SSN++: LAYERWISE SUBSPACE NETWORKS FOR LOW-BANDWIDTH PIPELINE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Subspace networks (SSNs) make pipeline-parallel training communication-efficient by constraining every layer to write into one shared low-dimensional subspace, so activations and gradients cross each stage boundary losslessly. This shared subspace, however, limits what the model can express, and with it the loss an SSN can reach relative to an uncompressed transformer. We propose SSN++, which removes this constraint without additional communication. We introduce a norm-preserving orthogonal transform that losslessly transports each compressed activation from one layer's subspace into the next, so every layer can write into its own subspace. We extend this separation within each layer, giving attention and the MLP separate subspaces and dividing each into groups with their own subspaces, which together exceed the rank of one subspace at equal parameters, and we add learned gates that choose what each layer passes on. Because the compressed activations of preceding tokens have already crossed the boundary, the receiving layer also decodes its input from them with a learned causal filter, raising its rank beyond one subspace. Random subspaces spread activations like rotation-based quantisers, which makes the communication amenable to quantisation, and we spend the saved bits on a higher rank. Approximation theory and measurements during training support each choice. From 229M to 2B parameters, SSN++ lowers loss and raises eval accuracy over a parameter-matched SSN at equal communication, and with quantised communication it closes 63 to 78% of the matched SSN's loss gap to the uncompressed transformer at 229M and 1B and 58% at 2B, while training 12 to 26 times faster per update than it over 200 Mb/s links. The loss gap of SSNs thus stems largely from how capacity is organised around the compressed activations, and closing it without extra communication makes pipeline-parallel training over internet-grade links far more efficient.

1 INTRODUCTION

Pipeline parallelism trains a transformer across devices by placing successive layers on different accelerators, so activations and gradients cross device boundaries throughout training (Huang et al., 2019). Over limited bandwidth, as in heterogeneous and geographically distributed settings (Ryabinin et al., 2023; Avraham et al., 2026), this communication can leave the available compute idle. Subspace networks (SSNs) remove most of this cost by construction (Ramasinghe et al., 2025). Every attention and MLP output projection writes into one shared subspace of dimension k, much smaller than the model width, so the residual stream between any two layers is the token embedding plus k coefficients in that subspace. Stages exchange only these compressed activations and their gradients, and because the receiving stage rebuilds the residual stream exactly, the compression is lossless in both the forward and the backward pass.

The shared subspace that makes compression lossless also limits what the model can represent. Every layer, and both of its sublayers, attention and the MLP, must write into the same k directions, although different computations generally prefer different directions, and features superposed in few dimensions (Elhage et al., 2022) interfere more when all computations share one space. At the ranks that make communication feasible over internet-links, SSNs therefore reach a higher loss than uncompressed transformers trained on the same number of tokens. We ask whether the capacity of the model can be increased at all while its communication stays lossless and as small as before, and how much of the loss gap this would close.

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.

Our work, SSN++ (Figure 1), relaxes the shared-subspace restriction where each layer writes, reads, and communicates. Where a layer writes, attention and the MLP write into separate subspaces that differ between layers, and the output projection of each sublayer is split further into groups of hidden features, each writing into its own subspace. This grouping applies to any attention or MLP layer. With grouped-query attention (GQA) (Ainslie et al., 2023), the key–value heads give a natural number of attention groups, whereas the MLP group count is a free choice. Where a layer reads, an SSN decodes each token from its own compressed activation alone. SSN++ adds a learned causal finite impulse response (FIR) filter over the compressed activations of the preceding tokens, which have already crossed the boundary, so the layer decodes richer inputs without extra communication. In what it communicates, learned gates aggregate the group outputs into the k transmitted coefficients, and an orthogonal transform between consecutive subspaces preserves every compressed activation and gradient, so communication stays lossless. Because SSN subspaces are random orthonormal bases, which spread activations across hidden channels in the way rotation-based quantisers do (Ashkboos et al., 2024b), the compressed activations also tolerate integer quantisation, and we spend the saved bits on a higher rank at the same number of transmitted bytes. SSN++ thus becomes a layerwise subspace network whose layers have many more output directions than k, while the volume communicated between layers is unchanged. We make the following contributions:

- We introduce SSN++, which increases the capacity of SSNs inside each layer while keeping their communication lossless at the same volume.
- We motivate each change with measurements of rank and superposition, and show that during training SSN++ uses increasingly many directions beyond the transmitted rank.
- We establish approximation and capacity results for separate and grouped output projections, decoding from preceding tokens, and the propagation of quantisation error.
- At three model sizes, SSN++ lowers cross-entropy loss and raises eval accuracy at equal communication volume, and over 200 Mb/s links it trains 12 to 26 times faster than the uncompressed transformer.

1.1 Related Work

Low-rank gradient and parameter methods reduce optimisation or adaptation costs (Vogels et al., 2019; Zhao et al., 2024; Hu et al., 2022), while data-aware compression and layer-specific rotations motivate different approximation spaces across a transformer (Wang et al., 2025; Ashkboos et al., 2024a). MAPL and ResBM instead learn narrow representations between layers (Janson et al., 2026; Aboudib et al., 2026), whereas our focus is the capacity and training dynamics of the computation around lossless SSN communication. Appendix A discusses related work in more detail.

2 Preliminaries

Why pipeline compression must be lossless. In pipeline-parallel training each stage boundary carries every token's d-dimensional residual stream forward and its gradient backward, $4d$ bytes per token in BF16 precision. Compressing these tensors is harder than compressing parameter gradients in data parallelism, where an error only perturbs the update and can be corrected by error feedback (Vogels et al., 2019). In a pipeline, an error at one boundary changes the input of every later layer in the forward pass and the gradient of every earlier layer in the backward pass, errors from successive boundaries compound, and they recur at every update, which is why the effect of activation compression on training must be evaluated separately from its savings (Bian et al., 2024). Exact compression, such as SSNs (Ramasinghe et al., 2025), avoids this altogether, because the partitioned model then computes the same function and gradients as the unpartitioned one.

SSNs. Let d be the residual width, $k \ll d$ the rank of the compressed activations, and $U \in \mathbb{R}^{d\times k}$ an orthonormal basis with $U^\top U = I_k$. An SSN makes compression exact by constraining every attention and MLP output projection to write into the span of U (Ramasinghe et al., 2025). Residual addition then keeps the residual stream $h_t \in \mathbb{R}^{1\times d}$ of token t in the form $h_t = e_t + z_t U^\top$ with $z_t \in \mathbb{R}^{1\times k}$, where $e_t \in \mathbb{R}^{1\times d}$ is a fixed embedding of the token and z_t collects the coefficients, in the basis U, of all outputs added so far, which we call the compressed activation of token t. Each stage recomputes e_t from the token identifier, so the forward boundary needs only the k coefficients

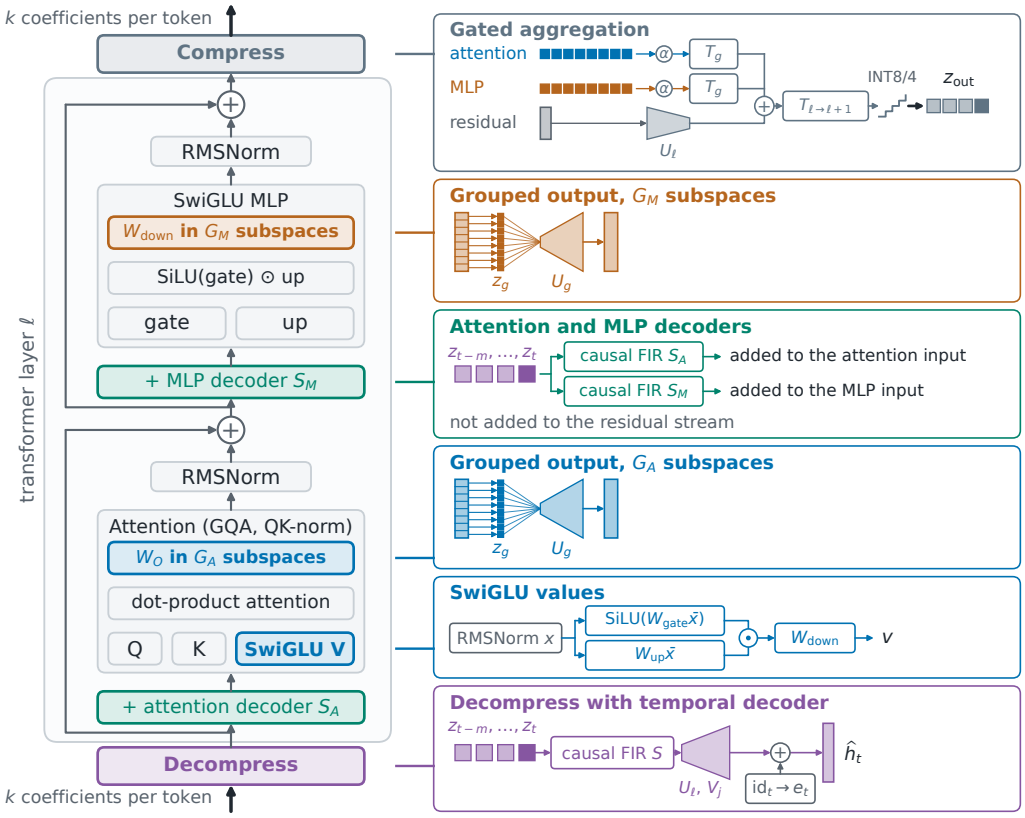


Figure 1: One SSN++ layer between two stage boundaries. Grey is the SSN layer, which normalises each sublayer output with a parameter-free RMSNorm and uses QK-norm, and colour marks the SSN++ additions. The receiving layer decompresses its input in its own basis U_ℓ from the locally reconstructed token embedding, the current compressed activation, and a temporal decoder S over the last m ones (Eq. 6), while separate attention and MLP decoders S_A and S_M, causal filters of the same form, add to the sublayer inputs only. Attention uses a SwiGLU value map, and both output projections write through G_A and G_M group subspaces (Eq. 3). The sending side aggregates three inputs, the gated attention and MLP group outputs and the rest of the residual stream projected onto U_ℓ, and transforms the result into the basis of the next layer (Eq. 5).

z_t. The training loss L depends on the sending stage only through z_t, so the backward boundary needs only $\partial L/\partial z_t = (\partial L/\partial h_t)U$, which is again k numbers per token. Both directions are exact and together cost $4k$ bytes per token. The price is that every output projection of every layer must share the same k directions, which is the constraint that SSN++ relaxes while keeping the exact k-dimensional boundary (Section 3).

Projection and reparametrisation. The original SSN enforces the constraint by projection, storing full output matrices $W \in \mathbb{R}^{d\times p}$ and projecting them back after each optimiser step, $W \leftarrow UU^\top W$. The reparametrised version of SSN (Avraham et al., 2026) instead writes each output projection as $W = UB$ with $B \in \mathbb{R}^{k\times p}$, so that $xW^\top = (xB^\top)U^\top$, and learns B directly. Both describe the same set of weight matrices, since every projected matrix equals UB with $B = U^\top W$ and every UB is unchanged by projection, and under plain gradient descent they follow the same trajectory, because a step on B moves $W = UB$ by $-\eta UU^\top \nabla_W L$, exactly the projected step. The reparametrisation stores kp trainable coefficients instead of dp, a factor of d/k fewer, shrinks the optimiser state by the same factor, and needs no projection, although AdamW then normalises updates in the coordinates of B rather than W. We adopt the reparametrised form, which trains without a loss penalty (Avraham et al., 2026), draw U from the Haar distribution and keep it fixed, factorise the trainable embedding in the same way (Appendix B.1), and spend the saved parameters on richer computation inside each layer.

Measuring expressivity through superposition. A rank-k subspace always has k independent directions, so its rank alone cannot tell how much a constrained layer expresses. Networks represent more features than they have dimensions by superposing sparse features along overlapping directions, at the price of interference that grows with the overlap (Proposition H.3; Elhage et al., 2022). What a rank-k subspace can express therefore depends on how many features it carries at once and how cleanly it separates them, and separate subspaces should improve both. Treating the output direction of each MLP hidden unit as a feature, we measure the first as the number of features active at once, the inverse participation ratio of their per-token energy shares divided by k ($\bar{s}$ in Eq. (59)), and the second as the Welch ratio (W in Eq. (59)), the mean squared overlap of the unit output directions divided by its lower bound in Eq. (57) (Welch, 1974), which is one for the tightest possible packing. For the computation inside a layer, which SSN++ no longer confines to k directions, we track the participation ratio $r_{\mathrm{PR}}(C) = \operatorname{tr}(C)^2/\operatorname{tr}(C^2)$ of an output covariance C, its effective number of directions, and the fraction $\tau_k(C)/\operatorname{tr} C$ of its spectral mass beyond the leading k (Theorem C.3).

Which heads should share a subspace. The same view decides how to divide attention among subspaces. The shared-subspace constraint costs nothing only when the computations use the same leading directions (Theorem C.3), and superposed outputs interfere only when they are active together. For the contributions $z_{h,t}$ of heads h at tokens t, with covariances C_h, we therefore measure both conditions, through a geometric affinity, the normalised overlap of two heads' covariances, and a co-activation affinity, the correlation of their output energies across tokens,

$$A^{\mathrm{geo}}_{hh'} = \frac{\operatorname{tr}(C_h C_{h'})}{\sqrt{\operatorname{tr}(C_h^2)\operatorname{tr}(C_{h'}^2)}}, \qquad A^{\mathrm{co}}_{hh'} = \operatorname{corr}_t\big(\|z_{h,t}\|^2, \|z_{h',t}\|^2\big). \tag{1}$$

Heads with a high geometric affinity can share a subspace at little cost, and grouping heads with a high co-activation affinity leaves different groups less often active together, so that their outputs can be superposed in the one transmitted activation with less interference (Appendix J.3). Spectral clustering (Ng et al., 2001) of either affinity partitions the heads of a layer into G groups. We compare partitions with the adjusted Rand index (ARI; Hubert & Arabie, 1985), which is one for identical partitions and zero in expectation for random ones, and score a consensus partition by its mean ARI with the individual partitions (Appendix I.2). Because overlap cannot tell whether superposed outputs reinforce or cancel, we also measure their signed interactions and intervene on the computation that uses each output (Appendices H and J.2).

3 SSN++

SSN++ keeps the lossless exchange of k coefficients per token between layers and removes the requirement that every output projection share one subspace, adding capacity inside each layer without changing what is communicated (Figure 1 and Appendix B).

3.1 Layer-specific subspaces without loss of information

Low rank does not imply a common orientation. For output projections with update second moments C_j and $\tau_k(C) = \sum_{i>k}\lambda_i(C)$, the sum of all but the k largest eigenvalues $\lambda_i(C)$ of C, the optimal shared subspace incurs an extra expected projection error of exactly $\Gamma_k = \tau_k(\sum_j C_j) - \sum_j \tau_k(C_j) \geq 0$, which vanishes only when all projections admit a common optimal rank-k projector (Theorem C.3), and relaxing the shared-subspace constraint from the whole network to layers, sublayers, and groups can only reduce the best error (Theorem C.1). SSN++ therefore first gives each layer its own subspace, although for random subspaces this benefit is measured rather than implied by Γ_k. A compressed activation must then change basis between layers, and projecting it, which multiplies it by $U_s^\top U_r$, would shrink it by the cosines of the principal angles between the two subspaces. We instead align the two bases with the closed-form solution of the orthogonal Procrustes problem $T_{s\to r} = \arg\min_{T\in O(k)} \|U_s - U_r T^\top\|_F$ (Schönemann, 1966), which for the singular value decomposition $U_s^\top U_r = P\Sigma Q^\top$ is

$$T_{s\to r} = PQ^\top = \operatorname{polar}(U_s^\top U_r), \qquad z_r = z_s T_{s\to r}. \tag{2}$$

This rotation preserves the norms of the compressed activation and its gradient and is exactly invertible, so the change of basis loses no information, and the receiving layer expresses the same coefficients in its own subspace (Appendix B.2, Proposition E.1). In trained 1B models (Appendix I.1),

one subspace per layer makes the output directions of the MLP interfere less, lowering the Welch ratio from 1.33 to 1.25, and the compressed activation carries 1.52 rather than 1.44 simultaneously active features per dimension. The loss stays within 0.01 nats of the SSN, however, because attention and the MLP of each layer still compete for one subspace.

3.2 SUBLAYER AND GROUP SUBSPACES

Separate subspaces $U_{\ell,A}$ and $U_{\ell,M}$ for attention and the MLP of each layer turn this better packing into lower loss. The MLP outputs then pack more tightly than those of an uncompressed model projected to the same rank (Figure 8), removing the direction that attention and the MLP share most strongly costs 0.55 nats instead of about four, and the loss falls by 0.13 nats (Appendix I.1). The MLP also reads the attention output in attention's own subspace, before anything is compressed.

Each sublayer can also split its output projection into groups with subspaces of their own. For sublayer $b \in \{A, M\}$, the input of the output projection is partitioned into G_b groups $x_{b,g}$, and

$$z_{b,g} = x_{b,g} B_{b,g}^\top, \qquad y_b = \sum_{g=1}^{G_b} z_{b,g} U_{b,g}^\top. \tag{3}$$

The factors still hold kp_b coefficients for input width p_b, but the output can reach rank $\min\{d, \sum_g \min(k, p_g)\}$ instead of k (Proposition C.5). This applies to any attention or MLP layer, and only the number of groups must be chosen. For attention, clustering the 32 query heads of a layer by the affinities of Section 2, with no knowledge of the architecture, is most reproducible at eight groups, with a consensus ARI of 0.57, and the consensus partition matches the key–value groups of GQA head for head (Figure 9, Appendix I.2). Heads that retrieve the same values also produce related outputs, so SSN++ uses these eight groups for attention. The MLP has no comparable structure and its outputs show no stable partition, so its eight equal groups are a capacity choice rather than a recovered structure.

The group outputs must still leave the layer as one compressed activation. Their sum contains signed cross terms, so simultaneously active groups can reinforce or cancel one another, and a small nonlinear gate weights each group according to the relative energies of all groups at the current token. For sublayer b with G_b groups, the gate maps the log-energy of each group output relative to the geometric mean over the groups, $u_{b,t} \in \mathbb{R}^{G_b}$, through one tanh layer to a scalar gain per group,

$$\alpha_{b,g,t} = 1 + \delta \tanh\Big(\beta_{b,g} + \sum_{h=1}^{G_b} A_{b,gh}\, u_{b,h,t}\Big), \qquad u_{b,g,t} = \log \frac{\|z_{b,g,t}\|_2 + \epsilon}{\big(\prod_{h=1}^{G_b}(\|z_{b,h,t}\|_2 + \epsilon)\big)^{1/G_b}}, \tag{4}$$

with learned $A_b \in \mathbb{R}^{G_b \times G_b}$ and $\beta_b \in \mathbb{R}^{G_b}$, $\epsilon = 10^{-5}$, and $\delta = 1$, so every gain lies in $(0, 2)$. Zero initialisation of A_b and β_b starts every gain at one, so the aggregation begins as a plain sum. Each $\alpha_{b,g,t}$ multiplies the whole k-dimensional output $z_{b,g,t}$ of its group, and with $G_A = G_M = 8$ the gates add $2G(G+1) = 144$ parameters per layer, or 2016 in the 1B model. Let U_ℓ be the basis in which layer ℓ receives its input. If c_t is the residual stream projected onto U_ℓ after removing the layer's own outputs and the fixed embedding, the transmitted activation is

$$z_{\text{out}} = \Big(c_t + \sum_{b,g} \alpha_{b,g,t} z_{b,g,t} T_{b,g}\Big) T_{\ell \to \ell+1}, \tag{5}$$

where $T_{b,g}$ transforms group g of sublayer b into U_ℓ and $T_{\ell\to\ell+1}$ transforms the result into the basis of the next layer, both as in Eq. (2). Unlike a change of basis, this gated aggregation is many-to-one, and it is where each layer selects which part of its richer computation to pass on (Appendix B.4).

3.3 RICHER INPUTS AT THE RECEIVING LAYER

Expanding the received activation only as $z_t U_\ell^\top$ would confine the communicated part of the next layer's input to one k-dimensional image. The compressed activations of preceding tokens have already crossed the same boundary and provide further coefficients without extra communication, and a temporal decoder turns them into additional features,

$$\widehat{y}_t = z_t U_\ell^\top + S(H_t), \qquad S(H_t) = \sum_{j=0}^{m} z_{t-j} G_j^\top V_j^\top, \tag{6}$$

where $H_t = (z_t, \ldots, z_{t-m})$ collects the compressed activations of the current token and the m preceding ones, with $m = 8$, G_j is learned, and V_j is a fixed orthonormal basis. This causal filter raises the attainable rank of the decoded inputs from k to $(m + 1)k$ (Proposition D.1). Attention and MLP decoders S_A and S_M of the same form add to the two sublayer inputs without entering the residual stream, so each sublayer can read the same history differently (Appendices B.5 and D). The parameters saved by the reparametrisation also pay for a SwiGLU value map, the gated form of the model's own MLPs (Shazeer, 2020), which removes the input-rank ceiling of a linear value projection while queries and keys stay linear, and more than doubles the stable rank of the values, from 7.6 to 16.9 (Appendices B.6, I.3, and J.5).

3.4 Higher rank at the same communication volume

The compressed activations are the coefficients of each output in a random orthonormal basis, which spreads large hidden channels over all k coefficients in the way rotation-based quantisers do (Ashkboos et al., 2024b), and SSNs avoid a known source of outliers (Bondarenko et al., 2023), since their random subspaces rule out massive activations by construction (Appendix J.5). SSN activations should therefore tolerate low precision, so we quantise the communicated activations and gradients with per-token integer scales, while all matrix multiplications stay in floating point. Rounding makes the communicated values inexact, but its measured effect on the loss is negligible (Section 4.3). With one two-byte scale per quantised direction, the forward-plus-backward volume per token and stage boundary is $4k$ bytes in BF16, $2k+4$ with INT8, and $k+\lceil k/2 \rceil+4$ with INT8 activations and INT4 gradients, so INT8/INT4 carries rank $k = 104$ in the 160 bytes that BF16 needs for $k = 40$, which raises capacity once more without increasing communication (Appendix B.7).

3.5 What the theory establishes

Beyond the results cited in Sections 3.1 to 3.4, the appendices show that a fixed random subspace loses little to an optimised one when the target spectrum is nearly flat (Propositions C.6 and C.7), that separate attention and MLP decoders improve on a shared one by the disagreement between the targets of the two sublayers (Appendix D.3), and that architectural and quantisation errors propagate through depth as a sum of per-boundary errors amplified by later layers (Theorem E.3), which sets how rank trades against precision at a fixed byte budget. Together, these results show that in theory SSN++ has a much larger capacity than an SSN with one shared subspace at equal communication, and Section 4 and Appendix J.2 measure how much of it is reachable during training.

4 Experiments and Results

4.1 Evaluation setting

We train Llama-3-family models (Grattafiori et al., 2024) on FineWeb-Edu (Penedo et al., 2024) and validate on C4 (Raffel et al., 2020), with GPT-2 tokenisation and length-1024 sequences (Appendix F.1). The 1B budget of 20.97B tokens is approximately compute-optimal under the 20-token-per-parameter heuristic (Hoffmann et al., 2022), whereas 229M is trained much longer.

We compare SSN++ with the SSN of Section 2, with a parameter-matched SSN that has the same SwiGLU values and embedding factorisation as SSN++ and differs from it in parameter count by less than 0.06% (Appendix F.1), with MAPL and ResBM implemented in the same backbone (Appendix F.2), and with uncompressed transformers, labelled Dense in tables and figures. All compressed models keep one dense final layer and communicate about 50 times fewer bytes per token than the dense residual stream at every stage boundary (Tables 1 and 4). We report cross-entropy losses on FineWeb-Edu training data and C4 validation data (Appendix F.3) and the mean eval accuracy over four tasks from the Language Model Evaluation Harness (Biderman et al., 2024). Training runs each model whole on eight data-parallel GPUs, so the compressed activations pass between transformer layers within a GPU exactly as they would cross stage boundaries, and throughput measures local computational cost (Appendix F.4). Section 4.4 also times every model as a pipeline whose stage boundaries are 200 Mb/s links (Appendix F.5).

4.2 CAPACITY AT THE SAME COMMUNICATION VOLUME

SSN++ lowers both losses at all three sizes (Table 1, Figure 2). With BF16 communication at the same rank as the parameter-matched SSN, it reduces C4 validation loss by 0.252, 0.128, and 0.116 nats at 229M, 1B, and 2B, and FineWeb-Edu training loss by 0.234, 0.123, and 0.111 nats. At 1B, this closes 33.0% of the matched SSN's validation-loss gap to the uncompressed transformer, so the rank of the communicated activations alone does not explain the loss of SSNs. At every size, SSN++ is ahead of the matched SSN at every validation evaluation after update 500, and SSN++ INT8/INT4 is ahead after the first.

Table 1: Cross-entropy loss (nats/token) on FineWeb-Edu training data (mean of the final 500 updates) and C4 validation data (mean of the final six evaluations), with the forward-plus-backward bytes per token that cross each stage boundary. Every compressed model communicates roughly 50 times fewer bytes than the dense transformer. SSN++ uses ranks 20, 40, and 50 with BF16, 40, 80, and 100 with INT8, and 52, 104, and 130 with INT8/INT4 communication.

	229M (24.12B tokens)			1B (20.97B tokens)			2B (39.85B tokens)		
Model	Bytes	FineWeb	C4	Bytes	FineWeb	C4	Bytes	FineWeb	C4
Dense	4096	2.9311	3.4126	8192	2.6428	3.1227	10240	2.4487	2.9437
SSN	80	3.5196	3.9964	160	3.1264	3.5980	204	2.9168	3.3998
SSN, matched params	80	3.4020	3.8865	160	3.0321	3.5102	204	2.8336	3.3269
ResBM	80	3.3944	3.8609	160	3.0971	3.5699	200	2.8894	3.3760
MAPL	80	3.2404	3.6970	160	3.0902	3.5490	200	2.9089	3.3797
SSN++	80	3.1680	3.6345	160	2.9089	3.3824	200	2.7231	3.2110
SSN++ INT8	84	3.0698	3.5378	164	2.8157	3.2898	204	2.6361	3.1252
SSN++ INT8/INT4	82	**3.0459**	**3.5167**	160	**2.7915**	**3.2677**	199	**2.6159**	**3.1060**

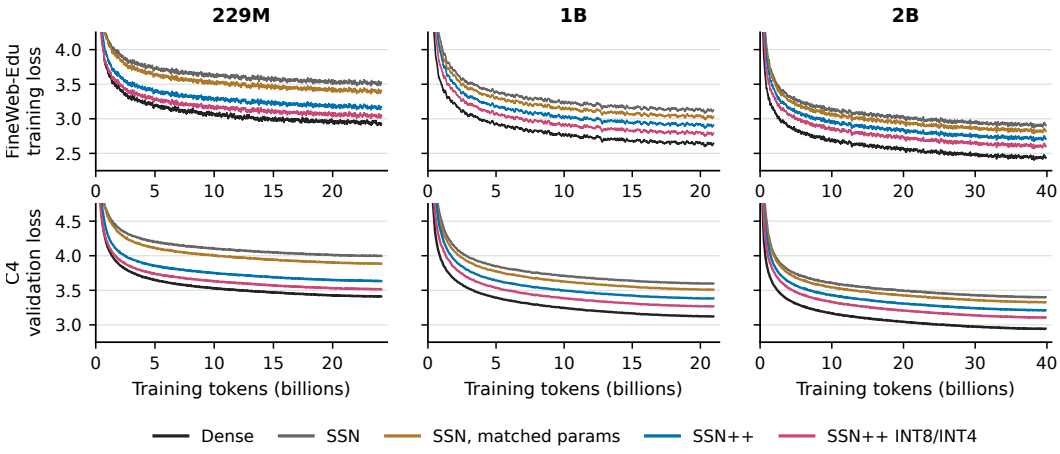


Figure 2: FineWeb-Edu training loss (top), as a mean over the preceding 100 updates, and C4 validation loss (bottom), on loss axes cut to show the final differences.

MAPL and ResBM, the two other methods that compress the activations exchanged between stages, trail SSN++ by 0.06 to 0.23 nats of validation loss as whole methods implemented in our backbone (Appendix F.2).

4.3 HIGHER RANK AT THE SAME COMMUNICATION VOLUME

At fixed rank $k = 40$, the 10,000-update 1B ablations show validation-loss increases of only 0.0023 nats with INT8 in both directions and 0.0063 nats with INT8 activations and INT4 gradients (Table 8, Appendix G.2). Quantisation is therefore essentially free at these precisions, without specialised objectives or low-bit matrix multiplication, and the quantised gradients stay aligned with the exact ones throughout training (Appendix J.6). Spending the saved bytes on rank completes SSN++. At 1B,

SSN++ INT8/INT4 at $k = 104$ costs the same 160 bytes and reaches validation loss 3.2677, 0.242 nats below the matched SSN, which closes 62.6% of its gap to the uncompressed transformer with 1.093B trainable parameters against 1.077B. Across the three sizes, the higher rank lowers validation loss by a further 0.118, 0.115, and 0.105 nats relative to SSN++ with BF16 communication, at 82, 160, and 199 bytes against 80, 160, and 200, gains that include the parameters higher ranks add (Table 5).

4.4 Transfer and computational cost

Lower loss is accompanied by higher mean eval accuracy (Table 2). SSN++ improves on the matched SSN by 2.43, 1.94, and 2.20 percentage points with BF16 communication and by 3.74, 3.97, and 4.29 points as SSN++ INT8/INT4, with gains that vary by task (Appendix G.1).

Table 2: Mean eval accuracy (%) on ARC-Easy, ARC-Challenge, PIQA, and HellaSwag.

Model	229M	1B	2B
Dense	43.68	49.99	56.76
SSN	37.88	41.67	43.98
SSN, matched params	38.53	42.20	45.94
ResBM	38.48	41.78	44.93
MAPL	39.73	41.61	44.16
SSN++	40.96	44.14	48.14
SSN++ INT8	42.14	46.06	49.01
SSN++ INT8/INT4	**42.27**	**46.17**	**50.23**

Table 3: Measured 1B seconds per update on one node of eight H100s and over 200 Mb/s pipeline links (Appendix F.5).

Model	One node	200 Mb/s
Dense	2.97	486.1
SSN	3.26	18.6
SSN, matched params	4.14	19.1
ResBM	3.14	18.4
MAPL	3.43	18.5
SSN++	4.97	19.6
SSN++ INT8	5.15	20.0
SSN++ INT8/INT4	5.27	20.3

The additional computation inside each layer has a cost on one node (Table 3), but it pays for itself even there, since by the time each compressed baseline finishes its budget, SSN++ has already reached a lower validation loss, by 0.04 to 0.33 nats at 229M, 0.11 to 0.16 nats at 1B, and 0.08 to 0.12 nats at 2B, although the uncompressed model remains both faster and more accurate on one node (Figures 4–6, Appendix F.5).

Training time under a 200 Mb/s pipeline. We also ran the 1B models as a 15-stage pipeline across two nodes, with a cut at each of the 14 stage boundaries, the one-forward-one-backward (1F1B) schedule (Narayanan et al., 2019), and 32 microbatches per update, where every stage boundary is a 200 Mb/s link with 50 ms latency that carries activations and gradients at the volumes of Section 3.4 (Appendix F.5), with steady update times in Table 3. The 8192-byte residual stream of the uncompressed model makes each update 25 times slower than SSN++, so the 20,000-update budget takes 2,701 hours against 109, and SSN++ INT8/INT4 at $k = 104$ trains in 113 hours at the same 160 bytes. Among compressed models, the additional computation of SSN++ costs 2.6% per update relative to the matched SSN.[1] At this bandwidth, communication rather than local arithmetic sets the training time.

4.5 How SSN++ uses the added capacity

For ablations we train 1B models for 10,000 updates. Adding to the SSN the layer and sublayer subspaces with grouped attention, grouped MLP outputs with aggregation gates, the temporal decoder with attention and MLP decoders, and SwiGLU values with a factorised embedding lowers the validation loss at every step, from 3.6941 to 3.4976 overall (Appendix G.3 lists the steps, some of which combine components).

The learned representations increasingly use the added capacity (Figure 3). Between updates 1000 and 10,000, the participation ratio of the grouped outputs of SSN++ at $k = 40$ rises from 34.6 to 57.4, above the transmitted rank, compared with 20.0 to 22.3 for the matched SSN, and their variance beyond rank 40 grows from 27.7% to 37.2% while remaining zero in the SSN. The activation

[1] We wrote no specialised kernels for SSN++. Quantisation, for example, runs as separate floating-point scaling and rounding operations rather than fused low-bit kernels, so this overhead could be reduced further.

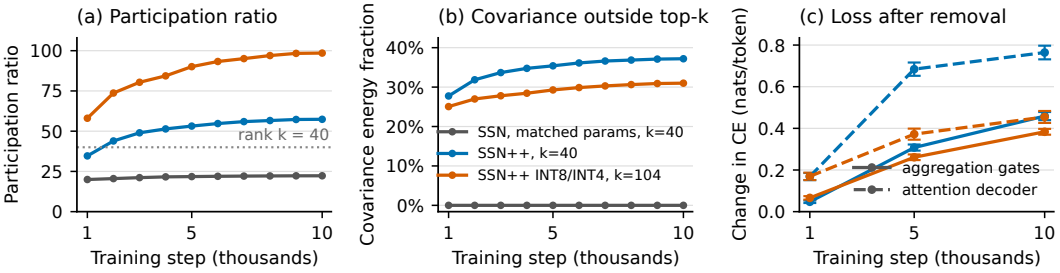


Figure 3: Representation dynamics in the 10,000-update 1B ablations. Over training, SSN++ uses more directions beyond rank k and relies more on its gates and attention decoder. (a) Participation ratio of the grouped outputs, with the rank $k = 40$ dotted, and (b) their covariance energy outside the best rank-k subspace. (c) Loss increase when the aggregation gates are deactivated (solid) or the attention decoder is removed (dashed), with standard errors over 32 sequences as bars.

received by the next layer keeps a nearly constant effective rank, 34.2 to 33.1, so the added capacity accumulates inside the layers rather than in communication (Appendix J.2), and the aggregation becomes more selective as the mean gate falls from 0.93 to 0.68 and the transmitted share of the output energy from 0.98 to 0.54 (Appendix J.3). Setting the gates to neutral raises the loss by 0.047 nats initially and 0.459 finally, and removing the attention decoder raises it by 0.169 and 0.764 nats, which measures reliance on these computations rather than their value in training (Appendix J.7).

5 Discussion

The loss gap of SSNs therefore comes less from the rank of the communicated activations than from how capacity is organised around them, since giving each computation its own subspace lowers the loss at nearly equal parameter counts and the same communication volume. The pipeline measurement also changes which resource the trade-off concerns. On one node the uncompressed model is both faster and more accurate, but over 200 Mb/s links the bytes per token set the training time, so the extra computation of SSN++ costs little and trading precision for rank at a fixed volume costs nothing in communication.

Our results cover one architectural family, dense Llama-style decoders with grouped-query attention and SwiGLU MLPs from 229M to 2B parameters. Hybrid architectures, other attention mechanisms, and mixture-of-experts layers are left to future work. The idea of separate subspaces should carry over to them, although where and in what form it helps most remains to be tested. Learned subspaces are a natural extension, but in preliminary experiments they gave no loss benefit over fixed subspaces at equal rank. After recent prior work on large-scale SSN training (Avraham et al., 2026), fixed subspaces are preferred, and the approximation theory bounds their error directly (Appendix C.3). Two directions appear more promising. Ranks need not be equal across depth, since the projection penalties and the use of local directions vary with layer position, and a deployment over real wide-area links between clusters, as in Avraham et al. (2026), would test what the fixed-rate links only approximate.

6 Conclusion

SSNs showed that pipeline-parallel training can communicate a few coefficients per token without loss, at the price of a shared subspace that limits capacity. SSN++ keeps that lossless communication at the same volume and relaxes the shared subspace in deliberate steps, each following from measurements of rank and superposition, and at every model size studied it lowers training and validation loss and improves eval accuracy, closing 63% of the gap between a parameter-matched SSN and the uncompressed transformer at 1B. Communication-efficient training therefore need not give up as much capacity as a single shared subspace implies. More broadly, this work aims to let large models be trained on compute spread across many sites, by bringing low-bandwidth pipeline training closer to the quality of a single data centre.

Reproducibility statement

Appendix B defines every component of SSN++, and Appendices C, D, and E state the assumptions and complete proofs of the theoretical results. The model configurations, parameter counts, and communication volumes are in Appendix F.1 and Tables 4 and 5, the MAPL and ResBM implementations in Appendix F.2, and the data, training budgets, optimiser, evaluation protocol, and aggregation of the reported losses in Appendix F.3. Appendices F.4 and F.5 describe the byte accounting and the timing protocols. The populations, statistics, and definitions behind the representation analyses and the training-dynamics measurements are in Appendices H, I, and J.1.

AI use statement

We used generative AI tools to draft and edit parts of the manuscript. We reviewed all AI-assisted text and take responsibility for the final content of this work.

References

Alan Aboudib, Rodrigo Lopez Portillo A., Kalei Brady, and Steffen Cruz. ResBM: Residual bottleneck models for low-bandwidth pipeline parallelism, 2026. URL https://arxiv.org/abs/2604.11947.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/6c340f25839e6acdc73414517203f5f0-Abstract.html.

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*, 2024a. URL https://arxiv.org/abs/2401.15024.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b5b939436789f76f08b9d0da5e81af7c-Abstract-Conference.html.

Gil Avraham, Violetta Shevchenko, Hadi Mohaghegh Dolatabadi, Karol Pajak, James Snewin, Harry Xi, Rodney O'Donnell, Thalaiyasingam Ajanthan, Sameera Ramasinghe, Chamin Hewa Koneputugodage, Shamane Siriwardhana, and Alexander Long. Agora: Collective and permissionless internet-scale pretraining of large language models, 2026. URL https://arxiv.org/abs/2607.13332.

Song Bian, Dacheng Li, Hongyi Wang, Eric P. Xing, and Shivaram Venkataraman. Does compressing activations help model parallel training? In *Proceedings of Machine Learning and Systems*, volume 6, pp. 239–252, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/71381211d0abef73ed1887b83c4547b1-Abstract-Conference.html.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, et al. Lessons from the trenches on reproducible evaluation of language models, 2024. URL https://arxiv.org/abs/2405.14782.

M. Sh. Birman and M. Z. Solomyak. Piecewise-polynomial approximations of functions of the classes W_p^α. *Mathematics of the USSR-Sbornik*, 2(3):295–317, 1967. doi: 10.1070/SM1967v002n03ABEH002343. URL https://www.mathnet.ru/eng/sm4132.

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*, 2020. URL https://arxiv.org/abs/1911.11641.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/edbcb7583fd8921dad78adecfe06a99b-Abstract-Conference.html.

Patrick Chen, Hsiang-Fu Yu, Inderjit S. Dhillon, and Cho-Jui Hsieh. DRONE: Data-aware low-rank compression for large NLP models. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f56de5ef149cf0aedcc8f4797031e229-Abstract.html.

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, 2018. URL https://arxiv.org/abs/1803.05457.

Albert Cohen, Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Optimal stable nonlinear approximation. *Foundations of Computational Mathematics*, 22(3):607–648, 2022. doi: 10.1007/s10208-021-09494-z. URL https://link.springer.com/article/10.1007/s10208-021-09494-z.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 933–941. PMLR, 2017. URL https://proceedings.mlr.press/v70/dauphin17a.html.

Ronald A. DeVore, Ralph Howard, and Charles Micchelli. Optimal nonlinear approximation. *Manuscripta Mathematica*, 63(4):469–478, 1989. doi: 10.1007/BF01171759. URL https://link.springer.com/article/10.1007/BF01171759.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Ky Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949. doi: 10.1073/pnas.35.11.652.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Mikhael Gromov. Entropy, homology and semialgebraic geometry. In *Séminaire Bourbaki, Vol. 1985/86*, volume 145–146 of *Astérisque*, pp. 225–240. 1987.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace, 2018. URL https://arxiv.org/abs/1812.04754.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2203.15556.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. GPipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075.

Paul Janson, Edouard Oyallon, and Eugene Belilovsky. Learned subspace compression for communication-efficient pipeline parallelism, 2026. URL https://arxiv.org/abs/2606.05484.

A. Kolmogoroff. Über die beste annäherung von funktionen einer gegebenen funktionenklasse. *Annals of Mathematics*, 37(1):107–110, 1936. doi: 10.2307/1968691. URL https://www.jstor.org/stable/1968691.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

V. A. Marchenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967. doi: 10.1070/SM1967v001n04ABEH001994.

Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R. Devanur, Gregory R. Ganger, Phillip B. Gibbons, and Matei Zaharia. PipeDream: Generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, pp. 1–15, 2019. doi: 10.1145/3341301.3359646.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pp. 849–856, 2001.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.17557.

Jonathan Pila and Alex J. Wilkie. The rational points of a definable set. *Duke Mathematical Journal*, 133(3):591–616, 2006.

Allan Pinkus. *n-Widths in Approximation Theory*, volume 7 of *Ergebnisse der Mathematik und ihrer Grenzgebiete, 3. Folge*. Springer, Berlin, Heidelberg, 1985. doi: 10.1007/978-3-642-69894-1. URL https://link.springer.com/book/10.1007/978-3-642-69894-1.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/904e89bb4e632e75fb47f093b620b257-Abstract-Conference.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://www.jmlr.org/papers/v21/20-074.html.

Sahar Rajabi, Nayeema Nonta, and Sirisha Rambhatla. Geometrically principled randomized optimization for efficient LLM training, 2026a. URL https://arxiv.org/abs/2510.01878v2. Version 2.

Sahar Rajabi, Nayeema Nonta, and Sirisha Rambhatla. Principled randomized exploration of gradient subspaces for efficient LLM training. In *ICLR Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026b. URL https://openreview.net/forum?id=vhd6SMnkI6.

Sahar Rajabi, Nayeema Nonta, and Sirisha Rambhatla. Random but right: A geometric explanation for efficient LLM training. In *ICLR Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2026c. URL https://openreview.net/forum?id=79FGd2bYnJ.

Sameera Ramasinghe, Thalaiyasingam Ajanthan, Gil Avraham, Yan Zuo, and Alexander Long. Subspace networks: Scaling decentralized training with communication-efficient model parallelism, 2025. URL https://arxiv.org/abs/2506.01260v3. Version 3; originally released as Protocol Models.

Max Ryabinin, Tim Dettmers, Michael Diskin, and Alexander Borzunov. SWARM parallelism: Training large models can be surprisingly communication-efficient. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29416–29440, 2023. URL https://proceedings.mlr.press/v202/ryabinin23a.html.

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451.

Noam Shazeer. GLU variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

David R. So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. Searching for efficient transformers for language modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2f3c6a4cd8af177f6456e7e51a916ff3-Abstract.html.

Noel Thomas. No subspace to track: Non-identifiability and optimizer state in low-rank training, 2026. URL https://arxiv.org/abs/2607.05872v2. Version 2.

Lou van den Dries. *Tame Topology and O-minimal Structures*, volume 248 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1998.

Lou van den Dries, Angus Macintyre, and David Marker. The elementary theory of restricted analytic fields with exponentiation. *Annals of Mathematics*, 140(1):183–205, 1994.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d9fbed9da256e344c1fa46bb46c34c5f-Abstract.html.

Hongyi Wang, Saurabh Agarwal, and Dimitris Papailiopoulos. Pufferfish: Communication-efficient models at no extra cost. In *Proceedings of Machine Learning and Systems*, volume 3, pp. 365–386, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/94cb28874a503f34b3c4a41bddcea2bd-Abstract.html.

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*, 2025. URL https://arxiv.org/abs/2403.07378v5.

L. R. Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information Theory*, 20(3):397–399, 1974. doi: 10.1109/TIT.1974.1055219.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.17453.

Yosef Yomdin. C^k-resolution of semialgebraic mappings. Addendum to: Volume growth and entropy. *Israel Journal of Mathematics*, 57(3):301–317, 1987.

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, 2025. URL https://arxiv.org/abs/2312.05821v5. Version 5; originally submitted in 2023.

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*, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Shiliang Zhang, Hui Jiang, Shifu Xiong, Si Wei, and Li-Rong Dai. Compact feedforward sequential memory networks for large vocabulary continuous speech recognition. In *Proceedings of Interspeech*, pp. 3389–3393, 2016. doi: 10.21437/Interspeech.2016-121. URL https://www.isca-archive.org/interspeech_2016/zhang16i_interspeech.html.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61121–61143. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhao24s.html.

ORGANISATION OF THE APPENDICES

Appendix A extends the related work, and Appendix B gives the implementation details of SSN++. Appendices C, D, and E develop the theory of grouped output projections, of decoding from preceding tokens, and of the stage boundary, each followed by its proofs. Appendices F and G give the experimental setup and the additional results. Appendices H, I, and J interpret the superposition measurements, report the measurements behind the design choices, and follow SSN++ through training.

A EXTENDED RELATED WORK

Communication-efficient training. GPipe improves pipeline utilisation through microbatch scheduling (Huang et al., 2019), while SWARM constructs rebalanced pipelines for heterogeneous, unreliable devices (Ryabinin et al., 2023). Communication compression addresses a complementary constraint. QSGD quantises stochastic gradients (Alistarh et al., 2017), and PowerSGD uses low-rank gradient factors compatible with all-reduce (Vogels et al., 2019). Pufferfish trains factorised networks to reduce distributed-training costs (Wang et al., 2021). These approaches principally concern parameter-gradient exchange. Bian et al. (2024) study activation compression in model parallelism and show why its costs and effects require separate evaluation. Subspace networks (SSNs) instead constrain the attention and MLP output projections to a common subspace, which enables exact transmission of compressed activations within that architecture (Ramasinghe et al., 2025). The reparametrised version of SSN (Avraham et al., 2026) parametrises the constrained maps directly, reducing the trainable state while keeping the constraint inside the computation graph. We adopt this formulation and study how the remaining parameter and communication budgets can support richer local representations. Our integer formats quantise only the communication across stage boundaries, while matrix multiplications remain in floating point.

ResBM learns nonlinear encoder–decoder bottlenecks across pipeline boundaries while retaining a low-dimensional residual identity path (Aboudib et al., 2026). MAPL learns an orthonormal projection at each boundary using Stiefel-constrained optimisation and reconstructs activations with factorised token embeddings (Janson et al., 2026). We compare adaptations of these boundary constructions within the same Llama-3 backbone. Our MAPL baseline uses the original architecture without its optional vector-quantisation extension, which combines grouped residual vector quantisation with per-stage codebooks and streaming dictionary synchronisation. SSN++ instead quantises the compressed activations and their gradients with per-token scalar quantisers that need no codebook.

Rotations and integer quantisation. QuaRot uses randomised Hadamard rotations to spread large activation coordinates and make integer quantisation more accurate while preserving the full-precision computation (Ashkboos et al., 2024b). Its objective is low-bit inference, including weights, activations, and the key–value cache. In an SSN, the orthonormal basis U already mixes low-dimensional updates across hidden channels while preserving their norms. SSN++ additionally applies orthogonal transforms to compressed activations that pass between its subspaces. Only the compressed activations and their gradients are quantised, while local matrix multiplications remain in floating point. This provides a connection between the choice of representation coordinates and the precision needed to communicate them.

Low-rank weights, activations, and updates. Low-dimensional structure has been studied in several distinct objects. Gur-Ari et al. (2018) observe gradient concentration in leading Hessian eigenspaces in classification networks. GaLore uses periodically updated low-rank gradient projections while learning full-parameter models (Zhao et al., 2024). LoRA parameterises low-rank changes to frozen pretrained weights (Hu et al., 2022), and AdaLoRA distributes adaptation capacity across matrices according to importance (Zhang et al., 2023). These results motivate structured optimisation, but do not imply that complete weights or residual activations share one subspace across depth.

For model compression, DRONE uses data-aware matrix decomposition to preserve layer outputs (Chen et al., 2021), and SVD-LLM relates truncation after whitening to activation reconstruction error (Wang et al., 2025). ASVD incorporates activation statistics and matrix-specific compres-

sion sensitivity (Yuan et al., 2025). Most directly, SliceGPT rotates the signals at successive attention and feed-forward sublayers into their own principal coordinates before removing low-energy directions (Ashkboos et al., 2024a). Together, these studies motivate approximation spaces adapted to each local computation. They concern post-training compression, whereas we study learning with separate output subspaces and fixed-rank communication across stage boundaries.

Approximation and representation capacity. Classical low-rank approximation and spectral variational principles (Eckart & Young, 1936; Fan, 1949) underlie the projection objectives used here. Kolmogorov widths characterise the best uniform approximation of a set by a subspace of prescribed dimension (Kolmogoroff, 1936; Pinkus, 1985). Their row-space restriction captures the requirement that every token share the same output subspace, which makes the difference between a shared subspace and separate subspaces an approximation-theoretic question. Polynomial constructions for smooth function classes (Birman & Solomyak, 1967) motivate the smooth-image bound in our appendix, with separate assumptions for matrix-valued target sets. Nonlinear widths extend approximation beyond fixed linear spaces (DeVore et al., 1989), while stable manifold widths impose regularity on the encoder and decoder (Cohen et al., 2022). These theories provide context for richer local decoding, whereas the guarantees here concern the specific causal-filter class implemented by the temporal decoder of SSN++.

Superposition and feature organisation. Toy models show how sparse features can occupy overlapping directions and produce polysemantic representations (Elhage et al., 2022). Classical coherence bounds constrain the packing of unit vectors in a finite-dimensional space (Welch, 1974). In SSN and SSN++, the normalised columns of the output projections provide accessible directions for such geometric measurements, but those parameters are not automatically semantic features. We combine overlap and the number of active features per dimension with signed moments and downstream sensitivity to distinguish useful superposition from harmful interference.

Temporal filtering and local features. Compact feedforward sequential memory networks combine low-rank projections with learned filters over neighbouring hidden states for speech recognition (Zhang et al., 2016). Gated convolutional networks establish finite causal contexts as an effective basis for language modelling (Dauphin et al., 2017), and Primer combines short convolutions with transformer attention (So et al., 2021). These works motivate local temporal features alongside a compact representation or an attention mechanism. In our setting, the filtered inputs are compressed activations that have already crossed a stage boundary. The temporal decoder and the attention and MLP decoders use separate subspaces to map the causal history of these compressed activations into residual-stream, attention, and MLP features, while the number of transmitted coefficients per token stays fixed. This use of filtering complements the separation of output subspaces with richer representations in the receiving layer.

Attention sinks and nonlinear values. StreamingLLM identifies disproportionate attention to initial tokens and uses their retained key–value states to support streaming inference (Xiao et al., 2024). Bondarenko et al. (2023) connect activation outliers to attention heads that suppress residual updates by concentrating attention on tokens whose value vectors have small magnitudes. Their clipped-softmax and gated-attention variants reduce outliers and improve integer quantisation. This links attention behaviour to numerical precision, although attention concentration and coordinate outliers are distinct quantities. Gated-attention experiments distinguish the benefit of nonlinear processing from query-dependent sparse gating after scaled dot-product attention, linking the latter to reduced attention sinks (Qiu et al., 2025). SwiGLU was introduced as a gated feed-forward transformation (Shazeer, 2020). Our SwiGLU values place this transformation in the value path before attention aggregation. They therefore change the available value features at a different location from post-attention output gating. In our measurements attention sinks are already weak in SSNs with linear values, so the SwiGLU values do not act by removing sinks (Appendix J.5).

B Architectural details

This appendix records the implementation details of SSN++ that Section 3 leaves out. Appendix B.1 describes the SSN layer that SSN++ builds on, Appendix B.2 the fixed bases and the orthogonal transforms between them, and Appendices B.3 and B.4 the grouped output projections and the gated

aggregation. Appendices B.5, B.6, and B.7 give the decoders, the SwiGLU values, and the quantisers.

B.1 THE SSN LAYER

Every compressed layer normalises the output of each sublayer rather than its input. Attention and the MLP read the residual stream directly, their outputs pass through a parameter-free RMSNorm before the residual addition, and attention applies a parameter-free RMSNorm to every query and key head (QK-norm). All norms use $\epsilon = 10^{-5}$. Normalising an output scales the d-dimensional output of a sublayer and the coefficients of its groups by the same factor, so the coefficients that enter the gated aggregation stay consistent with the normalised output. The trainable token embedding is factorised as $EU_e^\top$ with $E \in \mathbb{R}^{V\times k}$ for vocabulary size V and its own fixed basis U_e, and its coefficients are mapped into the subspace of the first layer by an orthogonal transform before the fixed embedding e_t is added. The final layer is dense and keeps linear values, and the last compressed layer passes its full residual stream to it without aggregation.

B.2 BASES AND ORTHOGONAL TRANSFORMS

All bases are drawn at random from the Haar distribution and stay fixed during training. Each layer ℓ has a backbone basis U_ℓ in which it receives and transmits its compressed activation, a separate basis for the output of each group of each sublayer, a basis for every lag of each decoder, and the embedding its own basis U_e. The group bases $U_{b,g}$ are mapped into U_ℓ by $T_{b,g} = \mathrm{polar}(U_{b,g}^\top U_\ell)$, and the aggregated activation is mapped into the backbone basis of the next layer by $T_{\ell\to\ell+1} = \mathrm{polar}(U_\ell^\top U_{\ell+1})$. Because the bases are fixed, every transform is computed once before training. At equal ranks each transform is orthogonal, so the backward pass multiplies gradients by its transpose and preserves their norm as well. Appendix E bounds how far the d-dimensional vectors that the same coefficients expand to in two subspaces can differ.

B.3 GROUPED OUTPUT PROJECTIONS

Attention has eight key–value heads at every size, so there are $G_A = 8$ groups, and group g collects the outputs of the query heads that read key–value head g, which are four of the 32 query heads at 1B and two of the 16 at 229M. The MLP hidden features are split into $G_M = 8$ blocks of equal size. Group g of sublayer b has its own reduced factor $B_{b,g}$ and basis $U_{b,g}$ as in Eq. (3), so the grouped projection holds the same kp_b trainable coefficients as an ungrouped one, and $G_A = G_M = 1$ recovers one attention subspace and one MLP subspace per layer. The MLP receives the full d-dimensional attention output, including every group, before the attention groups are aggregated for transmission.

B.4 GATED AGGREGATION

The gates of Eq. (4) are computed in FP32 precision from the coefficients of each group after output normalisation. Their input is the log-energy of each group relative to the geometric mean over the groups of the same sublayer, so the gates respond to the relative energies of the groups rather than to their common scale. Gating changes only what is transmitted, not the d-dimensional output that the MLP and the residual stream receive, which lets the network separate a feature's local use from its importance to the next layer. With Y the layer output, e the fixed embedding, and y_A and y_M the two sublayer outputs, the transmitted activation of Eq. (5) is

$$z_{\mathrm{out}} = \Big[(Y - y_A - y_M - e)\,U_\ell + \sum_{b\in\{A,M\}}\sum_{g} \alpha_{b,g} z_{b,g} T_{b,g}\Big] T_{\ell\to\ell+1}. \tag{7}$$

Subtracting the two sublayer outputs before projecting the residual stream onto U_ℓ avoids projecting them a second time, because their group coefficients enter through the gated sum instead. Gates exist only in the compressed layers that transmit.

B.5 DECODERS

The temporal decoder of Eq. (6) and the attention and MLP decoders use $m = 8$ lags, so each holds nine learned matrices $G_j \in \mathbb{R}^{k\times k}$ with fixed bases $V_j \in \mathbb{R}^{d\times k}$, the three decoders adding $3 \cdot 9k^2$ parameters per receiving layer. All G_j start at zero, so every layer initially uses the direct expansion $z_t U_\ell^\top$ and learns how much of the history to add. Positions before the start of a sequence are zero-padded. In a streaming implementation the receiving layer keeps the compressed activations of the preceding m tokens, and during sequence training they are already present in the received tensor. The bases V_j need not be orthogonal to U_ℓ, so the decoders can both modify existing directions and add new ones. The attention decoder adds $S_A(H_t)$ to the input of attention and the MLP decoder adds $S_M(H_t)$ to the input of the MLP, while only the temporal decoder writes into the residual stream.

B.6 SWIGLU VALUES

In compressed layers the linear value projection is replaced by

$$v = W_{\text{down}}\left[\text{SiLU}(W_{\text{gate}}\bar{x}) \odot W_{\text{up}}\bar{x}\right], \qquad \bar{x} = \text{RMSNorm}(x), \tag{8}$$

in column notation, with a parameter-free input normalisation that controls the scale presented to the quadratic form. Queries and keys stay linear. The hidden width, 5024 at 229M, 5688 at 1B, and 6648 at 2B, is chosen within the parameter budget freed by reparametrising the output projections and the trainable embedding, and the dense final layer keeps ordinary linear values. We use SwiGLU because it is the gated unit of the backbone's own MLPs and outperforms ungated activations in transformer feed-forward layers (Shazeer, 2020). Any nonlinear value map removes the rank ceiling of a linear one, and we did not consider alternatives.

B.7 QUANTISERS

Compressed activations are quantised per token with symmetric scalar quantisation and one BF16 scale per token and direction, using nearest rounding for INT8 activations and stochastic rounding for INT4 gradients. The quantisers are active throughout ordinary language-model training and need no learned codebook, calibration, or auxiliary objective. Section 3.4 motivates such simple quantisers by the random bases and the weak attention sinks of SSNs, and the measured tolerance is consistent with this motivation, although our experiments do not isolate the contribution of either property. Received activations and gradients are dequantised for floating-point computation, so no low-bit matrix-multiplication kernels are needed. Section 3.4 gives the resulting communication volumes and Appendix E.3.6 derives them, and Appendix E.3.5 bounds the rounding and saturation errors.

C KOLMOGOROV WIDTHS AND THE SHARED-SUBSPACE CONSTRAINT

SSNs transmit their compressed activations exactly (Ramasinghe et al., 2025), but exact transmission does not make them as expressive as an uncompressed transformer, because every output projection must write into one shared subspace even when the directions that each would prefer differ. Kolmogorov widths make the cost of this shared-subspace constraint precise. They measure the best approximation that a linear space of a given dimension offers a whole set of targets, and holding the targets fixed separates the geometric cost of the constraint from the training of a particular network. Appendix C.1 defines the row-space width that describes an SSN, and Appendix C.2 compares one global subspace with per-layer, per-sublayer and group subspaces, both through worst-case widths and through the exact expected cost of Theorem C.3 that Section 3.1 quotes. Appendix C.3 bounds how much a subspace adapted to its targets can improve on a fixed random one, and Appendix C.4 contains the proofs.

C.1 KOLMOGOROV WIDTHS AND THE ROW-SPACE CONSTRAINT

For a nonempty compact set $\mathcal{F}$ in a normed space X and an integer $r \geq 0$, its Kolmogorov r-width is

$$d_r(\mathcal{F}; X) = \inf_{\substack{V \subseteq X \text{ linear} \\ \dim V \leq r}} \sup_{f\in\mathcal{F}} \inf_{v\in V} \|f - v\|_X. \tag{9}$$

The approximating space is common to the entire target set, while the coefficients of the approximation may depend on each target (Kolmogoroff, 1936; Pinkus, 1985). This is exactly the situation of subspace communication, in which one fixed subspace serves every input and each token has its own coefficients in it.

An SSN imposes additional structure when the target is a matrix of n token updates in $\mathbb{R}^{n\times d}$. For an orthonormal basis $U \in \mathbb{R}^{d\times k}$ spanning S, the admissible updates form

$$V_S = \{ZU^\top \mid Z \in \mathbb{R}^{n\times k}\} = \{Y \in \mathbb{R}^{n\times d} \mid \mathrm{Row}(Y) \subseteq S\}. \tag{10}$$

Every row uses the same output space S, giving k communicated coefficients per token and nk coefficients for the sequence. Thus $\dim V_S = nk$, although V_S belongs to a restricted family of nk-dimensional matrix subspaces. The corresponding *row-space Kolmogorov width* is

$$\begin{aligned} d_k^{\mathrm{row}}(\mathcal{F}) &= \min_{\substack{S\subseteq\mathbb{R}^d \\ \dim S=k}} \sup_{\Delta\in\mathcal{F}} \inf_{Y\in V_S} \|\Delta - Y\|_F \\ &= \min_{\dim S=k} \sup_{\Delta\in\mathcal{F}} \|\Delta(I - P_S)\|_F. \end{aligned} \tag{11}$$

Here $P_S = UU^\top$, the orthogonal projection ΔP_S is the best approximant in the Frobenius norm, and the squared Frobenius error sums the squared errors of all token rows. The minimum exists for integers $0 \le k \le d$ because both the target set and the Grassmannian of candidate spaces are compact.

Because the spaces V_S are only some of the nk-dimensional subspaces of $\mathbb{R}^{n\times d}$, the row-space width is at least the ordinary width of the same dimension,

$$d_{nk}(\mathcal{F}; \mathbb{R}^{n\times d}) \le d_k^{\mathrm{row}}(\mathcal{F}), \tag{12}$$

and the two coincide with the usual vector formulation when $n = 1$ (Lemma C.8). The difference between them is the structural restriction that an SSN imposes at a stage boundary, where each token communicates its own k coefficients but all tokens must use the same k-dimensional output space. The finite-dimensional Hilbert setting also permits optimal linear projection, whereas Kolmogorov and linear-operator widths are distinct notions in general normed spaces (Pinkus, 1985, Chapters II, IV and VI). The SSN approximation problem is the structured Kolmogorov problem in Eq. (11), whose admissible ranges have the specific form V_S. Its projection formula does not remove that structural restriction.

C.2 The approximation cost of the shared-subspace constraint

Targets and subspace partitions. Let $\mathcal{J}$ be a nonempty finite set of output projections and let $\mathcal{F}_j \subset \mathbb{R}^{n\times d}$ be the nonempty compact set of target updates of output projection j. An output projection can be indexed by a layer, a sublayer, or an individual attention or MLP group, and all comparisons below use the same atomic output projections and target sets. The worst-case error of output projection j for a prescribed output space is

$$e_j(S) = \sup_{\Delta\in\mathcal{F}_j} \|\Delta(I - P_S)\|_F, \qquad d_k^{\mathrm{row}}(\mathcal{F}_j) = \min_{\dim S=k} e_j(S). \tag{13}$$

A partition Π of $\mathcal{J}$ specifies which output projections share a subspace. Its optimal worst-case error across output projections is

$$E_\Pi(k) = \min_{\substack{S_C\subseteq\mathbb{R}^d,\ \dim S_C=k \\ C\in\Pi}} \max_{C\in\Pi} \max_{j\in C} e_j(S_C). \tag{14}$$

Since each cell chooses its space independently, the objective has the equivalent width representation

$$E_\Pi(k) = \max_{C\in\Pi} d_k^{\mathrm{row}}\left(\bigcup_{j\in C} \mathcal{F}_j\right). \tag{15}$$

In particular, one global subspace and a separate subspace for every atomic output projection give

$$E_{\mathrm{global}}(k) = d_k^{\mathrm{row}}\left(\bigcup_j \mathcal{F}_j\right), \qquad E_{\mathrm{private}}(k) = \max_j d_k^{\mathrm{row}}(\mathcal{F}_j). \tag{16}$$

The global subspace must approximate the union of all target updates, whereas the private subspaces can follow the directions that each output projection requires, which is why updates that are individually low-dimensional need not have an equally accurate common low-dimensional representation.

An atomic index $j = (\ell, b, g)$ records a layer, a sublayer and a group, and one global subspace and per-layer, per-sublayer and group subspaces correspond to partitions that retain successively more of these indices. SSN++ relaxes the shared-subspace constraint along exactly this sequence (Sections 3.1 and 3.2). With per-sublayer subspaces, a sublayer that contains several groups is approximated through the union of their targets.

Theorem C.1 (Relaxing the shared-subspace constraint)**.** *If Π' refines Π, meaning every cell of Π' is contained in a cell of Π, then $E_{\Pi'}(k) \le E_{\Pi}(k)$. In particular, the optimal errors with group, per-sublayer, per-layer and global subspaces satisfy*

$$E_{\text{group}}(k) \le E_{\text{sublayer}}(k) \le E_{\text{layer}}(k) \le E_{\text{global}}(k). \tag{17}$$

The inequality holds because a finer partition can always keep its parent's subspace. It compares the updates at equal rank *per atomic output projection*, and the finer partition pays only for storing more bases. Independently sampled fixed subspaces cannot keep a parent's subspace, and Proposition C.4 below shows what replaces the ordering for them.

The width allows the best coefficients for every target, and the reparametrised SSN layer can realise them. For features H_j and an update $\Delta_j = H_j W_j^\top$, the reduced factor $B_j = U_j^\top W_j$ gives $H_j B_j^\top U_j^\top = \Delta_j P_{S_j}$, the optimal projection of the update onto the subspace.

Angular mismatch. Let $S_j^\star$ attain $d_j = d_k^{\text{row}}(\mathcal{F}_j)$ and set $a_j = \sup_{\Delta \in \mathcal{F}_j} \|\Delta P_{S_j^\star}\|_F$. With $\theta_{\max}$ denoting the largest principal angle, define

$$A_\Pi = \min_{\dim S_C = k} \max_{C \in \Pi} \max_{j \in C} a_j \sin \theta_{\max}(S_j^\star, S_C). \tag{18}$$

For $1 \le k < d$, these angles bound the cost of the shared-subspace constraint.

Proposition C.2 (Mismatch upper bound)**.** $0 \le E_\Pi(k) - \max_j d_j \le A_\Pi$.

The constraint therefore costs at most the largest component of each output projection's targets in their optimal subspace, scaled by the sine of the largest angle between that subspace and the shared one. Although both errors decrease with k, this cost need not, as the example in Appendix C.4.6 shows.

An exact expected-error diagnostic. Worst-case widths control every target in a set. The expected squared projection error instead weights the targets by their distribution, has an exact spectral expression through the variational principle of Fan (1949), and can be estimated from measured second moments. For random target updates with finite second moments, define the uncentred matrices $C_j = \mathbb{E}[\Delta_j^\top \Delta_j]$ and fix weights $w_j > 0$. For a cell C, let $C_C = \sum_{j \in C} w_j C_j$, and write $\tau_k(M) = \sum_{i>k} \lambda_i(M)$ for the eigenvalue tail of a positive semidefinite matrix, with eigenvalues in decreasing order. The expected projection objective is

$$R_\Pi(k) = \min_{\dim S_C = k} \sum_{C \in \Pi} \sum_{j \in C} w_j \mathbb{E}\|\Delta_j (I - P_{S_C})\|_F^2. \tag{19}$$

Theorem C.3 (Spectral cost of the shared-subspace constraint)**.** *The exact optimum is $R_\Pi(k) = \sum_{C \in \Pi} \tau_k(C_C)$. If a cell C is split into $C_1, \dots, C_r$, its reduction in optimal error is*

$$\tau_k\left(\sum_{a=1}^{r} C_{C_a}\right) - \sum_{a=1}^{r} \tau_k(C_{C_a}) \;\ge\; 0. \tag{20}$$

Equality holds exactly when there is one rank-k orthogonal projector maximising $\operatorname{tr}(P C_{C_a})$ for every child a. Consequently, distinct unique top-k eigenspaces of two children imply a strictly positive reduction.

With unit weights, splitting the global cell into single output projections gives the cost $\tau_k(\sum_j C_j) - \sum_j \tau_k(C_j)$ of the shared-subspace constraint, which is the energy that the output projections could capture in their own top-k subspaces but lose in the best common one. It vanishes only when all of them prefer the same principal directions, and it is largest when their preferred subspaces are mutually orthogonal.

The effect of fixed-subspace alignment. The spectral objective also separates the cost of the shared-subspace constraint from the cost of a particular fixed subspace. For fixed rank-k orthogonal projectors $P = (P_C)_{C\in\Pi}$, let $\widehat{R}_\Pi(P)$ be the objective in Eq. (19) evaluated at those projectors.

Proposition C.4 (Fixed-subspace excess error)**.** *With the second moments defined above,*

$$\widehat{R}_\Pi(P) = R_\Pi(k) + \mathcal{D}_\Pi(P), \qquad \mathcal{D}_\Pi(P) = \sum_{C\in\Pi}\left[\sum_{i=1}^{k}\lambda_i(C_C) - \operatorname{tr}(P_C C_C)\right] \geq 0. \tag{21}$$

The deficit measures the target energy missed because the fixed subspace is not an optimal principal subspace. With unit weights on all output projections, the difference between the actual shared and fully private projection errors is

$$\widehat{R}_{\text{global}} - \widehat{R}_{\text{private}} = \Gamma_{k,\mathcal{J}} + \mathcal{D}_{\text{global}} - \mathcal{D}_{\text{private}}, \tag{22}$$

where $\Gamma_{k,\mathcal{J}} = \tau_k(\sum_{j\in\mathcal{J}} C_j) - \sum_{j\in\mathcal{J}}\tau_k(C_j)$. With the attention and MLP sublayers as the atomic output projections, $\Gamma_{k,\mathcal{J}}$ is the cost Γ_k quoted in Section 3.1. Theorem C.3 guarantees $\Gamma_{k,\mathcal{J}} \geq 0$ only for optimally chosen subspaces. For independently sampled fixed subspaces, Eq. (22) applies instead, and relaxing the shared-subspace constraint lowers the error only when $\Gamma_{k,\mathcal{J}}$ exceeds the increase in alignment deficit $\mathcal{D}_{\text{private}} - \mathcal{D}_{\text{global}}$.

Grouped matrices and the parameter budget. In column-vector notation, split the input of a grouped output projection into G blocks of sizes p_g, with $p = \sum_g p_g$. Given orthonormal bases $U_g \in \mathbb{R}^{d\times k}$, its linear map before the output normalisation is

$$\widehat{W} = [U_1 B_1 \ \cdots \ U_G B_G], \qquad B_g \in \mathbb{R}^{k\times p_g}. \tag{23}$$

Proposition C.5 (Fixed-subspace grouped approximation)**.** *For a target $W = [W_1 \ \cdots \ W_G]$,*

$$\min_{B_1,\ldots,B_G} \|W - \widehat{W}\|_F^2 = \sum_g \|(I - U_g U_g^\top)W_g\|_F^2. \tag{24}$$

The minimum is attained at $B_g = U_g^\top W_g$. The reduced factors contain kp trainable scalars, and $\operatorname{rank}(\widehat{W}) \leq \min\{d, \sum_g \min(k, p_g)\}$.

A grouped output projection can therefore write into more than k ambient directions with the same kp trainable coefficients as a single $k \times p$ factor. It is not an arbitrary matrix of rank Gk, however, because each input block writes only into its own subspace. The fixed bases add Gdk stored values and, like the orthogonal transforms between them, are not trained. The gates and decoders add trainable parameters, which the counts of Table 5 include. Proposition C.5 describes this linear map before the output normalisation. The normalisation only rescales each token's output, so the outputs keep the range of $\widehat{W}$, and the gates act only on the transmitted coefficients, whose aggregation Proposition E.2 bounds. Locally the groups together hold Gk coefficients, and how much of this information survives their gated aggregation into the k communicated coefficients depends on the aggregation rather than on the ambient rank.

C.3 Random subspaces and the value of subspace adaptation

The optima above describe subspaces that adapt to their targets, as the Grassmannian updates of the original SSN formulation do (Ramasinghe et al., 2025). SSNs with fixed subspaces instead learn only the reduced factors, inside output spaces drawn at random, which raises the question of how much expected projection error adapting the subspaces could remove. Work on gradient compression suggests that it can be little. The studies of Rajabi et al. (2026b;c), developed further by Rajabi et al. (2026a), find randomised gradient subspaces competitive when optimiser states and discarded gradient components are handled appropriately, and Thomas (2026) finds that estimated gradient subspaces vary substantially across minibatches beyond a small reproducible core. What matters is therefore the energy that distinguishes candidate subspaces, rather than any particular basis.

The expected cost of a random subspace. For a fixed positive semidefinite second moment $C \in \mathbb{R}^{d\times d}$, define $K_k(C) = \sum_{i=1}^{k} \lambda_i(C)$ and

$$\begin{aligned} R_C(P) &= \operatorname{tr}((I-P)C), \qquad R_C^\star = \operatorname{tr} C - K_k(C), \\ D_C(P) &= R_C(P) - R_C^\star = K_k(C) - \operatorname{tr}(PC). \end{aligned} \tag{25}$$

Here P is a rank-k orthogonal projector and $1 \le k < d$. Orthonormalising an independent Gaussian $d \times k$ matrix, as the SSN initialisation does, gives a uniformly distributed subspace, whose projector is Haar distributed whatever signs the QR factorisation selects.

Proposition C.6 (Random-subspace excess error)**.** *If P is a Haar-distributed rank-k orthogonal projector independent of C, then*

$$\mathbb{E}_P R_C(P) = \left(1 - \frac{k}{d}\right) \operatorname{tr} C, \qquad \mathbb{E}_P D_C(P) = K_k(C) - \frac{k}{d} \operatorname{tr} C. \tag{26}$$

Writing $L_C = \lambda_1(C) - \lambda_d(C)$ gives

$$0 \le \mathbb{E}_P D_C(P) \le \frac{k(d-k)}{d} L_C. \tag{27}$$

The expected deficit is the gain of an optimally chosen subspace over a random one on the same targets. It vanishes for an isotropic second moment, although the unavoidable rank-k error then still retains the fraction $1 - k/d$ of the target energy, so a small benefit from adapting the subspace is compatible with a large error from compression.

A uniform guarantee under near-isotropy. Spectral spread also controls every candidate subspace, including one selected after observing the targets.

Proposition C.7 (Limited gain from subspace adaptation)**.** *For every rank-k orthogonal projector P, with $q = \min\{k, d-k\}$,*

$$0 \le D_C(P) \le K_k(C) - \sum_{i=d-k+1}^{d} \lambda_i(C) \le q L_C. \tag{28}$$

If $\|C - \alpha I\|_2 \le \varepsilon\alpha$ for $\alpha > 0$ and $0 \le \varepsilon < 1$, then

$$\frac{D_C(P)}{R_C^\star} \le \frac{2\varepsilon q}{(d-k)(1-\varepsilon)}. \tag{29}$$

For an independent Haar projector, the corresponding expectation satisfies

$$\frac{\mathbb{E}_P D_C(P)}{R_C^\star} \le \frac{2\varepsilon k}{d(1-\varepsilon)}. \tag{30}$$

These bounds make precise when optimising a subspace barely changes its approximation error, namely when the whole target spectrum is nearly flat, which a small eigengap at the rank cutoff does not ensure. For several output projections, the absolute bounds add over the cell moments C_C of Proposition C.4 and bound the total gain from adapting every subspace.

A stable component and an isotropic remainder. Adapting the subspace matters when a few directions carry substantial energy. Let $C = \alpha I + S$, where $\alpha > 0$, $S \succeq 0$, and $\operatorname{rank} S = s \le k$. Then

$$R_C^\star = \alpha(d-k), \qquad D_C(P) = \operatorname{tr}((I-P)S), \qquad \mathbb{E}_P D_C(P) = \left(1 - \frac{k}{d}\right) \operatorname{tr} S. \tag{31}$$

Any rank-k space that contains $\operatorname{Range}(S)$ is optimal, whatever its remaining directions, whereas a fully random space generally misses part of this stable component, at an expected cost of $(1 - k/d) \operatorname{tr} S$. The model thus separates an identifiable core from an interchangeable remainder, as Thomas (2026) observes for gradient subspaces, and shows that the energy of the core rather than its dimension decides whether adapting the subspace pays off. Appendix C.4.4 proves these identities and the preceding propositions.

A reference spectrum for random targets. The flatness that Proposition C.7 requires is a property of the targets, and randomness alone does not provide it. Let $X \in \mathbb{R}^{d\times m}$ hold m independent standard Gaussian samples in $\mathbb{R}^d$, with second moment

$$C_{\mathrm{G}} = \frac{\alpha}{m} X X^\top, \qquad \mathbb{E}_X C_{\mathrm{G}} = \alpha I \tag{32}$$

Although it is isotropic in expectation, its empirical eigenvalue distribution converges to the Marchenko–Pastur law as $d, m \to \infty$ with $d/m \to \gamma \in (0, 1)$ (Marchenko & Pastur, 1967), whose support is

$$[\lambda_-, \lambda_+] = \left[\alpha(1 - \sqrt{\gamma})^2, \alpha(1 + \sqrt{\gamma})^2\right]. \tag{33}$$

The spectrum is therefore nearly flat only when there are many samples per dimension. In the notation of Proposition C.7, ε tends to $2\sqrt{\gamma} + \gamma$, which is below one only for $\gamma < 3 - 2\sqrt{2} \approx 0.17$, that is, with more than about six samples per dimension. A subspace fitted to such a sample can exploit this spread although the population has no preferred directions, so an adapted subspace may beat a random one on the data it was fitted to without generalising. Real targets can also carry isolated spikes outside the bulk, like the stable component above.

The SSN bases do not have this spectrum, because orthonormalising the Gaussian draw leaves a projector with eigenvalues zero and one, and so Proposition C.6 rests on their random orientation alone.

Consequences for SSN++. Fixed random subspaces are therefore justified when the target spectrum is nearly flat or its stable component is weak, since changing their orientation can then gain little, and both conditions can be measured.

Separating random subspaces plays a different role. For fixed target moments that are independent of the sampled subspaces, linearity gives

$$\mathbb{E}_P \widehat{R}_\Pi(P) = \left(1 - \frac{k}{d}\right) \sum_{j\in\mathcal{J}} w_j \operatorname{tr} C_j \tag{34}$$

for every partition whose cell subspaces have Haar marginals, so shared and separate random subspaces have the same expected error while the targets stay fixed. In training, however, the features and updates of each layer adapt to the subspaces it is given, and grouping, gated aggregation and the decoders change how the local representations combine. Any benefit of separate random subspaces therefore arises from these effects, which Sections 3.1 and 3.2 measure in trained models, whereas the cost of the shared-subspace constraint describes what optimally adapted subspaces would gain. In preliminary experiments, learned subspaces gave no loss benefit over fixed ones at equal rank (Section 5).

C.4 Proofs

This subsection proves the results above. Appendices C.4.1 and C.4.2 prove the refinement and angular-mismatch bounds and the spectral cost of the shared-subspace constraint, Appendix C.4.3 treats grouped output projections, and Appendix C.4.4 proves the random-subspace results. Appendix C.4.5 bounds the width of smooth images of a manifold, and Appendix C.4.6 bounds the width of a shared subspace by dividing its dimensions among the output projections and shows that the cost of the constraint need not decrease with the rank.

C.4.1 Projection geometry and partition refinement

Lemma C.8 (Row-space projection)**.** *For $S \subseteq \mathbb{R}^d$, define $V_S = \{G \in \mathbb{R}^{n\times d} : \mathrm{Row}(G) \subseteq S\}$. Then $\dim V_S = n \dim S$, and for every $\Delta \in \mathbb{R}^{n\times d}$,*

$$\inf_{G\in V_S} \|\Delta - G\|_F = \|\Delta(I - P_S)\|_F. \tag{35}$$

If $\mathcal{F}$ is nonempty and compact, the minimum over k-dimensional S exists for each integer $0 \le k \le d$. Furthermore, $d_k^{\mathrm{row}}(\mathcal{F}) \ge d_{nk}(\mathcal{F}, \mathbb{R}^{n\times d})$, where the right-hand side is the ordinary Kolmogorov width with the Frobenius norm.

Proof. Every row of G can independently occupy $\dim S$ coordinates, proving the dimension statement. For $G \in V_S$, $G = GP_S$, and the Frobenius inner product between $\Delta(I - P_S)$ and $\Delta P_S - G$ is zero. Hence

$$\|\Delta - G\|_F^2 = \|\Delta(I - P_S)\|_F^2 + \|\Delta P_S - G\|_F^2,$$

which is minimised at $G = \Delta P_S$. To show attainment, put $M = \sup_{\Delta \in \mathcal{F}} \|\Delta\|_F < \infty$. For projectors P, Q,

$$\left| \sup_{\Delta \in \mathcal{F}} \|\Delta(I - P)\|_F - \sup_{\Delta \in \mathcal{F}} \|\Delta(I - Q)\|_F \right| \leq M\|P - Q\|_2.$$

The objective is therefore continuous on the compact set of rank-k orthogonal projectors. Finally, the family $\{V_S : \dim S = k\}$ is a subset of all nk-dimensional linear subspaces of $\mathbb{R}^{n \times d}$, so its approximation infimum cannot be smaller. □

For each cell C, the supremum over $\mathcal{F}_C = \bigcup_{j \in C} \mathcal{F}_j$ equals the maximum of the suprema over its constituent sets. Every choice of cell subspaces therefore has a worst-case error across output projections of at least $\max_C d_k^{\text{row}}(\mathcal{F}_C)$. Conversely, the finite union $\mathcal{F}_C$ is compact and has an optimal subspace by Lemma C.8. Choosing these minimisers independently for all cells attains that maximum. This proves Eq. (15) and its global and private specialisations.

Proof of Theorem C.1. For each $C' \in \Pi'$, choose the subspace of its unique parent $C \in \Pi$. Every atomic output projection then has exactly the same approximating subspace and error as before refinement. This constructs a feasible refined solution with the coarse optimum's objective value, proving the inequality. The displayed hierarchy follows by successive refinements. The compactness argument in Lemma C.8 applies to the finite product of Grassmannians, so all minima exist. □

Equality in Theorem C.1 has a direct characterisation. Let $e' = E_{\Pi'}(k)$. Then $E_\Pi(k) = e'$ if and only if each coarse cell C admits a k-dimensional subspace S_C such that $e_j(S_C) \leq e'$ for every $j \in C$. Necessity uses an attained coarse optimum. Sufficiency constructs a coarse solution with objective at most e' and then uses refinement. Unlike the expected-error criterion below, this condition allows a coarse subspace to be suboptimal for an output projection whose individual optimum is strictly below the threshold set by the worst output projection.

Proof of Proposition C.2. Fix j, set $P_j = P_{S_j^\star}$, and let $P = P_{S_C}$ for its assigned cell. For any $\Delta \in \mathcal{F}_j$,

$$\begin{aligned}
\|\Delta(I - P)\|_F &\leq \|\Delta P_j(I - P)\|_F + \|\Delta(I - P_j)(I - P)\|_F \\
&\leq \|\Delta P_j\|_F \|P_j(I - P)\|_2 + \|\Delta(I - P_j)\|_F \\
&\leq a_j \sin\theta_{\max}(S_j^\star, S_C) + d_j.
\end{aligned}$$

For equal-dimensional subspaces, the projector norm in the second line equals the sine of the largest principal angle. Taking the supremum, the maximum over output projections, and then the minimum over cell subspaces gives $E_\Pi(k) \leq \max_j d_j + A_\Pi$. Conversely, every output projection's error under any subspace is at least d_j, which proves the lower inequality. The decomposition uses a triangle inequality because the two residual components need not be orthogonal. □

C.4.2 Expected errors and strictness

Proof of Theorem C.3. For any orthogonal projector P, cyclicity of the trace and $(I - P)^2 = I - P$ give

$$\mathbb{E}\|\Delta_j(I - P)\|_F^2 = \operatorname{tr}\big((I - P)C_j\big).$$

No independence between output projections is needed, since the objective is a sum of individual squared errors rather than the squared error of a sum. Thus the problem separates over cells, with cell objective $\operatorname{tr}(C_C) - \operatorname{tr}(PC_C)$.

The spectral maximisation follows from the eigendecomposition. If $M = \sum_{i=1}^d \lambda_i u_i u_i^\top$ is positive semidefinite and P has rank k, let $q_i = u_i^\top P u_i$. Then $0 \leq q_i \leq 1$, $\sum_i q_i = k$, and $\operatorname{tr}(PM) = \sum_i \lambda_i q_i \leq \sum_{i=1}^k \lambda_i$. For example, the last inequality follows by moving any available mass from

an index above k to a deficient index at most k, which cannot decrease the sum. The projector onto $\mathrm{span}(u_1, \ldots, u_k)$ attains the bound. Subtracting from $\mathrm{tr}(M)$ proves the tail formula.

For a split, put $M_a = C_{C_a}$ and $K_k(M) = \max_{\mathrm{rank}\, P=k} \mathrm{tr}(PM)$, where all projectors are orthogonal. Then

$$K_k\left(\sum_a M_a\right) = \max_P \sum_a \mathrm{tr}(PM_a) \le \sum_a K_k(M_a).$$

The trace terms cancel when subtracting the fine risk from the coarse risk, yielding Eq. (20). If one projector maximises every summand, it attains equality. Conversely let $P_\star$ maximise the sum. Under equality,

$$0 = \sum_a \left[K_k(M_a) - \mathrm{tr}(P_\star M_a)\right].$$

Each bracket is nonnegative and must therefore be zero, which makes $P_\star$ a simultaneous maximiser. If $\lambda_k(M_a) > \lambda_{k+1}(M_a)$, its maximising subspace is unique. Two different such subspaces preclude a simultaneous maximiser and therefore force strict inequality. □

Proof of Proposition C.4. The fixed-subspace objective is $\widehat{R}_\Pi(P) = \sum_{C\in\Pi}[\mathrm{tr}(C_C) - \mathrm{tr}(P_C C_C)]$. Subtracting $R_\Pi(k) = \sum_C[\mathrm{tr}(C_C) - \sum_{i=1}^k \lambda_i(C_C)]$ gives Eq. (21). Every summand is nonnegative by the same spectral variational principle. Taking the difference of this identity for global and fully private partitions, with $w_j = 1$, gives Eq. (22). □

C.4.3 Grouped output projections

Proof of Proposition C.5. The columns belonging to different input blocks are disjoint in the matrix Frobenius norm. Within a block,

$$\|W_g - U_g B_g\|_F^2 = \|(I - U_g U_g^\top)W_g\|_F^2 + \|U_g^\top W_g - B_g\|_F^2,$$

using orthogonality and $U_g^\top U_g = I$. The unique minimiser is $B_g = U_g^\top W_g$, giving Eq. (24). Counting its entries gives $\sum_g kp_g = kp$. Each block has rank at most $\min(k, p_g)$ and the column space of the concatenation is the sum of the block column spaces. The rank of that sum is at most the sum of the ranks and at most d. □

If the U_g may also be chosen optimally, applying the spectral argument above to each $W_g W_g^\top$ yields

$$\min_{\substack{U_g^\top U_g = I \\ B_g}} \|W - [U_1 B_1 \ \cdots \ U_G B_G]\|_F^2 = \sum_g \tau_k(W_g W_g^\top). \tag{36}$$

The corresponding shared-subspace optimum is $\tau_k(WW^\top) = \tau_k(\sum_g W_g W_g^\top)$, so Theorem C.3 gives the exact benefit of optimal group subspaces over one optimal shared subspace for this matrix objective. For fixed subspaces the ordering can go either way. With $d = 2$, $k = 1$, two scalar input groups, target $W_1 = e_1$, $W_2 = 0$, shared basis $U = e_1$, and group bases $U_1 = e_2$, $U_2 = e_1$, the fixed shared error is zero and the fixed grouped error is one, and reversing the target to $W_1 = e_2$, $W_2 = 0$ reverses the ordering.

C.4.4 Random subspaces, spectral spread, and stable components

Proof of Proposition C.6. Orthogonal invariance implies that $\mathbb{E}_P P$ commutes with every orthogonal matrix and hence is a scalar multiple of I. Taking its trace gives $\mathbb{E}_P P = (k/d)I$. Independence from the target second moment then gives $\mathbb{E}_P \mathrm{tr}(PC) = (k/d)\,\mathrm{tr}\, C$, which proves Eq. (26). With eigenvalues in decreasing order,

$$K_k(C) - \frac{k}{d}\,\mathrm{tr}\, C = \frac{1}{d}\sum_{i=1}^{k}\sum_{j=k+1}^{d}(\lambda_i(C) - \lambda_j(C)).$$

Every summand lies between zero and L_C, proving Eq. (27). Applying the expectation identity to each cell and summing proves Eq. (34) without requiring independence between cell projectors. Gaussian QR gives the stated Haar law because left multiplication by an orthogonal matrix preserves the Gaussian distribution and rotates its column space by that same matrix. □

Proof of Proposition C.7. The spectral variational principle bounds $\mathrm{tr}(PC)$ between the sums of the smallest and largest k eigenvalues. Their difference is at most kL_C. Rewriting the same difference as the sum of the largest $d-k$ eigenvalues minus the smallest $d-k$ eigenvalues also bounds it by $(d-k)L_C$. This proves Eq. (28). The near-isotropy assumption gives $L_C \le 2\varepsilon\alpha$ and $R_C^\star \ge (d-k)\alpha(1-\varepsilon) > 0$. Substitution proves Eq. (29), and the expectation bound in Eq. (27) proves Eq. (30). □

For Eq. (31), all eigenvalues of C are α plus the corresponding eigenvalues of S. Since $s \le k$, $K_k(C) = k\alpha + \mathrm{tr}\, S$ and $R_C^\star = (d-k)\alpha$. Subtracting $\mathrm{tr}(PC) = k\alpha + \mathrm{tr}(PS)$ gives $D_C(P) = \mathrm{tr}((I-P)S)$. Its expectation follows from $\mathbb{E}_P P = (k/d)I$. Finally,

$$\mathrm{tr}((I-P)S) = \|(I-P)S^{1/2}\|_F^2$$

vanishes exactly when $\mathrm{Range}(S) \subseteq \mathrm{Range}(P)$.

C.4.5 A SMOOTH-MANIFOLD UPPER BOUND WITH EXPLICIT CHART ASSUMPTIONS

Kolmogorov widths relate the approximation error to the dimension of the output space, but their definition gives no rate at which the error decays with that dimension. Smooth targets supply one. When the updates depend smoothly on a few underlying parameters, the row-space width decays polynomially in the rank. The construction adapts the local polynomial approximation principle developed for Sobolev classes by Birman & Solomyak (1967), surveyed with the classical width theory by Pinkus (1985, Chapter VII), to a matrix-valued image of a compact manifold that one subspace of $\mathbb{R}^d$ approximates for all token rows.

Assumption C.9 (Quantitative atlas and smooth target map)**.** Let M be a compact manifold of positive dimension m, and let $s \ge 1$ be an integer. There are finitely many parametrisations $\psi_a : \Omega_a \subset \mathbb{R}^m \to M$ and compact coordinate sets $K_a \subset \Omega_a$ such that $M = \bigcup_a \psi_a(K_a)$. For some $0 < \varepsilon_0 \le 1$ and $C_0 \ge 1$, at each $0 < \varepsilon \le \varepsilon_0$ the sets K_a admit a combined collection of at most $C_0\varepsilon^{-m}$ centres in their respective K_a, so every $z \in K_a$ is within ε of a centre from the same chart. The ε_0-neighbourhood of each K_a lies inside Ω_a. Let $\Psi : M \to \mathbb{R}^{n\times d}$ be C^s, and write its rows in chart a as $f_{a,i} : \Omega_a \to \mathbb{R}^d$. On these neighbourhoods assume

$$\max_{a,i} \max_{0\le q\le s} \sup_z \|D^q f_{a,i}(z)\|_{\mathrm{op}} \le M_s < \infty,$$

where $D^0 f = f$ is measured in Euclidean norm. All derivatives are derivatives of the chart pull-backs, and bounds on $D\psi_a$ alone are not substituted for bounds on higher chart derivatives.

Such an atlas exists on every compact smooth manifold, by restricting finitely many charts to smaller compact sets, and the constants depend on that choice. Smoothness bounds in ambient coordinates give M_s through the chain rule, which then needs bounds on $D^q\psi_a$ for every $1 \le q \le s$.

Theorem C.10 (Row-space widths of smooth images)**.** *Under Assumption C.9, let* $P = \binom{m+s-1}{s-1}$ *and* $K_0 = C_0 n P \varepsilon_0^{-m}$*. For every integer* $1 \le k < d$ *satisfying* $k \ge K_0$*,*

$$d_k^{\mathrm{row}}(\Psi(M)) \le \frac{\sqrt{n} M_s}{s!} (C_0 n P)^{s/m} k^{-s/m}. \tag{37}$$

For all $1 \le k \le d$*, a valid, possibly loose bound is*

$$d_k^{\mathrm{row}}(\Psi(M)) \le \sqrt{n} M_s K_0^{s/m} k^{-s/m}, \qquad d_d^{\mathrm{row}}(\Psi(M)) = 0. \tag{38}$$

Proof. Fix a resolution $\varepsilon \le \varepsilon_0$ and choose the centres in Assumption C.9. Expanding each row function about a centre $z_{a,t}$ gives

$$f_{a,i}(z) = \sum_{|\beta|\le s-1} \frac{\partial^\beta f_{a,i}(z_{a,t})}{\beta!} (z - z_{a,t})^\beta + R_{a,t,i}(z).$$

For $z \in K_a$ within distance ε, the segment to the centre lies in the stated neighbourhood. Taylor's integral remainder along this segment therefore gives $\|R_{a,t,i}(z)\|_2 \le M_s \varepsilon^s / s!$. Let S span every

vector coefficient of these Taylor polynomials for all centres and all n rows. There are P multi-indices of degree at most $s-1$, so

$$\dim S \le nPC_0\varepsilon^{-m}.$$

At any point of M, select a covering chart and centre. Each row's Taylor polynomial belongs to S. Applying the projection lemma and summing the n squared remainder bounds gives

$$\|\Psi(x)(I-P_S)\|_F \le \sqrt{n}M_s\varepsilon^s/s!.$$

When $k \ge K_0$, choose $\varepsilon = (C_0nP/k)^{1/m} \le \varepsilon_0$. Then $\dim S \le k$, and extending S to dimension exactly k cannot increase the error. Substitution proves Eq. (37). For $1 \le k < K_0$, the zeroth-order bound gives $d_k^{\mathrm{row}}(\Psi(M)) \le \sqrt{n}M_s$, which is bounded above by the right-hand side of Eq. (38). For $k \ge K_0$, that right-hand side is at least the bound in Eq. (37), since $\varepsilon_0 \le 1$ and $s! \ge 1$. Finally the whole ambient space gives exact representation at $k = d$. □

The factor n in the dimension budget reflects that one output subspace must serve every token position, and the joint sequence manifold accounts for attention's dependence on all tokens without assuming independence between positions.

For a merely Lipschitz chart pullback with rowwise Lipschitz bound L, spanning the row values at the centres instead gives error $\sqrt{n}L\varepsilon$ with at most $nC_0\varepsilon^{-m}$ vectors, and hence a $k^{-1/m}$ rate over the admissible resolutions. Faster rates for nonsmooth networks require further regularity assumptions. In particular, a transformer containing a ReLU is not globally piecewise affine simply because its MLP activation is piecewise affine.

The target map enters the constants of Theorem C.10 through the derivative bound M_s and the covering constant C_0 of Assumption C.9. Estimates of M_s for a trained network scale with products of the weight norms along its layers, so the bound can lie far above the actual width, and the atlas itself has to be assumed. Tameness removes both limitations. A structure on the real numbers is o-minimal if every subset of $\mathbb{R}$ that it defines is a finite union of points and intervals (van den Dries, 1998), and the sets and maps definable in an o-minimal structure are called tame. Semialgebraic sets are tame, and so are sets built with the exponential and with analytic functions restricted to bounded domains, which together form the o-minimal structure $\mathbb{R}_{\mathrm{an,exp}}$ (van den Dries et al., 1994). Tameness excludes infinite oscillation, such as that of $\sin(1/x)$ near zero, and it bounds the complexity of all members of a definable family uniformly, which is what replaces the weight-dependent constants of Theorem C.10.

Remark C.11 (Tame networks). Networks built from polynomial, ReLU, exponential, softmax and normalisation operations are definable in $\mathbb{R}_{\mathrm{an,exp}}$ and hence tame, whether or not they are smooth. By the o-minimal form of the Yomdin–Gromov lemma (Yomdin, 1987; Gromov, 1987; Pila & Wilkie, 2006), every bounded definable set of dimension m is covered, for every order s, by finitely many C^s maps from the unit cube whose derivatives up to order s are bounded by the size of the set, and the number of maps is uniform over a definable family, such as all weight settings of one architecture. An analogue of Assumption C.9 therefore holds for every such network, with the derivative bound M_s replaced by the size of the targets and the covering constant C_0 by the number of maps. The rate remains $k^{-s/m}$, but its constant depends on the architecture rather than on the weights.

C.4.6 Allocation bounds and the dependence on rank

Proposition C.12 (A valid dimension-allocation upper bound)**.** *For finitely many nonempty compact sets $\mathcal{F}_j \subset \mathbb{R}^{n\times d}$ and nonnegative integers k_j with $\sum_j k_j \le k \le d$,*

$$d_k^{\mathrm{row}}\left(\bigcup_j \mathcal{F}_j\right) \le \max_j d_{k_j}^{\mathrm{row}}(\mathcal{F}_j), \tag{39}$$

where $d_0^{\mathrm{row}}(\mathcal{F}_j) = \sup_{\Delta\in\mathcal{F}_j}\|\Delta\|_F$.

Proof. Choose an optimal k_j-dimensional subspace S_j for each set. Their sum has dimension at most $\sum_j k_j \le k$. Extend it to a k-dimensional subspace S. Since $S_j \subseteq S$, projecting onto S is no worse for any element of $\mathcal{F}_j$ than projecting onto S_j. Taking suprema proves the claim. □

A shared subspace of rank k is therefore never worse than private subspaces whose ranks add up to k, so the shared-subspace constraint costs at most what dividing the k dimensions among the output projections costs. For example, if there are J sets and $d_r^{\mathrm{row}}(\mathcal{F}_j) \le c_j r^{-a}$ for positive ranks, with $a > 0$ and $c_j > 0$, giving every set $\lfloor k/J \rfloor$ dimensions for $k \ge J$ yields

$$d_k^{\mathrm{row}}\left(\bigcup_j \mathcal{F}_j\right) \le \max_j c_j \lfloor k/J \rfloor^{-a}.$$

In the positive-real relaxation of the allocation, the optimum gives the sets dimensions in proportion to $c_j^{1/a}$,

$$k_j^\star = \frac{k c_j^{1/a}}{\sum_i c_i^{1/a}}, \qquad \lambda_\star = \left(\frac{\sum_j c_j^{1/a}}{k}\right)^a,$$

because a maximum error of at most λ requires $k_j \ge (c_j/\lambda)^{1/a}$, and summing these requirements proves both formulas. The relaxed optimum $\lambda_\star$ bounds the best integer allocation from below.

The cost of the constraint need not decrease with rank. Take $n = 1$, $d = 3$, and let $\mathcal{F}_1$ and $\mathcal{F}_2$ be the Euclidean unit balls of $\mathrm{span}(e_1, e_2)$ and $\mathrm{span}(e_2, e_3)$, respectively. At rank one, each ball has width one because every one-dimensional candidate leaves a unit vector in the ball orthogonal to it. The shared width is also one, since all targets have norm at most one. At rank two, both individual widths are zero. Write a shared plane as $v^\perp$, $\|v\|_2 = 1$. Its errors on the two balls are

$$\sqrt{v_1^2 + v_2^2}, \qquad \sqrt{v_2^2 + v_3^2}.$$

The squared maximum is at least their squared average $(1 + v_2^2)/2 \ge 1/2$, attained at $v_2 = 0$ and $|v_1| = |v_3| = 1/\sqrt{2}$. Thus the shared-minus-private gap at ranks one, two, and three is, respectively, 0, $1/\sqrt{2}$, and 0. The constraint costs most at the intermediate rank, where each set fits its own subspace exactly but no common subspace fits both.

D Decoding from preceding tokens

The compressed activations of preceding tokens have already crossed the stage boundary, so the receiving layer can build a richer representation of each token from its causal history without additional communication. The analysis below characterises what the temporal decoder and the attention and MLP decoders can represent in this way. Appendix D.1 bounds the rank that a decoder over received history can reach, and Appendix D.2 characterises the part of a residual that the implemented decoder family can predict. Appendix D.3 shows when separate attention and MLP decoders improve on a single shared decoder. Appendix D.4 gives the proofs.

D.1 A decoder over received history

In column-vector notation, consider one stage boundary at which the receiving layer has orthonormal basis $U \in \mathbb{R}^{d\times k}$. Let $z_t \in \mathbb{R}^k$ be the compressed activation already expressed in the receiving layer's basis, and set $z_t = 0$ for $t \le 0$. With token embeddings subtracted, the implemented linear reconstruction is

$$\widehat{y}_t = U z_t + S_G(H_t), \qquad S_G(H_t) = \sum_{j=0}^{m} V_j G_j z_{t-j}, \qquad H_t = (z_t, \dots, z_{t-m}). \tag{40}$$

In the temporal decoder, each $V_j \in \mathbb{R}^{d\times k}$ is a fixed orthonormal basis and the $G_j \in \mathbb{R}^{k\times k}$ are learned. The bases V_j need not be orthogonal to U, and the output of the decoder is not projected onto $U^\perp$. Since gated aggregation also need not produce $z_t = U^\top y_t$, the residual that the decoder can fill in is $r_t = y_t - U z_t$ rather than $(I - UU^\top)y_t$.

Proposition D.1 (Linear history rank)**.** *For fixed decoder parameters, let $\widehat{Y} \in \mathbb{R}^{n\times d}$ contain the reconstructed vectors in Eq.* (40)*. Then*

$$\mathrm{rank}(\widehat{Y}) \le \min\{n, d, (m+1)k\}. \tag{41}$$

The terms Uz_t and $V_0G_0z_t$ both depend on the current compressed activation, so together they contribute only one set of k features. The further mk directions come from the history, which lets the decoder raise the attainable rank above k while each token still transmits only k coefficients, drawing on information that has already crossed the boundary.

D.2 Predictable residuals within the decoder family

Fix the encoder, backbone, bases, and distribution of (r, H), with finite second moments, define $\|f\|_{L^2}^2 = \mathbb{E}\|f\|_2^2$, and let $\mathcal{C}$ be the linear space of FIR functions $S_G(H)$ in Eq. (40), a finite-dimensional subspace of L^2.

Theorem D.2 (Predictable residual in the implemented class)**.** *If $\Pi_{\mathcal{C}}$ denotes orthogonal projection in L^2, then*

$$\inf_G \mathbb{E}\|r - S_G(H)\|_2^2 = \mathbb{E}\|r\|_2^2 - \|\Pi_{\mathcal{C}} r\|_{L^2}^2. \tag{42}$$

The improvement is strict exactly when some $S \in \mathcal{C}$ satisfies $\mathbb{E}\langle r, S(H)\rangle \neq 0$.

The predictable part of the residual thus depends both on the information in the compressed activations and on how well the fixed bases of the decoder can express it. More generally, put $\mu(H) = \mathbb{E}[r \mid H]$. For any family $\mathcal{F}$ of square-integrable functions of H,

$$\inf_{f\in\mathcal{F}} \mathbb{E}\|r - f(H)\|_2^2 = \mathbb{E}\|r - \mu(H)\|_2^2 + \inf_{f\in\mathcal{F}} \mathbb{E}\|\mu(H) - f(H)\|_2^2. \tag{43}$$

The two terms separate information that the history does not contain from the approximation error of the decoder class. Reconstruction is only one way in which the decoder can help, since it can also build useful features from the token history without reproducing any reference residual, and the lag statistics of Appendix J.4 are consistent with this use.

D.3 Attention and MLP decoders

SSN++ adds to the temporal decoder an attention decoder and an MLP decoder, two further linear FIR filters over the same history with fixed bases of their own. Attention takes $x + S_A(H)$ as input and the MLP takes $h + S_M(H)$ after the attention update, so the two decoders reach the residual stream only through the outputs of their sublayers.

A least-squares model in which every decoder uses the same linear family makes the potential benefit precise. For targets r_a of the temporal, attention, and MLP decoders, weights $w_a > 0$, and a common linear decoder space $\mathcal{C}$, let $u_a = \Pi_{\mathcal{C}} r_a$ and $\bar{u} = (\sum_a w_a u_a)/(\sum_a w_a)$. Then

$$\underbrace{\inf_{s\in\mathcal{C}} \sum_a w_a \|r_a - s\|_{L^2}^2}_{R_{\text{shared}}} - \underbrace{\sum_a w_a \inf_{s_a\in\mathcal{C}} \|r_a - s_a\|_{L^2}^2}_{R_{\text{private}}} = \sum_a w_a \|u_a - \bar{u}\|_{L^2}^2. \tag{44}$$

Separate decoders therefore gain exactly the disagreement between the representable targets of the sublayers. In SSN++ the fixed bases of the attention and MLP decoders also differ, so each draws on a function space of its own, and the architecture retains every function of the temporal decoder alone, since both added decoders can be set to zero. The removal interventions of Appendix J.7, in which the attention decoder matters far more than the MLP decoder, indicate that the two sublayers do ask for different decoded inputs. Separate decoders also keep separate gradients, whereas tied parameters would sum them and could cancel. Appendix D.4.3 extends this comparison to any number of decoders and to the separate function spaces of SSN++.

D.4 Proofs

This subsection proves the results above. Appendix D.4.1 proves the rank bound for a decoder with fixed parameters, and Appendix D.4.2 derives the optimal residual prediction within a fixed decoder family. Appendix D.4.3 extends the comparison of shared and private decoders to any number of targets.

D.4.1 FIXED-PARAMETER RANK

Proof of Proposition D.1. With compressed activations as column vectors, define

$$M_0 = U + V_0 G_0, \qquad M_j = V_j G_j \quad (1 \le j \le m), \qquad M = [M_0\ M_1\ \cdots\ M_m]. \tag{45}$$

Then $M \in \mathbb{R}^{d\times(m+1)k}$ and $\widehat{y}_t = M h_t$, where $h_t = [z_t^\top, z_{t-1}^\top, \ldots, z_{t-m}^\top]^\top$. Let $H_n \in \mathbb{R}^{n\times(m+1)k}$ have rows $h_t^\top$. Equation (40) gives $\widehat{Y} = H_n M^\top$, hence

$$\operatorname{rank}(\widehat{Y}) \le \min\{\operatorname{rank}(H_n), \operatorname{rank}(M)\} \le \min\{n, d, (m+1)k\}.$$

Zero padding at the beginning of the sequence does not change this factorisation. No independence assumption on the compressed activations or mutual orthogonality assumption on the bases is needed. □

The bound concerns the linear reconstruction after the known token embedding is subtracted, and later nonlinear computation can add further directions to the hidden states. Within the linear decoder, the extra rank comes only from using the history, since setting all $G_j = 0$ recovers the rank-k expansion.

D.4.2 OPTIMAL RESIDUAL PREDICTION IN A FIXED DECODER FAMILY

Let $(\Omega, \mathcal{A}, \mathbb{P})$ describe the fixed distribution of examples and token positions under consideration. Let $r : \Omega \to \mathbb{R}^d$ and $H = (z_0, \ldots, z_m) : \Omega \to \mathbb{R}^{(m+1)k}$ have finite second moments. Here z_j denotes the compressed activation at lag j, not that of absolute token j. The space $L^2(\Omega; \mathbb{R}^d)$ has inner product $\langle f, g\rangle_{L^2} = \mathbb{E}[f^\top g]$, with functions that are equal almost surely identified.

Proof of Theorem D.2. For fixed bases, define the linear map

$$T : \left(\mathbb{R}^{k\times k}\right)^{m+1} \longrightarrow L^2(\Omega; \mathbb{R}^d), \qquad T(G_0, \ldots, G_m) = \sum_{j=0}^{m} V_j G_j z_j.$$

Its image is $\mathcal{C}$. Every image has a finite second moment because the compressed activations do, and $\mathcal{C}$ has dimension at most $(m+1)k^2$. Every finite-dimensional subspace of a Hilbert space is closed; thus the orthogonal projection $p = \Pi_{\mathcal{C}} r$ exists and belongs to $\mathcal{C}$. In particular, some collection of mixer matrices realises p, although those matrices need not be unique.

For every $s \in \mathcal{C}$, orthogonality of $r - p$ to $\mathcal{C}$ gives

$$\|r - s\|_{L^2}^2 = \|r - p\|_{L^2}^2 + \|p - s\|_{L^2}^2.$$

The minimum is attained at $s = p$. Applying the same orthogonality to $r = (r - p) + p$ gives $\|r - p\|_{L^2}^2 = \|r\|_{L^2}^2 - \|p\|_{L^2}^2$, proving Eq. (42).

The gain is strict if and only if $p \neq 0$. If $p = 0$, then r is orthogonal to every $s \in \mathcal{C}$. Conversely, if $\langle r, s\rangle_{L^2} = 0$ for every such s, then in particular $\langle r, p\rangle_{L^2} = \|p\|_{L^2}^2 = 0$, so $p = 0$. This proves the stated correlation criterion. □

In terms of the fixed bases, the decoder family predicts part of the residual exactly when the residual, seen through some basis V_j, correlates with the compressed activation at lag j,

$$\Pi_{\mathcal{C}} r \neq 0 \quad \Longleftrightarrow \quad \text{there exists } j \text{ such that } \mathbb{E}[V_j^\top r\, z_j^\top] \neq 0, \tag{46}$$

because $\mathbb{E}\langle r, V_j G_j z_j\rangle$ is the Frobenius inner product between G_j and $\mathbb{E}[V_j^\top r\, z_j^\top]$ and the G_j vary independently. These moments, which exist by Cauchy–Schwarz, are uncentred because the linear FIR has no intercept.

Proof of the conditional-expectation identity in Eq. (43). Let $\mathcal{H} = \sigma(H)$ and $\mu = \mathbb{E}[r \mid \mathcal{H}]$. Conditional Jensen's inequality gives $\mu \in L^2$. For every $\mathcal{H}$-measurable square-integrable vector $f(H)$,

$$\mathbb{E}\langle r - \mu, \mu - f(H)\rangle = 0.$$

This follows from the defining conditional-expectation property, extended from bounded to square-integrable test functions by truncation and Cauchy–Schwarz. Expanding the square therefore yields

$$\mathbb{E}\|r - f(H)\|_2^2 = \mathbb{E}\|r - \mu\|_2^2 + \mathbb{E}\|\mu - f(H)\|_2^2.$$

The first term is independent of f; taking the infimum over $\mathcal{F}$ proves Eq. (43), whether or not the infimum in that family is attained. In the unrestricted family of square-integrable functions of H, choosing $f = \mu$ makes the second term zero. Taking $f = 0$ also gives

$$\mathbb{E}\|r - \mu\|_2^2 = \mathbb{E}\|r\|_2^2 - \mathbb{E}\|\mu\|_2^2.$$

□

D.4.3 Shared versus private decoder targets

The common-family comparison in Eq. (44) extends to any finite number of decoder targets, including those of the temporal, attention, and MLP decoders.

Proposition D.3 (Weighted disagreement cost)**.** *Let $\mathcal{C} \subseteq L^2(\Omega; \mathbb{R}^d)$ be a closed linear subspace, let $r_1, \dots, r_q \in L^2$, and let $w_a > 0$ be constant weights. Define $u_a = \Pi_{\mathcal{C}} r_a$, $W_{\mathrm{tot}} = \sum_a w_a$, and $\bar{u} = W_{\mathrm{tot}}^{-1} \sum_a w_a u_a$. Then Eq.* (44) *holds, and equivalently*

$$R_{\mathrm{shared}} - R_{\mathrm{private}} = \frac{1}{W_{\mathrm{tot}}} \sum_{a<b} w_a w_b \|u_a - u_b\|_{L^2}^2. \tag{47}$$

In particular, the gap is positive exactly when some representable targets differ in L^2.

Proof. Since $r_a - u_a$ is orthogonal to $\mathcal{C}$, for every $s \in \mathcal{C}$

$$\|r_a - s\|_{L^2}^2 = \|r_a - u_a\|_{L^2}^2 + \|u_a - s\|_{L^2}^2.$$

Independent minimisation over s_a gives $R_{\mathrm{private}} = \sum_a w_a \|r_a - u_a\|_{L^2}^2$. For the common s, the identity $\sum_a w_a (u_a - \bar{u}) = 0$ gives

$$\sum_a w_a \|u_a - s\|_{L^2}^2 = \sum_a w_a \|u_a - \bar{u}\|_{L^2}^2 + W_{\mathrm{tot}} \|\bar{u} - s\|_{L^2}^2.$$

Because $\bar{u} \in \mathcal{C}$, the last term can be made zero. This proves Eq. (44). Expanding both sides of

$$\sum_{a<b} w_a w_b \|u_a - u_b\|_{L^2}^2 = W_{\mathrm{tot}} \sum_a w_a \|u_a\|_{L^2}^2 - \Big\| \sum_a w_a u_a \Big\|_{L^2}^2$$

proves Eq. (47). All weights are positive, so this sum vanishes precisely when every u_a is the same element of L^2. □

The identity also covers the temporal decoder with additive attention and MLP decoders when all three use the same linear family, since any decoded inputs $(s_0, s_A, s_M) \in \mathcal{C}^3$ are realised by the temporal decoder output s_0 and the attention and MLP decoder outputs $s_A - s_0$ and $s_M - s_0$, which belong to $\mathcal{C}$ by linearity.

In SSN++ the fixed bases define different spaces $\mathcal{C}_0$, $\mathcal{C}_A$, and $\mathcal{C}_M$, and the attainable tuples are

$$\mathcal{T} = \{(s, s + t_A, s + t_M) : s \in \mathcal{C}_0,\ t_A \in \mathcal{C}_A,\ t_M \in \mathcal{C}_M\}. \tag{48}$$

Both added decoders can output zero, so $\{(s, s, s) : s \in \mathcal{C}_0\} \subseteq \mathcal{T}$, and for any objective on these tuples the infimum over $\mathcal{T}$ is no larger than over the shared tuples. The separate decoders are therefore never worse than a shared one at the optimum, and they gain the full disagreement of Eq. (44) whenever $\mathcal{C}_A$ and $\mathcal{C}_M$ contain the differences $s_A - s_0$ and $s_M - s_0$ of the optimal decoded inputs.

E Orthogonal transforms and quantised communication

A compressed activation transmitted exactly between two layers expands to different vectors in their two subspaces, and quantised communication adds a rounding error on top of this architectural change. This appendix separates the two effects and follows them through depth. Appendix E.1 bounds the ambient defect of an orthogonal transform (Proposition E.1) and the norm and Lipschitz constant of gated aggregation (Proposition E.2), and propagates both kinds of error through depth (Theorem E.3). Appendix E.2 relates the rounding of the communicated activations and the rank allowed by a fixed byte budget to that bound, and Appendix E.3 gives the proofs.

E.1 TRANSFORMS, AGGREGATION, AND ERROR PROPAGATION

Proposition E.1 (Orthogonal transforms preserve norms of compressed activations, not ambient vectors)**.** *Let $U, V \in \mathbb{R}^{d\times k}$ have orthonormal columns and let $T = \mathrm{polar}(U^\top V) \in O(k)$. Then $\|bT\|_F = \|b\|_F$, but*

$$\|bU^\top - bTV^\top\|_F \le 2\sin(\theta_{\max}(U,V)/2)\|b\|_F. \tag{49}$$

The coefficient is the exact operator norm of $U - VT^\top$.

The defect vanishes when the two bases span the same subspace and is close to its largest value for independent random subspaces, since the principal angles between two random 40-dimensional subspaces of $\mathbb{R}^{2048}$ lie between about 74 and 90 degrees. The transform therefore keeps the coefficients exactly while changing the ambient vectors they expand to, and each receiving layer learns to read them in its own subspace. Unlike an ordinary residual sum, which can accumulate a union of layer subspaces, SSN++ replaces the residual stream at every boundary by a transformed k-dimensional compressed activation.

Proposition E.2 (An explicit scalar-gate bound)**.** *For the gate in Eq.* (4)*, $\epsilon > 0$, $0 \le \delta \le 1$, orthogonal T_g, and $F(Z) = \sum_g \alpha_g(Z) z_g T_g$, one has $\|F(Z)\|_2 \le (1+\delta)\sqrt{G}\|Z\|_F$. On $\|Z\|_F \le R$, a Lipschitz bound is*

$$L_F \le \sqrt{G}\left(1 + \delta + \frac{R\delta\|A\|_2}{\epsilon}\right). \tag{50}$$

Because every gain lies between $1-\delta$ and $1+\delta$, aggregation can amplify the norm of the group outputs by at most $(1+\delta)\sqrt{G}$, but its sensitivity to a perturbation also grows with the gate weights $\|A\|_2$ and with R/ϵ, since the gate reads log-energies that change quickly for small group outputs.

Theorem E.3 (Architectural and communication error through depth)**.** *Let $f_\ell, \widetilde{f}_\ell$ be reference and modified maps on full sequence states in common domains containing both trajectories. Suppose $\sup_x \|f_\ell(x) - \widetilde{f}_\ell(x)\| \le \varepsilon_\ell$ and $\mathrm{Lip}(f_\ell) \le K_\ell$. Suppose quantisation contributes an additional discrepancy $\sup_x \|\widetilde{f}_\ell^{Q}(x) - \widetilde{f}_\ell(x)\| \le q_\ell$ on the same domain. From identical inputs,*

$$\|h_L - \widehat{h}_L\| \le \sum_{\ell=1}^{L} (\varepsilon_\ell + q_\ell) \prod_{j=\ell+1}^{L} K_j. \tag{51}$$

An error introduced at boundary ℓ is thus amplified by the Lipschitz constants of all later layers, so that with constants above one the early boundaries weigh most, and exact communication removes the rounding terms q_ℓ altogether. The sequence states include the history of compressed activations that the decoders read, and a linear FIR decoder changes its output by at most $(1+\sum_{j=0}^{m}\|G_j\|_2)\|\eta\|$ under an input perturbation η in sequence Frobenius norm, so a bound on q_ℓ covers the retained history, the attention and MLP decoders, and the rest of the stage.

E.2 QUANTISED COMMUNICATION AND RANK

Quantisation acts on the transmitted compressed activation and on its gradient in the backward pass, and the receiving layer dequantises both for floating-point computation, so the analysis concerns only the stage boundary. For nearest rounding to a grid of step s with $2M+1$ levels, no clipping, and exact arithmetic, a token's compressed activation of k coefficients satisfies $\|Q(z) - z\|_2 \le \sqrt{k}s/2$, and this rounding error is the term q_ℓ of Theorem E.3. With one scale per token, the step s is set by the largest coefficient, and because random subspaces spread large hidden channels over all coefficients (Section 3.4), that coefficient is of typical size and the error stays small relative to the activation. The implementation rounds scales through BF16 and clips to the finite grid, and Appendix E.3.5 includes the resulting saturation error. In the backward pass, straight-through and stochastic-rounding rules optimise through the rounded forward map, and stochastic rounding is unbiased in the absence of clipping.

With a fixed readout A_{out}, per-token cross-entropy changes by at most $\sqrt{2}\|A_{\mathrm{out}}\|_2\|h - \widehat{h}\|_2$, with any final normalisation adding its Lipschitz factor, which turns the state bound of Theorem E.3 into a bound on the loss.

Rank at a fixed byte budget. A larger rank lowers the best attainable approximation error, while rounding adds the term q_ℓ of Eq. (51), so rank and precision trade off at a fixed number of bytes. In our models the larger rank wins at equal bytes (Section 4.3), and Appendix E.3.6 derives the byte constraint and the corresponding error bound.

E.3 Proofs and integer communication

This subsection proves the results above and treats the integer communication format in more detail. Appendices E.3.1, E.3.2, and E.3.4 prove Propositions E.1 and E.2 and Theorem E.3, and Appendix E.3.3 shows that models with gated aggregation or a temporal decoder contain their simpler counterparts. Appendix E.3.5 bounds rounding and saturation errors and discusses the backward approximations, and Appendix E.3.6 derives the byte constraint that sets the largest rank at a fixed budget.

E.3.1 Proof of Proposition E.1

Let $U^\top V = A\Sigma B^\top$ be a singular value decomposition, with $A, B \in O(k)$ and $\Sigma = \operatorname{diag}(\sigma_i)$, where $\sigma_i = \cos\theta_i$. Choose $T = AB^\top$, including any orthogonal completion at zero singular values. Orthogonality gives $\|bT\|_F = \|b\|_F$. Moreover,

$$(U - VT^\top)^\top (U - VT^\top) = 2I - U^\top VT^\top - TV^\top U = 2A(I - \Sigma)A^\top.$$

Its largest eigenvalue is $2(1 - \cos\theta_{\max})$, proving the stated operator norm and Frobenius inequality. Procrustes also minimises $\|U^\top V - Q\|_F$ over $Q \in O(k)$, with minimum squared distance $\sum_i (1 - \sigma_i)^2$. Indeed, expanding the square reduces the optimisation to maximising $\operatorname{tr}(Q^\top A\Sigma B^\top) \le \sum_i \sigma_i$. The transform is unique when all $\sigma_i > 0$, which holds almost surely for independent random subspaces. Summing the transformed compressed activations of several groups, as the gated aggregation does, is nevertheless many-to-one even though each transform is orthogonal.

E.3.2 Proof of Proposition E.2

Let $r_g(Z) = \|z_g\|_2$ and $C = I - \mathbf{1}\mathbf{1}^\top/G$. The maps $Z \mapsto r(Z)$ and C have Lipschitz constant at most one; $r \mapsto \log(r + \epsilon)$ has Lipschitz constant at most $1/\epsilon$ on $r \ge 0$. Since componentwise $\tanh$ is one-Lipschitz,

$$\|\alpha(Z) - \alpha(Z')\|_2 \le \delta \|A\|_2 \|Z - Z'\|_F / \epsilon.$$

The linear operation that applies each group's orthogonal transform and then sums the results has norm $\sqrt{G}$. Further, $|\alpha_g| \le 1 + \delta$ and

$$\begin{aligned}\|\alpha(Z) \odot Z - \alpha(Z') \odot Z'\|_F &\le (1 + \delta)\|Z - Z'\|_F + \|[\alpha(Z) - \alpha(Z')] \odot Z'\|_F \\ &\le (1 + \delta + R\delta\|A\|_2/\epsilon)\, \|Z - Z'\|_F.\end{aligned}$$

Here $\odot$ scales complete rows, completing both bounds.

E.3.3 Baseline inclusion

Fix all bases, ranks, normalisation choices, and upstream sublayers. A model with gated aggregation includes its ungated version by setting the gate matrix and bias to zero. A model with a temporal decoder includes the same architecture without that decoder by setting all $G_j = 0$, and the attention and MLP decoders may independently be zero. Thus, whenever $\mathcal{H}_0 \subseteq \mathcal{H}_1$ is obtained by these settings, for any well-defined task risk R,

$$\inf_{h \in \mathcal{H}_1} R(h) \le \inf_{h \in \mathcal{H}_0} R(h).$$

The inequality holds for population and empirical risk alike, because it takes the infimum over a larger function class. Since the gates start at a gain of one and all decoder mixers at zero (Appendices B.4 and B.5), each richer model also begins training exactly as its simpler counterpart.

E.3.4 Proof of Theorem E.3

Let $e_\ell = \|h_\ell - \widehat{h}_\ell\|$, and denote the actually communicated modified map by $\widetilde{f}_\ell^{Q}$. The assumptions give

$$\begin{aligned} e_\ell &\le \|f_\ell(h_{\ell-1}) - f_\ell(\widehat{h}_{\ell-1})\| + \|f_\ell(\widehat{h}_{\ell-1}) - \widetilde{f}_\ell(\widehat{h}_{\ell-1})\| + \|\widetilde{f}_\ell(\widehat{h}_{\ell-1}) - \widetilde{f}_\ell^{Q}(\widehat{h}_{\ell-1})\| \\ &\le K_\ell e_{\ell-1} + \varepsilon_\ell + q_\ell. \end{aligned}$$

Unrolling from $e_0 = 0$ proves the result. The common domain allows models of different ranks to be compared after embedding their representations in a common ambient or augmented sequence state, and the bounds hold at perturbed states as well as along the reference trajectory.

For a causal shift P_j with left zero padding, $\|P_j Z\|_F \le \|Z\|_F$. The orthonormal bases of the temporal decoder have norm one, so

$$\|D(Z) - D(Z')\|_F \le \left(1 + \sum_{j=0}^{m} \|G_j\|_2\right) \|Z - Z'\|_F.$$

Finite memory bounds the shift operator, while the learned mixer norms remain explicit in the decoder's Lipschitz constant.

For logits a, cross-entropy for class y has gradient $\mathrm{softmax}(a) - e_y$ of Euclidean norm at most $\sqrt{2}$. The mean-value inequality and a fixed linear readout yield the loss bound in Appendix E.2. For mean loss over n tokens, Cauchy–Schwarz gives $\sqrt{2/n}\|A_{\mathrm{out}}\|_2\|H - \widehat{H}\|_F$.

E.3.5 Rounding, saturation, and backward approximations

Let $s > 0$ be the *stored* scale and $Q_s(x) = s\,\mathrm{clip}(\mathrm{round}(x/s), -M, M)$. In exact arithmetic,

$$|Q_s(x) - x| \le s/2 + (|x| - Ms)_+. \tag{52}$$

Projection onto $[-Ms, Ms]$ moves x by $(|x| - Ms)_+$, and subsequent rounding within that interval adds at most $s/2$. Thus for a group of q coefficients,

$$\|Q_s(z) - z\|_2 \le \sqrt{q}\, s/2 + \|(|z| - Ms\mathbf{1})_+\|_2.$$

For independent scale groups, the squared group bounds add. Rounding the scale to BF16 can reduce Ms below the group's original maximum, so the saturation term can be nonzero even when the scale was derived from that maximum, and the final floating-point cast adds a further rounding error. The implementation uses a positive scale floor, so zero rows remain zero.

Without clipping, stochastic rounding to adjacent grid points satisfies $\mathbb{E}[Q_s(x) \mid x, s] = x$ and $\mathrm{Var}(Q_s(x) \mid x, s) \le s^2/4$. For independent repetitions of the *same fixed quantity*, averaging N draws divides this variance by N. A straight-through backward rule differentiates through the piecewise-constant forward quantiser as if it were the identity.

E.3.6 Increasing the rank at a fixed communication budget

Ignoring token identifiers, packet headers, padding, and alignment, uniform BF16 communication uses $2k$ bytes per token per direction. Per-token INT8 quantisation with one BF16 scale uses $k + 2$ bytes. Packed per-token INT4 quantisation with one BF16 scale uses $\lceil k/2 \rceil + 2$ bytes, and INT4 with groups of eight and one scale per group uses $k/2 + 2k/8$ when $8 \mid k$. A forward-INT8/backward-INT4 format therefore uses $k + 2$ and $k/2 + 2$ bytes in the two directions, respectively, for even k. For communication precisions b_F, b_B and scale overheads s_F, s_B bytes per token, the forward-plus-backward constraint at each stage boundary is

$$B(k, b_F, b_B) = \left\lceil \frac{k b_F}{8} \right\rceil + s_F + \left\lceil \frac{k b_B}{8} \right\rceil + s_B \le B_0. \tag{53}$$

Here $s_F = s_B = 0$ for the BF16 reference and $s_F = s_B = 2$ for the per-token INT8/INT4 format. At $B_0 = 160$, the maximal ranks are 40 and 104, respectively. The latter uses 106 forward and 54 backward bytes, compared with 80 in each direction for BF16.

The comparison of designs uses a common padded state space, norms, reference maps, and reference Lipschitz bounds K_j valid for every candidate rank. Let $\eta_\ell(k, b_F)$ uniformly bound the forward perturbation of the history of compressed activations in the chosen sequence norm at stage boundary ℓ. Let $D_{\ell,k}$ map this input to *all affected coordinates* of the state used in Theorem E.3, including the retained history, the attention and MLP decoders, and any subsequent computation assigned to that stage. Holding other inputs fixed, suppose $D_{\ell,k}$ has Lipschitz constant at most $L_{D,\ell}(k)$ uniformly on the common domain. Then $q_\ell \le L_{D,\ell}(k)\eta_\ell(k, b_F)$, and each feasible design has the certificate

$$\|h_L - \widehat{h}_L\| \le \Phi(k, b_F) := \sum_{\ell=1}^{L} \big(\varepsilon_\ell(k) + L_{D,\ell}(k)\eta_\ell(k, b_F)\big) \prod_{j=\ell+1}^{L} K_j. \tag{54}$$

Because $D_{\ell,k}$ covers more than the temporal decoder, its constant combines those of all affected outputs. For their Euclidean concatenation with constants L_0, L_A, L_M, L_H a bound is $\sqrt{L_0^2 + L_A^2 + L_M^2 + L_H^2}$, and any subsequent map multiplies its own Lipschitz factor. A per-token error bound e implies a sequence Frobenius error of at most $\sqrt{n}\, e$, and stacking $m + 1$ lags adds at most a factor $\sqrt{m+1}$, unless η_ℓ already measures the stacked history.

Stage boundaries without quantisation have $\eta_\ell = 0$, so the certificate trades the architectural approximation error of a larger rank against the forward rounding error of a lower precision, subject to Eq. (53). The backward precision enters through the byte constraint and through the training trajectory.

F Experimental setup and measurement protocols

This appendix describes how the experiments of Section 4 were configured and measured. Appendix F.1 gives the model configurations, the parameter-matched SSNs, and the trainable parameters of every model, Appendix F.2 the MAPL and ResBM implementations, and Appendix F.3 the training budgets, optimiser, loss aggregation and downstream evaluation. Appendix F.4 explains the byte accounting and how measured throughput becomes training time, and Appendix F.5 describes the 200 Mb/s pipeline and reports its results at all three sizes. The separate 10,000-update 1B study behind the ablations and the training dynamics is described in Appendix G.2.

F.1 From communication savings to model quality

The central empirical question is whether richer local representations narrow the loss gap to an uncompressed transformer while preserving the communication savings of the SSN. Comparisons at similar parameter counts distinguish this benefit from simply increasing model size, while training throughput and downstream evaluation establish its computational cost and transfer beyond language-modelling loss. We study Llama-3-family transformers (Grattafiori et al., 2024) at three sizes. The 229M setting has eight transformer layers and residual width 1024, the 1B setting has 16 layers and width 2048, and the 2B setting has 20 layers and width 2560. All use head width 64 and eight key–value heads. Every compressed model keeps its final transformer layer dense, and the output head is dense in every model. The last compressed layer does not compress its output but passes its full residual stream to that dense layer, so the two share the final pipeline stage, and a stage boundary follows each earlier transformer layer, giving $L - 2$ boundaries for L layers. The compressed models therefore have six, fourteen, and eighteen compressed stage boundaries at the three sizes. With BF16 communication, SSN++ uses ranks $k = 20, 40, 50$, respectively, and Table 4 lists the rank, bytes, and compression of every model. Local computation remains floating point throughout.

At 229M, the backbone has 16 query heads, an MLP hidden width of 4096, and a vocabulary of 50,257 tokens. The nonlinear value network has hidden width 5024 and input normalisation in both SSN++ and the parameter-matched SSN. SSN++ uses eight groups in each attention and MLP output projection. Its six compressed stage boundaries follow transformer layers 0–5, and layer 6, the last compressed layer, keeps its compressed output projections.

The SSN establishes the performance of a single shared subspace. The parameter-matched SSN uses the same nonlinear value network and embedding reparametrisation as SSN++, separating additional parameter capacity from the choice of local subspaces and decoders. At every size the parameter-

Table 4: Communication at every stage boundary. Rank is the number of coefficients per token and direction, the subspace rank k for SSN and SSN++ and the bottleneck width for ResBM and MAPL. Bytes count the activations and gradients per token and stage boundary, and compression is the $4d$ bytes of the dense residual stream divided by a model's bytes. The 229M, 1B, and 2B models have 8, 16, and 20 transformer layers of width 1024, 2048, and 2560, and six, fourteen, and eighteen compressed stage boundaries.

	229M			1B			2B		
Model	Rank	Bytes	Compression	Rank	Bytes	Compression	Rank	Bytes	Compression
Dense	–	4,096	1.0	–	8,192	1.0	–	10,240	1.0
SSN	20	80	51.2	40	160	51.2	51	204	50.2
SSN, matched params	20	80	51.2	40	160	51.2	51	204	50.2
ResBM	20	80	51.2	40	160	51.2	50	200	51.2
MAPL	20	80	51.2	40	160	51.2	50	200	51.2
SSN++	20	80	51.2	40	160	51.2	50	200	51.2
SSN++ INT8	40	84	48.8	80	164	50.0	100	204	50.2
SSN++ INT8/INT4	52	82	50.0	104	160	51.2	130	199	51.5

matched SSN and SSN++ differ by less than 0.06% in trainable parameters (Table 5). Uncompressed transformers provide the quality reference at all three sizes.

Table 5: Trainable parameters (millions) of the models in the main comparisons, excluding the fixed embeddings and the fixed orthonormal bases. The higher ranks of the quantised SSN++ models add parameters to their output projections and decoders.

Model	229M	1B	2B
Dense	228.77	1078.34	1987.57
SSN	192.77	800.68	1426.10
SSN, matched params	228.68	1077.19	1986.49
ResBM	229.01	1080.61	1992.17
MAPL	234.91	1107.56	2035.00
SSN++	228.74	1077.80	1987.43
SSN++ INT8	230.66	1087.15	2004.77
SSN++ INT8/INT4	231.87	1093.35	2016.35

F.2 MAPL AND RESBM COMPARISONS

MAPL and ResBM (Janson et al., 2026; Aboudib et al., 2026) provide alternative ways to communicate compressed activations between the same backbone layers. Both retain dense attention and MLP maps and linear value projections. The comparisons therefore concern these methods implemented in our backbone with their own optimisation rules, rather than reproductions of the papers' original training setups.

In the 229M comparison, ResBM uses bias-free $d \to 20 \to 20$ and $20 \to 20 \to d$ maps with an intermediate ReLU, together with truncated and zero-padded residual paths. MAPL uses a learned orthonormal $d \times 20$ projector at each stage boundary and rank-20 token anchors with learned coefficients and fixed expansion maps. These models share the rank, BF16 communication, and training-token budget of the SSN and SSN++, while differing in their local computation.

For this comparison, MAPL updates its projectors using SPEL with peak step size 2.5×10^{-3} and momentum 0.95. Polar factors are computed by thin SVD. Anchor coefficients use AdamW with a $28\times$ learning-rate multiplier, giving a peak rate of 8.4×10^{-3}, and zero weight decay. All MAPL AdamW groups use $\epsilon = 10^{-20}$ instead of the common 10^{-8}. The common warmup and cosine schedule scale each group's own initial rate.

F.3 Training budgets and evaluation

All models are trained on FineWeb-Edu (Penedo et al., 2024), using GPT-2 tokenisation and sequences of 1024 tokens, and validated on C4 (Raffel et al., 2020). The 229M experiments complete 46,000 updates with 512 sequences per update, or 24.117B tokens, the 1B experiments 20,000 updates with 1024 sequences per update, or 20.972B tokens, and the 2B experiments 38,000 updates with 1024 sequences per update, or 39.846B tokens. At 1B and 2B this is approximately 20 tokens per parameter, following the compute-allocation heuristic of Hoffmann et al. (2022), whereas the 229M schedule reaches approximately 105 tokens per parameter, well beyond this allocation. The three sizes therefore test the architecture under different training budgets, following primarily the Chinchilla scaling law at 1B and 2B, and with a larger budget at 229M to explore the effect of a long training horizon on the loss gap.

The common AdamW configuration (Loshchilov & Hutter, 2019) uses a peak learning rate of 3×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, $\epsilon = 10^{-8}$, 150 warmup updates, and cosine decay to 0.1 of the peak rate. The MAPL-specific settings are given in Appendix F.2. All reported losses are cross-entropies in nats/token. C4 validation loss averages a common final window of six validation values spaced every 100 updates, from 45,500–46,000 at 229M, 19,500–20,000 at 1B, and 37,500–38,000 at 2B. FineWeb-Edu training loss averages the final 500 updates of each run. This averaging summarises the final part of each trajectory. Each validation evaluation uses 20 batches of four length-1024 sequences per replica. Training parameters use BF16 and gradient reduction uses FP32.

To establish whether lower language-modelling loss transfers beyond the validation corpus, we measure the eval accuracy of the final models with the Language Model Evaluation Harness (Biderman et al., 2024), without in-context examples. The four tasks are ARC-Easy and ARC-Challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), and HellaSwag (Zellers et al., 2019). We use normalised accuracy and report the unweighted mean across the four. Evaluation uses the native model implementation, BF16 autocast, and the training tokeniser, with no conversion of the SSN and SSN++ architectures. All examples in each benchmark split are scored.

F.4 Communication and computation

The communication volume counts the compressed activations and their gradients per token and stage boundary, including the scales of the integer formats (Section 3.4, derived in Appendix E.3.6), and Table 1 lists it for every model. It excludes token identifiers, framing overhead, and MAPL projector synchronisation. Reconstruction in SSN, SSN++ and MAPL depends on the token, so it assumes that receivers have the token identifiers. At 2B, the SSN and the parameter-matched SSN use rank 51, inherited from their width-to-rank ratio, and communicate 204 bytes against 200 for SSN++ at rank 50.

Training runs use eight data-parallel replicas of the whole model, so the compressed activations cross every stage boundary within a GPU, which isolates the architectural and numerical effects of narrow communication from network behaviour. The accompanying throughput measurements characterise the local computational cost of each architecture. Equivalent training time is the number of updates multiplied by tokens per update and divided by measured aggregate throughput. For 1B, throughput is the median recorded during training on eight H100 GPUs. SSN++, for example, processes 26,363 tokens per second per GPU against 31,662 for the parameter-matched SSN, which gives 27.6 and 23.0 hours for the 20,000 updates. For 229M and 2B, throughput is measured in separate short runs of the optimised implementation on eight L40S and eight H100 GPUs, respectively, with equal local batches within each size. Equivalent time excludes interruptions, validation, and other job overheads, so the resulting curves describe computational cost at the measured throughput.

F.5 Training time under a 200 MB/s pipeline

Every model of the three comparisons was run as a pipeline-parallel job with `torch.distributed.pipelining` and one stage per GPU, seven stages on the seven L40S GPUs of one node at 229M, 15 stages on 15 H100 GPUs of two nodes at 1B, and 19 stages on 19 H100 GPUs of three nodes at 2B. Each stage of a compressed model holds one transformer layer, and its last stage also holds the dense final layer, so that every stage boundary carries the rank-k compressed activations and the pipeline cuts at all six stage boundaries at 229M, all fourteen

at 1B, and all eighteen at 2B. The uncompressed transformer runs with the same number of stages. The jobs use the 1F1B schedule with 32 microbatches per update, each holding 16 sequences of 1,024 tokens at 229M and 32 at 1B and 2B, so that an update processes the full training batch. Each stage boundary acts as a link at 200 Mb/s with 50 ms latency. Each rank's inbound link is a serial queue. Every activation and gradient transfer pays its bytes at the link rate plus the latency, and the receiving stage waits until the transfer would have arrived. Bytes are counted by the communication-volume convention of Section 3.4, with $4d$ bytes for the uncompressed residual stream, rather than by the tensors actually moved, whose transfers within a node and between nodes are far faster than the 200 Mb/s links. The steady update time is the median of the last three 10-update windows after warm-up over 44 updates (8 for the uncompressed model, whose update over the links takes several minutes). Activation checkpointing is off at 229M, as in the one-node runs, and recomputes every layer at 1B and 2B, where the activations of the microbatches in flight would otherwise exceed the memory of a GPU. Recomputation lengthens the compressed models' updates, while the update of the uncompressed transformer is set by its link. Table 6 lists all three sizes. Figures 5, 4, and 6 place the 229M, 1B, and 2B training and validation losses on the measured time axes, on one node and over the links. Over the links, the uncompressed transformer needs 598, 1,161, and 2,404 hours at 229M, 1B, and 2B to reach the final validation loss that SSN++ INT8/INT4 reaches in 110, 113, and 278 hours, although at 229M it already matches the final loss of the matched SSN within the 105 hours that the matched SSN takes.

Table 6: Measured training time as data-parallel replicas on one node and as a pipeline on one node at 229M, two at 1B, and three at 2B, with every stage boundary a 200 Mb/s link with 50 ms latency. Budgets are 46,000 updates at 229M, 20,000 at 1B, and 38,000 at 2B. Bytes are the forward-plus-backward communication volume per token per stage boundary.

			Seconds per update		Hours for the budget	
Size	Model	Bytes	One node	200 Mb/s	One node	200 Mb/s
229M	Dense	4096	1.59	103.4	20.3	1,321.6
229M	SSN	80	1.64	7.9	20.9	100.9
229M	SSN, matched params	80	2.07	8.2	26.4	105.4
229M	ResBM	80	1.63	7.8	20.8	100.1
229M	MAPL	80	1.73	7.9	22.1	101.0
229M	SSN++	80	2.27	8.4	29.1	107.8
229M	SSN++ INT8	84	2.30	8.6	29.4	109.4
229M	SSN++ INT8/INT4	82	2.33	8.6	29.7	110.1
1B	Dense	8192	2.97	486.1	16.5	2,700.6
1B	SSN	160	3.26	18.6	18.1	103.2
1B	SSN, matched params	160	4.14	19.1	23.0	106.2
1B	ResBM	160	3.14	18.4	17.5	102.5
1B	MAPL	160	3.43	18.5	19.0	102.9
1B	SSN++	160	4.97	19.6	27.6	109.0
1B	SSN++ INT8	164	5.15	20.0	28.6	111.0
1B	SSN++ INT8/INT4	160	5.27	20.3	29.3	112.6
2B	Dense	10240	5.25	660.6	55.4	6,973.5
2B	SSN	204	5.73	24.0	60.5	253.1
2B	SSN, matched params	204	7.18	24.7	75.8	261.2
2B	ResBM	200	5.53	23.6	58.4	248.8
2B	MAPL	200	6.03	23.7	63.6	250.0
2B	SSN++	200	9.60	25.2	101.3	266.5
2B	SSN++ INT8	204	10.10	25.8	106.6	272.6
2B	SSN++ INT8/INT4	199	10.45	26.3	110.3	278.0

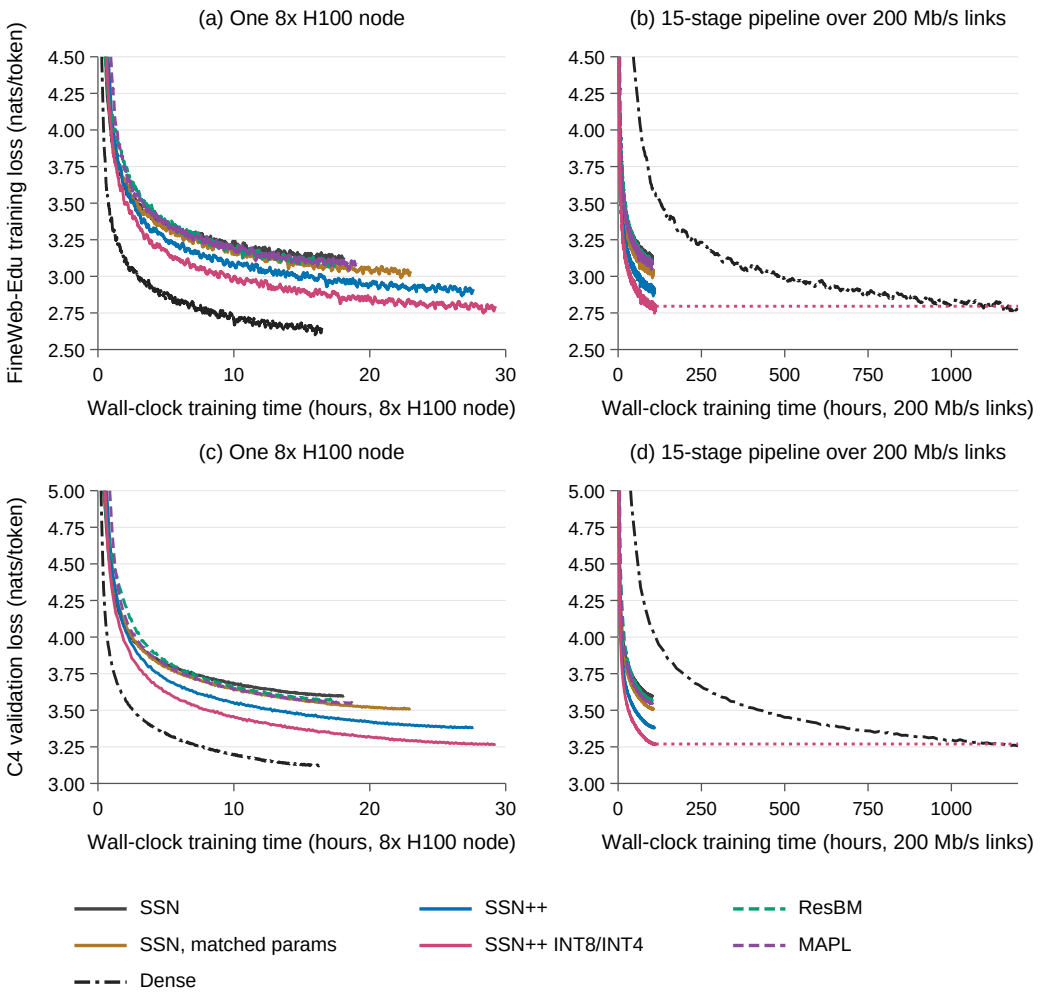


Figure 4: FineWeb-Edu training loss, as a mean over the preceding 100 updates (top), and C4 validation loss (bottom) against measured training time at 1B, on one eight-H100 node (left) and as a 15-stage pipeline on two nodes over 200 Mb/s links (right). The dotted lines extend the final losses of SSN++ INT8/INT4, which the dense transformer reaches over the links after 1,094 and 1,161 hours. The loss axes are cut to show the final differences.

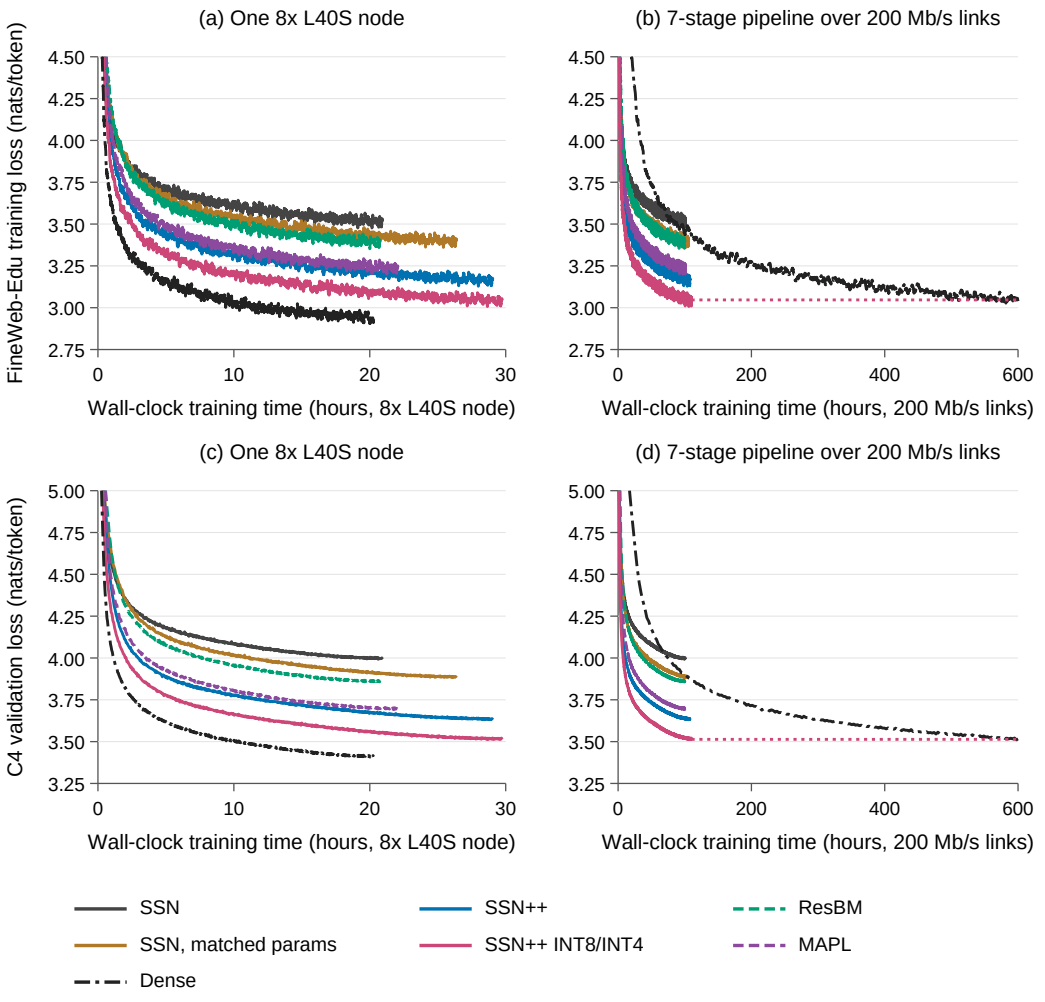


Figure 5: FineWeb-Edu training loss, as a mean over the preceding 100 updates (top), and C4 validation loss (bottom) against measured training time at 229M, on one eight-L40S node (left) and as a seven-stage pipeline over 200 Mb/s links (right). The dotted lines extend the final losses of SSN++ INT8/INT4, which the dense transformer reaches over the links after 495 and 598 hours. The loss axes are cut to show the final differences.

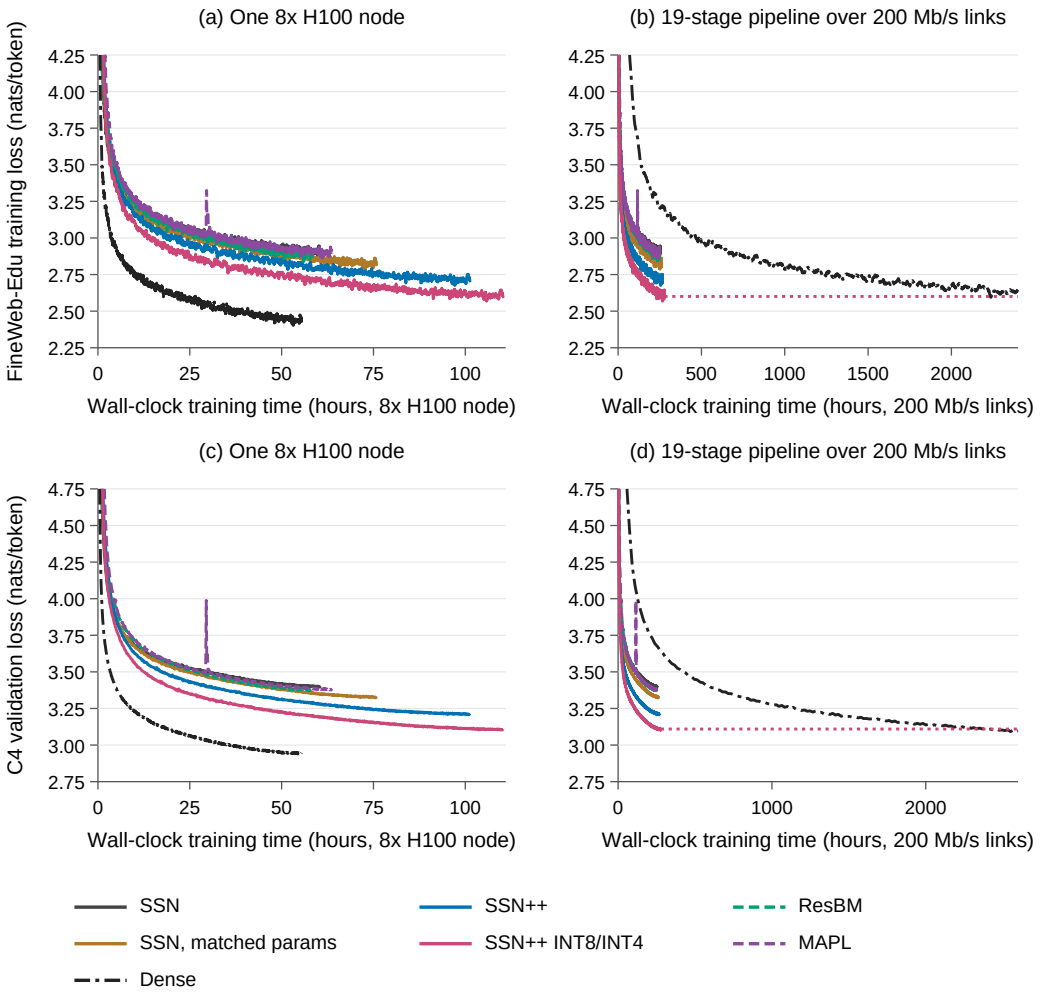


Figure 6: FineWeb-Edu training loss, as a mean over the preceding 100 updates (top), and C4 validation loss (bottom) against measured training time at 2B, on one eight-H100 node (left) and as a 19-stage pipeline on three nodes over 200 Mb/s links (right). The dotted lines extend the final losses of SSN++ INT8/INT4, which the dense transformer reaches over the links after 2,235 and 2,404 hours. The loss axes are cut to show the final differences.

G ADDITIONAL RESULTS

This appendix collects the results that the main text summarises. Appendix G.1 breaks the mean eval accuracy into its four tasks and follows each task across training checkpoints, Appendix G.2 describes the separate 10,000-update 1B study and compares communication precision and rank within it, and Appendix G.3 reports its sequence of architectural additions.

G.1 EVAL ACCURACY BY TASK

Table 7 breaks the mean eval accuracy of Table 2 into its four tasks, each scored by normalised accuracy on the full benchmark split (Appendix F.3). SSN++ improves on the parameter-matched SSN on ARC-Easy and HellaSwag at 229M and 1B, by 3.9 to 4.3 and 1.5 to 2.1 points, and with quantised communication at higher rank the HellaSwag gain grows to 2.5 to 5.1 points. ARC-Challenge and PIQA are noisier, with standard errors of about 1.3 and 1.1 points against 0.5 for HellaSwag, and the only case in which an SSN++ model falls below the matched SSN, on ARC-Challenge at 1B, is smaller than one standard error. At 2B, SSN++ improves on the matched SSN by 2.6 points on ARC-Easy and 3.1 on HellaSwag, SSN++ INT8 by 2.8 and 6.5, and SSN++ INT8/INT4 by 4.8 and 7.5.

Figure 7 follows each task across the evaluated checkpoints at the three sizes. At 229M and 1B, SSN++ INT8/INT4 scores above every compressed baseline on ARC-Easy, PIQA and HellaSwag at every checkpoint, and on ARC-Challenge at 17 of the 20; at 2B it does so on ARC-Easy, ARC-Challenge and HellaSwag at every checkpoint and on PIQA at 12 of the 13. The dense transformer stays ahead of every compressed model on ARC-Easy, PIQA and HellaSwag throughout training at 1B, and of every compressed model on all four tasks at 2B.

Table 7: Eval accuracy (%) by task. The mean is the four-task average of Table 2.

Size	Model	ARC-Easy	ARC-Challenge	PIQA	HellaSwag	Mean
229M	Dense	50.55	26.02	63.87	34.26	43.68
	SSN	41.04	22.78	59.03	28.69	37.88
	SSN, matched params	41.41	22.78	60.23	29.71	38.53
	ResBM	44.07	22.44	58.87	28.56	38.48
	MAPL	44.82	23.38	60.50	30.20	39.73
	SSN++	45.66	24.66	62.35	31.18	40.96
	SSN++ INT8	46.42	26.28	**63.66**	32.21	42.14
	SSN++ INT8/INT4	**46.46**	**26.79**	63.38	**32.44**	**42.27**
1B	Dense	55.72	30.55	70.08	43.60	49.99
	SSN	46.46	25.85	62.62	31.76	41.67
	SSN, matched params	45.58	26.02	63.55	33.65	42.20
	ResBM	47.10	25.68	62.24	32.12	41.78
	MAPL	46.46	24.83	63.06	32.09	41.61
	SSN++	49.49	25.68	65.61	35.76	44.14
	SSN++ INT8	50.93	**28.33**	66.43	38.56	46.06
	SSN++ INT8/INT4	**51.73**	27.56	**66.65**	**38.76**	**46.17**
2B	Dense	63.09	37.37	73.01	53.55	56.76
	SSN	47.47	26.02	66.54	35.88	43.98
	SSN, matched params	51.14	27.65	66.59	38.38	45.94
	ResBM	50.04	27.90	65.78	36.00	44.93
	MAPL	49.87	26.71	64.42	35.64	44.16
	SSN++	53.70	28.84	68.55	41.46	48.14
	SSN++ INT8	53.96	28.33	68.82	44.92	49.01
	SSN++ INT8/INT4	**55.89**	**29.86**	**69.31**	**45.84**	**50.23**

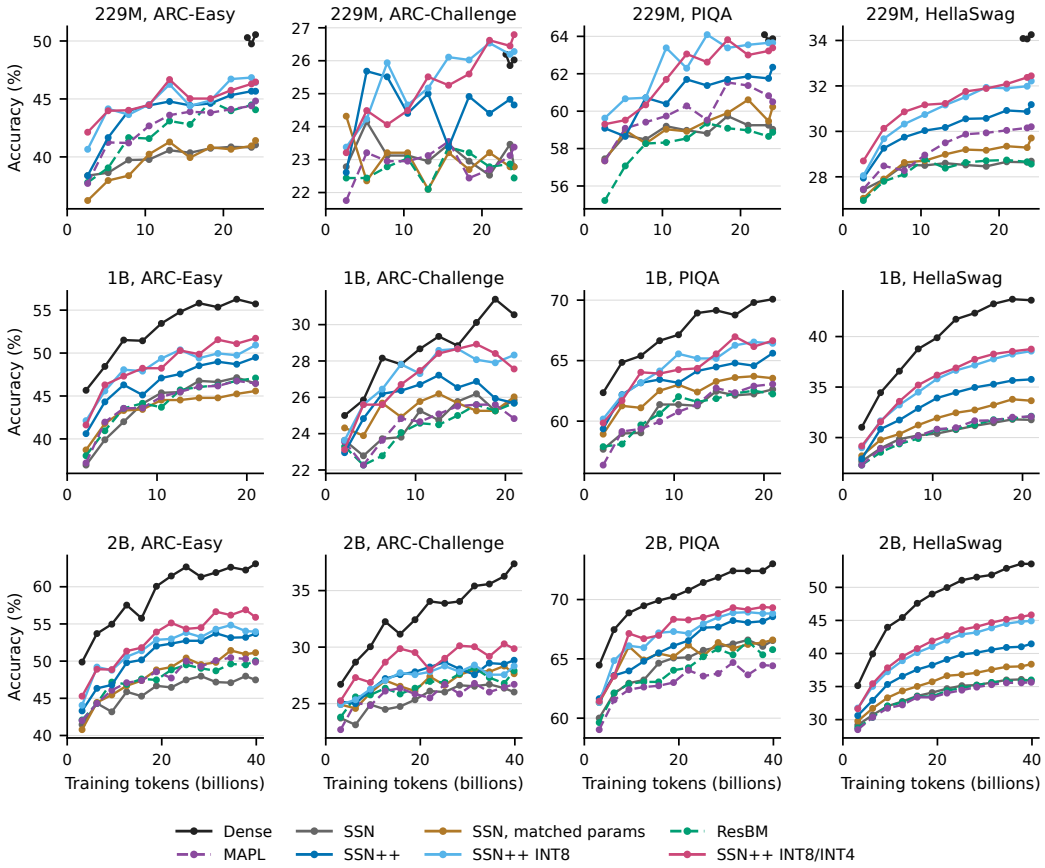


Figure 7: Eval accuracy on each task across the evaluated checkpoints, at 229M (every 5,000 updates and the final update), 1B (every 2,000 updates), and 2B (every 3,000 updates and the final update). For the 229M dense transformer, we provide only the last three checkpoints. Standard errors are about 1.0 points on ARC-Easy, 1.3 on ARC-Challenge, 1.1 on PIQA, and 0.5 on HellaSwag.

G.2 1B ABLATION STUDY

The ablations, the fixed-rank controls of communication precision, and the training dynamics of Appendix J come from a separate set of 1B models trained for 10,000 updates, or 10.49B tokens, with the 1B configuration of Appendix F.1 and the training setup of Appendix F.3. Every model in the study completes this schedule, so all share a common endpoint. Losses average the six C4 validation evaluations at updates 9500 to 10,000 and the 51 logged training losses at updates $9500, 9510, \ldots, 10000$, without smoothing, minimum selection, or interpolation across runs, and the validation loss at update 10,000 differs from this mean by at most 0.006 nats. Table 8 compares communication precision at the fixed rank $k = 40$ with the higher ranks that the same bytes allow.

Table 8: Cross-entropy loss (nats/token) in the 10,000-update 1B study on FineWeb-Edu training data and C4 validation data, averaged over updates 9500 to 10,000. F/B gives the communication precision of the activations and of their gradients, and bytes are per token and stage boundary. The last column is the difference in C4 loss from the SSN, computed from the unrounded losses.

Model	k	Bits F/B	Bytes F+B	FineWeb	C4	Δ C4 vs SSN (nats)
SSN	40	16/16	160	3.2238	3.6941	—
SSN, matched params	40	16/16	160	3.1426	3.6215	-0.0726
SSN++	40	16/16	160	3.0255	3.4993	-0.1948
SSN++ INT8	40	8/8	84	3.0274	3.5016	-0.1926
SSN++ INT8/INT4	40	8/4	64	3.0325	3.5056	-0.1885
SSN++ INT8	80	8/8	164	2.9373	3.4116	-0.2825
SSN++ INT8/INT4	104	8/4	160	**2.9102**	**3.3843**	$\mathbf{-0.3098}$

G.3 ARCHITECTURAL ABLATIONS

Table 9 adds the components of SSN++ to the SSN in stages within the 10,000-update study, keeping rank $k = 40$ and BF16 communication, so that every row communicates 160 bytes per token and stage boundary.

Table 9: Cumulative architectural additions in the 10,000-update 1B study, each row keeping the preceding additions. C4 loss averages the evaluations at updates 9500 to 10,000, and the last column is the change from the previous row, computed from the unrounded losses.

Architectural addition	Params (M)	Loss	Δ vs previous (nats)
SSN	800.684	3.6941	—
Layer and sublayer subspaces, grouped attention	800.684	3.6566	-0.0376
Grouped MLP outputs and aggregation gates, $\delta = 1$	800.686	3.6017	-0.0549
Temporal decoder, attention and MLP decoders	801.291	3.5799	-0.0218
Embedding reparametrisation and SwiGLU values	1077.801	3.4976	-0.0822
Value-input RMS normalisation	1077.801	3.4993	$+0.0016$

Interpretation of the additions. The first comparison jointly relaxes the shared-subspace constraint across layers and across the attention and MLP sublayers and adds grouped attention, so it does not isolate those choices. Adding eight MLP groups and aggregation gates for both sublayers with amplitude $\delta = 1$ jointly reduces loss by 0.0549 nats. The temporal decoder and the attention and MLP decoders reduce it by a further 0.0218 nats. Those decoders include terms in the current compressed activation as well as lagged terms, so this comparison does not isolate direct access to history. The next addition combines embedding reparametrisation with width-5688 SwiGLU value networks and lowers the loss by 0.0822 nats while adding 276,509,864 parameters. This comparison does not separate the two, and the parameter-matched SSN of Table 8 controls for the added value capacity when the complete SSN++ is evaluated.

Value-input normalisation. Normalising the input of the value network adds no parameters and mainly speeds up early training. C4 loss is lower by 0.051, 0.025, and 0.005 nats at updates 500,

1000, and 2000, and by 0.034 and 0.033 nats on average over the 11 validation evaluations and the 101 logged training losses through update 1000, although the unnormalised model is ahead at update 200. The final-window means differ by only 0.0016 nats.

Parameter accounting. Starting from 800,684,032 trainable parameters, the aggregation gates add $14 \cdot 2 \cdot (8^2 + 8) = 2016$, and the temporal decoder with the attention and MLP decoders adds $3 \cdot 14 \cdot 9 \cdot 40^2 = 604{,}800$. Embedding reparametrisation removes $50{,}257(2048 - 40) = 100{,}916{,}056$ parameters, and SwiGLU values add $15[5688(2 \cdot 2048 + 512) - 2048 \cdot 512] = 377{,}425{,}920$. The fixed orthonormal bases are not trainable and are excluded from these counts.

H Geometry and interpretation of superposition measurements

The normalised columns of an output projection form a dictionary of output directions, and its geometry shows how the learned contributions of a layer share the space they write into. The overlaps of these directions can be measured without identifying them as semantic features (Elhage et al., 2022), and covariance rank likewise describes how activity energy spreads rather than how many concepts are represented. Covariance alone also misses sparsity, since a standard Gaussian vector and a coordinate vector chosen uniformly at random, with a random sign and scaled by $\sqrt{k}$, both have covariance I_k. The measurements therefore characterise how tightly the directions pack, how many features are active at once, and how the activations are distributed. The first two paragraphs below compare directions across the local, ambient, and aggregated spaces and connect packing to the approximation objective of Appendix C through signed moments. The next two separate average from maximal overlap with the Welch bound and define the two superposition measures of Section 2, with a random-sign model that makes their combined interference an exact decoding error. Appendix J applies these statistics through training.

Local, ambient, and aggregated directions. Let $f_{gi} \in \mathbb{R}^k$ be a unit direction in group g's local dictionary, with ambient orthonormal basis $U_g \in \mathbb{R}^{d \times k}$. Its ambient direction is $U_g f_{gi}$, while its ungated direction in the aggregated compressed activation is $R_g f_{gi}$, where $R_g = T_g^\top \in O(k)$ under the convention that orthogonal transforms act on row vectors. The local dictionaries, their ambient union, and their aggregated dictionary in the space of compressed activations therefore define three distinct objects. Across groups,

$$\langle U_g f_{gi}, U_h f_{hj} \rangle = f_{gi}^\top U_g^\top U_h f_{hj}, \qquad \langle R_g f_{gi}, R_h f_{hj} \rangle = f_{gi}^\top R_g^\top R_h f_{hj}. \tag{55}$$

Cross-group comparisons depend on the common space in which the directions are expressed, since independently chosen local bases assign different meanings to the same coordinates. A common orthogonal change $F \mapsto QF$, $z \mapsto Qz$ preserves the Gram matrices of the directions, norms, covariance spectra, and covariance overlap when both arguments are transformed. Indeed, $(QF)^\top QF = F^\top F$ and transformed covariances are $QCQ^\top$. Trace cyclicity proves the remaining claims. Independent orthogonal transforms of different groups need not preserve cross-group overlaps.

Signed moments connect packing to approximation. Let $z = Fa$, with unit columns f_i of $F \in \mathbb{R}^{k \times N}$. The signed coefficients a_i include activation magnitudes, the column norms of the output projection and, when appropriate, scalar normalisation factors.

Proposition H.1 (Full and diagonal moments)**.** *For fixed F and square-integrable a, put $M_a = \mathbb{E}[aa^\top]$ and $C_a = \mathrm{Cov}(a)$. Then*

$$M_z = F M_a F^\top, \qquad C_z = F C_a F^\top, \qquad M_z = C_z + (F\mathbb{E}a)(F\mathbb{E}a)^\top. \tag{56}$$

The diagonal proxy $\sum_i \mathbb{E}[a_i^2] f_i f_i^\top$ omits $\sum_{i \neq j} \mathbb{E}[a_i a_j] f_i f_j^\top$.

Proof. Expand $(Fa)(Fa)^\top$ and take expectations. Applying the same argument to $a - \mathbb{E}a$ proves the covariance identity; splitting M_a into its diagonal and off-diagonal entries proves the final statement. □

Thus the exact spectral-tail objective in Theorem C.3 uses full, uncentred target moments. A normalised weighted sum of the outer products of the directions, which keeps only diagonal terms, instead summarises marginal contributions while omitting their signed interactions. For the operators $S_b = \sum_i p_{bi} f_{bi} f_{bi}^\top$ of the attention and MLP sublayers, expressed in a common basis, the normalised overlap is $\operatorname{tr}(S_A S_M)/\sqrt{\operatorname{tr}(S_A^2)\operatorname{tr}(S_M^2)}$. This compares marginal weighted directions, not the signed cross-covariance between the activations of the two sublayers.

Tightness, coherence, and signed interference. For unit directions let $\rho_{ij} = f_i^\top f_j$ and $\mu = \max_{i \neq j} |\rho_{ij}|$. Their distinction is explicit in the Welch bound (Welch, 1974).

Proposition H.2 (Average and maximum coherence)**.** *If $N > k$, then*

$$\mu^2 \;\geq\; \frac{\sum_{i \neq j} \rho_{ij}^2}{N(N-1)} \;\geq\; \frac{N-k}{k(N-1)}. \tag{57}$$

Equality in the second inequality holds exactly when $FF^\top = (N/k) I_k$. Equality throughout additionally requires equal absolute off-diagonal inner products.

Proof. For $S = FF^\top$, $\operatorname{tr} S = N$ and $\sum_{ij} \rho_{ij}^2 = \operatorname{tr}(S^2) \geq N^2/k$ by Cauchy–Schwarz on its k eigenvalues. Equality requires all eigenvalues to equal N/k. Subtract the N diagonal terms and divide by $N(N-1)$. The maximum is at least the average, with equality exactly when all terms in that average coincide. □

A tight dictionary need not be equiangular, as repeating each vector of an orthonormal basis twice gives a tight dictionary with $\mu = 1$. Low average squared overlap therefore does not exclude highly overlapping pairs. Moreover,

$$\|Fa\|_2^2 = \sum_i a_i^2 + 2 \sum_{i<j} a_i a_j \rho_{ij}. \tag{58}$$

Squared-overlap metrics discard these signs and correlations. The same per-direction energies can produce either constructive or destructive interaction.

Active features per dimension and the Welch ratio. Write the MLP output projection as $W = UB$ with $B \in \mathbb{R}^{k \times p}$, so that hidden unit i writes along the unit direction $f_i = b_i/\|b_i\|_2$ of its column b_i, and let $h_{i,t}$ be its activation at token t. The energy shares $p_{i,t} = h_{i,t}^2 \|b_i\|_2^2 / \sum_j h_{j,t}^2 \|b_j\|_2^2$ give the number of active features per dimension, and the N directions of the nonzero columns give the Welch ratio,

$$\bar{s} = \frac{1}{k}\, \mathbb{E}_t\Big[\Big(\sum_i p_{i,t}^2\Big)^{-1}\Big], \qquad W = \frac{1}{N(N-1)} \sum_{i \neq j} \rho_{ij}^2 \Big/ \frac{N-k}{k(N-1)}, \tag{59}$$

where the expectation runs over the tokens with a nonzero output. By Proposition H.2, $W \geq 1$, with equality exactly for a tight dictionary. The two measures matter together because interference depends on both. For a nonzero token with energy shares p_i, the statistic $I(p, F) = \sum_{i \neq j} p_i p_j \rho_{ij}^2$ weights the squared overlap of every pair of directions by how strongly the pair is active at once, and the following reference model makes it an exact decoding error.

Proposition H.3 (Random-sign reference model)**.** *Fix F and $p_i \geq 0$ with $\sum_i p_i = 1$. Set $a_i = \epsilon_i \sqrt{p_i}$ with independent equiprobable signs, and use the hypothetical matched-filter decoder $\widehat{a} = F^\top F a$. Then*

$$\mathbb{E}_\epsilon \sum_i p_i (\widehat{a}_i - a_i)^2 = I(p, F). \tag{60}$$

If the unit directions are also independent isotropic draws, independently of p, then $\mathbb{E}_F I(p, F) = (1 - \sum_i p_i^2)/k$.

Proof. Unit diagonal entries give $\widehat{a}_i - a_i = \sum_{j \neq i} \rho_{ij} \epsilon_j \sqrt{p_j}$. Independence and zero sign means eliminate cross terms after squaring. Multiplying by p_i and summing proves Eq. (60). For independent isotropic unit directions, $\mathbb{E}[\rho_{ij}^2] = \operatorname{tr}((I_k/k)^2) = 1/k$; sum $p_i p_j$ over $i \neq j$ to obtain the reference value. □

Interference therefore grows with the overlap of the directions and, for random directions, with the number of active features $1/\sum_i p_i^2$, while it falls as $1/k$ with the dimension. The model assumes independent signs and a matched-filter decoder, so it interprets the statistic rather than predicting the loss of the trained network.

I Representation analysis underlying the architectural choices

The architectural development was informed by measurements of the representations produced by trained networks. The measurements examine which computations have sufficiently distinct structure to benefit from their own subspaces and which local features remain restricted after those subspaces have been introduced. Appendix I.1 compares superposition under a shared subspace, one subspace per layer, and separate attention and MLP subspaces in each layer, which motivates the separate attention and MLP subspaces of Section 3.1. Appendix I.2 clusters attention heads by the affinities of their outputs and recovers the GQA groups used in Section 3.2. Appendix I.3 measures the value spectra behind the nonlinear values of Section 3.3, and Appendix J.5 compares their attention sinks with those of a dense transformer. These measurements concern development models that precede the final comparisons, whereas Appendix J follows the completed architecture through training.

I.1 Superposition across the subspace hierarchy

Before grouping heads, we measured how the subspace structure itself shapes superposition. Three 1B development models with $k = 40$ were trained with a common recipe to 10,000 updates and differ only in their subspaces. They use a shared subspace, one subspace per layer, or separate attention and MLP subspaces in each layer. The model with separate attention and MLP subspaces has none of the other SSN++ components, such as the grouped output projections, the aggregation gates, the temporal decoder, or the nonlinear values, so the comparison isolates the subspaces. The final checkpoints of the three models were evaluated on a frozen measurement corpus, and all values are means over the compressed layers. The number of active features per dimension and the Welch ratio of the MLP output are those of Eq. (59). A number of active features per dimension above one means more active features than the compressed activation has dimensions, and a lower Welch ratio means less interference, with one the tightest possible packing. The single-direction ablation removes, at every stage boundary, the direction that the attention and MLP outputs share most strongly, and reports the loss increase in excess of removing a random direction. The utilisation of the compressed activation is the participation ratio of its covariance divided by k, and the activation–gradient alignment compares the covariances of the compressed activation and of its gradient.

Table 10: Superposition in three 1B development models that differ only in their subspaces, as means over the compressed layers. The columns correspond to a shared subspace, one subspace per layer, and separate attention and MLP subspaces in each layer. The per-sublayer model has none of the other SSN++ components.

	Shared	Per layer	Per sublayer
Active features per dimension	1.44	1.52	1.52
Welch ratio of the MLP output	1.33	1.25	1.13
Output-direction pairs with $\lvert\cos\rvert > 0.25$ (%)	45.7	39.0	31.9
Excess loss from removing one direction (nats)	4.36	3.93	0.55
Utilisation r_{PR}/k of the compressed activation	0.68	0.67	0.76
Activation–gradient alignment	0.83	0.83	0.87
Loss on the measurement corpus (nats)	3.539	3.549	3.422

Giving each layer its own subspace lets the compressed activation superpose more features with less interference (Figure 8). With one subspace per layer, the model carries 1.52 active features per dimension against 1.44 with a shared subspace, packs them with a Welch ratio of 1.25 against 1.33, and has 39.0% rather than 45.7% of its output-direction pairs overlapping by more than 0.25 in absolute cosine. This better superposition alone leaves the use of the compressed activation and the loss unchanged. The utilisation and alignment of the model with one subspace per layer match those of the model with a shared subspace, its loss on the measurement corpus is within 0.01 nats of that

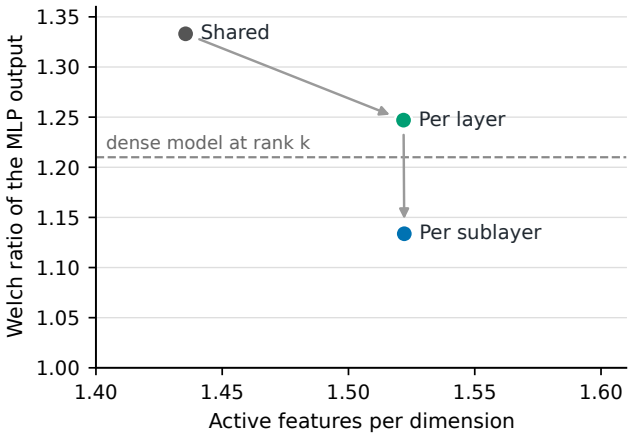


Figure 8: Superposition with shared, per-layer, and per-sublayer subspaces for the models of Table 10. Each point plots how many features the compressed activation carries at once against how tightly the MLP output packs them, where a lower Welch ratio means less interference, and the arrows follow the order in which the subspaces are separated. The dashed line is the dense model's MLP output projected to the same rank.

model's, and both route so much of the compressed activation through one direction that removing it costs about four nats more than removing a random one. Separate attention and MLP subspaces in each layer go further with the same number of active features per dimension. The Welch ratio falls to 1.13, below the 1.21 of the dense model's MLP output projected to the same rank, the utilisation and alignment rise, the critical direction disappears, and the loss falls by 0.13 nats. The freedom that improves superposition across depth therefore pays off in loss once it reaches the computations inside each layer, which motivates the separate attention and MLP subspaces of SSN++.

I.2 Attention groups and the structure of head outputs

For query head h, let $z_{h,t}$ denote its contribution to the compressed attention output at token t. Two heads can be related geometrically, through the directions they use, or statistically, through the tokens on which they are active. We measure these relationships using the geometric and co-activation affinities

$$A^{\mathrm{geo}}_{hh'} = \frac{\mathrm{tr}(C_h C_{h'})}{\sqrt{\mathrm{tr}(C_h^2)\,\mathrm{tr}(C_{h'}^2)}}, \qquad A^{\mathrm{co}}_{hh'} = \mathrm{corr}_t\big(\|z_{h,t}\|^2, \|z_{h',t}\|^2\big), \qquad C_h = \mathrm{Cov}_t(z_{h,t}).$$

The head outputs are compared in common coordinates. Both statistics are invariant to a common orthogonal change of coordinates, and the orthogonal transforms of the groups must therefore be applied before comparing heads with different subspaces.

The grouping study evaluates $G \in \{2, 4, 8, 12, 16\}$ for nine architectural variants with 32 query heads. Symmetrised affinities, with negative entries clipped to zero and diagonal entries removed, define a normalised graph Laplacian. Spectral clustering (Ng et al., 2001) uses its G smallest-eigenvalue eigenvectors, row normalisation, and seeded k-means with 20 restarts. A consensus partition is obtained by clustering the pairwise co-membership matrix across layer and architecture partitions. Its agreement is the mean adjusted Rand index (ARI) with those partitions. The two stability measures average pairwise ARIs across layers within each architecture, and across architectures at each common layer, respectively.

Figure 9 summarises the grouping measurements. For the co-activation affinity, the consensus ARIs at $G = 2, 4, 8, 12, 16$ are 0.066, 0.184, 0.566, 0.457, and 0.249. Eight groups achieve the strongest agreement, with layer and architecture stability of 0.323 and 0.364. The resulting partition is exactly the eight contiguous groups of four query heads that share keys and values. The geometric affinity recovers the same eight-group partition, although its agreement with individual partitions is weaker. This correspondence motivates giving each GQA group its own attention subspace. It does not

necessarily establish, however, that eight groups minimise training loss, since the study measures representation structure rather than retraining every candidate grouping.

An analogous study of MLP features provides a useful contrast. It does not recover a stable discrete group count, with agreement between corresponding feature partitions across trained variants remaining approximately 0.00–0.03. We therefore use eight equal MLP groups as an architectural choice rather than as an identified set of feature classes. Moreover, per-token and corpus-aggregate effective group counts are similar in the ungated networks. The motivation for gated aggregation is to learn how simultaneously active group contributions are combined, not to reproduce a pre-existing sparse routing pattern. Grouping preserves local width, while a gate makes the combination into the narrow compressed activation depend on its content.

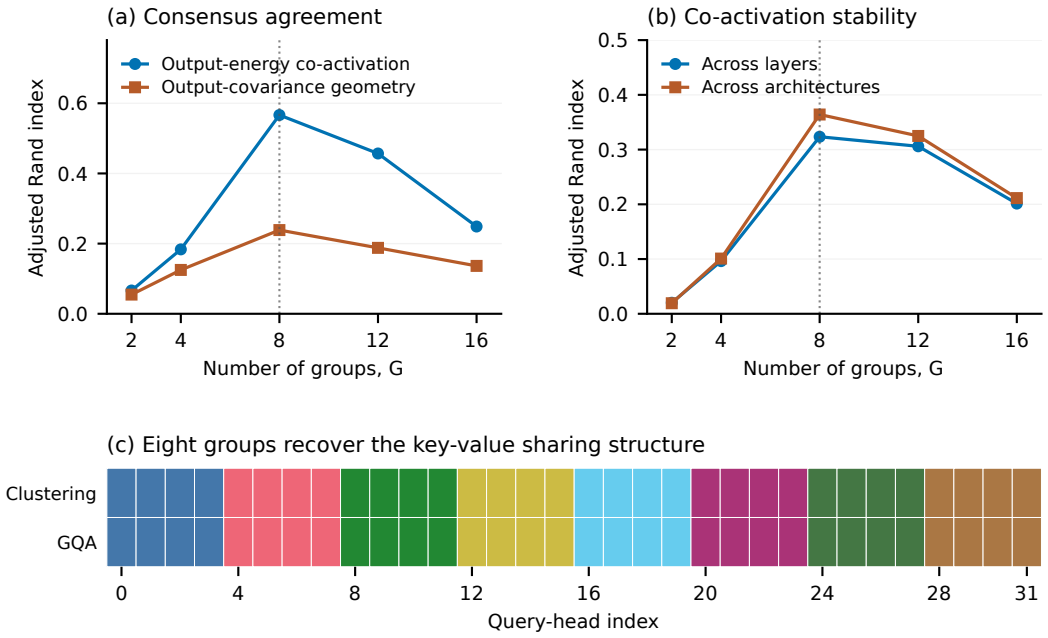


Figure 9: Head-output affinities recover the GQA groups. Panels (a,b) report partition agreement and stability, measured by adjusted Rand index, as a function of the number of clusters, and both peak at eight. Panel (c) compares the consensus eight-group partition (top) with the GQA key–value assignment (bottom), one cell per query head. Cluster labels are arbitrary, so the colours are matched between the rows and only which heads share a colour carries information, since a cluster with the right size but the wrong members would show its colour in separate segments of the row. The rows coincide for all 32 heads, so each recovered cluster is exactly one GQA group. The indices in (a,b) below one quantify how far individual layers and architecture variants deviate from this consensus. These measurements identify representation structure rather than the training-loss optimum over group counts.

I.3 Nonlinear values and activation spectra

The value comparison evaluates two trained configurations on the same 8×1024 C4 tokens after 10,000 updates. The nonlinear configuration precedes the addition of value-input normalisation. At each layer, the matrix $V \in \mathbb{R}^{8192\times512}$ contains the uncentred value activations before replication across query heads. We distinguish the stable rank from the singular-value entropy rank, defined respectively as

$$\mathrm{sr}(V) = \frac{\sum_i \sigma_i^2}{\sigma_1^2}, \qquad \mathrm{erank}(V) = \exp\left(-\sum_i p_i \log p_i\right), \quad p_i = \frac{\sigma_i}{\sum_j \sigma_j}.$$

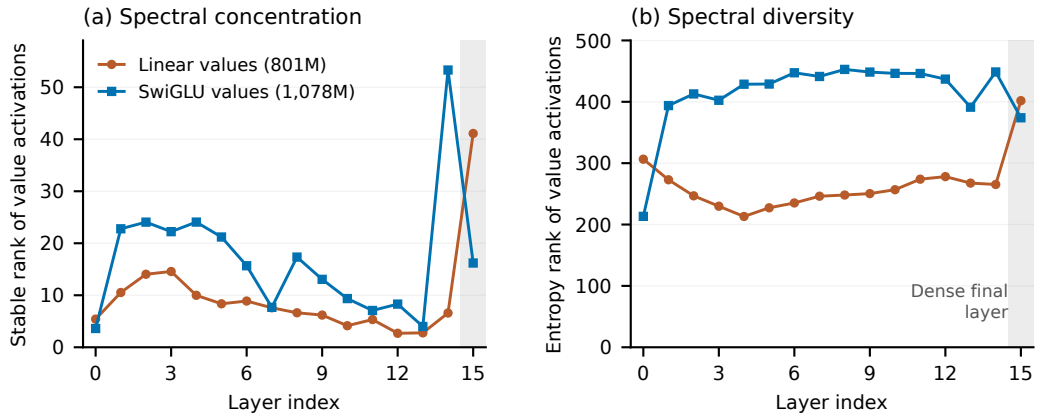


Figure 10: Nonlinear value features broaden the local activation spectrum. Stable rank and singular-value entropy rank are measured on the same eight sequences of 1,024 C4 tokens, and the shaded region marks the dense final layer. Lines show per-layer measurements for the 801M linear and 1,078M nonlinear variants, and the nonlinear model in this diagnostic precedes the addition of value-input normalisation. Parameter count and embedding parameterisation also change in this comparison, so the spectra establish a difference in the learned value representations rather than isolating the effect of nonlinearity on prediction loss. The text below gives the definitions.

These are activation statistics, distinct from the weight-matrix rank measurements that motivate compression in the original SSN work (Ramasinghe et al., 2025). They do not compare the ranks of values, queries, and output weights.

Figure 10 shows that nonlinear values have a less concentrated spectrum through most of the compressed depth. Averaged over layers 0–14, stable rank increases from 7.58 to 16.92, and entropy rank from 254.53 to 415.96. Stable rank increases in 14 of the 15 compressed layers. At the dense final layer, where both configurations retain a linear value map, the corresponding stable ranks are 41.11 and 16.20. The two models' inputs and trained weights differ, so the dense final layer offers a comparison of the same module type rather than a controlled intervention.

For a linear value map, $\mathrm{rank}(XW_V) \leq \mathrm{rank}(X)$ is an algebraic restriction, but it does not bound stable rank by the input's stable rank. A nonlinear map can generate linearly independent functions of the received representation without increasing the number of transmitted coefficients. The measured spectral change is consistent with this source of local capacity. It does not imply that missing information is recovered from the compressed activation.

The linear-value and nonlinear-value configurations have approximately 801M and 1,078M trainable parameters, respectively, and the latter also changes the embedding parameterisation. Their loss contrast therefore measures the combined allocation of local capacity, not an isolated effect of nonlinearity. The spectral measurements describe one batch per model.

J Training Dynamics

This appendix follows the 1B models through training with five families of measurements, each aimed at one part of the architecture, and with interventions that switch off individual components. Appendix J.1 gives the protocol that all measurements share. Appendix J.2 measures the capacity of the local outputs against the transmitted compressed activation, Appendix J.3 the gated aggregation, Appendix J.4 the decoders at the receiving layer, Appendix J.5 the value features and attention sinks, and Appendix J.6 the quantised communication and its gradients. Appendix J.7 reports the interventions. Figures 11, 12, 14, and 15 show every measurement at the saved checkpoints.

J.1 PROTOCOL

Models and evaluation data. We evaluate 40 model checkpoints from four configurations of the 10,000-update 1B study (Appendix G.2). These are the parameter-matched SSN at $k = 40$, SSN++ at $k = 40$ with BF16 communication, SSN++ INT8/INT4 at $k = 104$ with INT8 activations and INT4 gradients, and the fixed-rank SSN++ INT8/INT4 control at $k = 40$. Each configuration is evaluated at updates $1000, 2000, \ldots, 10000$. The first three configurations also have 30 value and attention evaluations and nine intervention evaluations, the latter at updates 1000, 5000, and 10000. The measurement corpus is a fixed set of 977 FineWeb-Edu evaluation sequences of length 1024. The dynamics measurements use the complete corpus, the value and attention analysis uses its first 128 sequences, and the interventions and gradient analysis use its first 32. FineWeb cross-entropy is evaluated on these tokens and is distinct from the C4 validation loss in the main performance comparison.

Numerical precision and measurement validation. The dynamics measurements, the value and attention measurements, and the interventions use FP32 forward computation, whereas the gradient measurements use BF16 autocast over FP32 stored parameters. Each model is evaluated with the communication format used in its training. No GEMM quantisation is evaluated. Checkpoint restoration requires matching model-tensor keys and shapes, allowing deterministic regeneration of the RoPE buffer. All 40 evaluations satisfy the reconstruction identities to the stated numerical tolerances, with maximum relative residual below 10^{-6}. Instrumented and uninstrumented forward passes produce bitwise-equal logits on an eight-sequence prefix at each checkpoint. This comparison covers eight sequences per checkpoint, whereas moment estimation uses all 977 sequences. Random-generator state, model parameters, and buffers remain unchanged during measurement in all 40 evaluations.

Layer indexing and aggregation. Indices are zero-based. The main summaries of the ambient outputs average over layers 0–13, the layers that transmit a compressed activation, and over the attention and MLP output projections with equal weight. They are averages of per-projection ratios, not ratios of energies pooled across output projections. The measurements also include the last compressed layer, 14, whose output passes to the dense final layer without another gated aggregation. Projection interventions include layers 0–14. Measurements of the temporal decoder and of the attention and MLP decoders cover the receiving layers 1–14.

J.2 LOCAL CAPACITY AND THE TRANSMITTED ACTIVATION

The local outputs and the compressed activation that crosses each stage boundary evolve differently (Figures 3a and 11a). Between updates 1,000 and 10,000 the participation ratio of the local outputs rises from 34.6 to 57.4 in SSN++ at $k = 40$ and from 58.0 to 98.5 in SSN++ INT8/INT4 at $k = 104$, whereas the participation ratio of the activation received at the next layer stays nearly constant, moving from 34.2 to 33.1 and from 77.1 to 75.6. In the matched SSN the participation ratio of the outputs stays between 20.0 and 22.3, and that of the received activation declines from 32.1 to 30.4. The local outputs also spread beyond one subspace (Figure 3b). The fraction of their centred covariance outside the best rank-k subspace grows from 27.7% to 37.2% at $k = 40$ and from 25.1% to 31.0% at $k = 104$, and it stays numerically zero in the matched SSN, whose outputs have rank k by construction. The corresponding uncentred energy tails at update 10,000 are 7.4% and 10.8%. The capacity that training adds therefore accumulates inside the layers while the compressed activation keeps the same effective rank. In the transmitted activation the attention and MLP contributions become increasingly anti-aligned, with their signed cosine falling from -0.29 to -0.59 at $k = 40$ and from -0.21 to -0.42 at $k = 104$ (Figure 11d). In the matched SSN the cosine starts lower and ends at -0.77, so the two sublayers cancel each other in the transmitted activation most strongly when they share one subspace.

Definitions. For a normalised ambient output y, define $M = \mathbb{E}[yy^\top]$ and $C = M - (\mathbb{E}y)(\mathbb{E}y)^\top$, where expectations are empirical averages over selected token positions. For positive trace, define the spectral tails

$$\tau_k^{\mathrm{cov}} = 1 - \frac{\sum_{i=1}^{k} \lambda_i(C)}{\operatorname{tr} C}, \qquad \tau_k^{\mathrm{raw}} = 1 - \frac{\sum_{i=1}^{k} \lambda_i(M)}{\operatorname{tr} M}, \tag{61}$$

with eigenvalues in descending order. Moments are accumulated in FP64, and negative roundoff eigenvalues are clipped for these ratios. The covariance tail measures centred variation outside the best fixed k-dimensional subspace. The raw tail measures residual uncentred energy. Neither counts semantic features or establishes a communication lower bound for the task.

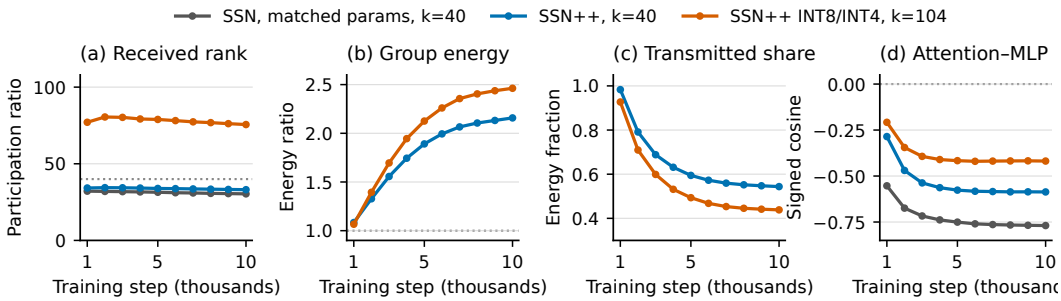


Figure 11: The transmitted compressed activation against the capacity inside the layers, whose participation ratio and covariance beyond rank k are panels (a) and (b) of Figure 3. (a) Participation ratio of the compressed activation received by the next layer, with the rank $k = 40$ dotted. (b) Summed energy of the group coefficients relative to the transmitted activation, with one dotted. (c) Share of the ambient output's energy that is transmitted. (d) Signed cosine between the attention and MLP contributions to the transmitted activation. Aggregation statistics average layers 0 to 13, and statistics of the received activation average receiving layers 1 to 14.

J.3 Gated aggregation

The gated energy of the eight groups concentrates within the first thousand updates, reaching $\sum_g p_g^2 = 0.73$ at $k = 40$ and 0.50 at $k = 104$ against $1/8$ for equal use (Figure 12d). The effective number of groups, the inverse of this concentration, is therefore about 1.4 and 2.0, although no group is ever unused. The aggregation gates, initialised at the neutral value of one, move steadily toward attenuation (Figure 12a). Their mean falls from 0.93 at update 1,000 to 0.68 at update 10,000 at $k = 40$ and from 0.89 to 0.59 at $k = 104$, and the fraction of the group-coefficient energy that they pass into the aggregation falls from 0.81 to 0.52 and from 0.79 to 0.41 (Figure 12b). Meanwhile the summed energy of the group coefficients rises from 1.08 to 2.16 times the energy of the transmitted activation at $k = 40$ and from 1.07 to 2.46 at $k = 104$ (Figure 11b), and the transmitted share of the ambient output's energy falls from 0.98 to 0.54 and from 0.93 to 0.44 (Figure 11c). The signed interference between the group contributions after their orthogonal transforms stays positive but falls from 0.25 to 0.11 at $k = 40$ and from 0.26 to 0.17 at $k = 104$ (Figure 12c), so the groups reinforce each other on average, and less so as training proceeds. At individual output projections the groups can interact negatively, but the means do not support globally destructive aggregation. The gated aggregation thus becomes more selective, passing a shrinking share of increasingly rich group coefficients into the compressed activation.

Definition. For gated group contributions c_g after their orthogonal transforms, the signed group interference in the aggregation is $I = (\mathbb{E}\|\sum_g c_g\|^2 - \sum_g \mathbb{E}\|c_g\|^2)/\sum_g \mathbb{E}\|c_g\|^2$. Negative and positive values indicate net destructive and constructive cross-group contributions, respectively. This estimate uses the full joint moments of the gated contributions and the model's orthogonal transforms, since input-dependent gates change the signed moments and cannot be averaged separately from them. An aggregate $I = 0$ means zero net interaction, not pairwise orthogonality. Ratios of the energies of the group coefficients, the ambient outputs, and the transmitted activation are descriptive scale statistics, not fractions of retained information.

J.4 Temporal, attention, and MLP decoders

The decoders at the receiving layer grow during the first half of training and then settle (Figure 12e). The output of the temporal decoder rises from 1.6% to 4.4% of the energy of the decoded residual stream at $k = 40$ and from 2.7% to 6.8% at $k = 104$, and the outputs of the attention and MLP

decoders rise from 4.7% to 10.9% and from 8.5% to 16.6% of the energy of the inputs to the sublayers that use them. Most of the energy of the temporal decoder's terms comes from past activations throughout. The lag-zero share rises only from 0.075 to 0.118 at $k = 40$ and from 0.14 to 0.27 at $k = 104$ (Figure 12f), the energy-weighted mean lag stays near 2.2 positions, and the cross-lag interference falls from 0.026 to -0.005, so the lagged terms become nearly orthogonal. The current activation linearly accounts for about half of the temporal decoder's output, rising from 0.47 to 0.52 at $k = 40$ and from 0.50 to 0.63 at $k = 104$. The attention decoder relies on history even more, with a lag-zero share of 0.027 and 0.065 and a current-activation score of 0.363 and 0.376 at update 10,000. These statistics document the use of lagged terms and are consistent with building features from token history, but they do not isolate the task benefit of direct access to past activations, which requires a lag-only intervention or a controlled retraining comparison.

Definitions. Let the decoder output be $s_t = \sum_{j=0}^{8} c_{j,t}$, where $c_{j,t}$ is the term at lag j, and let $H_{ij} = \mathbb{E}\langle c_{i,t}, c_{j,t}\rangle$. The lag-zero term-energy share is $H_{00}/\operatorname{tr} H$. Its denominator is the sum of individual term energies, whereas total output energy is $\mathbf{1}^\top H \mathbf{1}$ and includes cross-lag interactions. Consequently, lag shares do not partition total output energy. For the current received activation z_t, put $A = \mathbb{E}[s_t z_t^\top]$ and $S = \mathbb{E}[z_t z_t^\top]$. Define the current-activation linear-predictability score, for positive output and activation energy, as

$$P_{\mathrm{cur}} = \frac{\operatorname{tr}\left[A(S+\eta I)^{-1}A^\top\right]}{\mathbb{E}\|s_t\|^2}, \qquad \eta = 10^{-9}\operatorname{tr}(S)/k. \tag{62}$$

This uses uncentred moments without an intercept and the same sample for fitting and evaluation. Its numerator is the reduction in the optimal ridge-penalised squared-error objective relative to the zero predictor, so the score is not exactly the unpenalised explained-energy fraction or a centred test-set R^2. We note that, the score does not establish that past activations improve task loss, or that the current activation contains no nonlinearly accessible equivalent information.

J.5 Values and attention sinks

The value features widen during training in both SSN++ models but not in the matched SSN (Figure 14a). The mean stable rank of the value output rises from 9.1 to 15.8 at $k = 40$ and from 10.0 to 17.0 at $k = 104$, while in the matched SSN it peaks at 13.8 after 2,000 updates and ends at 12.5. All three models use the nonlinear value map, so the difference reflects how the surrounding architecture uses the value features rather than the map itself, and these measurements do not isolate the effect of nonlinear values over linear ones.

Attention sinks are weak in every SSN we measured. In the development models of Appendix I, the attention mass on the first four key positions, averaged over query positions 128 to 1023, heads, sequences, and layers of 2×1024 C4 tokens, is 5.93% for the dense transformer against 0.39% for SSN++ with linear values and 0.48% with SwiGLU values. The sinks are thus already weak with linear values, and sink removal does not explain the wider value spectrum of Appendix I.3. The first query positions are omitted because their mass is high by construction when only a few keys are available. Through training, first-token attention mass stays below 0.25% in every model (Figure 14b), and mean attention entropy decreases only slightly. The model with the lowest loss, SSN++ INT8/INT4 at $k = 104$, has the largest first-token mass, 0.24% at update 10,000 against 0.16% for the matched SSN and 0.15% for SSN++ at $k = 40$, which limits a monotonic sink-reduction explanation of the gains while leaving open how nonlinear values change useful content.

The residual streams show the same contrast (Figure 13). We ran a controlled experiment on the attention-sink phenomenon to the final SSN and SSN++ of the 10,000-update study and to a dense 1B transformer trained on the same data for as many updates, on 2×1024 C4 tokens. In the dense transformer the largest coordinate of the residual stream entering blocks 1 to 15 is 37 to 64 times the median coordinate, and on 70 to 90% of tokens it is the same coordinate, the fixed massive activation that accompanies attention sinks in dense models. From block 5 onward the dense model also has keys that collect more than 100 times the mean attention mass, and its later layers drain the first-token value norm to as little as 0.30 times the median. In SSN and SSN++ the ratio stays between 7.7 and 10.4 in every block, the largest coordinate changes from token to token, no key exceeds 100 times the mean attention mass, and the first-token value norm averages 1.30 and 1.34 times the median across layers, against 0.81 in the dense model. Every output projection of an SSN writes

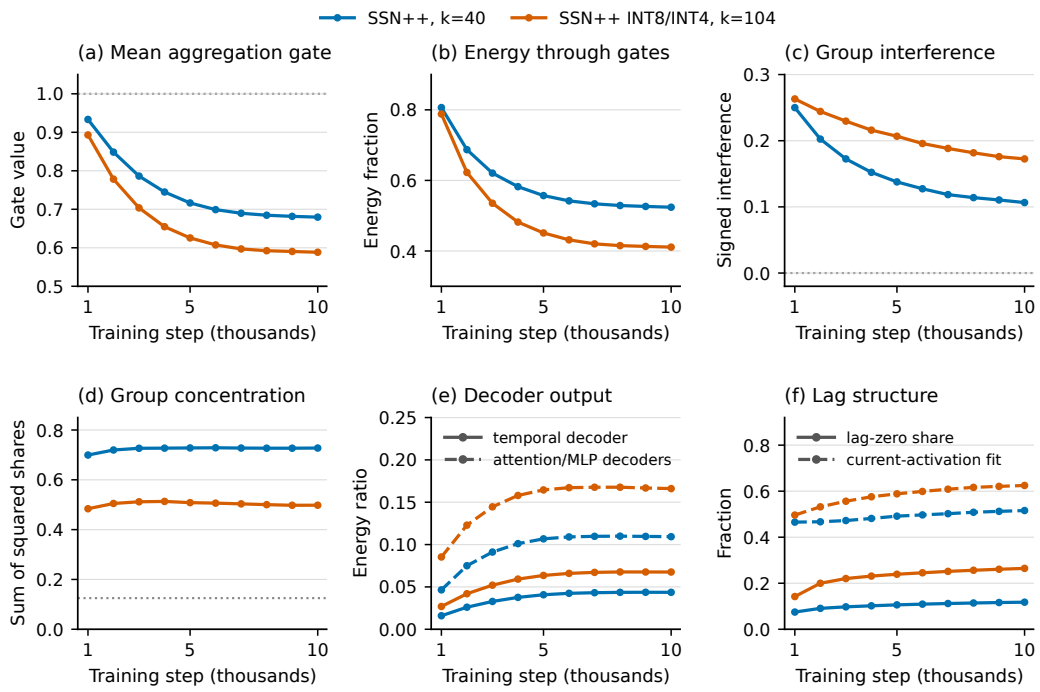


Figure 12: The gated aggregation at the sending layer and the decoders at the receiving layer in SSN++ at $k = 40$ and SSN++ INT8/INT4 at $k = 104$. (a) Mean aggregation gate, with the neutral value one dotted. (b) Fraction of the group-coefficient energy that the gates pass into the aggregation. (c) Signed interference between the group contributions after their orthogonal transforms. (d) Concentration of the gated group energies, with the value $1/8$ of equal use dotted. (e) Energy of the temporal decoder's output relative to the decoded residual stream (solid) and of the outputs of the attention and MLP decoders relative to the inputs of the sublayers that use them (dashed). (f) Lag-zero share of the temporal decoder's term energy (solid) and the share of its output that the current activation predicts linearly (dashed). Aggregation statistics average layers 0 to 13 and decoder statistics average receiving layers 1 to 14.

into the residual stream through an orthonormal basis whose rows have norm $\sqrt{k/d}$, so no single coordinate is favoured. A vector in a random 40-dimensional subspace of $\mathbb{R}^{2048}$ has a max/median coordinate ratio of at most about 15, or about 20 with the embedding's own subspace, against about 5 for a typical vector, so the residual stream of an SSN cannot form a massive activation, and the measured 7.7 to 10.4 lie within this bound. Normalising the sublayer outputs rescales each write uniformly and leaves this ratio unchanged. The absence of outliers is consistent with the small cost of quantising the compressed activations (Section 4.3).

J.6 Quantised communication and gradients

Quantised communication behaves the same at every checkpoint (Figure 14c–e). The forward INT8 error energy is 3.1×10^{-5} of the activation energy in the SSN++ INT8/INT4 control at $k = 40$ and 4.0×10^{-5} in SSN++ INT8/INT4 at $k = 104$, and no quantised coefficient saturates. One stochastically rounded INT4 gradient deviates from the unquantised gradient by 14% and 16% in relative norm at every checkpoint, with cosines of 0.990 and 0.987, and the mean of eight independent roundings deviates by 5.0% and 5.7%, smaller by the factor $\sqrt{8}$ expected of unbiased rounding. The temporal decoder's response to the forward error grows with the decoder, from 0.11 to 0.25 of the error's norm in the control and from 0.12 to 0.27 at $k = 104$, but stays well below one. The whole-model parameter gradient with the trained quantisers keeps a cosine between 0.95 and 0.98 with the gradient of the same weights under unquantised communication, and quantisation changes

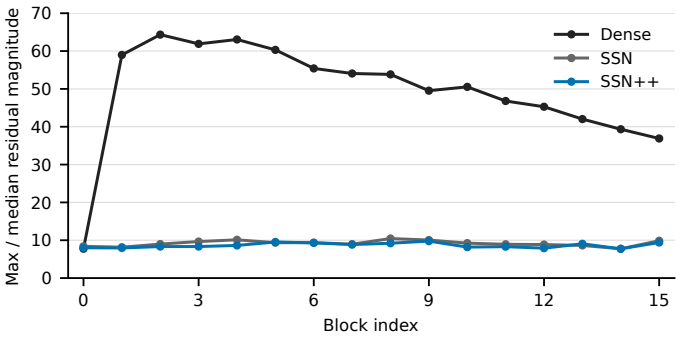


Figure 13: Largest over median coordinate magnitude of the residual stream entering each block, over 2×1024 C4 tokens, in 1B models after 10,000 updates. The dense transformer develops a fixed massive activation from block 1 onward, while SSN and SSN++ stay below 11 in every block.

the loss on the measurement corpus by between -2.2×10^{-4} and 6.5×10^{-4} nats per token at every checkpoint.

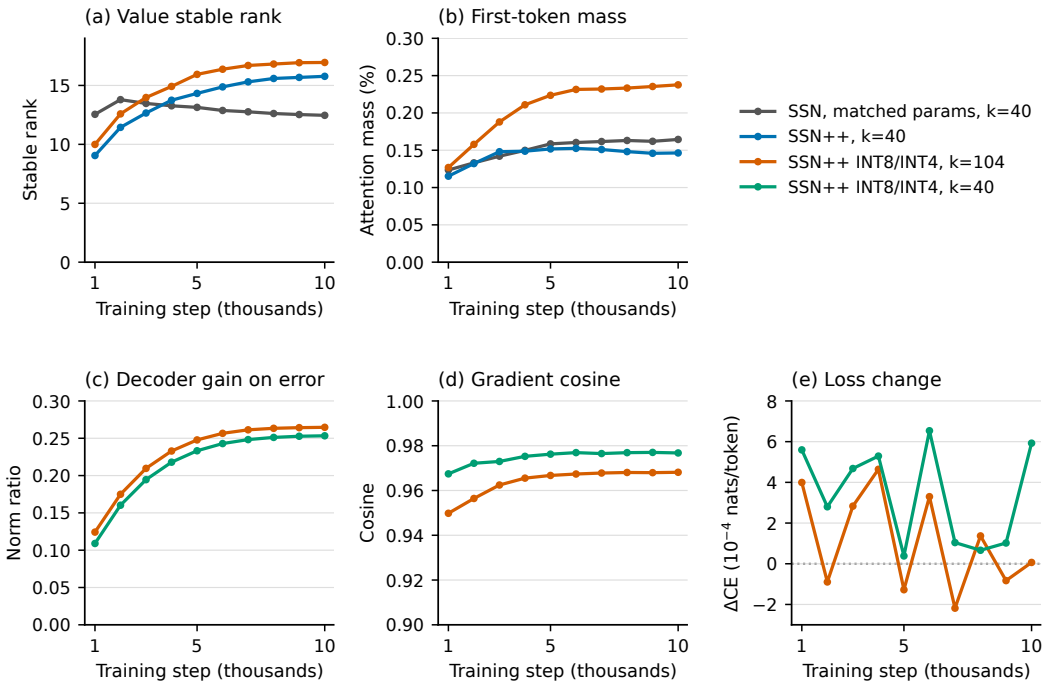


Figure 14: Values, attention sinks and quantised communication. (a) Mean stable rank of the value output and (b) mean attention mass on the first token, over all 16 layers. (c) Gain of the temporal decoder on the forward quantisation error, measured as the norm of its response relative to the error. (d) Cosine between the whole-model parameter gradient with the trained quantisers and that of the same weights under unquantised communication. (e) Change in cross-entropy on the measurement corpus caused by quantised communication. Panels (c) to (e) show the two models trained with INT8/INT4 communication, SSN++ INT8/INT4 at $k = 104$ and the SSN++ INT8/INT4 control at $k = 40$.

J.7 INTERVENTIONS OVER TRAINING

Reliance on every component grows during training (Figure 15). Setting the aggregation gates to neutral raises the loss by 0.047 nats at update 1,000 and 0.459 at update 10,000 in SSN++ at $k = 40$, and by 0.066 and 0.384 in SSN++ INT8/INT4 at $k = 104$. Removing the attention decoder costs 0.169 and 0.764 nats at $k = 40$ and 0.169 and 0.455 at $k = 104$, far more than removing the MLP decoder, which costs 0.046 to 0.097 and 0.152 to 0.199. Removing the temporal decoder costs 0.100 to 0.168 at $k = 40$ and 0.310 to 0.395 at $k = 104$, and replacing its output with norm-matched noise costs more, 0.126 to 0.231 and 0.380 to 0.550, so the models rely on the content of the temporal decoder's output and not only on its scale. Removing all three decoders together costs 0.699 to 2.301 nats at $k = 40$ and 1.580 to 2.721 at $k = 104$, more than the separate removals combined, so the decoders at the receiving layer depend on one another. For $k = 40$, projecting the attention output of one layer onto its top-k subspace at update 10,000 changes the loss by at most 0.021 nats at layers 0 to 13 and by 0.231 at layer 14, the last compressed layer, which feeds the dense final layer (Figure 15f). The growing sensitivity shows that the trained models rely more on these components as learning proceeds, which need not equal the gain from including a component throughout training.

Intervention estimates and controls. Each intervention is compared with the untouched checkpoint on the same 32 sequences, and the error bars of Figure 15 are the standard errors $s_{\Delta\ell}/\sqrt{32}$ of the paired per-sequence next-token losses, which summarise sequence variation for a fixed checkpoint. At update 10,000, removing the attention and MLP decoders together raises the loss by 1.29 ± 0.04 nats in SSN++ at $k = 40$ and by 1.04 ± 0.03 in SSN++ INT8/INT4 at $k = 104$, and projecting the attention outputs of all compressed layers onto their top-k subspaces raises it by 0.285 ± 0.014 and 0.031 ± 0.005 nats. The top-k projection uses leading eigenvectors of the *uncentred* second moment of the output, fitted on those same 32 sequences, so projector fitting and sensitivity evaluation use the same sample. The study also includes random-subspace and norm-matched noise controls. Removing a whole decoder also removes its current-activation term, so these removals do not isolate the contribution of history.

The MLP projection effect is localised to the last compressed layer. At a stage boundary with gated aggregation, compression subtracts the ambient MLP output from the residual stream and aggregates the cached group coefficients. The projection intervention changes the ambient output but leaves those cached coefficients intact. Accordingly, at step 10,000 the MLP top-k intervention at all layers of SSN++ at $k = 40$ changes the cross-entropy by 0.1803, and projecting layer 14 alone changes it by also 0.1803, whereas the individual effects at layers 0–13 are below 6×10^{-8} in magnitude. For SSN++ INT8/INT4 at $k = 104$, the corresponding changes are 0.0873 and 0.0876, with effects at earlier layers below 6.5×10^{-5} in magnitude. These results support reliance on the ambient MLP output at the last compressed layer, which feeds the dense final layer, and not throughout the layers with gated aggregation. Projecting an attention output also changes the MLP input in the same layer and therefore tests how that MLP uses the ambient attention output.

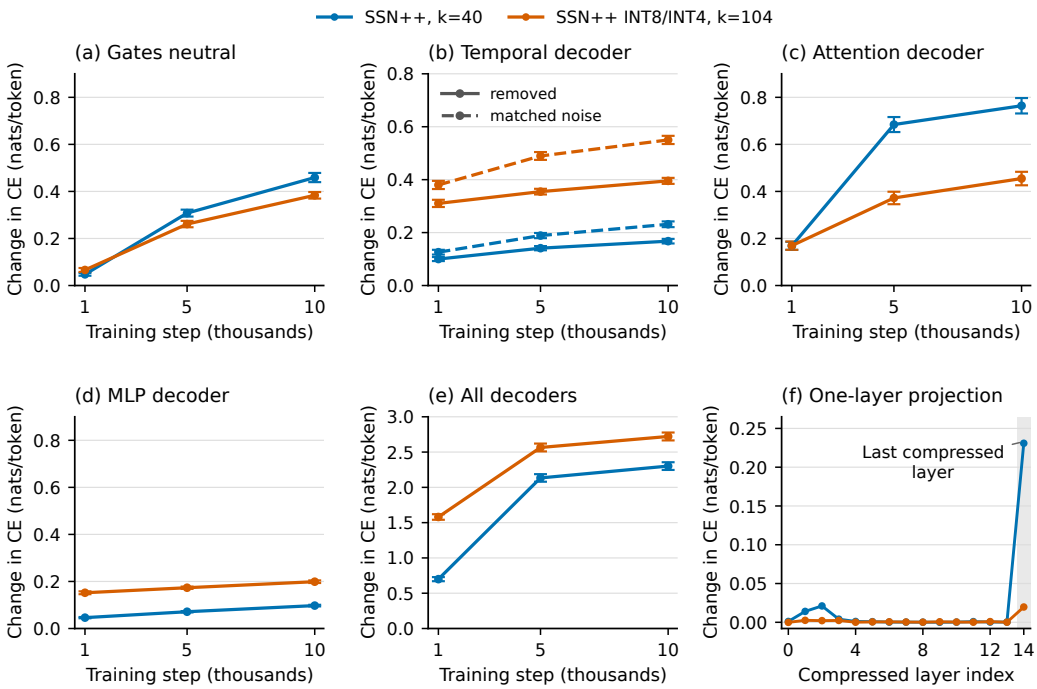


Figure 15: Interventions over training in SSN++ at $k = 40$ and SSN++ INT8/INT4 at $k = 104$, as paired increases in cross-entropy on 32 sequences of the measurement corpus, with sequence standard errors as bars. (a) Aggregation gates set to neutral. (b) Temporal decoder removed (solid) or its output replaced by norm-matched noise (dashed). (c) Attention decoder removed. (d) MLP decoder removed. (e) All three decoders removed together. (f) Projection of the attention output of one layer onto its top-k subspace at update 10,000, plotted against the compressed layer index, where layer 14 is the last compressed layer, which feeds the dense final layer.