93647-5_1.

Chiyoung Kim, Sanghyuk Roy Choi, and Minhyeok Lee. Recovering Aggressively Pruned Vision-Language-Action Models with Offline Hidden-State Distillation. arXiv:2609.19579, 2026.

Ping Li, Trevor J. Hastie, and Kenneth W. Church. Improving Random Projections Using Marginal Information. In *Learning Theory: 19th Annual Conference on Learning Theory, COLT 2006*, pp. 635–649. Springer Berlin Heidelberg, 2006. doi: 10.1007/11776420_46.

Ping Li, Michael Mitzenmacher, and Anshumali Shrivastava. Coding for Random Projections. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pp. 676–684. PMLR, 2014.

Shenggui Li, Chao Wang, Yikai Zhu, Yubo Wang, Fan Yin, Shuai Shi, Yefei Chen, Xiaomin Dong, Qiaoling Chen, Jin Pan, Ji Li, Yineng Zhang, Lei Yu, Yonggang Wen, Ivor Tsang, and Tianwei Zhang. SpecForge: A Flexible and Efficient Open-Source Training Framework for Speculative Decoding. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a.

Yanyi Li, Yimu Zhang, and Cong Fang. PRAC: Principal-Random Subspace for LLM Activation Compression and Memory-Efficient Training. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b.

Honghao Lin, Vahab Mirrokni, and David P. Woodruff. Pairwise-Independent Dithering for Single-Stage Hadamard Quantization. arXiv:2608.02564, 2026.

Siyuan Ma, Yutian Zhang, Boshi Zhang, Qinglian Wu, Jiaqi Zhai, Dong Wei, and Xiaojin Huang. ForeTime-VLA: Causal Future-Token Distillation from a World Action Model for Conveyor-Belt Manipulation. arXiv:2608.20735, 2026.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling Data-Constrained Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 50358–50376. Curran Associates, Inc., 2023. doi: 10.52202/075280-2191.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact Language Models via Pruning and Knowledge Distillation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 41076–41102. Curran Associates, Inc., 2024. doi: 10.52202/079017-1299.

OLMo Team. 2 OLMo 2 Furious. arXiv:2501.00656, 2025.

Jiyoung Park, Hankyu Jang, Changseok Song, and Wookeun Jung. TIDE: Temporal Incremental Draft Engine for Self-Improving LLM Inference. arXiv:2602.05145, 2026.

Hao Peng, Xin Lv, Yushi Bai, Zijun Yao, Jiajie Zhang, Lei Hou, and Juanzi Li. Pre-training Distillation for Large Language Models: A Design Space Exploration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3603–3618. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.181.

Qwen Team. Qwen3 Technical Report. arXiv:2505.09388, 2025.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. In *International Conference on Learning Representations*, 2015.

Bakbergen Ryskulov, Iker García-Ferrero, David Montero, David Jansen, Ali Hashemi, Jezabel R. Garcia, Antonio Tiene, and Román Orús. Efficient Knowledge Distillation for LLMs: Offline Top-K Logits and a Fused Chunked KL Loss. arXiv:2608.03796, 2026.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108, 2019.

L. Schuchman. Dither Signals and Their Effect on Quantization Noise. *IEEE Transactions on Communication Technology*, 12(4):162–165, 1964. doi: 10.1109/tcom.1964.1088973.

Yara Shamshoum, Nitzan Hodos, Yuval Sieradzki, and Assaf Schuster. CompAct: Compressed Activations for Memory-Efficient LLM Training. 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)*, pp. 1511–1524. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.71.

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling Diverse Vision Foundation Models for Robot Learning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, pp. 724–748. PMLR, 2025.

Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, Chenhan Yu, Wei-Chun Chen, Hayley Ross, Oluwatobi Olabiyi, Ashwath Aithal, Oleksii Kuchaiev, Daniel Korzekwa, Pavlo Molchanov, Mostofa Patwary, Mohammad Shoeybi, Jan Kautz, and Bryan Catanzaro. LLM Pruning and Distillation in Practice: The Minitron Approach. arXiv:2408.11796, 2024.

Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed Mean Estimation with Limited Communication. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3329–3337. PMLR, 2017.

Joel A. Tropp. Improved Analysis of the Subsampled Randomized Hadamard Transform. *Advances in Adaptive Data Analysis*, 3(1–2):115–126, 2011. doi: 10.1142/s1793536911000787.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 48630–48656. PMLR, 2024.

Shay Vargaftik, Ran Ben-Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben-Itzhak, and Michael Mitzenmacher. DRIVE: One-bit Distributed Mean Estimation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 362–377. Curran Associates, Inc., 2021.

Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben Itzhak, and Michael Mitzenmacher. EDEN: Communication-Efficient and Robust Distributed Mean Estimation for Federated Learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 21984–22014. PMLR, 2022.

L. Welch. Lower Bounds on the Maximum Cross Correlation of Signals. *IEEE Transactions on Information Theory*, 20(3):397–399, 1974. doi: 10.1109/tit.1974.1055219.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches. In *International Conference on Learning Representations*, 2018.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-Divergence Minimization for Sequence-Level Knowledge Distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10817–10834. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.605.

Zijun Wu, Yongchang Hao, and Lili Mou. Ultra-Low-Dimensional Prompt Tuning via Random Projection. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1284–1303. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.59.

Nengneng Yu, Sixian Xiong, Yibo Zhao, Wei Wang, and Zaoxing Liu. Enabling Performant and Flexible Model-Internal Observability for LLM Inference. arXiv:2605.11093, 2026.

Zony Yu, Yuqiao Wen, and Lili Mou. Revisiting Intermediate-Layer Matching in Knowledge Distillation: Layer-Selection Strategy Doesn't Matter (Much). In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 1686–1694. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-ijcnlp.105.

Amir Zandieh, Majid Daliri, and Insu Han. QJL: 1-Bit Quantized JL Transform for KV Cache Quantization with Zero Overhead. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25805–25813, 2025. doi: 10.1609/aaai.v39i24.34773.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online Vector Quantization with Near-optimal Distortion Rate. In *International Conference on Learning Representations*, 2026.

A Experimental details

A.1 Teachers and students

Table 4 gives the shapes. Each student scales its teacher's model width, number of attention heads and MLP width by three quarters, keeps the head size of 128, and keeps its teacher's tokenizer, vocabulary and tied or separate embeddings. Parameter counts and their ratio are totals, and in brackets without the embedding matrices.

A.2 The logit cache

As in Anshumann et al. (2025), the logit cache stores a sample of the teacher's next-token distribution rather than its top entries, which keeps the loss and its gradient unbiased. For each token we store $n = 128$ vocabulary entries. The head consists of the h largest entries that a sample of n drawn in proportion to probability would include with certainty, and they are stored exactly. The remaining $n - h$ entries are a systematic sample of the tail, drawn in proportion to probability with

Table 4: Teacher and student shapes: layers, width, parameters, and the teacher-to-student ratio (in brackets, without embeddings).

	teacher			student			
	layers	width	parameters	layers	width	parameters	ratio
Qwen3-1.7B	28	2048	1.72B (1.41B)	19	1536	0.77B (0.54B)	2.2× (2.6×)
Qwen3-4B	36	2560	4.02B (3.63B)	24	1920	1.65B (1.36B)	2.4× (2.7×)
Qwen3-8B	36	4096	8.19B (6.95B)	24	3072	3.54B (2.60B)	2.3× (2.7×)
OLMo2-1B	16	2048	1.48B (1.07B)	11	1536	0.72B (0.42B)	2.1× (2.6×)

a single uniform offset. With p and t the teacher's probabilities and log-probabilities, s the student's log-probabilities and R_h the tail mass, the per-token estimate

$$\widehat{\mathrm{KL}} = \sum_{v \in \text{head}} p_v \, (t_v - s_v) \; + \; \frac{R_h}{n-h} \sum_{\text{draws } v} (t_v - s_v)$$

is the exact head term plus a Horvitz–Thompson estimate of the tail term, so its expectation is the full KL. The student's log-probabilities enter linearly, with weights set by the teacher, so the gradient is unbiased as well. The draws are made once, when the cache is written. The record stores the entries' indices and log-probabilities, the head size and the tail mass. To check that sampling costs the student nothing, we trained the full-state and logits-only students of Table 1 at the Qwen3-1.7B teacher, with the teacher in the loop, once with the full-vocabulary KL and once with the sampled estimate, over three seeds. With the full KL the held-out loss changes by -0.0005 ± 0.0006 nat for the full-state student and by $+0.0006 \pm 0.0003$ nat for the logits-only student (mean and standard error of the paired differences). Both changes are below 0.001 nat, and neither is distinguishable from noise.

A.3 The hidden-state term

Projection. Hadamard matrices exist at every teacher width we use, by Sylvester's construction for $d = 2048$ and 4096, and for $d = 2560$ as the Kronecker product of a Paley matrix of order 20 and a Sylvester matrix of order 128.

Layer. We chose the matched layer at the Qwen3-1.7B teacher with full states and 300M tokens, pairing each teacher layer with the student layer at the same relative depth. Matching one layer at half depth kept 96% of the gain over logits only obtained by matching all 19 student layers, against 91% at quarter depth and 79% at three-quarter depth. These runs used a learning rate of 5×10^{-4}. At the Qwen3-4B teacher, with the learning rate of 10^{-3}, half depth gave the lowest held-out cross-entropy of the three depths, ahead of quarter depth by 0.003 nat and of three-quarter depth by 0.044 nat. Which layer is matched has been found to matter little in general (Yu et al., 2025).

Seeds. In our runs the projection and the signs were seeded per micro-batch of the cache as it was written, rather than per sequence as in Section 2.2. Each token's projection has the same distribution either way.

Learning rate. The learning rate of 10^{-3} was the best of $\{1, 2, 5, 10, 20\} \times 10^{-4}$ for the full-state student at the Qwen3-1.7B teacher over 100M tokens, and again the best of $\{5, 10, 20\} \times 10^{-4}$ at 4B. Every method uses it.

Weight. The weight is $\lambda = H \, \mathcal{L}_{\mathrm{KL}} / \mathcal{L}_{\mathrm{hid}}$, computed at every step from the current batch with no gradient through the ratio, where $\mathcal{L}_{\mathrm{hid}}$ is the sketched estimate when a sketch is used. Because λ divides by the batch's own estimate, the update is a ratio of two estimates from the same batch. Its bias is second order in the relative error of the batch estimate, which averages over the 64k tokens of a step. For full states we use $H = 5$, the best of $H \in \{0.2, 1, 5, 20, 100\}$ at the Qwen3-1.7B teacher and 300M tokens. Each cell of Table 1 takes the better of $H \in \{1, 5\}$ by held-out cross-entropy, chosen at the Qwen3-1.7B teacher and reused at 4B (Table 5). At $512\times$ the choice was made on the development loss after 2000 of the 4580 steps. Our sketch uses $H = 5$ at $32\times$ and $128\times$ and $H = 1$ at $512\times$ in every other experiment.

Table 5: The weight H of each cell of Table 1, identical at both teachers.

record	32×	128×	512×
rand. subspace	5	5	5
product quant.	5	1	1
PCA subspace	1	1	1
token subset	1	1	1
sparse sketch	5	1	1
dense sketch (ours)	5	5	1

A.4 THE GRADIENT PROBE

Figure 2 uses one frozen student, the full-state student of the Qwen3-1.7B long run after 13.5B tokens, with its trained adapter. Every gradient the hidden-state term sends into the student passes through the per-token gradients $\nabla_a \mathcal{L}_{\text{hid}}$, so we compare records by the gradient with respect to the adapter weight. The teacher states are an fp16 cache of the matched layer for 64 micro-batches of 8,192 tokens, grouped into 8 minibatches of 64k tokens. On each micro-batch the reference is the gradient of the exact loss equation 2, and each variant draws its projection, signs and dither as in training and forms its gradient from the same states. Both gradients are summed over each minibatch before the cosine is taken. Every variant stores the bytes of ours at 3 bits, with $k = 1360$, 340, 74 and 16 at $8\times$, $32\times$, $128\times$ and $512\times$. The left panel shows the mean cosine over the 8 minibatches with standard errors. The right panel takes, for each minibatch, the ratio of a variant's misalignment $1 - \cos$ to ours, and shows its mean minus one with a 95% t-interval.

B PROOFS

We sketch the proof of Proposition 1. Fix one token, write $M = P^\top P$, $c = \|a\|/\|t\|$ and $r = a - c\,t$, so that $\mathcal{L}_{\text{hid}} = \|r\|^2/(2\|a\|^2)$, and let the signs $D = \text{diag}(\epsilon_1, \ldots, \epsilon_d)$ be independent of P, of the dither and of a.

Unbiasedness. Since the columns of P have unit norm, $M_{ii} = 1$ and

$$\|PDr\|^2 = \|r\|^2 + Z, \qquad Z = \sum_{i \neq j} \epsilon_i \epsilon_j \, M_{ij} \, r_i r_j.$$

For $i \neq j$ the product $\epsilon_i \epsilon_j$ has mean zero, so $\mathbb{E}_D \|PDr\|^2 = \|r\|^2$ for every fixed P, and dividing by $2\|a\|^2$ gives $\mathbb{E}_D \hat{\mathcal{L}}_{\text{hid}} = \mathcal{L}_{\text{hid}}$. The expectation over D is a finite average of functions that are smooth in a, so it commutes with the gradient, and $\mathbb{E}_D \nabla_a \hat{\mathcal{L}}_{\text{hid}} = \nabla_a \mathcal{L}_{\text{hid}}$.

Quantization. With subtractive dither and no coordinate outside the grid, the stored sketch is $PDt + e$, where e has independent coordinates uniform on $[-\Delta/2, \Delta/2]$ and is independent of t, a, P and D (Schuchman, 1964; Gray & Stockham, 1993). The residual becomes $PDr - c\,e$, and $\|PDr - c\,e\|^2 = \|PDr\|^2 - 2c\,\langle PDr, e\rangle + c^2\|e\|^2$. The middle term is linear in e, so it has mean zero in value and in gradient. Since the grid spans a fixed multiple of $\|t\|/\sqrt{k}$, $\Delta = \gamma\|t\|/\sqrt{k}$ for a constant γ. After the division by $2\|a\|^2$ the last term becomes $\|e\|^2/(2\|t\|^2)$, with mean $\gamma^2/24$, which is the energy we subtract. Neither depends on a, so the corrected estimate is unbiased and its gradient is unchanged.

Variance. The sign products $\epsilon_i \epsilon_j$ of different pairs are uncorrelated, so for a fixed P, $\text{Var}_D \|PDr\|^2 = 2\sum_{i \neq j} M_{ij}^2 \, r_i^2 r_j^2$. Since $\mathbb{E}_D \|PDr\|^2$ does not depend on P, the variance over P and D is the average of this over P. For both families we use, $\mathbb{E}\, M_{ij}^2 = v_P/k$ for every pair $i \neq j$, with $v_P = 1$ for Rademacher rows by direct computation and $v_P = (1 - k/d)\, d/(d-1)$ for k random rows of a Hadamard matrix (Choromanski et al., 2017). Therefore

$$\text{Var}\, \hat{\mathcal{L}}_{\text{hid}} = \frac{2v_P}{k}\, \mathcal{L}_{\text{hid}}^2 \Big(1 - \frac{\sum_i r_i^4}{\|r\|^4}\Big) \leq \frac{2v_P}{k}\, \mathcal{L}_{\text{hid}}^2.$$

For Gaussian P with entries of variance $1/k$, $\|PDr\|^2$ is $\|r\|^2/k$ times a chi-squared variable with k degrees of freedom, so the variance is exactly $(2/k)\,\mathcal{L}_{\text{hid}}^2$.

C Additional results

C.1 Bit depth and width

Every record in Table 1 stores its values on the dithered grid of Section 2.5, at the bit depth that gave the best gradient alignment (Appendix A.4) for that record and budget, chosen at the Qwen3-1.7B teacher and reused at 4B. Records that store raw coordinates also store the two dominant channels exactly (Table 6).

Table 6: The operating point of every cell of Table 1: width (k coordinates, one token in N, or m product-quantization codes of one byte) and bit depth (b). +2: the record also stores the two dominant channels exactly in fp16.

	Qwen3-1.7B			Qwen3-4B		
record	32×	128×	512×	32×	128×	512×
rand. subspace	k 331, 3b	k 74, 3b	k 25, 2b	k 416, 3b	k 96, 3b	k 32, 2b
product quant.	m 128	m 32	m 8	m 160	m 40	m 8
PCA subspace	k 331, 3b	k 74, 3b	k 25, 2b	k 416, 3b	k 96, 3b	k 32, 2b
token subset, +2	N 6, 3b	N 28, 3b	N 96, 3b	N 6, 3b	N 24, 3b	N 96, 3b
sparse sketch, +2	k 256, 4b	k 64, 3b	k 8, 2b	k 304, 4b	k 85, 3b	k 16, 2b
dense sketch (ours)	k 340, 3b	k 74, 3b	k 16, 3b	k 416, 3b	k 96, 3b	k 21, 3b

Without quantization. We trained every record once more at the 32× budget with fp16 values and no grid (Table 7). Every record keeps less than with 3-bit values at the same bytes, and ours still keeps the most, followed by the sparse sketch. For our sketch at the Qwen3-1.7B teacher, fp16 values also keep less at 128×: 67% at 36 bytes against 84% at 30 bytes.

Table 7: The benefit kept (↑) at the 32× budget, with 3-bit values (Table 1) and with fp16 values and no grid. Records with exact channels keep them at fp16, and each fp16 cell uses the weight H of the same record at 128×. Product quantization has no fp16 variant.

	Qwen3-1.7B		Qwen3-4B	
record	3-bit	fp16	3-bit	fp16
rand. subspace	80%	<0%	85%	25%
PCA subspace	85%	15%	88%	48%
token subset	77%	44%	79%	48%
sparse sketch	93%	77%	93%	76%
dense sketch (ours)	96%	83%	98%	90%

C.2 Token ratio

The token ratio ρ of Section 3.3 is the number of training tokens a student needs to reach the held-out loss that the full-state student reaches with T tokens, divided by T. Near T the full-state loss is close to linear in $\ln T$, with slope $s = -\mathrm{d}L_{\text{full}}/\mathrm{d}\ln T$, so a student whose loss is higher by Δ on a parallel curve has $\rho = \exp(\Delta/s)$. For the long runs, s is the slope of a power law $a + bT^{-c}$ fitted to the five cooldown branches of the full-state run. The four-teacher runs end at 1B tokens without branches. For them, s is the slope of the full-state run's held-out loss over the second half of training, divided by 1.5, the ratio of the in-training slope to the slope of the cooled branches in the long runs. Over the range of that ratio in the long runs, 1.33 to 1.67, the 1B values of ρ move by at most 0.003 at 32× and 0.014 at 128×. With the paired seed noise of Section 3.1, $\sigma = 0.0010$ nat, ρ is uncertain by about ± 0.01 at 1B tokens and by up to ± 0.03 at 15B, where the slope is about six times smaller.

For the long runs we also fit the sketch's five branches with the full-state curve stretched along the token axis, $a + b(T/\rho)^{-c}$, which gives one ρ per sketch. A rule fixed before the fits would have rejected the single ρ if it predicted a held-out branch worse than separate curves by more than σ, or if the last two branches drifted the same way by more than twice their band. It rejected none. The logits-only curve has a different shape, so we fit its own branches with $a + bT^{-c}$ and read off the tokens at which it reaches the full-state loss of each branch from 1B to 8B tokens (Table 8). At 15B tokens this reading would extrapolate beyond the end of the run.

Table 8: Token ratios ρ (↓). At 1B tokens: one run per teacher. Long runs: one ρ fitted to all five branches (1B to 15B tokens) for each sketch, and for logits only the range over the branches from 1B to 8B tokens. At 15B tokens its reading would extrapolate beyond the run, so it is left out. The Qwen3-4B long run uses Rademacher rows.

	1B tokens		long runs, 1B to 15B tokens		
teacher	32×	128×	32×	128×	logits only
OLMo2-1B	1.02	1.11	—	—	—
Qwen3-1.7B	1.01	1.09	1.01	1.05	1.39–1.46
Qwen3-4B	1.01	1.07	1.01	1.04	1.47–1.73
Qwen3-8B	1.01	1.05	—	—	—

C.3 Cost

Table 9 gives the storage and step time behind Section 3.5.

Table 9: Cost on one H100. Storage: what each record adds to the logit cache over 15B tokens. Time: median per minibatch of 65,536 tokens, with the slowdown relative to logits only. The teacher in the loop computes its logits and states online, with full-vocabulary KL.

	Qwen3-1.7B		Qwen3-4B	
	storage (TB) ↓	ms per minibatch ↓	storage (TB) ↓	ms per minibatch ↓
logits only	0	868	0	1857
sketch 128×	+0.45	879 (1.01×)	+0.57	1865 (1.00×)
sketch 32×	+1.94	880 (1.01×)	+2.37	1866 (1.00×)
full states, fp16 cache	+61.44	895 (1.03×)	+76.80	1870 (1.01×)
full states, teacher in the loop	—	1338 (1.54×)	—	2928 (1.58×)

C.4 Dither and the grid

At this scale the dither makes no measurable difference. With plain rounding to the same 3-bit grid the student keeps the same 96% at 32×, and the two gradients agree within 0.003 in cosine at every compression: the random signs already make each stored value vary from token to token, so plain rounding behaves much like dithered rounding. We keep the dither because it costs nothing and makes the rounding error exactly zero-mean, which the guarantee of Section 2.6 needs. What the dither cannot remove is clipping at the edge of the $\pm 3\sigma$ grid. For our sketch it is small: with random signs every coordinate is sub-Gaussian, its tails no heavier than those of a Gaussian of the same variance whatever outlier channels the state has, and clipping lowers the loss value by 0.3%. Records that store raw channels clip more, since a few channels carry heavy tails, but a $\pm 5\sigma$ grid did not improve them in training.

C.5 Ablation

Table 10 gives the trained ablation of Section 3.6.

Table 10: One component removed at a time, trained as in Table 1 (Qwen3-1.7B teacher, 300M tokens, one seed): held-out cross-entropy (nats, ↓) and, in brackets, the benefit kept (↑). Naive estimator: equation 4. No per-token signs: one shared projection per micro-batch. Plain rounding was run at 32× only.

	32×	128×
full method	2.568 (96%)	2.586 (84%)
naive estimator	2.577 (90%)	2.623 (58%)
no per-token signs	2.638 (47%)	2.775 (<0%)
naive estimator, no signs	2.853 (<0%)	
plain rounding, no dither	2.568 (96%)	

C.6 Downstream tasks

We evaluate every student with lm-eval 0.4.11 on ten tasks. ARC-Easy, ARC-Challenge, HellaSwag, PIQA, WinoGrande, OpenBookQA, SciQ and BoolQ are 5-shot, scored by length-normalized accuracy where lm-eval reports it and by accuracy for WinoGrande and BoolQ. LAMBADA is 0-shot accuracy, and GSM8K is 8-shot with chain of thought, scored by strict exact match. The five-task mean averages ARC-Easy, HellaSwag, PIQA, SciQ and LAMBADA. A task score has a binomial standard error of about 0.5 to 2 points. At 300M and 1B tokens every student scores below the majority class of BoolQ (62%). Tables 11 and 12 give every task at 1B and 15B tokens, and Table 13 the five-task mean of every cell of Table 1.

Table 11: Downstream accuracy (%, ↑) of every task at 1B training tokens, one seed. The 5-task mean (the first five tasks) is the mean of Table 2; the last column averages all ten tasks.

	ARC-E	HS	PIQA	SciQ	LAMB	ARC-C	Wino	OBQA	BoolQ	GSM8K	5-task	all
OLMo2-1B teacher												
logits only	52.6	32.3	62.3	85.5	25.0	25.2	52.6	29.4	54.1	1.4	51.5	42.0
sketch 512×	54.8	34.2	63.8	88.1	27.7	27.0	50.5	27.6	55.2	1.6	53.7	43.1
sketch 128×	55.9	35.4	64.5	87.4	28.1	26.9	51.9	30.6	54.9	2.0	54.3	43.8
sketch 32×	57.3	36.0	63.9	89.3	29.1	28.0	51.7	30.8	53.1	2.4	55.1	44.2
full states	56.6	36.3	64.4	90.3	28.9	28.9	49.7	29.6	48.6	6.0	55.3	43.9
Qwen3-1.7B teacher												
logits only	54.2	34.0	63.2	87.2	27.8	27.6	51.9	29.2	49.4	1.2	53.3	42.6
sketch 512×	56.7	36.3	65.4	90.7	31.8	29.5	51.7	30.4	55.8	5.9	56.2	45.4
sketch 128×	57.1	37.4	64.6	91.5	33.2	29.3	51.7	28.6	52.4	10.0	56.8	45.6
sketch 32×	57.5	37.6	64.6	91.9	34.0	28.8	51.9	31.4	56.8	8.6	57.1	46.3
full states	57.2	37.9	64.1	91.8	34.1	29.8	51.6	29.0	61.1	7.8	57.0	46.4
Qwen3-4B teacher												
logits only	57.3	35.5	64.7	89.3	30.2	28.7	52.1	31.4	56.2	1.1	55.4	44.6
sketch 512×	58.6	39.5	65.8	91.6	35.4	30.6	53.2	29.0	49.0	15.0	58.2	46.8
sketch 128×	60.4	41.3	66.4	92.0	36.2	31.1	53.1	32.2	58.9	12.7	59.3	48.4
sketch 32×	62.6	41.8	66.3	92.2	36.8	30.8	53.0	31.4	56.3	11.3	59.9	48.2
full states	61.0	41.3	66.9	93.2	36.3	30.8	51.6	32.8	53.3	15.4	59.7	48.3
Qwen3-8B teacher												
logits only	56.6	36.3	65.0	90.7	28.4	30.0	51.9	31.2	42.5	4.3	55.4	43.7
sketch 512×	62.0	41.0	66.2	92.5	35.8	31.3	52.2	33.4	51.3	15.2	59.5	48.1
sketch 128×	63.3	43.3	67.6	92.5	38.3	32.4	54.1	33.4	51.6	17.5	61.0	49.4
sketch 32×	62.2	43.5	68.1	93.3	38.4	32.3	55.2	32.8	56.6	17.1	61.1	49.9
full states	61.5	44.0	68.6	91.9	37.8	31.2	54.1	34.0	57.3	16.0	60.8	49.6

Table 12: Downstream accuracy (%, ↑) of every task at 15B training tokens, one seed. The 5-task mean (the first five tasks) is the mean of Table 3; the last column averages all ten tasks.

	ARC-E	HS	PIQA	SciQ	LAMB	ARC-C	Wino	OBQA	BoolQ	GSM8K	5-task	all
Qwen3-1.7B teacher												
logits only	69.4	47.7	69.2	95.0	40.8	35.8	55.7	35.8	56.1	29.7	64.4	53.5
sketch 128×	69.4	48.9	70.0	95.1	44.1	36.9	58.9	33.2	63.8	34.6	65.5	55.5
sketch 32×	70.1	49.0	70.5	95.0	43.2	36.9	57.1	35.6	64.4	38.3	65.6	56.0
full states	69.7	49.3	70.0	95.2	43.2	37.3	57.5	34.0	67.2	37.8	65.5	56.1
Qwen3-4B teacher												
logits only	74.4	54.8	72.4	96.4	45.9	40.0	60.0	36.8	67.6	44.9	68.8	59.3
sketch 128×	75.9	57.7	74.0	96.4	49.9	44.5	60.5	37.2	73.5	51.8	70.8	62.1
sketch 32×	74.7	58.3	73.9	96.6	49.2	43.8	60.4	38.8	73.1	52.3	70.5	62.1
full states	75.6	57.7	72.4	96.0	50.6	44.6	60.7	38.6	73.0	52.2	70.5	62.1

Table 13: Five-task mean (%, ↑) of every cell of Table 1, 300M tokens, one seed.

	Qwen3-1.7B			Qwen3-4B		
record	32×	128×	512×	32×	128×	512×
logits only		46.3			47.8	
rand. subspace	49.7	47.4	45.6	51.3	49.9	46.7
product quant.	49.4	48.1	46.4	51.5	48.2	44.1
PCA subspace	50.7	48.2	46.5	51.4	49.8	47.3
token subset	49.7	48.7	46.3	51.2	50.3	47.8
sparse sketch	50.6	50.0	47.3	52.1	50.9	49.2
dense sketch (ours)	50.0	50.5	49.2	52.1	51.2	51.0
full states		50.4			52.0	