GARAK-2: SELECTIVE LOW-RANK COMPRESSION OF A KOREAN SPEECH FOUNDATION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

I introduce Garak-2, an efficiency-oriented successor to Garak-1, a Korean speech recognition foundation model trained on multi-domain Korean speech. Garak-1 initializes its cache-aware FastConformer encoder from the March 2026 NVIDIA Nemotron Speech Streaming 0.6B checkpoint and replaces the original RNN-T decoder with a Token-and-Duration Transducer (TDT) decoder. Both the encoder and decoder are trained on Korean speech, with the decoder initialized from scratch. Starting from this Korean-trained Garak-1 checkpoint, Garak-2 targets the encoder, which dominates model parameters and CPU inference time. I apply rank-512 factorization to the feed-forward projection matrices in the middle 12 layers of the 24-layer encoder, preserving the original expansion width, nonlinearities, cache-aware attention and convolution modules, and TDT formulation. This reduces the parameter count from 622.0 million to 546.5 million, a 12.1% reduction (546.8 million with the small correction path of the evaluated model). To account for the effects of additional training, I compare the compressed model with an uncompressed Garak-1 baseline under matched continued-training budgets. Preliminary experiments across 19 Korean speech datasets show that a short period of continued training brings the compressed model's macro-averaged character error rate close to that of the equally trained baseline, with a small remaining deficit at matched training steps (+0.04 CER points) that varies with the checkpoint. The same analysis reveals dormant feed-forward modules in the middle layers: replacing 12 of them by weight-derived constants removes 16.2% of the parameters without any training, at a small accuracy cost (about 0.5% relative on a held-out set). CPU inference benchmarking at batch size eight with FP32 computation on eight cores shows a 16–18% reduction in total processing time. These initial findings suggest that selective encoder compression is a promising approach to reducing the size and inference cost of Garak-1 at a small cost in average recognition performance.

1 INTRODUCTION

On CPUs, the encoder of a streaming speech recogniser dominates both parameters and latency. Garak-1 (Anonymous, 2026) is a 622.0M-parameter Korean streaming recogniser that pairs a 24-layer cache-aware FastConformer encoder (Gulati et al., 2020; Rekesh et al., 2023; Noroozi et al., 2024) with a Token-and-Duration Transducer (TDT) decoder (Xu et al., 2023). Its encoder was initialised from the English Nemotron Speech Streaming 0.6B checkpoint (NVIDIA, 2026), and its feed-forward networks (FFNs) alone hold 402.7M parameters. I compress this encoder in two steps, a registered compression and a follow-up analysis of why the compression works.

Registered compression. Following a pre-registered plan, I factorise the 48 FFN matrices of the middle 12 layers at rank 512 with activation-aware SVD (Wang et al., 2025c). This removes 12.1% of the parameters. After continued training, at the registered common step (35K; an early, reduced-scope stop of the registered 200K-step plan), the compressed model G is slightly worse than an uncompressed control N trained with the same plan and data (pooled CER $+0.041$, 95% interval $[+0.013, +0.069]$, 0.9% relative). 520 steps later the difference is $+0.008$ $[-0.019, +0.036]$ (19-domain macro CER 4.734 vs. 4.717), and G's own 35K and 35.5K checkpoints differ by 0.033, so the size of the gap depends on the checkpoint. G reduces CPU processing time by 16–18% at batch size 8 (two runs with the same benchmark settings: -18.1% and -16.1%).

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

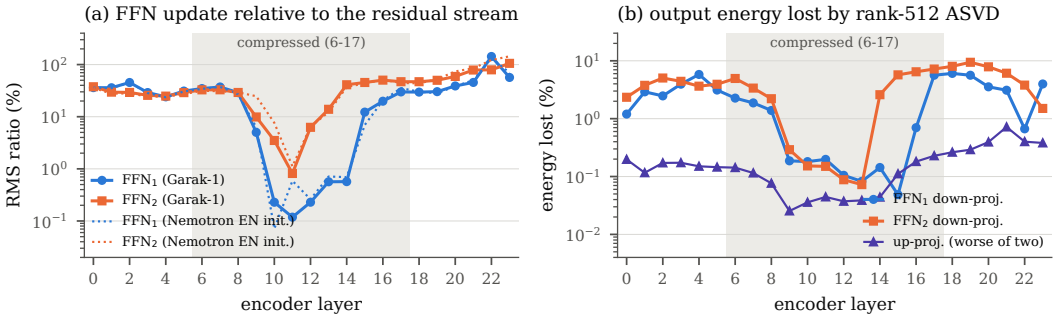


Figure 1: **The middle-layer FFNs of the encoder are dormant and highly compressible.** (a) RMS of each macaron FFN update ($0.5\,\mathrm{FFN}(x)$) relative to the RMS of the residual stream entering the layer (log scale), for Garak-1 (solid) and its English Nemotron initialisation (dotted), both measured on the same 120 calibration utterances (115 Korean, 5 English). (b) Share of activation-weighted output energy lost when each FFN matrix is truncated to rank 512 by activation-aware SVD. The loss is U-shaped over depth: at most 0.3% for every matrix in layers 9–13, but up to 5.8% (early layers) and 9.4% (late layers) for down-projections. Shading: layers 6–17, factorised in the registered model G.

Why the middle compresses. The macaron FFNs in the middle of the encoder are *dormant* (Figure 1); attention and convolution there remain active. In layers 10–14 the first macaron FFN adds only 0.1–0.6% of the residual-stream RMS, and all of its 4,096 hidden units sit deep in the negative saturation of Swish. I trace this to the *bias of the pre-FFN LayerNorm*. Scaling that bias down wakes the FFNs, and scaling it up in active layers largely switches them off (units off 63%→96%, contribution 0.30→0.13 times the residual RMS), so the bias causally controls the switch. The switch is inherited from English pretraining, and it can be read off the weights alone. A data-free detector finds dormant modules in the Nemotron streaming lineage (11–12 modules, about 15% of parameters), finds one module in one other public checkpoint, and finds none in 11 more.

The rank-0 limit. Because the outputs of dormant modules are dominated by a constant component, a rank-0 (constant) approximation is effective for them. The constant each module should output can itself be computed from the weights, so compression needs neither audio nor training. Replacing the 12 detected modules (16.2% of the parameters) by such constants leaves test CER indistinguishable from Garak-1 ($+0.002$; 95% interval $[-0.015, +0.019]$), stays within a pre-registered 1% margin on 40,000 held-out utterances at a small but significant 0.5% cost, and removing these modules, timed as a proxy, cuts CPU time by 13% (one run); at 0 ms lookahead a small accuracy loss is detected. Removing the dormant modules from public English checkpoints leaves LibriSpeech WER essentially unchanged.

Contributions. (1) A same-plan evaluation of registered rank-512 compression, with a same-step check (Section 2). (2) A mechanistic account of dormant macaron FFNs: a LayerNorm-bias switch, a causal dose–response, and its inheritance from pretraining (Section 3). (3) A data-free detector and a survey of 15 checkpoints (Section 4). (4) Data-free rank-0 compression by removal or weights-only constants, with a pre-registered held-out check, a latency analysis and public English models (Section 5). (5) Evidence that *where* to compress matters more than how (Sections 5–7, Appendix B).

2 Registered compression vs. an uncompressed control

Model. Each encoder layer of Garak-1 is a macaron Conformer block with model width 1024:

$$x_1 = x + \tfrac{1}{2}\mathrm{FFN}_1(x),\quad x_2 = x_1 + \mathrm{MHSA}(x_1),\quad x_3 = x_2 + \mathrm{Conv}(x_2),\quad y = \mathrm{LN}\big(x_3 + \tfrac{1}{2}\mathrm{FFN}_2(x_3)\big), \tag{1}$$

with $\mathrm{FFN}_j(x) = W_2^{(j)}\,\mathrm{Swish}\big(W_1^{(j)}\,\mathrm{LN}_j(x)\big)$, $W_1 \in \mathbb{R}^{4096\times1024}$, $W_2 \in \mathbb{R}^{1024\times4096}$, no biases, and $\mathrm{LN}_j(x) = \gamma_j \odot \hat{x} + \beta_j$ with $\hat{x}$ the standardised input. Garak-1 was trained for 380K steps on 36,289 h of Korean speech from 18 AI-Hub corpora, with LibriSpeech additionally sampled at 5%; details are in Appendix A.

Table 1: **Main comparison on the 19-domain test set** (CER %). Pooled-18 values and intervals come from the registered cluster bootstrap. G: 35,520 steps; N: 35,000 steps. The registered common-step comparison (both at 35K) is G−N $+0.041$ $[+0.013, +0.069]$ (Table 4).

Model	Params	Macro-19	Macro-18	Pooled-18 [95% int.]	463
Garak-1 (start, 380K)	622.0M	4.704	4.483	4.637	8.684
N: uncompressed, continued	622.0M	4.717	4.499	4.651	8.630
G: rank 512, L6–17, continued	546.8M	4.734	4.510	4.659	8.775
G − N (35.5K vs. 35K, unequal steps)		$+0.018$	$+0.011$	$+0.008\,[-0.019, +0.036]$	$+0.145$
G − Garak-1		$+0.030$	$+0.027$	$+0.022\,[-0.004, +0.049]$	$+0.091$

Activation-aware SVD. For $W \in \mathbb{R}^{m\times n}$ with input second moment $C = \mathbb{E}[xx^\top]$, estimated on 1,000 calibration utterances (3–15 s, sampled in proportion to the training mixture, disjoint from all evaluation data) and regularised by a small eigenvalue floor, let $WC^{1/2} = U\Sigma V^\top$. I set $W \approx BA$ with $B = U_R\Sigma_R^{1/2}$ and $A = \Sigma_R^{1/2}V_R^\top C^{-1/2}$. This choice minimises $\mathbb{E}\|Wx - BAx\|^2$ under the floored second moment, for which it equals $\sum_{i>R}\sigma_i^2$ (Wang et al., 2025c). The retained output energy is $E_R = \sum_{i\le R}\sigma_i^2 / \sum_i \sigma_i^2$.

G. As pre-registered, G factorises all 48 FFN matrices of layers 6–17 at $R = 512$, with nothing inserted between A and B. Each such matrix keeps 62.5% of its parameters and multiply–accumulates. The model shrinks from 622,025,866 to 546,528,394 parameters (-12.1%), and the mean activation-weighted relative error $\sqrt{1 - E_{512}}$ is 0.068 (range 0.014–0.269). The registered configuration also adds a zero-initialised, 0.29M-parameter multiplicative correction path to the joint network (546,823,306 parameters in total, still -12.1%). I analyse this path in Appendix D; about 83% of its development-set benefit is a constant logit offset.

Control and training. N is Garak-1 without compression, trained with the same continued-training plan as G. Both start from the 380K checkpoint and use the same data stream in nearly identical order (the two runs resumed from two false data-mismatch stops with a cursor difference of at most 4 micro-batches). The optimiser is AdamW with global batch 128 (4 H100 GPUs, gradient accumulation 4), bf16 mixed precision, 2,000 warm-up steps, and peak learning rates 10^{-5} (encoder), $3\cdot10^{-5}$ (prediction and joint networks) and 10^{-4} (correction path). Weight decay is 10^{-3} and gradients are clipped at 1.0. I compare G at 35,520 steps with N at 35,000 steps; at the common 35K step G is worse (Table 4). The registration specified a common step after 200K steps, or a common early-stop step reported as reduced scope. Training was stopped early, so this is a reduced-scope comparison and G has 520 (1.5%) more steps than N.

Evaluation. The test set has 19 Korean domains from AI-Hub with at most 2,000 utterances each (37,645 utterances). Eighteen domains are training domains (35,637 scored utterances). Domain 463 (broadcast conversation) is excluded from training and reported separately (column "463"). I report CER (%) after one fixed normalisation (garak-norm v1.1, tier T1, identical for all systems), macro-averaged over domains and pooled over the 18 training domains. Pooled differences use the pre-registered cluster bootstrap (Bisani & Ney, 2004): 10,000 resamples of 33,503 source recordings with seed 20260924. Only G vs. N, G vs. Garak-1 and the held-out check of W_{12} (Section 5) are interpreted inferentially; only the first and the last were pre-registered; all other intervals are exploratory and uncorrected for multiplicity. A separate 8,500-utterance development set (Garak-1: 4.753) is used for zero-shot and ablation studies.

Results. Table 1 reports the main comparisons. G is the full registered system, factorisation plus the 0.29M correction path, so the G−N comparison does not isolate factorisation alone; switching the path off at inference in trained G gives 4.775. After continued training (G at 35.5K), G's 19-domain macro CER (4.734) is within 0.018 points of N (4.717). At the registered common step (35K), G is worse than N: $+0.041$ $[+0.013, +0.069]$ (Table 4). For G at 35.5K the pooled difference is $+0.008$ $[-0.019, +0.036]$, which contains 0 (no difference detected, not equivalence); a post-hoc ±0.10-point equivalence test would pass. G's 35K and 35.5K checkpoints differ by 0.033, so I report the gap as checkpoint-dependent. G also does not differ detectably from Garak-1 ($+0.022$ $[-0.004, +0.049]$). Continued training does not improve the uncompressed model: N moves from 4.637 to 4.651 pooled, and its FFN weight norms change by at most 0.025%. The held-out broadcast domain shows a larger gap (G−N $+0.145$). On CPU at batch size 8 with FP32

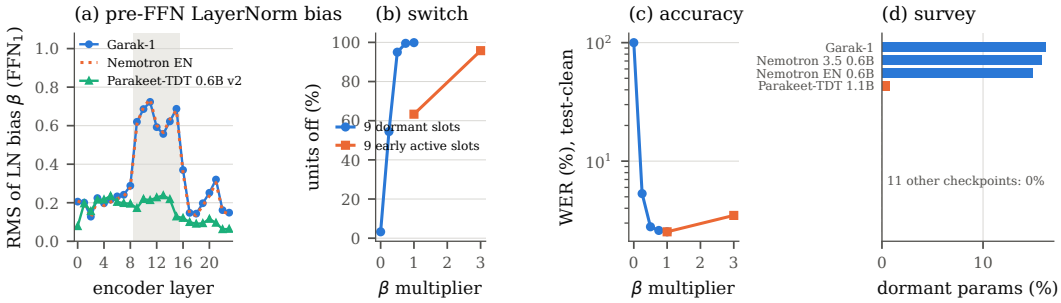


Figure 2: **A LayerNorm-bias switch.** (a) RMS of the pre-FFN$_1$ LayerNorm bias β. It is large in layers 9–15 (shaded) of Garak-1 and of its English initialisation, but not in Parakeet-TDT 0.6B v2. (b,c) Dose–response in Nemotron EN: scaling β of the nine dormant slots by 1, 0.75, 0.5, 0.25 and 0 (blue), or β of nine active early slots by 3 (orange). The measurements are the mean fraction of hidden units that are off (60 dev-clean utterances) and the WER on LibriSpeech test-clean. (d) Share of all model parameters that lie in FFN modules (with their LayerNorms) flagged dormant by the data-free detector, in 15 checkpoints (Table 2).

on eight cores, G reduces total processing time by 16–18% (two runs: -18.1% and -16.1%; Section 7); these are secondary measurements without intervals, and the pre-registered locked-engine speed test was not run.

3 Anatomy of the encoder: dormant macaron FFNs

Dormancy. I measure, per module, the RMS of the update $\frac{1}{2}\mathrm{FFN}_j(x)$ relative to the residual RMS, the mean pre-activation of each hidden unit, and activation-aware spectra (Appendix Table 9 lists the per-layer contributions, median pre-activations and off-unit shares; Figure 1b shows the spectra). FFN$_1$ in layers 10–14 contributes 0.1–0.6% of the residual RMS, and FFN$_2$ in layers 10–12 contributes 0.8–6.3%. Every module in layers 0–8 contributes 24–45%, and every module in layers 17–23 contributes 30–143%. In the dormant modules (FFN$_1$ 9–15, FFN$_2$ 9–13), 99.1–100% of hidden units have a mean pre-activation below -4, with a median of -14.2 to -17.5, where the Swish output is below 10^{-5} in magnitude; "off" refers to this mean over frames, not to every frame. Elsewhere the median is -2.5 to -7.5. The resulting activation-weighted spectra are extremely concentrated: every matrix in layers 9–13 keeps $\geq$99.7% of its output energy at rank 512 (Figure 1b). This concentration is why the registered middle-layer factorisation was cheap. Removing any single one of the nine modules that D removes (FFN$_1$ 9–14, FFN$_2$ 10–12) changes development CER by -0.006 to $+0.006$, whereas removing any one of the other 39 modules costs a median of $+0.094$ and up to $+0.480$ (layer-23 FFN$_1$). Across all 48 modules, the Spearman correlation between residual contribution and single-removal ΔCER is 0.888 (Appendix Figure 5). Skipping *whole* layers is harmful even in the dormant band (layer 11: $+0.123$; layers 9–13: $+0.794$), so attention and convolution there remain active; only the FFNs are asleep.

Mechanism: the pre-FFN LayerNorm bias. Why are the units off? The weights are not small: the Frobenius norms of the FFN$_1$ down-projections in layers 10–14 are 564–594, against 701–890 elsewhere. The inputs are the cause. The LayerNorm output is dominated by its time-constant component, which carries 97.8–98.8% of its energy in layers 9–15. This share is also high in most active layers (77–97% in layers 1–8 and 16–21), so its size alone does not decide dormancy; what it maps to does. In the dormant band, the LayerNorm bias is large: the RMS of β is 0.56–0.72 for FFN$_1$ in layers 9–15, against 0.09–0.32 in active modules (0.37 for the transitional FFN$_1$ in layer 16), while γ is unremarkable (0.17–0.28 for FFN$_1$ in layers 9–15, 0.16–0.34 elsewhere; Appendix Table 10); Figure 2a shows the β profile. The up-projection maps this near-constant input to large negative pre-activations for nearly all units. The residual stream in these layers also carries a massive-activation channel, channel 321, whose RMS is 31–42 times the median channel (Appendix Table 10).

A causal dose–response. I intervene on β alone in Nemotron EN, measuring off units and FFN contribution on 60 LibriSpeech dev-clean utterances and WER on test-clean (2,620 utterances; Figure 2b,c). Scaling β of the nine dormant slots by 1, 0.75, 0.5, 0.25 and 0 lowers the fraction of off

Table 2: **Data-free survey and zero-shot removal of the flagged modules.** Survey of 14 public NVIDIA NeMo checkpoints and Garak-1 (full list in Appendix Table 8). Removal WER is on LibriSpeech test-clean (2,620 utterances, greedy decoding); for Garak-1 it is Korean test CER (pooled-18).

Checkpoint	Streaming	Dormant	Params (share)	Zero-shot removal
Garak-1	cache-aware	12 / 48	100.7M (16.2%)	CER 4.637 → 4.664
Nemotron Streaming EN 0.6B	cache-aware	11 / 48	92.3M (14.9%)	WER 2.532 → 2.526
Nemotron 3.5 Streaming 0.6B	cache-aware	12 / 48	100.7M (15.8%)	WER 3.458 → 3.465[†]
Parakeet-TDT 1.1B	no	1 / 84	8.4M (0.8%)	WER 1.527 → 1.529
11 others[‡]	2 of 11	0	0	–

[†]Language tags appended to the output are removed before scoring. On Korean FLEURS test (382 utterances) the same removal changes CER from 9.727 to 9.614. [‡]Parakeet CTC/RNNT 0.6B, Parakeet-TDT 0.6B v2/v3, Canary 180M/1B Flash, and five 110–120M FastConformers, two of them cache-aware streaming.

units from 99.9% to 99.5%, 95.0%, 54.4% and 3.3%. Over the same scales, the FFN contribution rises from 0.021 to 0.044, 0.138, 0.477 and 4.307 times the residual RMS. WER degrades from 2.53 through 2.60, 2.79 and 5.32, and decoding breaks down completely at 0 (100%). Conversely, tripling β in nine *active* early slots pushes them toward dormancy: off units rise from 63.3% to 95.8%, contribution falls from 0.298 to 0.132 (dormant slots: 0.021), and WER rises to 3.48. The bias therefore moves the switch in both directions. Dormant FFNs cannot simply be woken up; this is consistent with W_1 being co-adapted to the large β.

Inherited from English pretraining. The same band exists in the English Nemotron checkpoint. Its FFN_1 contributions in layers 10–14 are 0.07–0.71% of the residual RMS, and 100% of units are off (median -14.6 to -16.4), close to Garak-1's 0.12–0.57%. Its β RMS profile is nearly identical (Figure 2a). The weights carry a "fossil" of this history. Relative to their English initialisation, the 24 dormant matrices of Garak-1 have norm ratios of 0.9760–0.9777, while the 72 active matrices have 0.9792–0.9926, with no overlap. This is consistent with dormant weights receiving little gradient and shrinking mainly under weight decay. Their directions have barely moved: after the best scalar fit, the dormant W_1 and W_2 differ from their English initialisation by 1.6% and 2.3% (cosine 0.9998 and 0.9997), against 9.4% and 11.2% for the active matrices. Parakeet-TDT 0.6B v2, an offline English model that shares the massive-activation channel indices, which suggests a shared initialisation (not documented in the model cards), has smaller β (0.10–0.24 in these layers) and live FFNs (FFN_1 contributions 33–39% of the residual RMS in layers 10–14). Swapping β between the two models is consistent with this co-adaptation. Parakeet's β in Nemotron's dormant slots wakes them (off units 99.9%→60%, contribution 0.02→2.2) but breaks recognition (WER 100%). Nemotron's β in Parakeet inflates the FFN contribution to 8.4 times the residual RMS.

4 A DATA-FREE DETECTOR AND A SURVEY OF PUBLIC CHECKPOINTS

The mechanism suggests a detector that needs no audio. If the LayerNorm output is approximated as $\beta + \gamma \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, the pre-activation of hidden unit i is Gaussian with mean $w_i^\top \beta$ and standard deviation $\|w_i \odot \gamma\|$ (the up-projection bias, absent in Garak-1 and the Nemotron checkpoints, is ignored, which makes the detector approximate for checkpoints with FFN biases). Its probability of being positive is therefore $\Phi\big(w_i^\top \beta / \|w_i \odot \gamma\|\big)$. I flag a module as dormant when fewer than 3% of its units have $P(z_i > 0) > 1\%$. Across all 48 modules, the predicted fraction of active units has a Spearman correlation with the measured residual contribution of 0.82 in Garak-1 and 0.83 in Nemotron EN (0.25 in Parakeet-TDT 0.6B v2, which has no dormant modules). In Garak-1, the detector selects exactly 12 modules: the nine of D plus FFN_2 in layers 9 and 13 and FFN_1 in layer 15.

Table 2 applies the detector to 14 public checkpoints and Garak-1. Dormancy is lineage-specific. The two Nemotron streaming checkpoints and Garak-1, which descends from one of them, have 11–12 dormant modules (15–16% of parameters). Parakeet-TDT 1.1B has a single dormant module (FFN_2 in layer 1). The detector flags none in the remaining eleven checkpoints. Streaming alone is not the cause, because the two small cache-aware streaming models have no dormant modules. Direct measurements agree with the detector: Nemotron 3.5 has FFN_1 contributions of 0.08–0.73%

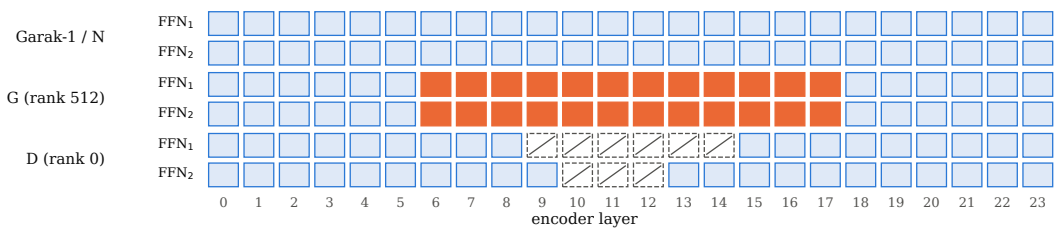


Figure 3: **Where each model is compressed.** Each cell is one macaron FFN module. G factorises every FFN in layers 6–17 at rank 512 (orange). D removes the nine dormant modules and their LayerNorms, or replaces each with a constant (dashed). Both remove 75.5M parameters.

Table 3: **Compression without training vs. continued training** on the Korean test set (pooled-18 CER, registered bootstrap procedure). Differences vs. Garak-1 for D- and W-variants and zero-shot factorisations are exploratory. "zs": zero-shot, no training after compression. W uses no audio at all; bold marks the weights-only rows.

Model	Removed	Train	Pooled-18	Δ vs. Garak-1 [95% int.]	dev
Garak-1	–	–	4.637	–	4.753
N (uncompressed control)	–	35K	4.651	$+0.014\,[-0.009,\ +0.038]$	–
G (rank 512, layers 6–17)	75.5M	35.5K	4.659	$+0.022\,[-0.004,\ +0.049]$	4.816
D-mean: mean-replace 9 dormant	75.5M	zs	4.643	$+0.006\,[-0.005,\ +0.018]$	4.762
D: remove 9 dormant	75.5M	zs	4.659	$+0.023\,[+0.010,\ +0.035]$	4.764
D_6: remove 6 dormant	50.3M	zs	4.642	$+0.005\,[-0.002,\ +0.012]$	4.741
D_{12}: remove 12 dormant	100.7M	zs	4.664	$+0.027\,[+0.011,\ +0.043]$	4.794
D_{12}-mean	100.7M	zs	4.651	$+0.014\,[-0.001,\ +0.029]$	4.778
W: weights-only constants, 9	75.5M	zs	**4.631**	$\mathbf{-0.005\,[-0.019,\ +0.008]}$	4.742
W_{12}: weights-only constants, 12	100.7M	zs	**4.638**	$\mathbf{+0.002\,[-0.015,\ +0.019]}$	4.771
rank 512, layers 6–17 (G untrained)	75.5M	zs	4.690	$+0.053\,[+0.033,\ +0.073]$	4.827
spectrum-guided ranks	75.5M	zs	4.661	$+0.024\,[+0.009,\ +0.040]$	4.784
rank 512, layers 0–5, 18–23	75.5M	zs	5.042	$+0.405\,[+0.371,\ +0.439]$	5.170
remove 9 early FFNs (layers 1–5)	75.5M	zs	9.102	$+4.466\,[+4.374,\ +4.558]$	9.343

of the residual RMS in layers 10–14, while Parakeet-TDT 0.6B v2 (33–39%) and FastConformer Transducer Large (30–243%) have none dormant.[1]

5 The rank-0 limit: removing or mean-replacing dormant FFNs

If a module's output is almost constant, the best rank-0 approximation is its mean. The first variant, **D (removal)**, deletes the nine dormant modules (FFN_1 in layers 9–14 and FFN_2 in layers 10–12, Figure 3) together with their pre-FFN LayerNorms. The residual branch in Eq. (1) then becomes the identity, leaving 546,509,962 parameters (-12.1%), the same budget as G. The second variant, **D-mean (mean replacement)**, instead replaces each module's contribution by its mean over 60 calibration utterances, a 1024-dimensional constant per module (9,216 floats in total), so each block computes $x + \mu_j$. The third, **W (weights-only constants)**, needs no audio: it draws 16,384 LayerNorm outputs from the detector's model $\beta + \gamma \odot \epsilon$ (Section 4) and uses the mean FFN output as the constant. No variant is trained. D_6 and D_{12} remove 6 and 12 modules in order of output energy; D_{12} equals the detector's selection, and W_{12} applies W to those 12 modules (521.3M parameters plus 12,288 constants).

Korean results. On this test set, Table 3 shows that constant (mean) replacement is the best way to spend the 75.5M budget (point estimates). With no training, D-mean is $+0.006\,[-0.005,\ +0.018]$ points from Garak-1, inside a post-hoc ± 0.10 margin. That is better than plain removal (D-mean$-$D: $-0.016\,[-0.027,\ -0.006]$, exploratory). Its point estimate is also below G after 35.5K training steps (D-mean$-$G: $-0.016\,[-0.043,\ +0.011]$) and N itself ($-0.008\,[-0.033,\ +0.015]$), although

[1] All public-model activation measurements used attention context [70, 13], which is not a native setting for the offline models (native full context) or for Nemotron 3.5 (native $[56, \cdot]$).

Table 4: **Continued training at matched steps** (pooled-18 / macro-18 CER on the Korean test set). All runs use the same data plan, schedule and hardware; step 0 is the zero-shot model (for G, a factorisation rebuilt with the same procedure, not the saved start checkpoint). †G at 35K; at 35.5K (35,520 steps, used in Table 1) it is 4.659 / 4.510.

Steps	0	2.5K	5K	10K	30K	35K
N (uncompressed, 622.0M)	4.637 / 4.483	–	4.651 / 4.497	4.654 / 4.494	4.657 / 4.499	4.651 / 4.499
G (rank 512, 546.8M)	4.690 / –	–	4.695 / 4.545	4.703 / 4.549	4.667 / 4.514	4.692 / 4.530†
D (removal, 546.5M)	4.659 / 4.504	4.635 / 4.485	4.662 / 4.503	4.646 / 4.492	–	–
D_{12} (removal, 521.3M)	4.664 / –	4.651 / 4.499	4.654 / 4.503	–	–	–

neither interval excludes zero (exploratory). At the detector's full selection of 12 modules (-16.2%), D_{12}-mean is $+0.014\,[-0.001,\ +0.029]$ from Garak-1 and has the same point estimate as N (D_{12}-mean$-$N: $-0.000\,[-0.024,\ +0.023]$, exploratory); on the held-out broadcast domain it scores 8.730 (Garak-1 8.684). The constant restores what the dormant modules still contribute, a nearly input-independent offset to the residual stream. Plain removal is a close second on this test set.

Constants from the weights alone. W needs neither audio nor training and does as well: W is $-0.005\,[-0.019,\ +0.008]$ and W_{12} (-16.2%) $+0.002\,[-0.015,\ +0.019]$ from Garak-1. On this test set W_{12} beats removal of the same 12 modules ($-0.026\,[-0.043,\ -0.008]$) and W beats trained G ($-0.028\,[-0.055,\ -0.001]$); both comparisons are exploratory. Whether a constant is there matters more than where it comes from: constants from English LibriSpeech audio (4.647), white noise (4.637), Korean audio (4.643) or the weights (4.631), and the weights-only constant scaled by $0.5\times$–$2\times$ (4.641–4.644), all lie below removal (4.659), with the unscaled weights-only constant lowest on this test set (point estimates); flipping its sign ($-1\times$: 4.663) is as bad as removal. At the standard lookahead, a dormant FFN therefore acts almost as a single learned bias vector, although three of the 12 detected modules still add 10–14% of the residual RMS, so W_{12} approximates rather than removes them.

A pre-registered held-out check. Because the test set was used during development, I registered one more test before decoding: on 40,000 utterances from a separate Korean test pool drawn from the same public corpora (60 h, 11 domains, excluding 463) that share no audio file with the test or development sets and were not used for any setting choice in this study, the 95% interval of $W_{12}-$Garak-1 must lie within $\pm1\%$ of Garak-1's CER. It does ($+0.026\,[+0.008,\ +0.045]$ against a margin of ±0.053; Garak-1 5.278). The interval excludes zero, so the replacement has a small but real cost of 0.5% relative, and the registered secondary contrast, removal worse than constants, did not replicate ($-0.018\,[-0.037,\ +0.001]$ for removal$-W_{12}$); which rank-0 variant is better is set-dependent. The registration listed no other contrasts; the W and R (Section 6) comparisons on this set were added afterwards and are exploratory: W (9 modules) costs $+0.016\,[+0.002,\ +0.031]$.

Latency dependence. W_{12} matches Garak-1 at the standard 1,040 ms lookahead, is not distinguishable from it at 480 ms ($+0.014\,[-0.003,\ +0.031]$), and is worse at 0 ms ($+0.036\,[+0.017,\ +0.055]$; Garak-1 4.741 and 5.133). At 0 ms, a constant measured at that setting (5.162) and removal (5.167) cost as much as W_{12} (5.169; point estimates), so with little right context these modules add input-dependent information: dormancy depends on the operating latency.

At a fixed budget, which modules are touched matters more than the rank (Appendix B): spending 75.5M on the edge layers at rank 512 costs $+0.41$, and removing nine early FFNs $+4.5$.

Public English models. Removal transfers without any training (Table 2). Removing the 11 detector-flagged modules from Nemotron EN (92.3M, -14.9%) leaves LibriSpeech test-clean WER essentially unchanged at every latency setting (single runs, no intervals); per-setting values and weights-only constants on the public models are in Appendix C.

6 Matched continued training of D and R

I continue training D with exactly the plan used for N and G (Table 4). At 30K steps G was still detectably worse than Garak-1 ($+0.030\,[+0.004,\ +0.057]$) and not detectably different from N ($+0.010\,[-0.017,\ +0.037]$); the 35K stopping point was chosen after this evaluation (Limitation ix). After 2.5K steps D reaches 4.635 pooled and 4.485 macro-18, close to Garak-1 (4.637 / 4.483), and 8.616 on the held-out broadcast domain (Garak-1 8.684, N 8.630, G 8.775). At 5K

Table 5: **Change in total processing time vs. Garak-1** (greedy decoding, FP32, median of 6 blocks from two order-reversed rounds; negative = faster; CPU: Xeon Platinum 8480+). Neither run is the pre-registered speed test; the timings carry no confidence intervals, and both runs shared the host with training jobs (protocol in Appendix C).

	G-only run	Comparison run		
Setting	G	G	spectrum-guided	D
CPU, batch 8, 8 cores	-18.1%	-16.1%	-7.2%	-8.7%
CPU, batch 1, 8 threads	-11.1%	-7.6%	-8.9%	-8.6%
GPU H100, batch 64, CUDA-graph decoder	-7.3%	-7.1%	-7.9%	-9.0%
GPU H100, batch 1, CUDA-graph decoder	$+4.8\%$	$+10.9\%$	$+3.4\%$	$+0.9\%$

steps it is at 4.662 / 4.503; the 5K−Garak-1 interval is $+0.026\,[+0.003,\ +0.048]$ (exploratory). At 7.5K steps it is at 4.652 / 4.489, with 8.684 on the broadcast domain (7.5K−N at 10K: $-0.002\,[-0.029,\ +0.024]$). D's checkpoints therefore bracket N (4.651); the 0.027 swing between 2.5K and 5K ($-0.027\,[-0.050,\ -0.006]$) is larger than the spread of N's checkpoints (at most 0.007, 5K–35K), so I do not read the 2.5K value as a stable gain over Garak-1. The 16.2%-smaller D_{12} (521.3M) reaches 4.651 / 4.499 after 2.5K steps, the same point estimate as N ($D_{12}-N$: $+0.000\,[-0.025,\ +0.025]$, exploratory). At matched steps D is ahead of G, which is at 4.695 at 5K and 4.703 at 10K ($D_{5K}-G_{5K}$: -0.033, point estimate; $D_{2.5K}-G_{5K}$: $-0.060\,[-0.086,\ -0.034]$, exploratory). Compressing by removal starts closer to the target and needs no recovery phase, whereas the recovery of factorisation is non-monotonic and checkpoint-dependent. At 10K steps D is at 4.646 / 4.492 and 8.652 on the broadcast domain ($D_{10K}-N_{10K}$: $-0.008\,[-0.031,\ +0.015]$), and D_{12} at 5K is at 4.654 / 4.503 ($D_{12,5K}-N_{10K}$: $+0.000\,[-0.024,\ +0.024]$); neither run moves away from Garak-1 or N (exploratory).

Reviving the dormant FFNs. Instead of removing the capacity, R reclaims it: for the nine modules it sets β to zero, re-initialises W_1, zeroes W_2 and adds an output bias equal to the module mean, so it starts exactly at D-mean (dev 4.762), and it trains these tensors at ten times the base learning rate with the same plan (622.0M parameters). After 2.5K steps R is at 4.631 / 4.480, $-0.006\,[-0.031,\ +0.020]$ from Garak-1 and $-0.020\,[-0.048,\ +0.008]$ from N at 5K (exploratory); reviving all 12 at three times that rate gives 4.660. On the held-out set of Section 5, R is worse than Garak-1 ($+0.039\,[+0.013,\ +0.065]$, exploratory), so its test-set point estimate does not generalise. At 5K steps R is at 4.662 / 4.510 and worse than Garak-1 on both the test set ($+0.026\,[+0.002,\ +0.048]$) and the held-out set ($+0.025\,[+0.001,\ +0.048]$; exploratory), the same as D at 5K. R at 2.5K equals W, which needs no training (R−W: $-0.000\,[-0.026,\ +0.026]$, exploratory); within this budget the dormant capacity is not worth reclaiming, which supports removing it.

7 Inference speed

The G-only run (Table 5) shows that G's gain depends on the regime. On CPU, G is faster at both batch sizes (-18.1% at batch 8, -11.1% at batch 1). Removal skips the FFN matrix multiplications instead: D is the fastest structure on the GPU at batch 64 (-9.0%) and roughly neutral at batch 1 ($+0.9\%$, against $+10.9\%$ for G in the same run). In the comparison run the CPU rows also favour G at batch 8 (-16.1%). Both table runs overlapped with other jobs on the host, including training, and in the comparison run a training job was not pinned away from the benchmark cores; I therefore report G's batch-8 CPU gain as the range of the two runs (16–18%) and treat the comparison-run CPU differences between structures as indicative. A later run on dedicated cores that were idle during the measurement (batch 8, two order-reversed rounds, no interval; Appendix C) times removal of the 9 and 12 dormant modules at -11.5% and -13.0% (39.35 s → 34.84 s and 34.22 s), against -8.7% for D in the comparison run; since constant replacement adds only one vector per module, I use these times as a proxy for W and W_{12}.

8 Related work

Activation-aware low-rank compression. FWSVD (Hsu et al., 2022), ASVD (Yuan et al., 2023) and SVD-LLM (Wang et al., 2025c) weight the SVD by input statistics. SVD-LLM V2 (Wang et al., 2025b), AdaSVD (Li et al., 2025) and Dobi-SVD (Wang et al., 2025a) allocate ranks non-uniformly. I use the whitening formulation and a simple greedy allocation and claim novelty for neither; LiteASR (Kamahori et al., 2025) applies activation-based low-rank compression to Whisper.

Speech encoder compression and redundancy. LiteASR reports that earlier Whisper layers are easier to compress, the opposite of the U-shape found here. Low-rank training (Fernandez-Lopez et al., 2025), metric-driven Conformer layer pruning (Bekal et al., 2023) and structured pruning of self-supervised models (Peng et al., 2023) compress with retraining. Redundancy in self-supervised speech stacks has been measured (Dorszewski et al., 2025), and concurrent work removes whole Whisper encoder layers (Aagaard & Detlefsen, 2026). I instead find dormant *sub-modules* whose neighbouring attention and convolution remain necessary.

Layer redundancy and dormant units. In LLMs, deeper layers can often be dropped (Gromov et al., 2025; Men et al., 2025; Lad et al., 2025; Zhong et al., 2025), attributed to under-used depth (Csordás et al., 2025) or Pre-LN variance growth (Sun et al., 2025), and attention is found to be more redundant than MLPs (He et al., 2024; Zhang et al., 2024). Dormant units are known in reinforcement learning (Sokar et al., 2023) and are controlled in Zipformer (Yao et al., 2024); weight decay induces low-rank structure (Kobayashi et al., 2024). The dormant components found here are whole FFN modules switched off by a LayerNorm bias and inherited from pretraining.

9 Limitations and conclusion

Limitations. (i) Dormancy is specific to one lineage. The survey shows that most public checkpoints have no dormant modules, so the rank-0 gains apply to the Nemotron family rather than to all Conformers, although the detector costs nothing to run. Dormancy is not a defect of these checkpoints: the flagged modules cost nothing in accuracy when removed from the public models tested, and I make no claim about why they arose during pretraining. (ii) Only G and the held-out check of W_{12} were pre-registered. D, D-mean, W, R and the spectrum-guided allocation were designed after I saw the anatomy, and their comparisons are exploratory and uncorrected for multiplicity. (iii) Trained comparisons, including the revival runs, use a single seed and short budgets; the 35K stop precedes the scheduled learning-rate decay, and gradient clipping was active at every step of the final training segment. (iv) The held-out broadcast domain 463 shows a larger gap for G. (v) CPU timings were measured on a host shared with GPU training jobs; order-reversed rounds reduce, but do not remove, this interference. (vi) Dormancy statistics and the D-mean constants use 60–120 calibration utterances; the SVD covariances, spectra and output energies use 1,000. (vii) The test set was used in earlier architecture development and is not an untouched hold-out; on the pre-registered held-out set W_{12} is within the 1% margin but 0.5% worse. (viii) Rank-0 replacement is latency-dependent: at 0 ms lookahead the 12 modules carry information and W_{12} costs $+0.036$. (ix) The 35K comparison is an early stop of the registered 200K plan, chosen after the 30K test evaluation, and compares G at 35,520 with N at 35,000 steps; at the common 35K step G is worse than N ($+0.041\,[+0.013,\ +0.069]$).

Conclusion. Registered rank-512 compression of the middle encoder layers of a Korean streaming ASR model removes 12.1% of its parameters and 16–18% of CPU processing time at batch size 8, and at the registered common step (35K) it is slightly worse than an uncompressed control trained with the same plan (G$-$N $+0.041\,[+0.013,\ +0.069]$, 0.9% relative); 520 steps later no difference is detected ($+0.008\,[-0.019,\ +0.036]$), so the gap is checkpoint-dependent. It works because the middle macaron FFNs are dormant. A large pre-FFN LayerNorm bias, inherited from English pretraining, switches them off; this can be shown causally and detected from the weights alone. At the rank-0 limit, the dormant FFNs can be replaced by constants computed from the weights alone, with no audio and no training: at the standard 1,040 ms lookahead, replacing 12 of them (16.2% of the parameters) by 12,288 constants is statistically indistinguishable from the uncompressed model on the test set and stays within a pre-registered 1% margin on held-out data at a measurable 0.5% cost, and removing the same modules cuts CPU time at batch size 8 by 13% (one run, no interval); at 0 ms lookahead the constants cost accuracy. Knowing *where* to compress matters more than how much.

AI USE STATEMENT

In this work, I used generative AI tools to aid and polish the writing, to retrieve and discover related work, for research ideation and execution (including writing and debugging analysis, training and evaluation code), and to draft sections of this paper. I have reviewed all AI-assisted work, checked every reported number against my experiment logs, verified cited works against their sources, and tested AI-generated code for correctness. The registered compression design (selective rank-512 factorisation of the middle encoder layers, the identically planned uncompressed control and the CPU-speed objective) is mine. Generative AI proposed several follow-up analyses (the activation anatomy, the LayerNorm-bias interventions, the data-free detector, the weights-only constants, the held-out check and the revival runs), which I directed, reviewed and verified. I take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All training and evaluation data are publicly available speech corpora (Korean AI-Hub corpora, LibriSpeech and FLEURS), which I used under their respective licenses. Public model checkpoints were used under their licenses. I release no audio, transcripts or other personal data. Cheaper speech recognition can support accessibility but also surveillance; I add no capability beyond that of the uncompressed models.

REPRODUCIBILITY STATEMENT

Sections 2 and 5 and Appendix A specify the factorisation, allocation, removal and mean-replacement procedures, all parameter counts and the training configuration. Section 2 specifies the evaluation and bootstrap protocol, and Section 7 and Appendix C the timing protocol. The data-free detector (Section 4) needs only the checkpoint weights. Garak-1 is not publicly released; Section 2 and Appendix A give its architecture, training data and recipe. Appendix E lists per-domain and per-layer values. I will release the code of the detector, the removal and mean-replacement procedures and the evaluation scripts.

ACKNOWLEDGMENTS

I thank the NVIDIA Nemotron team for releasing the Nemotron Speech Streaming models and pre-trained weights, on which Garak-1 and this work are built.

REFERENCES

Rasmus Aagaard and Nicki Skafte Detlefsen. Six layers less: Encoder pruning for Whisper with label-free recovery. *arXiv preprint arXiv:2609.27980*, 2026.

Anonymous. Garak-1: Cache-aware FastConformer–TDT for faster and more accurate Korean streaming ASR, 2026. Unpublished manuscript.

Dhanush Bekal, Karthik Gopalakrishnan, Karel Mundnich, Srikanth Ronanki, Sravan Bodapati, and Katrin Kirchhoff. A metric-driven approach to Conformer layer pruning for efficient ASR inference. In *Proc. Interspeech*, pp. 4079–4083, 2023.

Maximilian Bisani and Hermann Ney. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pp. I–409–I–412, 2004.

Róbert Csordás, Christopher D. Manning, and Christopher Potts. Do language models use their depth efficiently? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Teresa Dorszewski, Albert Kjøller Jacobsen, Lenka Tětková, and Lars Kai Hansen. How redundant is the Transformer stack in speech representation models? In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

Adriana Fernandez-Lopez, Shiwei Liu, Lu Yin, Stavros Petridis, and Maja Pantic. Full-rank no more: Low-rank weight training for modern speech recognition models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations (ICLR)*, 2025.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for speech recognition. In *Proc. Interspeech*, pp. 5036–5040, 2020.

Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in Transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations (ICLR)*, 2022.

Keisuke Kamahori, Jungo Kasai, Noriyuki Kojima, and Baris Kasikci. LiteASR: Efficient automatic speech recognition with low-rank approximation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3430–3442, 2025.

Seijin Kobayashi, Yassir Akram, and Johannes von Oswald. Weight decay induces low-rank attention layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. Remarkable robustness of LLMs: Stages of inference? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 130050–130083, 2025.

Zhiteng Li, Mingyuan Xia, Jingyuan Zhang, Zheng Hui, Haotong Qin, Linghe Kong, Yulun Zhang, and Xiaokang Yang. AdaSVD: Adaptive singular value decomposition for large language models. *arXiv preprint arXiv:2502.01403*, 2025.

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*, pp. 20192–20204, 2025.

Vahid Noroozi, Somshubra Majumdar, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. Stateful Conformer with cache-based inference for streaming automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12041–12045, 2024.

NVIDIA. nemotron-speech-streaming-en-0.6b (Nemotron ASR Streaming) model card. Hugging Face model card, 2026. URL `https://huggingface.co/nvidia/nemotron-speech-streaming-en-0.6b`. Accessed 2026-09-26.

Yifan Peng, Kwangyoun Kim, Felix Wu, Prashant Sridhar, and Shinji Watanabe. Structured pruning of self-supervised pre-trained models for speech recognition and understanding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. Fast Conformer with linearly scalable attention for efficient speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168, 2023.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *International Conference on Learning Representations (ICLR)*, 2025a.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. SVD-LLM V2: Optimizing singular value truncation for large language model compression. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4287–4296, 2025b.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations (ICLR)*, 2025c.

Hainan Xu, Fei Jia, Somshubra Majumdar, He Huang, Shinji Watanabe, and Boris Ginsburg. Efficient sequence transduction by jointly predicting tokens and durations. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38462–38484, 2023.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. Zipformer: A faster and better encoder for automatic speech recognition. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.

Yang Zhang, Yawei Li, Xinpeng Wang, Qianli Shen, Barbara Plank, Bernd Bischl, Mina Rezaei, and Kenji Kawaguchi. FinerCut: Finer-grained interpretable layer pruning for large language models. *arXiv preprint arXiv:2405.18218*, 2024.

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. BlockPruner: Fine-grained pruning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5065–5080, 2025.

A Model, measurement and allocation details

Garak-1. Garak-1 (Anonymous, 2026) has 622,025,866 parameters: encoder 609,141,760 (FFNs 402,653,184), prediction network 9,185,920 and joint network 3,698,186. The encoder is a 24-layer cache-aware FastConformer ($d_{\text{model}} = 1024$, FFN expansion 4096, Swish, no FFN biases, residual weight 0.5, attention context $[70, 13]$). The decoder is a TDT with durations 0–8 over a 4,096-token SentencePiece vocabulary. The encoder was initialised from Nemotron Speech Streaming EN 0.6B (NVIDIA, 2026); the decoder was trained from scratch.

Measurements. The ratio is the RMS of $0.5 \cdot \mathrm{FFN}(x)$ divided by the RMS of the layer input (120 calibration utterances, batch 1, no padding). A unit is "off" if its pre-activation averaged over all frames is below -4. Output energy is $\mathbb{E}\|W_2 h\|^2$ over the 1,000 calibration utterances used for the factorisation spectra. Single-module removal, whole-layer skipping and random removal are evaluated on the 8,500-utterance development set. Whole-layer skipping costs $+0.306$ at layer 4, $+0.123$ at layer 11, $+0.503$ at layers 10–12 and $+0.794$ at layers 9–13. Random nine-module removal gives 6.390, 6.768 and 6.177 (Garak-1 4.753).

Allocations. The greedy allocation lets each of the 96 matrices take a rank in $\{\text{full}, 768, 640, 512, 448, 384, 320, 256, 192, 128, 96, 64\}$. Starting from full rank everywhere, it repeatedly takes the rank-lowering step with the smallest increase in $1 - E_R$ per parameter saved, until the target budget is reached. At 75.5M it factorises 28 matrices, 24 of them up-projections; the summed energy loss is 0.036, against 0.479 for G and 1.082 for edge-layer rank 512.

Table 6: Zero-shot development-set sweep (Garak-1 4.753). Δ in CER points.

Removed	Method (units)	Units	dev CER	Δ
50.3M	remove dormant FFNs (D_6)	6	4.741	−0.012
75.5M	uniform rank 512, layers 6–17 (G, untrained)	48	4.827	+0.074
75.5M	uniform rank 512, layers 0–5, 18–23	48	5.170	+0.417
75.5M	spectrum-guided ranks 64–768	28	4.784	+0.031
75.5M	remove dormant FFNs (D)	9	4.764	+0.011
75.5M	mean-replace dormant FFNs (D-mean)	9	4.762	+0.009
75.5M	remove early FFNs (layers 1–5)	9	9.343	+4.590
75.5M	remove random FFNs (3 draws)	9	6.177–6.768	+1.424–+2.015
100.7M	remove dormant FFNs (D_{12})	12	4.794	+0.041
107.0M	uniform rank 384, layers 6–17	48	4.878	+0.125
107.0M	spectrum-guided	40	4.836	+0.083
138.4M	uniform rank 256, layers 6–17	48	5.036	+0.283
138.5M	spectrum-guided	49	4.927	+0.174
151.0M	uniform rank 512, all 24 layers	96	5.275	+0.522
151.2M	spectrum-guided	51	4.983	+0.230
75.5M	G after 35.5K training steps	48	4.816	+0.063

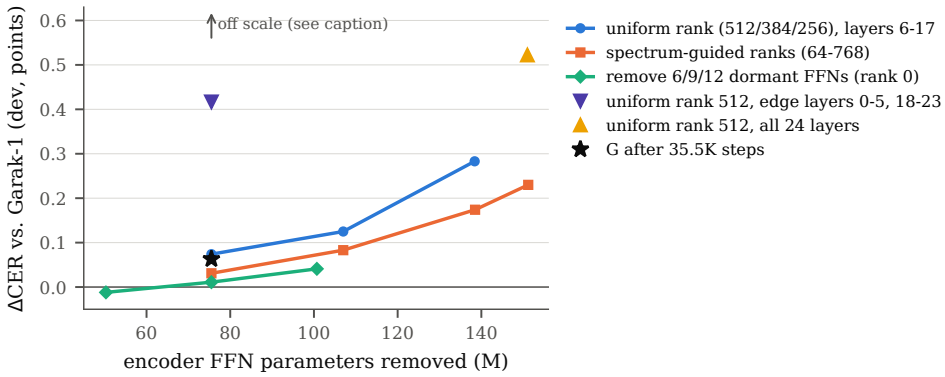


Figure 4: Zero-shot development CER change versus parameters removed (Table 6). Off scale at 75.5M (arrow): removing nine early FFNs (layers 1–5) costs +4.59, and removing nine random FFN modules costs +1.42 to +2.02 (three draws).

B Selection matters more than rank

At a fixed budget, the cost of compression depends on *which* modules are touched (Table 3; development-set sweeps in Appendix Table 6 and Figure 4). Spending 75.5M on the edge layers at rank 512 costs +0.41 on test, and removing nine early FFNs costs +4.5. On dev, removing nine random FFN modules costs +1.42 to +2.02 (three draws), while removing the nine dormant ones costs +0.011. A spectrum-guided greedy allocation, which lowers ranks where $1 - E_R$ grows least per parameter saved, beats uniform ranks at every dev budget tested. It costs +0.031 vs. +0.074 at 75.5M, +0.083 vs. +0.125 at 107M, +0.174 vs. +0.283 at 138M, and +0.230 vs. +0.522 at 151M. Its choices mirror the anatomy: at 75.5M it factorises up-projections widely, gives the up-projections of all nine dormant modules ranks of 96–192 (seven of their nine down-projections stay at full rank), and leaves the high-loss down-projections near the ends intact. Removal and mean replacement are the end point of this progression: once a module's spectrum is concentrated and its output is dominated by a constant, a rank-0 approximation becomes effective.

C Speed measurement details

The G-only run compares G with Garak-1 only; the comparison run measures all structures together. Neither run is the pre-registered speed test (locked CPU engine, batch 1, 10 paired blocks with a bootstrap interval), which does not yet support G and was not performed; the timings carry no

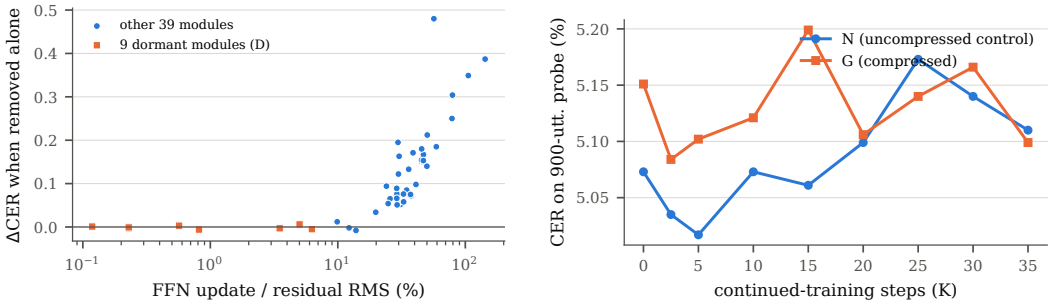


Figure 5: Left: development ΔCER when each of the 48 FFN modules is removed alone, versus its residual contribution (Spearman 0.888; 0.833 with output energy). Right: pooled CER on a 900-utterance probe during continued training of N and G; the probe is small and serves only to track dynamics.

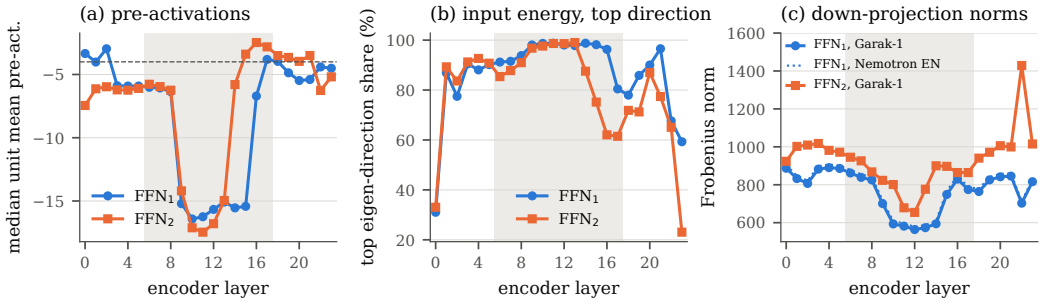


Figure 6: Further dormancy statistics for Garak-1. (a) Median over hidden units of the mean pre-activation (dashed: -4). (b) Share of the uncentred second moment of the normalised FFN input carried by its top direction. (c) Down-projection Frobenius norms; dotted: English initialisation.

confidence intervals. Both runs overlapped with other GPU jobs on the same host, including training jobs. In the G-only run the concurrent jobs were GPU jobs that, as far as I could verify, did not use the benchmark cores; in the comparison run a training job was not CPU-pinned, so the eight benchmark cores were not dedicated. CPU batch 8: 64 utterances (432 s of audio); CPU batch 1: 40 utterances (265 s); GPU batch 1: 60 utterances (408 s). Inputs are a fixed random sample (seed 20260926) of the 18-domain test set with 1 s of trailing silence; each model runs in its own process with one warm-up pass and 3 timed passes per round, 8 intra-op threads (PyTorch 2.10, NeMo 3.1). The idle-core CPU run (Section 7) used batch 8, 64 utterances, 8 intra-op threads on dedicated cores that were idle during the measurement, and 2 order-reversed rounds of 3 timed passes (FP32, greedy decoding, median of the 6 passes); removal was timed as a proxy for constant replacement, which adds one vector per module, and, as in the comparison run, each removed module returned a zero output while its LayerNorm still ran.

Checkpoint averages (Section 6): Garak-1 with N's 5K–35K checkpoints (4.634), the last five checkpoints of N, 25K–39.2K (4.641), and of G, 25K–35.5K (4.653).

Additional details. D costs $+0.023$ on test and $+0.011$ on dev, less than half of G's zero-shot cost ($+0.053$), and its point estimate equals that of G after 35.5K steps (G$-$D: $-0.000\,[-0.027,\,+0.027]$, exploratory). The multilingual Nemotron 3.5 (12 modules, -15.8%) goes from 3.458 to 3.465 on English and from 9.727 to 9.614 CER on Korean FLEURS; 92% of English and 88% of Korean outputs are character-identical to the full model's. Parakeet-TDT 1.1B (1 module) goes from 1.527 to 1.529. The constants survive uncompressed continued training (N 35K with W: 4.645 vs. 4.651) but not factorised training: on G they cost $+0.018\,[+0.006,\,+0.030]$, presumably because G's dormant slots have been trained at rank 512. Averaging checkpoints of Garak-1, N or G gives 4.634–4.653, none distinguishable from Garak-1 (exploratory). The factorised matrices have 37.5% fewer multiply–accumulates, but the saving is only about 12% of the encoder's weight multiply–accumulates, less than the batch-8 gain, and D, which removes the same number, gains only -8.7%

in the comparison run; I have not profiled where the remaining gain comes from. In Nemotron EN with its 11 flagged modules removed, at right-context lookahead of 0, 80, 480 and 1,040 ms, WER goes from 3.003 to 2.988, 2.769 to 2.773, 2.672 to 2.655, and 2.532 to 2.526. Removing only the nine slots that D removes in Garak-1 gives 2.522, whereas removing nine early-layer FFNs gives 3.287. Weights-only constants do not help where removal is already lossless: at context $[70, 13]$ (the Nemotron 3.5 values above use its native context), W gives 2.575 on Nemotron EN (full 2.532, removal 2.526), 3.604 on Nemotron 3.5 (full 3.604, removal 3.570) and 9.544 CER on Korean FLEURS (full 9.531, removal 9.492), whereas in Garak-1, where removal costs $+0.027$, they recover it. On a small-batch GPU, I hypothesise (without profiling) that the extra kernel launch of each factorised matrix (48 per encoder step) turns into a slowdown in latency-bound inference; G's batch-1 change varies from $+3.8\%$ to $+12.3\%$ across rounds. The timed D slightly understates the speed of D as defined in Section 5. D leaves the error profile unchanged: dev substitution/deletion/insertion rates are 3.096/0.956/0.712 for D against 3.085/0.957/0.711 for Garak-1, whereas trained G has more deletions (0.995). In the G-only run, CPU batch 8 goes from 40.08 s to 32.84 s for 64 utterances, and GPU batch 64 from 4.301 s to 3.988 s for 512 utterances (3,693 s of audio).

D THE JOINT CORRECTION PATH OF G

G adds $U_{\text{corr}}(a_t \odot b_u)$ to the joint hidden state before the token output layer, where $a_t = \tanh(P_h h_t)$ and $b_u = \tanh(P_g g_u)$ are 128-dimensional. U_{corr} is zero-initialised and the duration logits are unchanged (294,912 parameters). On 300 development utterances, conditioning on the reference transcripts (teacher forcing over the full TDT lattice), the path lowers the TDT loss per utterance by 1.6–2.2% (two runs: 8.260→8.125 and 8.311→8.130) and helps 62–64% of utterances. Its output has 0.135 times the norm of the joint state, 64% of its energy is a constant component, and it lowers the blank probability by 0.031 per lattice cell. Folding only the constant into the output bias, which removes the path at zero inference cost, gives dev CER 4.841, against 4.816 with the path and 4.966 without it, so 83% of the dev gain is a constant offset. On the test set, however, folding gives 4.708 against 4.659 with the path. All 300 utterances used for the analysis come from a single domain (571, low-quality telephone speech), so this failure may partly reflect domain mismatch; I do not treat it as a definitive negative. Without the path, G's dev deletion rate rises from 0.995 to 1.189.

Recovery methods for the untrained rank-512 model plateau at about -0.01 on dev. A closed-form refit gives 4.815 (test 4.678) against 4.827, and teacher-guided recovery for 800 steps gives 4.818 at learning rates 10^{-5} and $2 \cdot 10^{-6}$; 10^{-4} diverges. A decoding-time penalty on long duration skips helps both models I tested by about 0.02 on test (point estimates; γ chosen on dev for each model): G with $\gamma = 3$ goes from 4.659 to 4.637 and Garak-1 with $\gamma = 1$ from 4.637 to 4.619, so G still trails Garak-1 ($+0.017$). A blank-logit offset does not help (G: 4.820–5.082 for offsets 0.5–3; Garak-1: 4.749 at 0.5). I therefore report all main results without decoding adjustments.

E PER-DOMAIN AND PER-LAYER TABLES

Table 7: Per-domain test CER (%), computed from per-domain CERs; IDs are AI-Hub dataset identifiers. Domain 463 is excluded from training. Garak-1 hypotheses were scored by three pipelines during this work (pooled-18 4.6366, 4.6370 and 4.6395). Table 1 uses the registered-bootstrap hypotheses (macro-19 4.704, macro-18 4.483); the Garak-1 column of this table comes from the per-domain scoring run (macro-19 4.706, macro-18 4.485). The differences (at most 0.003) change no conclusion.

ID	Domain	Garak-1	N	G	G−N	G−Garak-1
100	Counseling speech	5.061	5.169	5.205	+0.036	+0.144
109	Free conversation (adults)	2.847	2.790	2.860	+0.070	+0.013
115	Korean lecture speech	5.342	5.325	5.311	-0.014	-0.031
123	KsponSpeech (spontaneous)	7.418	7.464	7.551	+0.087	+0.133
130	Korean conversational speech	4.563	4.540	4.563	+0.023	+0.000
131	Loanword utterances	2.678	2.664	2.671	+0.007	-0.007
132	KconfSpeech (meetings)	6.712	6.795	6.738	-0.056	+0.027
464	Meetings by domain	7.260	7.229	7.268	+0.039	+0.009
466	Emotion/style TTS speech	1.710	1.719	1.763	+0.043	+0.053
470	Welfare call-center counseling	2.183	2.250	2.285	+0.035	+0.102
542	Multi-speaker TTS speech	1.572	1.589	1.619	+0.030	+0.047
568	Noisy-environment speech	4.648	4.555	4.662	+0.107	+0.014
571	Low-quality telephone speech	5.730	5.818	5.775	-0.043	+0.045
71320	Age-group slang	3.361	3.408	3.411	+0.003	+0.050
71417	Extreme-noise speech	8.332	8.380	8.337	-0.043	+0.005
71556	Address speech	1.850	1.878	1.803	-0.074	-0.047
71557	News-script anchor speech	1.038	1.024	1.046	+0.021	+0.007
71627	University lectures	8.389	8.388	8.310	-0.077	-0.079
463	Broadcast conversation (held out)	8.684	8.630	8.775	+0.145	+0.091

Table 8: Data-free survey of all 15 checkpoints (detector of Section 4; dormant parameters include LayerNorms).

Checkpoint	Streaming	Params (M)	Layers	Dormant/FFN	Dormant (M)	Share (%)
Garak-1	yes	622.0	24	12/48	100.7	16.2
nemotron-3.5-asr-streaming-0.6b	yes	638.0	24	12/48	100.7	15.8
nemotron-speech-streaming-en-0.6b	yes	618.1	24	11/48	92.3	14.9
parakeet-tdt-1.1b	no	1070.5	42	1/84	8.4	0.8
canary-1b-flash	no	890.1	32	0/64	0.0	0.0
parakeet-tdt-0.6b-v3	no	627.1	24	0/48	0.0	0.0
parakeet-tdt-0.6b-v2	no	617.9	24	0/48	0.0	0.0
parakeet-rnnt-0.6b	no	616.8	24	0/48	0.0	0.0
parakeet-ctc-0.6b	no	608.9	24	0/48	0.0	0.0
canary-180m-flash	no	189.1	17	0/34	0.0	0.0
stt_en_fastconformer_ctc_large	no	115.6	18	0/36	0.0	0.0
parakeet_realtime_eou_120m-v1	yes	114.9	17	0/34	0.0	0.0
stt_en_fastconformer_hybrid_large_streaming_multi	yes	114.8	17	0/34	0.0	0.0
parakeet-tdt_ctc-110m	no	114.7	17	0/34	0.0	0.0
stt_en_fastconformer_transducer_large	no	114.1	17	0/34	0.0	0.0

Table 9: Per-layer FFN statistics measured on 120 calibration utterances. Ratio: RMS of the FFN update relative to the residual RMS (%). $\tilde{z}$: median over units of the mean pre-activation. Off: units with mean pre-activation < -4 (%). Top: top-eigen-direction share of the FFN input (%). The right block is the English Nemotron initialisation.

	Garak-1								Nemotron EN (init.)			
Layer	Ratio_1	Ratio_2	$\tilde{z}_1$	$\tilde{z}_2$	Off_1	Off_2	Top_1	Top_2	Ratio_1	Ratio_2	$\tilde{z}_1$	$\tilde{z}_2$
0	36.1	37.6	-3.3	-7.5	35.7	88.0	31.0	33.0	36.0	35.7	-3.7	-7.7
1	36.0	29.6	-4.0	-6.1	50.1	68.4	86.9	89.3	35.3	27.7	-4.1	-6.2
2	45.3	29.2	-3.0	-6.0	39.4	64.7	77.5	83.6	45.6	28.0	-3.0	-6.0
3	29.1	25.7	-5.9	-6.2	59.3	73.3	90.5	91.3	28.5	24.4	-5.9	-6.3
4	24.1	24.8	-5.9	-6.2	61.0	78.0	88.1	92.7	23.6	24.3	-6.0	-6.5
5	31.1	29.0	-5.9	-6.1	65.1	73.2	90.2	90.8	30.5	27.0	-6.0	-6.2
6	34.7	32.8	-6.0	-5.8	67.8	64.2	91.2	85.4	34.2	31.6	-6.1	-5.7
7	37.2	32.8	-6.1	-6.0	69.6	71.3	91.5	87.8	37.4	30.8	-6.1	-5.9
8	29.0	29.2	-6.4	-6.2	80.0	78.3	93.8	91.0	26.9	29.3	-6.5	-6.3
9	5.01	9.89	-15.2	-14.2	99.9	99.1	98.1	96.8	6.30	24.7	-10.7	-8.8
10	0.23	3.50	-16.4	-17.1	100.0	99.9	98.7	97.6	0.07	7.62	-16.4	-13.7
11	0.12	0.82	-16.2	-17.5	100.0	100.0	98.7	98.7	0.59	1.12	-15.5	-17.6
12	0.23	6.26	-15.7	-16.8	100.0	100.0	98.0	98.6	0.26	6.47	-15.4	-17.3
13	0.56	13.9	-15.1	-15.0	100.0	99.9	97.8	99.0	0.71	12.2	-14.6	-11.8
14	0.57	41.0	-15.5	-5.8	100.0	70.1	98.8	87.6	0.69	37.2	-15.7	-5.9
15	12.3	45.3	-15.4	-3.4	100.0	43.7	98.2	75.2	6.96	43.6	-10.7	-3.4
16	19.8	50.3	-6.7	-2.5	91.1	25.6	96.3	62.1	20.4	51.3	-7.2	-2.6
17	29.9	47.0	-3.8	-2.8	48.2	31.6	80.5	61.5	35.3	44.4	-3.6	-2.8
18	29.7	46.9	-4.0	-3.5	49.7	41.8	78.0	71.9	30.2	43.6	-3.6	-3.4
19	30.3	50.0	-4.9	-3.6	59.3	43.9	85.9	71.2	28.2	52.4	-4.4	-3.4
20	38.9	59.2	-5.5	-4.0	68.2	49.3	90.0	87.0	38.4	74.0	-4.7	-3.4
21	45.5	78.6	-5.4	-3.5	68.5	41.7	96.6	77.4	50.2	85.0	-4.8	-2.9
22	143.0	79.3	-4.4	-6.3	56.8	72.2	67.7	65.1	132.3	122.1	-3.4	-4.2
23	56.7	105.6	-4.5	-5.2	58.9	59.9	59.3	23.1	70.1	144.1	-3.3	-3.7

Table 10: Pre-FFN LayerNorm parameters (RMS of β and γ), the constant share of the FFN_1 LayerNorm output (%), and the residual stream's largest channel (index and RMS relative to the median channel; 60 utterances).

	Garak-1				Nemotron EN		Parakeet 0.6B v2		Garak-1		
Layer	β_1	β_2	γ_1	γ_2	β_1	β_2	β_1	β_2	const.	ch.	$\times$med.
0	0.21	0.26	0.34	0.46	0.21	0.25	0.08	0.31	30.5	161	8.9
1	0.20	0.23	0.18	0.11	0.19	0.22	0.20	0.19	86.4	161	11.7
2	0.13	0.19	0.18	0.11	0.12	0.17	0.16	0.15	76.8	161	12.9
3	0.22	0.26	0.17	0.11	0.21	0.25	0.22	0.19	90.2	761	12.8
4	0.20	0.28	0.16	0.12	0.18	0.27	0.22	0.19	88.0	761	11.4
5	0.22	0.25	0.16	0.12	0.20	0.24	0.23	0.16	90.1	40	11.7
6	0.23	0.19	0.17	0.12	0.22	0.19	0.20	0.13	91.1	40	12.0
7	0.24	0.21	0.18	0.12	0.23	0.21	0.20	0.13	91.5	321	13.8
8	0.29	0.26	0.18	0.13	0.28	0.25	0.19	0.12	93.6	321	21.1
9	0.62	0.42	0.19	0.13	0.61	0.41	0.17	0.14	98.1	321	31.8
10	0.69	0.50	0.23	0.15	0.70	0.49	0.22	0.15	98.7	321	36.4
11	0.72	0.66	0.26	0.15	0.74	0.66	0.21	0.15	98.8	321	42.5
12	0.59	0.73	0.28	0.16	0.60	0.74	0.23	0.12	98.1	321	37.5
13	0.56	0.64	0.18	0.09	0.56	0.64	0.24	0.11	97.8	321	33.8
14	0.62	0.20	0.21	0.11	0.63	0.19	0.22	0.10	98.8	321	39.3
15	0.69	0.14	0.17	0.12	0.68	0.13	0.13	0.10	98.1	321	30.6
16	0.37	0.10	0.18	0.11	0.36	0.10	0.12	0.08	96.3	321	33.7
17	0.15	0.09	0.17	0.11	0.14	0.09	0.10	0.09	80.5	973	30.2
18	0.14	0.13	0.18	0.12	0.13	0.13	0.09	0.12	77.8	973	35.7
19	0.20	0.14	0.20	0.13	0.19	0.13	0.09	0.12	85.7	973	38.0
20	0.25	0.27	0.21	0.14	0.23	0.26	0.12	0.14	90.0	973	33.6
21	0.32	0.23	0.20	0.14	0.30	0.22	0.10	0.10	96.5	918	34.7
22	0.16	0.23	0.18	0.19	0.15	0.23	0.06	0.14	67.9	314	23.4
23	0.15	0.09	0.20	0.46	0.14	0.10	0.06	0.05	59.5	262	20.1