Measuring Layer-wise Intrinsic Dimensionality of FFNs in LLMs via PCA

Daehee Kim
NAVER Cloud
daehee.k@navercorp.com

Abstract

Modern large language models (LLMs) keep the feed-forward network (FFN) intermediate width identical across every layer, a convention inherited from the original Transformer. Yet mechanistic work suggests layers at different depths do qualitatively different work, raising a natural question: are their capacity demands also non-uniform, and if so, can a fixed parameter budget be allocated more wisely than uniformly? To probe this, we apply principal component analysis (PCA) to the input of each FFN down-projection and measure, for every layer, its intrinsic dimensionality as the smallest number of principal components needed to reconstruct the activation up to a fixed cosine similarity. Scanning 58 open-weight LLMs spanning 14 families and 70M to 72B parameters, we find a recurring middle-heavy profile in which middle layers carry higher intrinsic dimensionality than edge layers; the profile is robust to the calibration corpus, even across English and Chinese, and is largely preserved through post-training fine-tuning. As a direct test that this signal is structurally meaningful, we use it to guide width pruning in a knowledge-distilled student under a matched parameter budget: the PCA-guided allocation outperforms the uniform-width convention by $+2.3$ points on Qwen3-8B and $+2.5$ on Qwen2.5-3B, while lower-dispersion teachers show smaller gaps ($+0.9$ on Gemma-3-4B, $+0.5$ on LLaMA-3-8B), and a budget-matched inverted control does worst on all four teachers. Across these four teachers, the size of the gap tracks how unevenly the teacher's intrinsic dimensionality is distributed across layers (a single-number summary we call σ_{99}), and the advantage persists through downstream supervised fine-tuning. Code is available at `https://github.com/naver-ai/lidf`.

1 Introduction

Every mainstream LLM [Touvron et al., 2023, Yang et al., 2024a, Jiang et al., 2023] applies the same FFN intermediate dimension to every transformer block, a convention inherited from the original Transformer [Vaswani et al., 2017]. This uniformity is computationally convenient but embeds a strong implicit assumption: that every layer performs representational work of the same complexity. A growing body of mechanistic work [Geva et al., 2021, Meng et al., 2022, Dai et al., 2022, Geva et al., 2022] suggests this assumption does not hold: layers at different depths process qualitatively different information. Yet whether the per-layer capacity demand is also non-uniform, and whether deviating from uniform width yields measurable benefits, has not been directly characterized at scale.

We explore this gap by introducing a simple, data-agnostic measurement protocol for layer-wise intrinsic dimensionality and by validating its predictions in a controlled width-pruning experiment. Specifically, we hook the input to each FFN's down-projection, the last intermediate activation of the layer in both gated (SwiGLU) [Shazeer, 2020] and non-gated (Gaussian Error Linear Unit, GELU) [Hendrycks and Gimpel, 2016] FFNs. We apply PCA [Jolliffe, 2002] to it and define d_θ as the smallest rank at which the cosine similarity between the activation and its top-d PCA reconstruction

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

reaches a threshold θ. The result is a per-layer integer, comparable across layers and architectures, that requires only a single forward pass on a small unlabeled corpus.

Three observations emerge from a 58-model scan covering 14 families at 70M–72B parameters: (i) the per-layer d_θ profile is recurringly middle-heavy: most models exhibit higher d_θ at intermediate depth and lower d_θ at the edges; (ii) the profile is largely robust to the calibration corpus, including across English and Chinese inputs (per-layer Pearson $r \geq 0.94$ on Qwen3-8B [Yang et al., 2025]); (iii) the layer-wise dispersion $\sigma_{99} := \mathrm{std}_l(d_{\theta,l}/D_l)$ at the default threshold $\theta = 0.99$, a single-number summary of how unevenly intrinsic dimensionality is distributed across layers, is a model-internal property that is approximately preserved through post-training such as supervised fine-tuning (SFT).

As a functional test of the measurement, we use d_θ directly as a width-allocation rule for a student trained by knowledge distillation (KD) [Hinton et al., 2015]. Three students with budget-matched FFN allocations—PCA-guided (each layer's width set to $d_{\theta,l}$), UNIFORM (flat width, current convention), and INVERTED (point-wise mirror around uniform, an adversarial control)—are distilled from the same teacher with identical optimizer, data, and learning-rate schedule. On the two high-σ_{99} teachers (Qwen2.5-3B [Qwen Team, 2024] and Qwen3-8B), PCA outperforms UNIFORM by $+2.3$ to $+2.5$ percentage points (pp) and INVERTED by $+4.6$ to $+5.5$ pp on a five-task average; on the lower-σ_{99} Gemma-3-4B [Gemma Team, 2025] and LLaMA-3-8B [Grattafiori et al., 2024] the gaps against UNIFORM are smaller ($+0.5$ to $+0.9$ pp). The advantage tracks layer-wise dispersion (σ_{99}) across the four teachers and is negligible at 1.5B and below; a downstream SFT round preserves the ordering, with PCA $-$ UNIFORM at $+11.0$ pp on ARC-Challenge (ARC-C) [Clark et al., 2018], $+4.9$ pp on MMLU [Hendrycks et al., 2021], and $+3.0$ pp on GSM8K [Cobbe et al., 2021].

Contributions.

1. **Measurement protocol.** We define a layer-wise intrinsic dimensionality d_θ for FFN down-projection input activations via PCA combined with a cosine-similarity threshold (§2). The protocol is data-agnostic and label-free.
2. **Empirical characterization across 58 LLMs.** We document a recurring middle-heavy profile, calibration-data robustness (Pearson $r \geq 0.94$ across all pairs of 5 English corpora and Chinese SkyPile [Wei et al., 2023]), a pretraining-trajectory growth phase in 20 of 21 trajectories across six families, and post-training stability of the layer-wise dispersion (§3).
3. **Functional validation: controlled KD.** PCA-guided width allocation outperforms a budget-matched uniform allocation on four teachers at moderate-to-large scale; the inverted control falls below the uniform allocation; the advantage persists through SFT (§4).

2 Measurement Protocol

Where to measure. For a transformer layer l, the intermediate activation just before the down-projection is

$$h_l \;=\; \begin{cases} \phi(W_g^{(l)} x_l) \;\odot\; (W_u^{(l)} x_l) & \text{(gated FFN)} \\ \phi(W_u^{(l)} x_l) & \text{(non-gated FFN)} \end{cases} \;\in\; \mathbb{R}^{D_l}, \tag{1}$$

where x_l is the residual-stream input, ϕ is the layer's activation function (e.g., SiLU [Elfwing et al., 2018] for SwiGLU, GELU for GeGLU [Shazeer, 2020] or non-gated GELU FFNs), $\odot$ is elementwise product, and D_l is the FFN intermediate dimension. We hook h_l at `model.layers[l].mlp.down_proj.input` and treat $\{h_l\}$ as the object whose intrinsic dimensionality we measure.

Calibration set and PCA. We collect $N \approx 100$K activation tokens per layer (50 sequences of 2,048 tokens) from SlimPajama [Cerebras Systems, 2023]. Writing $\bar{h}_l$ for the per-layer activation mean, we compute the eigendecomposition of the centered covariance $\tilde{X}_l^\top \tilde{X}_l/(N-1) \in \mathbb{R}^{D_l \times D_l}$, where $X_l \in \mathbb{R}^{N \times D_l}$ stacks the activations and $\tilde{X}_l = X_l - \mathbf{1}\bar{h}_l^\top$. Let $P_d^{(l)}$ denote the orthogonal projector onto the top-d principal directions of layer l.

Definition. For a threshold $\theta \in (0,1)$, the layer-l intrinsic dimensionality is

$$d_{\theta,l} \;:=\; \min\Big\{ d \;:\; \mathbb{E}_{h \sim X_l}\Big[\cos\big(h,\; \hat{h}^{(d)}\big) \Big] \;\geq\; \theta \Big\}, \qquad \hat{h}^{(d)} \;=\; P_d^{(l)}\big(h - \bar{h}_l\big) + \bar{h}_l. \tag{2}$$

Because the principal subspace maximizes explained variance rather than expected cosine, $d_{\theta,l}$ counts the principal directions needed to reach θ and is therefore an upper bound on the dimension of the best θ-accurate subspace. In practice, $d_{\theta,l}$ is obtained by linearly interpolating the mean cosine between $d/D_l \in \{0.1, \ldots, 0.9\}$. We use $\theta = 0.99$ as the default threshold throughout the paper. We treat θ as a hyperparameter that controls the resolution of the measurement and may be adjusted depending on the application. The middle-heavy profile, calibration robustness, and pretraining-trajectory results reported in this paper are obtained at $\theta = 0.99$, and Appendix A repeats the KD comparison at $\theta = 0.97$.

Cosine similarity as the reconstruction metric. The Eckart–Young theorem [Eckart and Young, 1936] makes top-d PCA the Frobenius-optimal rank-d approximation, which corresponds to maximizing explained variance. However, the down-projection input h_l contributes to the residual stream through $W_d^{(l)} h_l$, and the updated residual is renormalized at the next normalization step, which preserves direction while standardizing magnitude. Information is therefore propagated primarily through direction, which motivates a direction-based criterion over the variance-based one underlying PCA. We accordingly define $d_{\theta,l}$ via a cosine threshold on the reconstruction $\hat{h}^{(d)}$ rather than by cumulative explained variance [Jolliffe, 2002]. The same spectrum can also be read out with an explained-variance threshold; §4.3 and Appendix E compare the two.

Layer-wise dispersion. To summarize a model's per-layer profile $(d_{\theta,l}/D_l)_l$ with a single scalar, we fix the cosine threshold to $\theta = 0.99$ and define

$$\sigma_{99} \;:=\; \mathrm{std}_l(d_{0.99,\,l} \,/\, D_l)\,, \tag{3}$$

the standard deviation of layer-relative d_θ across layers (evaluated at the default threshold). We normalize by D_l so that σ_{99} is comparable across architectures with different FFN intermediate dimensions. Across the models we evaluate, σ_{99} ranges from ~ 0.06 (nearly flat profiles) to ~ 0.21 (highly dispersed profiles), reflecting substantial variation in how non-uniformly capacity is allocated across depth. σ_{99} serves as a compact descriptor of a model's structural heterogeneity and reappears in our analysis of post-training stability (§3.4) and in the KD validation, where it tracks the size of the gain (§4.4).

3 What d_θ Measures: Empirical Characterization

We scanned 58 LLMs spanning 14 families and 70M–72B parameters with the protocol of §2. Four findings emerge.

3.1 The profile is robust to calibration data, language, and measurement site

We test whether d_θ depends on the choice of calibration corpus along two axes: distributional variation within a single language, and variation across languages. On Qwen3-8B, we measure d_θ using six 100K-token corpora: five English (SlimPajama, MMLU, ARC-C, HellaSwag [Zellers et al., 2019], WikiText [Merity et al., 2017]) and one Chinese (SkyPile [Wei et al., 2023]). Across the five English corpora, all pairwise per-layer Pearson correlations lie in $[0.953, 0.990]$ (mean 0.975), indicating that d_θ is largely insensitive to the textual domain of the calibration data. A stronger test is cross-language: per-layer correlations between Chinese SkyPile and each English corpus lie in $[0.941, 0.993]$ (mean 0.973), comparable to the all-English pairwise correlations. The lowest pairwise correlation across all 15 pairs is 0.941 (Chinese SkyPile vs. HellaSwag), reflecting the outlier commonsense distribution of HellaSwag rather than the language gap. The mean of $d_{\theta,l}/D_l$ shifts by $+0.064$ under Chinese calibration—a near-uniform offset that preserves the profile shape (Figure 1). d_θ thus reflects a structural property of the weights rather than an artifact of the calibration distribution.

The profile is also stable across measurement sites. In addition to the down-projection input h_l, we compute the spectrum at four other points of Equation (1) on every layer of six models from four families: the gate pre-activation $W_g^{(l)} x_l$, the gate post-activation $\phi(W_g^{(l)} x_l)$, the up-projection output $W_u^{(l)} x_l$, and the FFN output $W_d^{(l)} h_l$. The FFN output, which is the quantity added to the residual stream, gives a middle-heavy profile in all six models that closely follows the h_l profile (Pearson $r = 0.70$–0.95 in five models; the exception is OLMo-2-7B [OLMo Team, 2024], the lowest-σ_{99} model in the set, whose h_l profile is not middle-heavy). Only the gate post-activation departs from it

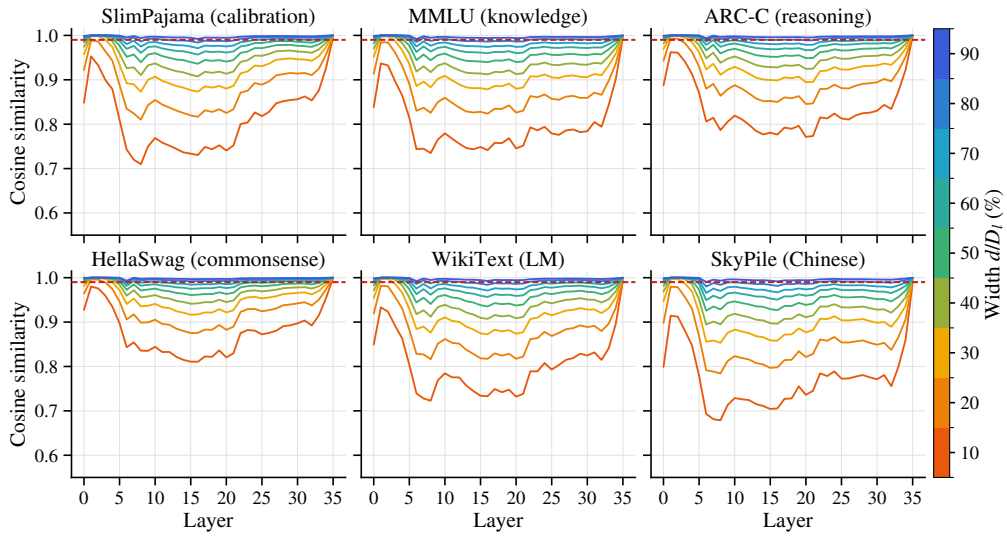


Figure 1: **Calibration robustness on Qwen3-8B.** Per-layer cosine similarity between h_l and its rank-d PCA reconstruction at width fractions $d/D_l \in \{10\%, 20\%, \ldots, 90\%\}$, measured on six 100K-token calibration corpora: five English (SlimPajama, MMLU, ARC-C, HellaSwag, WikiText) and Chinese SkyPile. Red dashed line marks the $\theta = 0.99$ threshold. All six panels show the same dip at middle layers when d/D_l is small, indicating that more principal directions are required to faithfully reconstruct middle-layer activations. Pairwise per-layer Pearson r on the extracted $d_{\theta,l}/D_l$ profiles: five English corpora min 0.953, mean 0.975; English $\leftrightarrow$ Chinese min 0.941, mean 0.973; all 15 pairs ≥ 0.941, with the minimum occurring between Chinese SkyPile and HellaSwag.

(not middle-heavy in five of six models, anticorrelated in three), as expected for a quantity that enters the FFN only through its elementwise product with $W_u^{(l)} x_l$ and never reaches the residual stream.

3.2 Per-layer profile is recurringly middle-heavy

Across the 58-model scan, the middle third of layers has higher mean $d_{\theta,l}/D_l$ than the edge thirds in 76% of the models (a mean-based criterion, not a strict bell shape). The exceptions are concentrated in two regimes: small models ($\leq$ 1.7B parameters; e.g., Qwen3-0.6B, LLaMA-3.2-1B, SmolLM2-135M/1.7B [Allal et al., 2025], BLOOM-560m/1b1 [BigScience Workshop, 2022], MAP-Neo-SL-250M [Zhang et al., 2024]), and families with low σ_{99} (OLMo-2 [OLMo Team, 2024] 1B/7B/13B, LLaMA-3.x [Grattafiori et al., 2024] 1B/3B/8B, Mistral-7B/Nemo [Jiang et al., 2023, Mistral AI, 2024]); Figure 2 shows the per-family σ_{99} distribution that motivates the latter regime. The shape is learned rather than architectural: a randomly initialized Qwen2.5-3B measured with the same protocol needs about 91% of the width on average and is not middle-heavy, whereas the trained model shows the middle-heavy profile.

3.3 σ_{99} trajectories show family-dependent dynamics during pretraining

Using publicly released checkpoints from pretraining runs across six families and two architectures, we track σ_{99} as a function of training compute (Figure 3). The runs cover BLOOM [BigScience Workshop, 2022] (560M–7.1B, GELU), OLMo-2 [OLMo Team, 2024] (1B/7B/13B, SwiGLU), Pythia [Biderman et al., 2023] (70M–12B, GELU), Amber [Liu et al., 2024] (7B, SwiGLU), MAP-Neo [Zhang et al., 2024] (7B, SwiGLU; decay phase after 3.7T tokens), and MAP-Neo-SL (250M/460M/980M, SwiGLU), for 21 trajectories in total, each with 5–8 checkpoints. Compute is normalized as the Chinchilla-optimal [Hoffmann et al., 2022] multiplier $T/20N$ (T training tokens, N parameters) to enable cross-scale comparison.

σ_{99} rises from the first checkpoint. Most trajectories exhibit a growth phase from the first checkpoint up to peak σ_{99} (20 of 21; the exception is MAP-Neo-SL-460M). Peak values lie in $\sigma_{99} \in [0.09, 0.23]$ depending on family and scale. The pattern holds across both gated SwiGLU and non-gated GELU

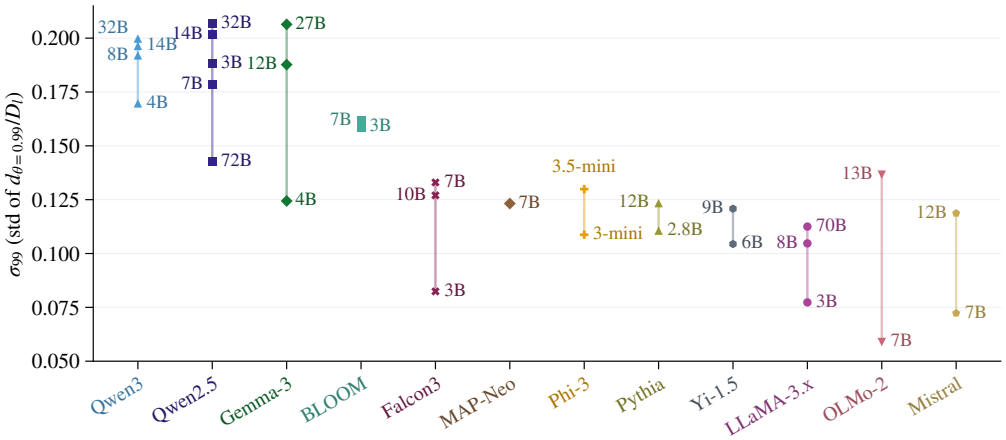


Figure 2: σ_{99} **by model family.** Each column is one family, ordered by median σ_{99}; markers are individual models of 2B parameters or more, labeled by size. Among the plotted models, σ_{99} shows a substantial association with model family. Within most families, σ_{99} also tends to rise with scale, although Qwen2.5 drops at 72B, the largest scanned model.

FFNs. This indicates progressive layer differentiation: initially similar layers diverge in d_θ as training proceeds.

Families share a profile shape. Within each family except BLOOM, the per-layer profiles have a similar shape (Figure 5 in Appendix B). σ_{99} also shows a substantial association with model family (family accounts for about half of its variance in the 58-model scan). At the 7B class, peak σ_{99} ranges from 0.103 (Amber) to 0.200 (BLOOM-7b1), with MAP-Neo-7B, OLMo-2-7B, and Pythia-6.9b in between.

Post-peak behavior differs across families. Past the peak, σ_{99} declines by more than 10% in 10 trajectories, spanning OLMo-2, BLOOM, MAP-Neo, and MAP-Neo-SL families (Figure 3b). Decline magnitudes range from -15% (MAP-Neo-SL-980M) to -60%, with the largest declines in OLMo-2-1B (-60% at $146\times$ Chinchilla) and OLMo-2-7B (-53% at $27.8\times$ Chinchilla). The remaining 11 trajectories (all 8 Pythia checkpoint sequences, Amber-7B, and MAP-Neo-SL-250M/460M) end within 10% of their peak. The decline-vs-no-decline distinction does not track total compute: small Pythia models trained at extremely high Chinchilla multipliers (93–$214\times$) show no decline, while several decline trajectories reach their peaks before Chinchilla-optimal and decline thereafter. The trajectory shape thus depends on factors beyond total compute, possibly including learning-rate (LR) schedule, data composition and architecture (§5).

3.4 σ_{99} is preserved through post-training

While pretraining substantially reshapes σ_{99} (§3.3), post-training does not. We measured general post-training (Qwen2.5-7B-Instruct [Qwen Team, 2024], Qwen3-8B [Yang et al., 2025]), domain-specific post-training (Qwen2.5-Coder-7B-Instruct [Hui et al., 2024], Qwen2.5-Math-7B-Instruct [Yang et al., 2024b]), and mid-training annealing (OLMo-2 [OLMo Team, 2024] Stage 2). In all cases σ_{99} changes by at most 10% relative to its base value (Table 6 in Appendix B), in contrast to the -53% post-peak decline of OLMo-2-7B during Stage 1 pretraining (§3.3). Domain-specific continued pretraining also leaves σ_{99} largely unchanged: Qwen2.5-Coder-7B and Qwen2.5-Math-7B, continued from Qwen2.5-7B on code and math corpora, differ from their base by -0.1% and $+4.8\%$. This stability suggests that how unevenly intrinsic dimensionality is distributed across layers is largely determined during pretraining. Distillation checkpoints fall outside this regime. DeepSeek-R1-Distill-Llama-8B [Guo et al., 2025a] has $\sigma_{99} = 0.075$, -26% relative to its base LLaMA-3.1-8B [Grattafiori et al., 2024] (0.101); such procedures involve substantial weight rewriting and are out of scope for this paper.

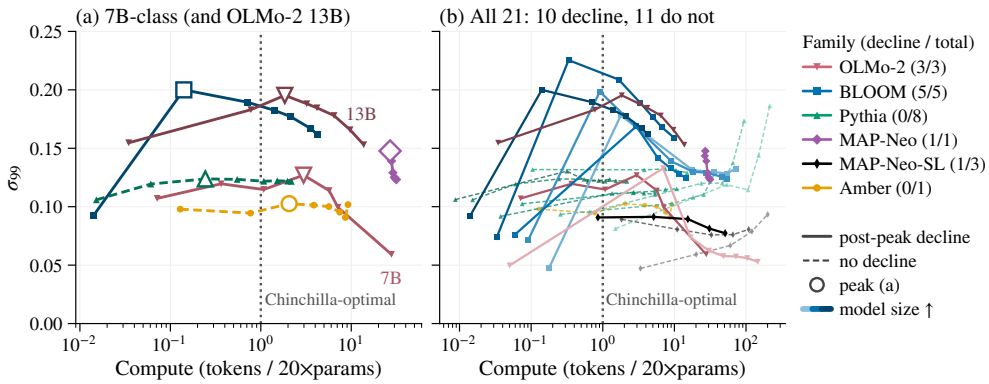


Figure 3: σ_{99} **trajectories across six families.** The horizontal axis is the Chinchilla-optimal compute multiplier: training tokens divided by $20\times$ the model's parameter count, where 20 tokens-per-parameter is the compute-optimal ratio identified by Hoffmann et al. [2022]. Values < 1 indicate under-trained models relative to compute-optimal; values > 1 indicate over-trained. Vertical dotted line marks the compute-optimal point. **(a)** 7B-class (and OLMo-2-13B) trajectories. OLMo-2 and BLOOM show growth followed by sustained decline, and MAP-Neo-7B declines during its decay phase; Pythia-6.9b (terminating at $2.2\times$ Chinchilla) and Amber show no decline. **(b)** All 21 trajectories.

4 Validation: Controlled Width Pruning under KD

We treat the 3-way KD experiment as a validation of the measurement protocol. The primary prediction is that PCA, with per-layer widths set to $d_{0.99}$, outperforms a budget-matched flat allocation UNIFORM; an ablation at the tighter $\theta = 0.97$ budget is reported in Appendix A. We additionally include an inverted allocation INVERTED (the per-layer mirror of PCA around uniform) as a falsifiability check: a finding of INVERTED $\geq$ UNIFORM would be consistent with the weaker hypothesis that any non-uniform allocation helps, while the observed INVERTED $<$ UNIFORM on all four teachers identifies the shape of the allocation, not its non-uniformity, as the relevant signal.

4.1 Setup

Models. We use four pretrained-only (base) teachers at moderate-to-large scale: Qwen2.5-3B, Qwen3-8B, Gemma-3-4B, and LLaMA-3-8B. For each teacher we distill three students under the same KD recipe.

Allocations. Let L be the number of transformer layers and D_l the teacher's FFN intermediate width at layer l. For each teacher, we fix the student's total FFN parameter budget B to the sum of the teacher's measured per-layer d_θ values: $B = \sum_l d_{\theta,l}$ at $\theta = 0.99$. The budget is therefore not a tuning hyperparameter but a direct consequence of the threshold choice.

Within budget B we construct three layer-width vectors:

- PCA: $w_l = d_{\theta,l}$ at $\theta = 0.99$, rounded to multiples of 128.
- UNIFORM: $w_l \approx B/L$.
- INVERTED: $w_l = 2\,\mathrm{uni}_l - \mathrm{pca}_l$ (the per-layer mirror of PCA around UNIFORM), clipped to $[128, D_l]$ and rescaled to B.

All three allocations match B up to rounding, and any residual (at most $+1.3\%$, for UNIFORM or INVERTED) favors the baseline; Figure 4 visualizes the per-layer width vectors for each teacher.

Initialization, distillation, and evaluation. We initialize each student via Minitron-style importance pruning [Muralidharan et al., 2024], scoring per-neuron importance by absolute activation magnitude over 128 SlimPajama calibration sequences of length 2,048, and retaining within each layer the top-scoring neurons up to that layer's allocated width. We then distill against a pure

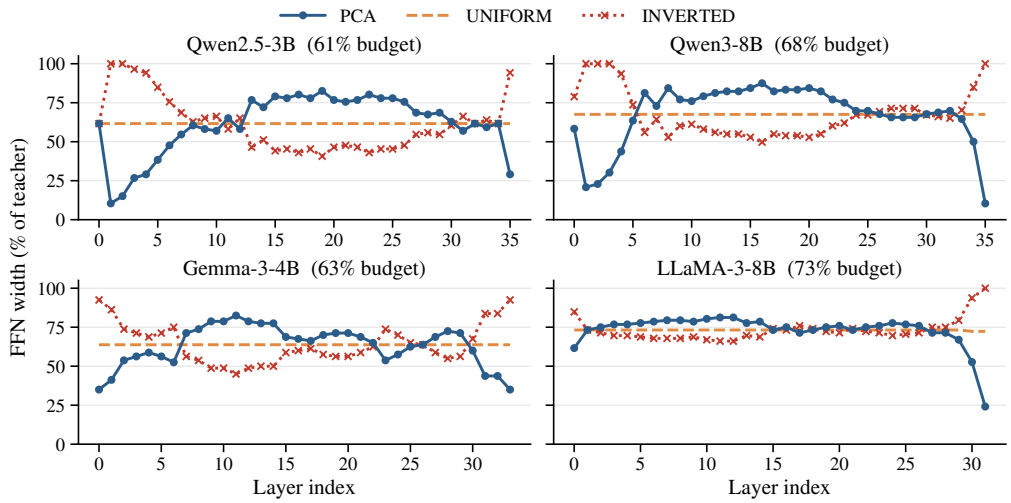


Figure 4: **Three width allocations under matched FFN budgets.** Per-layer FFN width as a percentage of the teacher's D_l; each title gives the total budget $\sum_l w_l/(LD)$. PCA (blue, solid) follows $d_{\theta,l}$ at $\theta = 0.99$; UNIFORM (orange, dashed) is B/L; INVERTED (red, dotted) is the mirror of PCA around UNIFORM, clipped at D_l. Within each panel the three curves are budget-matched (§4.1). Visual dispersion across the curves scales with σ_{99}: large in Qwen2.5-3B and Qwen3-8B ($\sigma_{99} \approx 0.19$), smaller in Gemma-3-4B ($\sigma_{99} = 0.124$) and LLaMA-3-8B ($\sigma_{99} = 0.105$).

Table 1: **3-way KD headline results at step 3K.** 5-task averages (MMLU, ARC-C, HellaSwag, WinoGrande, PIQA). Teacher avg is the unpruned model's same-protocol score. Rows are ordered by σ_{99}. Qwen2.5-3B and LLaMA-3-8B: means over three training seeds.

Model	σ_{99}	PCA avg	Δ vs. UNIFORM	Δ vs. INVERTED	Teacher avg
Qwen3-8B	0.192	67.48	+2.28	+4.63	72.23
Qwen2.5-3B	0.188	60.10	+2.50	+5.52	66.49
Gemma-3-4B	0.124	54.24	+0.88	+2.05	67.56
LLaMA-3-8B	0.105	61.83	+0.52	+0.99	67.42

Kullback–Leibler (KL) objective $\mathcal{L} = \mathrm{KL}(p_{\text{teacher}} \,\|\, p_{\text{student}})$ on SlimPajama at sequence length 2,048, effective batch 256 sequences, peak LR $1 \cdot 10^{-4}$ with a short warmup and cosine decay (constant for Qwen3-8B; Appendix C), 3K steps ($\approx$ 1.57B tokens). For evaluation we use five lm-eval-harness [Gao et al., 2024] tasks (MMLU [Hendrycks et al., 2021], ARC-Challenge [Clark et al., 2018], HellaSwag [Zellers et al., 2019], WinoGrande [Sakaguchi et al., 2021], PIQA [Bisk et al., 2020]) at harness-default few-shot settings, log-likelihood scoring without chat template, reporting the larger of `acc` and `acc_norm` per task and the unweighted 5-task average as the headline number.

4.2 The 3-way ordering: PCA > UNIFORM > INVERTED

Monotonic ordering across teachers. On all four teachers the three-way ordering PCA > UNIFORM > INVERTED holds at step 3K on the 5-task average, and on Qwen2.5-3B every PCA run beats every UNIFORM run, which beats every INVERTED run; it also holds at the 8K-step horizon on the three teachers trained that long (Appendices C and H). At the tighter $\theta = 0.97$ budget (42.9% of the teacher's FFN parameters), the advantage over UNIFORM is +2.86 pp, no smaller than at $\theta = 0.99$ (Appendix A). Table 1 reports headline numbers at step 3K and Table 2 the per-task breakdown for Qwen2.5-3B, where PCA achieves the highest score on all five tasks, with the largest margin on MMLU (+4.71 pp vs. UNIFORM) and the smallest on PIQA (+0.80 pp). The ordering also holds without a teacher (§4.5).

Init-vs-trained reversal rules out a static init-advantage account. At step 0, before any KD, the ordering on Qwen3-8B is UNIFORM (51.47) > PCA (49.92) > INVERTED (47.84). After 1K steps the ordering reverses to PCA (66.10) > UNIFORM (63.05) > INVERTED (60.35) and persists

Table 2: **Per-task results for Qwen2.5-3B at step 3K**, means over three training seeds per allocation. The Δ(PCA $-$ INVERTED) column is led by MMLU, where the inverted allocation reaches 37.93 against 52.25 for PCA.

Task	PCA	UNIFORM	INVERTED	Δ(PCA $-$ UNIFORM)	Δ(PCA $-$ INVERTED)
MMLU	**52.25**	47.54	37.93	+4.71	+14.32
ARC-C	**42.01**	41.18	38.99	+0.82	+3.01
HellaSwag	**65.77**	63.79	61.16	+1.98	+4.61
WinoGrande	**64.85**	60.64	60.04	+4.21	+4.81
PIQA	**75.61**	74.81	74.77	+0.80	+0.83
Avg	**60.10**	57.59	54.58	+2.50	+5.52

Table 3: **PCA against two other allocation rules at step 3K.** Qwen2.5-3B: mean $\pm$ SD over three training seeds per allocation. UNIFORM scores 57.59 on Qwen2.5-3B and 53.36 on Gemma-3-4B (Table 7).

Teacher	Allocation	5-task avg	MMLU	Δ (PCA $-$ alloc.)
Qwen2.5-3B	PCA	60.10 ± 0.22	52.25	—
	PCA-EV	$\mathbf{60.18} \pm 0.10$	**52.56**	-0.08
	SLIMLLM	56.69 ± 0.05	44.62	+3.41
Gemma-3-4B	PCA	**54.24**	**42.59**	—
	PCA-EV	52.60	38.76	+1.64
	SLIMLLM	52.88	38.75	+1.36

at steps 2K and 3K. We observe the same pattern on LLaMA-3-8B, where UNIFORM leads at init and PCA leads the three-seed mean at step 3K and the single longer run at steps 5K and 8K (Table 7 in Appendix C). On Qwen3-8B and LLaMA-3-8B, PCA's advantage therefore does not come from a better post-pruning initialization; it arises during KD training.

4.3 Comparison with other allocation rules

We compare PCA with two other budget-matched rules on Qwen2.5-3B and Gemma-3-4B (Table 3). PCA-EV reads the same spectrum with the cumulative explained-variance rule [Jolliffe, 2002], the criterion behind layer saturation [Shenk et al., 2019, Richter et al., 2021]; SLIMLLM applies the layer-ratio rule of Guo et al. [2025b] ($\alpha = 8$), a redundancy signal based on the cosine similarity between each layer's input and output. PCA-EV is comparable to PCA on Qwen2.5-3B but falls below it on Gemma-3-4B, and SLIMLLM trails PCA on both teachers, most on MMLU. The two readouts differ most on layers dominated by a single direction (Figure 6), where the explained-variance rule assigns near-minimal width; this has no measurable cost on Qwen2.5-3B (one early layer) but hurts on Gemma-3-4B (two mid-depth layers). The cosine readout keeps width on such layers and appears more robust across the two teachers.

The rules above all read a profile from the model being pruned; a further question is whether any sufficiently middle-heavy profile would do. Transferring the more strongly middle-heavy profile measured on Qwen3-8B onto LLaMA-3-8B at the same budget gives 60.53 at step 3K, below UNIFORM (60.85) and level with INVERTED (60.48), whereas LLaMA-3-8B's own PCA profile reaches 61.72 (single runs; Appendix G). This supports using the model's own measured profile rather than a generic middle-heavy allocation.

4.4 σ_{99} tracks the size of the gain

The 3-way ordering holds on each of the four teachers, but its magnitude varies, and the variation tracks σ_{99} (Table 1). On the two high-σ_{99} teachers (Qwen2.5-3B, 0.188; Qwen3-8B, 0.192), Δ(PCA $-$ UNIFORM) is +2.50 and +2.28 pp and Δ(PCA $-$ INVERTED) is +5.52 and +4.63 pp; on the lowest-σ_{99} teacher, LLaMA-3-8B (0.105), the gaps shrink to $+0.52 \pm 0.26$ and $+0.99 \pm 0.46$ pp, still positive in every seed. The gaps differ far more than σ_{99} does: when the profile is more dispersed, following it helps more and inverting it costs more.

Table 4: **Qwen3-8B step-3K students after SFT (`tulu-3-100k`, 2 epochs, best-of-epoch).** PCA > UNIFORM > INVERTED on every task.

Config	ARC-C	MMLU	GSM8K
PCA	**85.58**	**67.75**	**73.46**
UNIFORM	74.57	62.83	70.43
INVERTED	73.81	57.41	62.55
Δ(PCA − UNIFORM)	+11.01	+4.92	+3.03
Δ(PCA − INVERTED)	+11.77	+10.34	+10.91

Scale alone does not account for the gap. LLaMA-3-8B matches Qwen3-8B in size but shows a much smaller gap. The 1B-class teachers (LLaMA-3.2-1B, Qwen2.5-1.5B, Gemma-3-1B) also give $\Delta(\text{PCA} - \text{UNIFORM}) < 0.6$ pp (Appendix D), but they likewise have low σ_{99}, so size and dispersion cannot be separated at that scale. Every small-gap teacher, including Gemma-3-4B, has a lower σ_{99} than either large-gap teacher. Four teachers show a trend rather than a functional form, and σ_{99} is confounded with family and recipe (§5).

4.5 The ordering holds without distillation

To test whether the ordering depends on the teacher, we train the Minitron-pruned Qwen2.5-3B students with the language-modeling cross-entropy in place of the KD loss, keeping the data, the 3K steps, the three allocations and the Qwen2.5-3B KD learning-rate recipe (§4.1). The ordering is the same as under KD: PCA > UNIFORM > INVERTED holds at every checkpoint from 1K to 3K, with gaps of $+2.31$ and $+5.63$ pp at step 3K (56.69, 54.37 and 51.06; gaps from unrounded values). The advantage of the measured allocation is therefore not specific to teacher supervision.

4.6 The advantage persists through SFT

To test whether the KD-time advantage survives a downstream instruction-tuning stage, we applied identical SFT to the three Qwen3-8B step-3K students (Table 4). SFT setup: tulu-3-100k (a 100K-example stratified subset of `tulu-3` [Lambert et al., 2025] preserving the original category proportions), 2 epochs, batch 128, learning rate 2×10^{-5}. We evaluate on `arc_challenge_chat` [Clark et al., 2018], `mmlu_redux_generative` [Gema et al., 2025], and GSM8K [Cobbe et al., 2021] from lm-eval-harness, zero-shot with the chat template. After SFT, PCA leads UNIFORM by $+11.0$, $+4.9$ and $+3.0$ pp on ARC-C, MMLU and GSM8K and leads INVERTED by $+10.3$ to $+11.8$ pp; the PCA > UNIFORM > INVERTED ordering also holds at each individual epoch.

The width-allocation choice influences a property that survives a 2-epoch SFT round on a fresh dataset, so it is not a transient KD-time artifact. This is a deployment-relevant signal: in production pipelines that combine KD-pruning with downstream SFT, the choice of per-layer width carries through to user-facing benchmark performance.

5 Discussion and Limitations

What drives the family-level σ_{99} signatures? σ_{99} shows a substantial association with model family and spans more than $2\times$ across families (Figure 2: 0.18–0.21 for Qwen2.5 3B–32B and Qwen3 8B–32B vs. LLaMA 0.08–0.11). Identifying the responsible recipe ingredient would require iso-architecture iso-data ablations that public checkpoints cannot provide, and the same limitation restricts the σ_{99}-vs-gap pattern to the four teachers tested ($\sigma_{99} \in [0.105, 0.192]$); iso-family iso-scale variants across a wider σ_{99} range would help disentangle recipe from σ_{99}.

Limitations. Two limitations concern scale and deployment. Our runs cover 1.57–4.2B training tokens; whether the advantage persists at Minitron-scale budgets (70–100B) is untested. A non-uniform student also carries a serving cost: in our vLLM [Kwon et al., 2023] integration the trained PCA student is 5.3% slower than UNIFORM at a 0.4% parameter difference (Appendix F).

Toward d_θ-guided width re-allocation beyond KD. Two natural extensions are re-shaping the same pretrained model under a preserved budget with continued pretraining, and selective capacity augmentation at high-d_θ layers of an already-trained model, motivated by the observation that σ_{99} changes little after pretraining (§3.4). Whether the advantage persists under either scheme is the top future ablation. A layer-output perturbation or pruning-sensitivity study would further test whether high-d_θ layers are also more capacity-sensitive.

6 Related Work

Functional heterogeneity of FFN layers. FFN layers act as key-value memories [Geva et al., 2021] whose updates promote vocabulary concepts [Geva et al., 2022]; causal tracing localizes factual associations to middle multi-layer perceptrons (MLPs) [Meng et al., 2022], and Dai et al. [2022] identify knowledge neurons in BERT [Devlin et al., 2019]. These studies suggest that FFN layers at different depths perform qualitatively different functions; our d_θ provides a complementary, quantitative signal of how many activation directions each layer uses.

Width-aware compression and pruning. WANDA [Sun et al., 2024] and SparseGPT [Frantar and Alistarh, 2023] induce unstructured sparsity, and LLM-Pruner [Ma et al., 2023], Sheared LLaMA [Xia et al., 2024], ShortGPT [Men et al., 2025], SliceGPT [Ashkboos et al., 2024] and Olica [He and Lin, 2025] remove neuron groups, heads, layers or dimensions; none of them allocates per-layer FFN width from a measured intrinsic-dimensionality profile under a fixed budget. SlimLLM [Guo et al., 2025b] does allocate a per-layer ratio, and we compare against its rule in §4.3. W-PCA [Wang, 2025] counts PCA components of FFN states in untrained BERT-scale networks as a zero-shot neural architecture search (NAS) proxy; we measure trained LLMs and use the profile directly as the allocation. Minitron [Muralidharan et al., 2024] couples width pruning with distillation under a uniform ratio; we adopt its recipe and replace the uniform allocation with d_θ. Closest in motivation, Ikeda et al. [2025] report a binary middle-heavy FFN allocation trained from scratch on a single architecture family at 285M–1.2B parameters; we differ in (i) producing a non-binary d_θ per layer and (ii) validating with a controlled KD experiment that isolates width allocation from training procedure.

Effective rank and intrinsic dimensionality. Effective rank [Roy and Vetterli, 2007] and nonlinear intrinsic-dimension estimators [Ansuini et al., 2019] characterize layer representations; Aghajanyan et al. [2021] instead use a parameter-space intrinsic dimension to explain fine-tuning. Ansuini et al. [2019] report a "hunchback" profile in convolutional neural networks (CNNs), in which intrinsic dimensionality first increases and then progressively decreases toward the output, and note that linear estimates such as PCA do not recover it; our PCA-based measure recovers an analogous middle-heavy profile in LLM FFNs. Shenk et al. [2019] and Richter et al. [2021] define layer "saturation" as the share of dimensions needed to reach a fixed cumulative explained variance and use it to diagnose over-provisioned layers during training. Our d_θ is in this family but specialized to FFN down-projection input activations, and uses a cosine-similarity threshold rather than explained variance, motivated by the direction-additive way in which FFN outputs enter the residual stream; it separates from saturation on layers dominated by a single direction, where the cosine readout retains substantially more width (§4.3, Appendix E).

7 Conclusion

We define d_θ via cosine-thresholded PCA on the FFN down-projection input and document a recurring middle-heavy, calibration-robust, post-training-stable profile across 58 LLMs spanning 14 families from 70M to 72B parameters. The single-number dispersion σ_{99} derived from the profile is a model-internal property that survives the shift from English to Chinese calibration and changes little across post-training fine-tuning. We validate the signal in a 3-way KD experiment in which budget-matched PCA-guided allocation beats UNIFORM on four teachers and a budget-matched INVERTED control does worst, with the magnitude of the advantage tracking σ_{99} across teachers and persisting through downstream supervised fine-tuning.

Acknowledgments

We thank Geewook Kim for helpful discussions during the review process and for proofreading the manuscript.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 7319–7328, 2021.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. SmolLM2: When Smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*, 2024.

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

BigScience Workshop. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 7432–7439, 2020.

Cerebras Systems. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. `https://huggingface.co/datasets/cerebras-archive/SlimPajama-627B`, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8493–8502, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning (ICML)*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024. URL `https://zenodo.org/records/12608602`.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 5069–5096, 2025.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5484–5495, 2021.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 30–45, 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025a.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. SlimLLM: Accurate structured pruning for large language models. In *International Conference on Machine Learning (ICML)*, 2025b.

Jiujun He and Huazhen Lin. Olica: Efficient structured pruning of large language models without retraining. In *International Conference on Machine Learning (ICML)*, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 30016–30030, 2022.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Wataru Ikeda, Kazuki Yano, Ryosuke Takahashi, Jaesung Lee, Keigo Shibata, and Jun Suzuki. Layerwise importance analysis of feed-forward networks in transformer-based language models. In *Conference on Language Modeling (COLM)*, 2025. URL `https://openreview.net/forum?id=oP3b5YBFoP`.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2nd edition, 2002.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, et al. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling (COLM)*, 2025.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. LLM360: Towards fully transparent open-source LLMs. In *Conference on Language Modeling (COLM)*, 2024. URL `https://openreview.net/forum?id=QdWhj0QZFw`.

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

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 17359–17372, 2022.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Mistral AI. Mistral NeMo. `https://mistral.ai/news/mistral-nemo`, 2024.

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 (NeurIPS)*, volume 37, pages 41076–41102, 2024.

OLMo Team. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Mats L. Richter, Justin Shenk, Wolf Byttner, Anders Arpteg, and Mikael Huss. Feature space saturation during training. In *British Machine Vision Conference (BMVC)*, 2021.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference (EUSIPCO)*, pages 606–610, 2007.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Noam Shazeer. GLU variants improve Transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Justin Shenk, Mats L. Richter, Anders Arpteg, and Mikael Huss. Spectral analysis of latent representations. *arXiv preprint arXiv:1907.08589*, 2019.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Shang Wang. W-PCA based gradient-free proxy for efficient search of lightweight language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*, 2023.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations (ICLR)*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4791–4800, 2019.

Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, et al. MAP-Neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv:2405.19327*, 2024.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12):3848–3860, 2023.

A θ Sweep

Higher-compression $\theta = 0.97$ ablation on Qwen2.5-3B. The body experiments use the cosine threshold $\theta = 0.99$, which on Qwen2.5-3B yields a 61.3% FFN-parameter budget ($\sum_l d_{\theta,l} = 242{,}816$ relative to the teacher's $L \cdot D = 396{,}288$). To test whether the headline ordering depends on the size of that budget, we re-run the 3-way KD recipe at $\theta = 0.97$, which produces a tighter PCA allocation summing to $169{,}984$ FFN-intermediate units (42.9% of the teacher budget, $1.43\times$ smaller than at $\theta = 0.99$). The UNIFORM and INVERTED controls are budget-matched to this $\theta = 0.97$ PCA sum so the comparison isolates the shape of the allocation while holding total FFN parameters fixed. All other hyperparameters (teacher, optimizer, KD loss, batch size, $3{,}000$-step horizon, evaluation suite) are identical to the body $\theta = 0.99$ runs, including the learning-rate recipe (§4.1).

Table 5 reports the 5-task average at every evaluated KD step. The PCA > UNIFORM > INVERTED ordering holds at every post-init step, as for the $\theta = 0.99$ runs in Table 7. At step 3K the gap to UNIFORM does not shrink with the smaller budget (Δ(PCA − UNIFORM) $= +2.86$ pp at $\theta = 0.97$ against $+2.50$ pp at $\theta = 0.99$, Table 1), while the gap to INVERTED narrows ($+4.46$ pp against $+5.52$ pp). The $\theta = 0.97$ sweep is a single training run per allocation; the matching single $\theta = 0.99$ runs give $+2.44$ and $+5.47$ pp, so the comparison is the same against them.

Table 5: **Qwen2.5-3B 3-way KD at $\theta = 0.97$ (5-task average).** Bold = best per step; task scores follow the body's protocol (§4.1). The PCA allocation sums to $169{,}984$ FFN units (42.9% of the teacher's $L \cdot D = 396{,}288$), a $1.43\times$ smaller budget than the body's $\theta = 0.99$ runs (61.3% budget). UNIFORM and INVERTED are budget-matched to the same $\theta = 0.97$ sum. The PCA > UNIFORM > INVERTED ordering holds at every post-init step, matching the $\theta = 0.99$ post-init ordering on the same teacher in Table 7.

Config	step 1K	step 2K	step 3K
PCA	**51.22**	**52.71**	**53.42**
UNIFORM	48.73	50.23	50.56
INVERTED	46.81	48.32	48.96
Δ(PCA − UNIFORM)	+2.49	+2.48	+2.86
Δ(PCA − INVERTED)	+4.41	+4.39	+4.46

B 58-Model PCA Scan Summary

The full 58-model scan covers 14 model families ranging from 70M (Pythia-70m) to 72B (Qwen2.5-72B) parameters. For each model we extract parameter count, hidden dimension, FFN intermediate width D, number of transformer layers L, the dispersion descriptor σ_{99} (standard deviation of $d_{\theta,l}/D_l$ at $\theta = 0.99$), the saturation ratio (mean $d_{\theta,l}/D_l$), the prunable fraction (1 minus the saturation ratio), and the mean $d_{\theta,l}/D_l$ of the first, middle and last thirds of layers. The summary statistics in the body (§3) are computed over this set, and the per-family σ_{99} distribution and per-layer profiles are visualized in Figure 2 (§3.2) and Figure 5 below (profiles of all 58 models are in the supplementary material); pretraining trajectories are visualized in Figure 3 (§3.3). Sixteen of the 58 models (BLOOM-560m/1b1/1b7/3b/7b1, Pythia-70m/160m/410m/1b/1.4b/2.8b/12b, MAP-Neo-7B (end of its decay phase) and MAP-Neo-SL-250M/460M/980M) are the final checkpoints of the pretraining trajectories in §3.3, measured with the same protocol as the other models.

C Per-Step KD Results Across Four Teachers

All KD runs use AdamW [Loshchilov and Hutter, 2019] ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1) with gradient clipping at 1.0 and peak LR $1 \cdot 10^{-4}$. Qwen2.5-3B and Gemma-3-4B use warmup 300 and cosine decay to $1 \cdot 10^{-5}$ over an 8K-step schedule, LLaMA-3-8B warmup 500 and cosine decay to $1 \cdot 10^{-6}$ over 8K steps, and Qwen3-8B warmup 300 and a constant learning rate thereafter; results are reported at step 3K unless noted.

Table 7 reports the 5-task average at init and at KD steps 3K, 5K and 8K for each of the four teachers and three width allocations. Two patterns are visible: (i) in the reported values the PCA

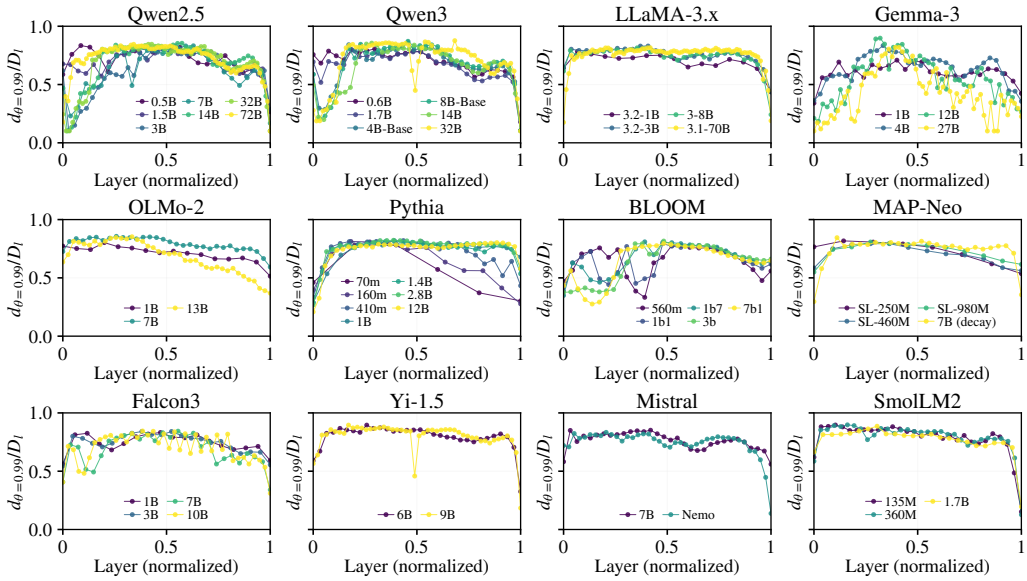


Figure 5: **Per-layer $d_{\theta,l}/D_l$ profiles for 51 of the 58 models in the scan, grouped by family.** Profiles of all 58 models are in the supplementary material. Horizontal axis is normalized layer index; vertical axis is $d_{\theta,l}/D_l$ at $\theta = 0.99$. The middle-heavy profile recurs across architecture, training corpus, and scale. How strongly middle-heavy it is varies by family (Qwen2.5/Qwen3 most pronounced, LLaMA-3.x, Mistral, and Yi-1.5 [Young et al., 2024] flatter); checkpoints that are visibly not middle-heavy are concentrated in the $\leq$ 1.7B-parameter regime and the low-σ_{99} families flagged in the body.

Table 6: **Post-training σ_{99} stability.** Across general and domain-specific post-training, domain-specific continued pretraining (CPT), and mid-training annealing, σ_{99} changes by at most 10% relative to its base model; Δ is computed from unrounded values. Δ is relative to the row's base: Qwen2.5-7B for Qwen2.5-7B-Instruct and the two CPT rows, and Qwen2.5-Coder-7B / Qwen2.5-Math-7B for their -Instruct rows. This contrasts with the -53% post-peak decline of OLMo-2-7B during Stage 1 pretraining (Figure 3).

Model	Stage	σ_{99}	Δ vs. base
Qwen2.5-7B	pretrain only	0.178	—
Qwen2.5-7B-Instruct	+ post-training	0.176	-1.5%
Qwen2.5-Coder-7B	+ code CPT	0.178	-0.1%
Qwen2.5-Coder-7B-Instruct	+ code CPT + post-training	0.161	-9.4%
Qwen2.5-Math-7B	+ math CPT	0.187	$+4.8\%$
Qwen2.5-Math-7B-Instruct	+ math CPT + post-training	0.185	-1.0%
Qwen3-8B (base)	pretrain only	0.192	—
Qwen3-8B (post)	+ post-training	0.183	-4.5%
OLMo-2-7B Stage 1	pretrain (final)	0.059	—
OLMo-2-7B main	+ 50B Stage 2 anneal	0.059	-0.6%

$>$ UNIFORM $>$ INVERTED ordering holds at every post-init step on every teacher, while at step 0 (before any KD training) the cold-start best is UNIFORM on Qwen3-8B and LLaMA-3-8B and PCA on Gemma-3-4B and Qwen2.5-3B; (ii) at every step the gaps are larger on the high-σ_{99} teachers than on LLaMA-3-8B; on Qwen3-8B Δ(PCA $-$ UNIFORM) narrows slightly from $+3.04$ pp at step 1K (§4.2; from unrounded values) to $+2.28$ pp at step 3K, while on Qwen2.5-3B Δ(PCA $-$ UNIFORM) and Δ(PCA $-$ INVERTED) are $+2.50$ and $+5.52$ pp at step 3K and, in the single longer run, $+2.28$ and $+5.18$ pp at step 5K and $+2.16$ and $+5.03$ pp at step 8K.

Tables 8 and 9 report the per-task breakdown for Qwen3-8B and LLaMA-3-8B at step 3K.

Table 7: **5-task average at init and at KD steps 3K, 5K and 8K for all four teachers $\times$ three width allocations.** Init values are the post-pruning, pre-KD checkpoints, shared across runs. Step-3K entries for Qwen2.5-3B and LLaMA-3-8B are means over three training seeds. At every post-init step shown the ordering is PCA $>$ UNIFORM $>$ INVERTED on every teacher.

Teacher (σ_{99})	Config	init	step 3K	step 5K	step 8K
Qwen2.5-3B (0.188)	PCA	**44.06**	**60.10**	**60.34**	**60.22**
	UNIFORM	43.02	57.59	58.06	58.06
	INVERTED	40.97	54.58	55.16	55.19
Qwen3-8B (0.192)	PCA	49.92	**67.48**	—	—
	UNIFORM	**51.47**	65.20	—	—
	INVERTED	47.84	62.85	—	—
Gemma-3-4B (0.124)	PCA	**39.41**	**54.24**	**54.82**	**55.26**
	UNIFORM	39.21	53.36	53.63	54.23
	INVERTED	38.91	52.19	52.92	53.17
LLaMA-3-8B (0.105)	PCA	55.10	**61.83**	**63.49**	**64.22**
	UNIFORM	**56.86**	61.31	62.71	63.42
	INVERTED	55.82	60.84	62.40	62.54

Table 8: **Per-task results for Qwen3-8B at step 3K.** PCA achieves the highest score on four of five tasks; UNIFORM narrowly leads on PIQA.

Task	PCA	UNIFORM	INVERTED	Δ(PCA $-$ UNIFORM)	Δ(PCA $-$ INVERTED)
MMLU	**65.69**	60.94	56.79	$+4.75$	$+8.90$
ARC-C	**52.73**	49.06	46.59	$+3.67$	$+6.14$
HellaSwag	**72.33**	71.23	69.40	$+1.10$	$+2.93$
WinoGrande	**69.85**	67.25	65.51	$+2.60$	$+4.34$
PIQA	76.82	**77.53**	75.95	-0.71	$+0.87$
Avg	**67.48**	65.20	62.85	$+2.28$	$+4.63$

D 1B-Class Scale Ablation

We extend the 3-way KD validation of §4 to three 1B-class teachers: Qwen2.5-1.5B [Qwen Team, 2024], LLaMA-3.2-1B [Grattafiori et al., 2024], and Gemma-3-1B [Gemma Team, 2025]. All three are low-σ_{99} teachers ($\sigma_{99} \leq 0.11$); Qwen2.5-1.5B sits at the bottom of the Qwen family's σ_{99} distribution despite the family's overall high-σ_{99} tendency. For each teacher we distill three students with the body's KD setup at $\theta = 0.99$ but a different learning-rate recipe: budget-matched PCA/UNIFORM/INVERTED allocations summing to $\sum_l d_{\theta,l}$, Minitron-style importance pruning at init, pure KL distillation on SlimPajama at sequence length $2{,}048$, effective batch 256 sequences, peak LR $3 \cdot 10^{-4}$ with warmup 300 and a constant learning rate thereafter, 3K steps. We report the 5-task average (MMLU, ARC-C, HellaSwag, WinoGrande, PIQA) at step 3K with the same headline-number convention as the body.

Table 10 reports the results. PCA scores highest on all three teachers, but every gap is under $+0.6$ pp, and on Qwen2.5-1.5B PCA leads INVERTED by only 0.04 pp while INVERTED edges UNIFORM. We do not read these as positive effects.

E Explained-Variance versus Cosine Readout

Construction. Both readouts use the same centered PCA of h_l (§2). The explained-variance allocation follows the cumulative explained-variance rule [Jolliffe, 2002]: it sets w_l to the smallest rank at which the cumulative explained variance of layer l reaches τ, with τ chosen so that $\sum_l w_l$ equals the PCA budget B ($\tau^\star = 0.983$ on Qwen2.5-3B and 0.984 on Gemma-3-4B), rounded to multiples of 128 with a floor of 128. At the nominal $\tau = 0.99$ the explained-variance rule asks for 11–12% more total width than the cosine rule on Qwen2.5-3B and Gemma-3-4B.

Table 9: **Per-task results for LLaMA-3-8B at step 3K**, means over three training seeds. PCA achieves the highest score on four of five tasks; UNIFORM narrowly leads on PIQA. The per-task gaps are smaller on average than on the high-σ_{99} teachers, and Δ(PCA − INVERTED) is smaller on every task, consistent with the σ_{99} trend in §4.4.

Task	PCA	UNIFORM	INVERTED	Δ(PCA − UNIFORM)	Δ(PCA − INVERTED)
MMLU	**44.11**	43.15	41.62	+0.96	+2.49
ARC-C	**43.63**	43.46	42.75	+0.17	+0.88
HellaSwag	**74.16**	73.58	73.50	+0.58	+0.66
WinoGrande	**69.01**	67.90	68.22	+1.11	+0.79
PIQA	78.22	**78.45**	78.11	−0.23	+0.11
Avg	**61.83**	61.31	60.84	+0.52	+0.99

Table 10: **1B-class 3-way KD results at step 3K.** 5-task averages (MMLU, ARC-C, HellaSwag, WinoGrande, PIQA). PCA scores highest on every teacher, but every gap is under +0.6 pp and INVERTED edges UNIFORM on Qwen2.5-1.5B; we do not observe a meaningful effect at 1B scale.

Teacher (σ_{99})	PCA avg	UNIFORM avg	INVERTED avg	Δ vs. UNIFORM	Δ vs. INVERTED
Qwen2.5-1.5B (0.105)	**48.25**	48.12	48.21	+0.13	+0.04
LLaMA-3.2-1B (0.079)	**43.10**	42.88	42.72	+0.22	+0.38
Gemma-3-1B (0.079)	**43.50**	43.06	42.97	+0.44	+0.53

Where the readouts differ. On Qwen2.5-3B and Gemma-3-4B the largest relative differences between the two readouts fall on the layers whose covariance is dominated by a single direction (one on Qwen2.5-3B, two on Gemma-3-4B; Figure 6), where the explained-variance rule cuts width to 2–27% of the cosine width; elsewhere the two allocations differ by smaller factors (0.44–1.37×), for example on layers 30 and 35 of Qwen2.5-3B and layers 31–33 of Gemma-3-4B. On a dominant-direction layer the explained-variance rule saturates early, while reconstructing token directions to cosine 0.99 still requires the remaining components; on Qwen2.5-3B layer 2, for instance, the explained-variance rule is satisfied at one dimension while the cosine at 10% of the width is only 0.73. The location of these layers differs between the two teachers: early on Qwen2.5-3B and at mid-depth on Gemma-3-4B.

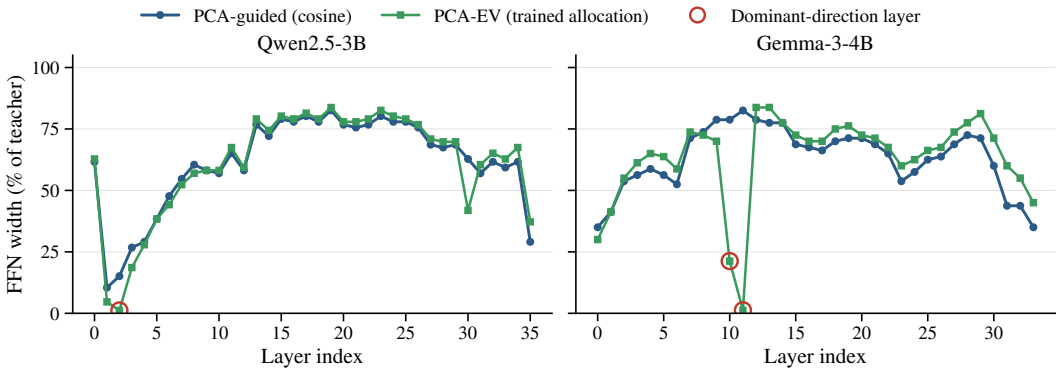


Figure 6: **Per-layer widths of the cosine (PCA) and explained-variance (PCA-EV) readouts on Qwen2.5-3B and Gemma-3-4B.** Both panels show the trained allocations, budget-matched on each teacher. Open red circles mark the layers whose covariance is dominated by a single direction (top-1 eigenvalue share above 0.9): layer 2 of Qwen2.5-3B and layers 10 and 11 of Gemma-3-4B. Per-layer correlation of the two width vectors: $r = 0.975$ (Qwen2.5-3B) and 0.31 (Gemma-3-4B).

Effect under KD. Under the recipes of the body, PCA-EV is comparable to PCA on Qwen2.5-3B but falls below it on Gemma-3-4B. On Qwen2.5-3B, PCA-EV reaches 60.18 ± 0.10 against 60.10 ± 0.22 for PCA at step 3K, a difference of means of -0.08 pp (§4.3). On Gemma-3-4B, PCA-EV reaches 52.60 at step 3K against 54.24 for PCA and 53.36 for UNIFORM, and stays below

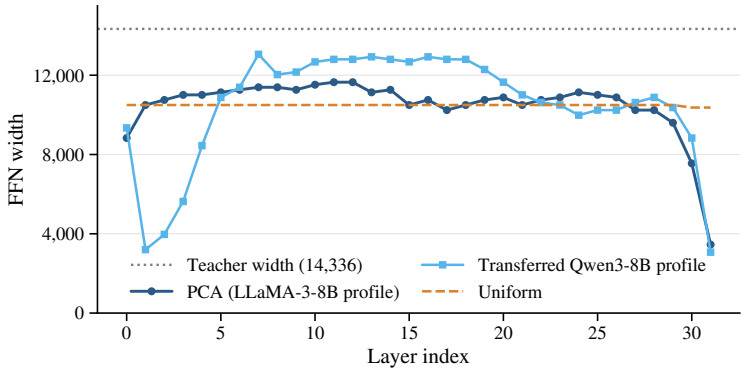


Figure 7: **Per-layer FFN widths of the LLaMA-3-8B students in the transfer control.** All three vectors sum to the same budget (335,616); the transferred Qwen3-8B profile cuts layers 1–3 to 3,200–5,632 units, where LLaMA-3-8B's own profile gives 10,496–11,008.

both at all eight checkpoints from 1K to 8K (mean -1.60 pp vs. PCA), with the largest loss on MMLU (-3.83 pp at 3K). Sharply reducing the width of a dominant-direction layer thus has no measurable cost when that layer is layer 2 of Qwen2.5-3B but hurts when the layers sit at mid-depth of Gemma-3-4B, so layer position appears to matter.

Correlation with the measured profile. The per-layer Pearson r between width vectors (Figure 6) is a coarse summary: it is dominated by the few layers whose widths differ most, and it is undefined for UNIFORM. On Qwen2.5-3B, SLIMLLM ($r = -0.37$ with PCA) scores below both readouts, but on Gemma-3-4B it ($r = -0.61$) scores slightly above PCA-EV ($r = 0.31$), and both fall below UNIFORM; PCA-EV cuts the two dominant-direction layers at mid-depth (layers 10 and 11) to 2,176 and 128 of $D_l = 10{,}240$ units, against 8,064 and 8,448 for PCA. A single correlation coefficient does not capture such layer-specific cuts. The cosine readout, which keeps width on these layers, therefore appears to be the more robust allocation rule across teachers.

F Serving Cost of a Non-Uniform Student

A non-uniform width vector is only useful if it can be served. We implemented the adaptive-width student as a custom vLLM [Kwon et al., 2023] model and measured output throughput on the trained Qwen2.5-3B checkpoints, on one V100-SXM2-32GB with tensor parallelism 1 and CUDA-graph decode: teacher 499.16 tok/s, PCA student 592.83 (+18.8% over the teacher), UNIFORM student 626.30 (+25.5%). No custom kernel is required: the layers remain dense GEMMs that differ in one dimension. The quality gain therefore carries an explicit serving cost: PCA is 5.3% slower than UNIFORM at a 0.4% parameter difference (2.143B vs. 2.152B); we do not read this as a floor, since the integration performs no per-shape kernel tuning.

G Cross-Model Transfer Control

To separate the measured profile from middle-heaviness as such, we interpolated the profile measured on Qwen3-8B onto the 32 layers of LLaMA-3-8B and scaled it to the PCA budget of LLaMA-3-8B (335,616); the transferred vector is more strongly middle-heavy than LLaMA-3-8B's own profile and correlates with it at $r = 0.53$ (Figure 7). Trained with the LLaMA-3-8B recipe of §4.1 alongside the first of the three training seeds in Table 1, it reaches 60.53 at step 3K, against 61.72 for PCA, 60.85 for UNIFORM and 60.48 for INVERTED: a more middle-heavy allocation borrowed from another model lands below UNIFORM and level with the inverted control.

H Evaluation-Seed Robustness on LLaMA-3-8B at Step 8K

To verify that the 3-way KD ordering on LLaMA-3-8B is stable under evaluation-time stochasticity (e.g., few-shot sampling, batching, log-likelihood shuffling) at the longer 8,000-step horizon, we

Table 11: **LLaMA-3-8B 3-way KD at step 8K, multi-seed evaluation.** n denotes the number of independent evaluation seeds applied to the same single training-run weights for each allocation. Evaluation-seed standard deviation on the 5-task average is under 0.05 pp for every allocation, an order of magnitude below the inter-allocation gaps ($\Delta(\text{PCA} - \text{UNIFORM}) = +0.80$ pp, $\Delta(\text{PCA} - \text{INVERTED}) = +1.68$ pp). Per-task means and standard deviations are reported for MMLU, ARC-C and WinoGrande; HellaSwag and PIQA vary by at most 0.03 (std). This isolates evaluation-time variance only; training-seed variance at step 3K is reported in §4.4.

Config	5-task avg		MMLU	ARC-C	WinoGrande
	mean	std	mean ± std	mean ± std	mean ± std
PCA (n=3)	**64.22**	0.00	**50.24** ± 0.00	**45.82** ± 0.00	**70.24** ± 0.00
UNIFORM (n=3)	63.42	0.04	48.13 ± 0.04	44.97 ± 0.00	69.74 ± 0.15
INVERTED (n=3)	62.54	0.03	47.86 ± 0.12	43.43 ± 0.00	67.80 ± 0.00

re-evaluate the same step-8K PCA/UNIFORM/INVERTED weights under multiple independent lm-eval-harness seeds. Each allocation's weights come from a single KD training run; the multi-seed entries in Table 11 differ only in evaluation seed, not in training. Table 11 reports the resulting 5-task average mean and standard deviation. The PCA $>$ UNIFORM $>$ INVERTED ordering holds with evaluation-seed standard deviations under 0.05 pp on the 5-task average, an order of magnitude below the inter-allocation gaps. The longer $8{,}000$-step horizon leaves $\Delta(\text{PCA} - \text{UNIFORM})$ essentially unchanged relative to step 3K of the same run ($+0.87 \rightarrow +0.80$ pp) and widens $\Delta(\text{PCA} - \text{INVERTED})$ ($+1.24 \rightarrow +1.68$ pp), both remaining below the high-σ_{99} teachers' gaps.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and §1 state the three contributions (measurement protocol, 58-LLM characterization, 3-way KD validation), developed in §2, §3 and §4; the four-teacher scope of the σ_{99}-dependent KD finding is stated in §1 and §4.4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: §5 discusses the lack of iso-architecture iso-data ablations, which limits the σ_{99}-vs-gap pattern to four teachers ($\sigma_{99} \in [0.105, 0.192]$), the short KD horizon (1.57–4.2B tokens vs. Minitron's 70–100B), and the serving cost of a non-uniform student.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper is empirical; the only formal definition is the cosine-thresholded d_θ in Equation (2), which is a measurement protocol rather than a theorem.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: §2 specifies the PCA protocol (hook position, calibration size, threshold), §4.1 and Appendix C all KD hyperparameters, and §4.6 the SFT recipe and lm-eval-harness task names.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The PCA measurement code, width-allocation and KD training scripts with per-teacher recipes, and the 58-model scan results are at `https://github.com/naver-ai/lidf`; the scan results and pretraining-trajectory measurements are also in the supplementary material. All data (SlimPajama, the Tülu 3 SFT mixture; subset in §4.6) are public on Hugging Face.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: §4 setup paragraph specifies the KD objective, effective batch size, learning-rate schedule, training steps, token count, and per-task evaluation protocol; optimizer settings are in Appendix C; the SFT setup is in §4.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: On Qwen2.5-3B and LLaMA-3-8B, the step-3K KD results in Tables 1–3 are means over three training seeds, with standard deviations in Table 3 and §4.4; all other runs are single training runs. Evaluation-seed variance is reported in Appendix H; standard deviations use n in the denominator.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: PCA measurement takes about 10 minutes per 8B model on one A100. KD uses one 8×A100-80GB node with FSDP [Zhao et al., 2023] SHARD_GRAD_OP; one 3K-step student takes about 20h at 8B (Qwen3-8B, LLaMA-3-8B), 17h for Gemma-3-4B, 8h for Qwen2.5-3B and 4–8h for 1B-class teachers, and each SFT round 5–7h, totaling roughly 7,000 A100 GPU-hours; serving uses one V100-SXM2-32GB (Appendix F). The full project used more compute for preliminary and discarded runs, including Qwen2.5-3B KD runs re-run after a fix to the student attention implementation; all reported results use the fixed implementation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work analyzes and compresses publicly released LLMs using public datasets; no human subjects, sensitive data, or restricted resources are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Positive: cheaper LLM inference with less accuracy loss than uniform-width pruning at the same budget. Negative: we do not foresee impacts beyond those of the open-weight models being compressed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The released artifacts are the measurement and training code and the 58-model scan results; they contain no model weights and do not introduce new high-risk capabilities beyond the public open-weight models they analyze.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All measured or distilled models (Appendix B) are open-weight and used under their published licenses (Apache-2.0, MIT, the Llama 3 Community License, the Gemma Terms of Use, the Qwen, DeepSeek and Falcon model licenses, and the BigScience RAIL License); all datasets and lm-eval-harness are public and cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The released code and 58-model scan results include README documentation of the per-layer d_θ schema, calibration setup, and KD hyperparameters (§2, §4).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: No human-subjects research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are the subject of study (we measure and compress LLMs); they were not used as a non-standard component of the methodology itself, and editorial assistance does not require declaration under the policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.