WHAT DOES MODEL GROWTH REALLY ADD? FUNCTIONAL CAPACITY BEYOND PARAMETER COUNT

Anonymous authors
Paper under double-blind review

ABSTRACT

Model growth adds parameters to a trained checkpoint and continues training, aiming to increase capacity while reusing the existing model instead of training from scratch. But does growing a trained model actually create new functional capacity that survives subsequent training? Parameter count alone cannot answer this question, and "capacity" is itself ambiguous; it can refer to parameter count, effective dimensionality, or the ability to represent new functions. We measure it directly, as the number of new, functionally independent directions a growth step adds, formalized as the rank increment of the model's functional Jacobian and estimated matrix-free at scale. For zero-gated function-preserving growth, this increment is exactly K under a transversality condition, i.e. $\mathrm{rank}(J_{\mathrm{grown}}) = \mathrm{rank}(J_{\mathrm{old}}) + K$. Continually training a Transformer grown from $253\mathrm{M}$ to $857\mathrm{M}$ parameters across three domains, we find these added directions persist, with all $K = 36$ remaining independent after continual learning, even when the model catastrophically forgets (perplexity $28 \to 481$) or we deliberately destroy the old function ($25.9 \to 127$). Geometric capacity and retained function are therefore decoupled. The functional capacity introduced by growth persists, while the previously learned function is not retained. A second, geometrically distinct mechanism, Net2Net, reaches the same K-dimensional capacity from directions that are dormant at initialization ($0 \to K$) and then retains it, so the decoupling is not specific to the zero-gate parameterization. The implication is precise. Catastrophic forgetting need not reflect a loss of functional capacity, and preserving dimensionality alone is insufficient to preserve the learned function.

1 INTRODUCTION

Motivation. "Capacity" means three different things in this literature. When a growth method (Chen et al., 2016; Wang et al., 2024; 2023; Chen et al., 2022) "adds capacity cheaply," capacity is *parameters and FLOPs*, a claim about size that says nothing about what survives training. When the plasticity literature reports that continual training "loses capacity" (Dohare et al., 2024; Liu et al., 2025), capacity is *effective rank*, a collapsing ability to keep learning. When continual-learning theory reports that "model capacity deteriorates" (Chakraborty & Raghavan, 2025), capacity is the *ability to represent tasks under the forgetting loss*. Three quantities, one word, and the conflation has a cost: it is unclear whether a growth operator's advertised capacity persists or is capacity you pay for and lose. Concretely, three things go unmeasured. First, what a growth operator actually adds: parameter count only upper-bounds delivered capacity, yet the growth literature reports parameters and loss, not the functional degrees of freedom a growth step contributes (Maddox et al., 2020). Second, whether that capacity survives continual learning: prior work measures function-value capacity (plasticity, retention), so whether the *geometric* capacity growth adds persists, and how it relates to what is retained, is unknown. Third, whether this depends on the growth mechanism or holds for function-preserving growth in general.

What we measure. We characterize the *geometric* capacity of growth: the number of new, functionally-independent directions a growth step adds, defined as the rank the operator adds to the network's functional Jacobian. The functional Jacobian is a standard analysis lens; prior work uses its rank to study, for instance, trainability at initialization (Everett, 2026). We apply the same lens

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.

to model growth: the functional capacity a *growth operator* adds, and whether that capacity persists through *continual learning*. For zero-gated function-preserving (FP) growth, the operator adds exactly K independent directions under a transversality condition, i.e. $\mathrm{rank}(J_{\mathrm{grown}}) = \mathrm{rank}(J_{\mathrm{old}}) + K$ (Lemma 1 and Theorem 1). We verify it with a matrix-free estimator built on Jacobian–vector products, which recovers the rank without ever materializing the Jacobian; a degeneracy stress test confirms the increment is exact and non-trivial (zeroing a fraction of new blocks drops the rank by exactly that fraction: $36 \to 27 \to 18 \to 9$). On a Transformer grown from 253M to 857M parameters ($K = 36$ new blocks), this gives an added rank of exactly 36.

Capacity is decoupled from retained function. We then continually train the grown model across three domains (WikiText-103 $\to$ BookCorpus $\to$ code) and track the added subspace. It is *preserved*, with hard rank staying $36 \to 36$ and remaining transversal to the old parameters, whether the old domain is catastrophically forgotten (perplexity $28 \to 481$) or deliberately destroyed by replay onto a conflicting objective ($25.9 \to 127$). At the same time, the subspace is *re-tasked*: the directions specialized to one domain are near-orthogonal to those specialized to the next. Geometric functional capacity is therefore **decoupled** from retained function. For a residual branch with a linear output map, FP-at-init depth growth admits two constructions: zero the scalar gate (Bachlechner et al., 2021; Yao et al., 2024; Han et al., 2025) or zero the output weights (Chen et al., 2016; Evci et al., 2022; Wang et al., 2024). The experiments show the decoupling applies to both, suggesting it is a property of function-preserving depth growth itself. Since the added capacity is intact even in a model that has completely forgotten, forgetting is not a shortage of capacity: keeping capacity high cannot, on its own, prevent it.

Contributions. Our contributions are a geometric analysis of model growth and its persistence under continual learning, measured through functional-Jacobian rank.

1. **A decomposition of capacity.** We define the *geometric* capacity of growth, the number of new functionally-independent directions it adds (functional-Jacobian rank), and separate it from the *function-value* capacity, plasticity and retention, that the continual-learning literature measures. This is what makes the decoupling below a measured result, not the trivial observation that rank and loss differ.
2. **The decoupling result.** The added K-dimensional capacity is preserved ($36 \to 36$) under catastrophic forgetting and under deliberate function destruction, while being re-tasked across domains, which is direct evidence that geometric capacity and retained function are independent. The decoupling holds for both FP mechanisms (gate-zero and Net2Net), showing its generality across function-preserving depth growth.
3. **A scalable measurement of functional capacity.** We compute functional-Jacobian rank matrix-free using Jacobian–vector products, avoiding explicit Jacobian materialization. We validate the measurement against exact rank and controlled degeneracy tests, making functional-capacity analysis practical at the scale of hundreds-of-millions-parameter Transformers.

2 Related Work

2.1 Model growth and function-preserving initialization

Model growth adds parameters to a trained checkpoint and continues training, reusing prior compute. We organize growth methods by their function-preserving (FP) mechanism (Wei et al., 2016): *how* the new component is made to contribute exactly zero at initialization so the grown model matches the base. On the **depth** axis there are exactly two FP-at-init constructions. *Class 1 (zero the gate):* a scalar or structured gate on the new branch is initialized to zero, annihilating the branch output regardless of its weights, as in MSG (Yao et al., 2024) and LOIRE (Han et al., 2025). Our operator, gate-zero, instantiates this mechanism, inspired by ReZero's zero-initialized residual gate (Bachlechner et al., 2021), and is our Class-1 representative. *Class 2 (zero the output weights):* the outgoing weight matrix is initialized to zero (Chen et al., 2016; Evci et al., 2022; Wang et al., 2024); these differ only in the input-weight initialization, not the FP mechanism, with Net2Net as representative. The same principle appears outside growth: Fixup zero-initializes each residual branch (Zhang et al., 2019), ControlNet uses zero-convolutions (Zhang et al., 2023), and LoRA zero-initializes its low-rank update (Hu et al., 2022).

A third mechanism, *Class 3 (canceling duplication)*, duplicates neurons or layers and rescales downstream weights to preserve the function without zeroing (Chen et al., 2016; 2022); it is **width-native**, so we leave it to future work. We exclude methods that are *not* FP at initialization: LiGO (Wang et al., 2023) learns a small-to-large linear map rather than a closed-form FP construction, and stacking (G_{stack}) (Du et al., 2024) and depth up-scaling (Kim et al., 2023) stack layers without preserving the function, recovering the drop through continued pretraining. Whether growth is even necessary for continual learning is contested: a fixed model with replay can control forgetting (Ibrahim et al., 2024), while FP growth has been applied directly to lifelong learning (Han et al., 2025). Growth's value therefore rests on its adding real, durable functional capacity, which is what we measure.

2.2 Rank, effective rank, and "capacity"

"Rank" and "capacity" name several distinct quantities, which we separate along two axes: the *object* (which matrix's spectrum is measured) and the *operator* (how the spectrum is turned into a number). The operators are largely shared: *effective rank* as spectral entropy (Roy & Vetterli, 2007), which we borrow for our soft measure; *effective dimensionality* from the loss Hessian (Maddox et al., 2020); and *intrinsic dimension* via random projection (Li et al., 2018). What differs is the object. Prior work measures the rank of **weights**, i.e. how far a model can be low-rank factorized while retaining accuracy (Zerihun, 2025); of **representations**, the hidden-state matrix, which collapses toward rank one with depth in pure-attention networks (Dong et al., 2021; Huh et al., 2023); or of the **loss Hessian / parameter curvature**, counting the parameter directions the data constrains (Maddox et al., 2020; Li et al., 2018; Gur-Ari et al., 2018). These established that parameter count overstates usable capacity, but only for *fixed* models.

In contrast, Transforming Rank (Everett, 2026) recasts skip connections and normalization as mechanisms that preserve the branch / input–output Jacobian rank across depth, and shows the *initialization* Jacobian rank predicts trainability. Instead of asking how a fixed architecture behaves at initialization, we measure what a *growth operator adds* (the parameter Jacobian $\partial f/\partial\theta$) and how it evolves *through continual learning*. This enables our analysis: the exact $+K$ increment growth adds, and its decoupling from retained function under continual learning.

2.3 Capacity in continual learning

A parallel line uses "capacity" for a function-value quantity and finds it **deteriorates** under continual training: representation effective rank collapses, costing plasticity (Dohare et al., 2024), which motivates growing or regenerating units to restore it (Liu et al., 2025), an idea with roots in task-wise network expansion (Rusu et al., 2016); and model capacity defined through the forgetting loss is provably non-stationary and degrades as tasks drift (Chakraborty & Raghavan, 2025). Our result separates the two senses these works conflate under one word: the *geometric* capacity a growth operator adds (functional Jacobian rank) is **preserved** even as this function-value capacity is lost; the two are decoupled.

3 Functional Rank of Growth Operators

Setup and notation. Let $f_\theta : \mathcal{X} \to \mathbb{R}^{d_{\text{out}}}$ be the model (for a language model, d_{out} is vocabulary size times sequence positions on a probe input), with base parameters $\theta \in \mathbb{R}^p$ trained on domain A. *Gate-zero growth* inserts K residual blocks, each of the form $x \mapsto x + \alpha_i\, B_{\phi_i}(x)$ with scalar gate α_i and block weights ϕ_i, appended after the original blocks, with $\alpha = 0$ at initialization; the grown parameters are $\theta' = (\theta, \phi, \alpha)$.

Definition 1 (Functional Jacobian and its rank)**.** On a probe input x, the functional Jacobian is $J_\theta(x) = \partial f_\theta(x)/\partial\theta \in \mathbb{R}^{d_{\text{out}} \times p}$, stacked over a probe set.

- The *functional rank* is $\operatorname{rank} J$, measured matrix-free via forward-mode Jacobian–vector products — intuitively, how many independent directions in parameter space the model can use to change its output.
- Two measures from the singular values of J:

- *soft rank* (Shannon rank): the effective rank (Roy & Vetterli, 2007) $\mathrm{erank}(J) = \exp(H(\sigma))$, with H the Shannon entropy of the L_1-normalized singular values σ_i (Appendix B).
- *hard rank* (numerical rank): $\#\{\sigma_i > \tau\,\sigma_{\max}\}$, the number of singular values above a relative tolerance τ (we use $\tau = 10^{-3}$).

All proofs are deferred to Appendix A.

3.1 What growth adds

Lemma 1 (Function preservation at initialization, and the live direction)**.** *For gate-zero growth with $\alpha = 0$:*

(i) $f_{\theta'}(x) = f_\theta(x)$ *for every input x;*

(ii) the new-weight block of the Jacobian vanishes, i.e. $\partial f_{\theta'}/\partial\phi\big|_{\alpha=0} = 0$*;*

(iii) the gate direction is live, $\partial f/\partial\alpha_i = (\partial f/\partial \mathrm{out}_i)\, B_{\phi_i}(h_i(x))$*, which is generally nonzero since $B_{\phi_i}(h_i(x))$ is the output of block i on its input activation.*

Hence the functional directions opened by growth are the K gate directions, and the rank analysis is carried out over the K gate columns.

Theorem 1 (Rank additivity under transversality)**.** *Let J_{old} have rank r, and let $J_\alpha = [\,\partial f/\partial\alpha_1 \mid \cdots \mid \partial f/\partial\alpha_K\,]$ be the K gate columns.*

- *If the columns of J_α are linearly independent of* $\mathrm{col}(J_{\mathrm{old}})$ *(*transversality*), then* $\mathrm{rank}[\,J_{\mathrm{old}} \mid J_\alpha\,] = r + K$.
- *If transversality fails, the rank increases by strictly less than K.*

Corollary 1 (Parameter count is not instantaneous functional dimension)**.** *At the gate-zero growth point, the first-order functional dimension added by growth satisfies:*

(i) $\Delta\,\mathrm{rank} \le K \le \min(P_{\mathrm{new}}, d_{\mathrm{out}})$*, where P_{new} is the number of new parameters, with equality $\Delta\,\mathrm{rank} = K$ under transversality (Theorem 1);*

(ii) $K \ll P_{\mathrm{new}}$ *whenever each gate opens a block of many weights;*

(iii) the remaining $P_{\mathrm{new}} - K$ new directions lie in the Jacobian null space at the growth point (Lemma 1(ii)).

This is a first-order statement at initialization, not a claim that those directions are permanently redundant: they become functional once the gates open. Parameter count therefore bounds from above, and in general overstates, the instantaneous functional tangent dimension.

3.2 What functional rank does not determine

Theorem 2 (Functional rank is a tangent dimension, not a position)**.** *Let $F : \Theta \to \mathcal{F}$ be the parameter-to-function map, where $\mathcal{F}$ is the finite-dimensional space of outputs on a fixed probe set, and let F be C^1 on an open set $U \subset \Theta$. Suppose $J(\theta) = DF_\theta$ has constant rank $r > 0$ on U. Then:*

(i) $F(U)$ is locally an r-dimensional embedded submanifold M of $\mathcal{F}$, and for every $\theta \in U$, $\mathrm{rank}\,J(\theta) = r = \dim T_{F(\theta)}M$*, so the functional rank is the dimension of the tangent space of M at $F(\theta)$;*

(ii) F is not constant on U, so there exist $\theta_1, \theta_2 \in U$ with $\mathrm{rank}\,J(\theta_1) = \mathrm{rank}\,J(\theta_2) = r$ *and $F(\theta_1) \neq F(\theta_2)$, and moreover, for any task loss L_T nonconstant on $F(U)$, such a pair can be chosen with $L_T(\theta_1) \neq L_T(\theta_2)$.*

Consequently, functional rank determines the local dimension of available functions, not which function is represented, and therefore does not determine retention.

Remark. The loss gap can be unbounded: for the linear model $f_\theta(x) = \theta^\top x$, the Jacobian $J = x^\top$ is identical for all θ (rank and column space fixed) while squared-error loss grows without bound as θ scales. Same functional rank, different function, arbitrarily different loss: rank fixes the tangent dimension, not the position on M.

4 EXPERIMENTS

Our experiments measure whether growth adds the functional capacity the theory predicts, whether that capacity survives continual learning, and how it relates to what the model retains.

4.1 SETUP

Base model and growth. The base is a 253M-parameter, 12-layer decoder-only Transformer trained on WikiText-103 (domain A); all growth experiments start from this single shared checkpoint, so differences between grown models are attributable to the growth operator and the subsequent training, not to the base. With growth factor $g = 4$, the model goes from 12 to 48 layers, adding $K = 36$ new blocks and reaching 857M parameters. Smaller factors ($g = 2$, $g = 3$; $K = 12$, 24; 454M, 656M) are used for the scale check in Table 2.

Growth operators. We grow by appending K residual blocks after the original stack (append layout), initialized by cloning the original blocks in round-robin order; this cloning is what makes the added directions *redundant* rather than random, a point we return to in Section 4.4. A function-preserving (FP) operator makes each new block contribute exactly zero at initialization, so the grown model reproduces the base model on every input. On the depth axis, there are two ways to do this, and we implement one representative of each (Section 2):

- **Gate-zero (Class 1, zero the gate).** Each new block is wrapped as $x \mapsto x + \alpha_i \, B_{\phi_i}(x)$ with a scalar gate α_i initialized to zero. The block weights ϕ_i are live but their contribution is annihilated by the gate; the gate itself is the live direction at initialization (Lemma 1).
- **Net2Net (Class 2, zero the output weights).** Each new block's output projections (the attention output projection and the MLP down projection) are initialized to zero, and the gate is fixed at one. The block contributes zero through its weights rather than through a gate. At initialization, its directions are therefore dormant (rank zero), and they activate only as the output weights train.

For both mechanisms, we verify function preservation numerically at the moment of growth: the maximum absolute difference in logits between base and grown model is 0.00 (exact to float precision) on a held-out batch.

Clean function preservation. Our production recipe for gate-zero growth includes three stabilizing knobs: a distillation term toward the base model, a small gate warm-up, and clone noise. To isolate the growth operator from these aids, all persistence and generality runs use a *clean-FP* configuration with all three disabled. The one exception is the function-destruction counterexample (Section 4.3, E5), which deliberately uses the replay recipe *with* these knobs to match the original run that produced it.

Continual-learning protocol. Three domains are used in sequence: WikiText-103 (A, the base domain), BookCorpus (B), and a code corpus (C). The two new domains are each capped at 118M tokens to match the WikiText-103 base ($\approx$ 118M tokens), so the three are size-balanced. We run a two-domain sequence (A→B) and a three-domain sequence (A→B→C). Each continual-learning phase trains for 10 epochs of 900 steps.

- **Isolation.** The original parameters are frozen and only the new blocks (and, for gate-zero, their gates) are trained on the new domain. This is the regime for the persistence and generality results.
- **Replay onto a conflicting objective.** The full model is trained on the new domain with a replay buffer of old-domain data at fraction α. Under our settings, this regime *destroys* the old-domain function (perplexity on A rises from 25.9 to 127) while the model continues to learn; it is used only as the function-destruction counterexample (E5).

Experiments. We report eight experiments, referenced by identifier throughout and mapped to specific rows and columns in each table.

- **What growth adds (Table 1). E1**: at the moment of growth the added rank equals K and is transversal to the old parameters. **E2**: a degeneracy stress test that makes new blocks contribute zero and confirms the rank counts live functional directions, not blocks.
- **Does it survive continual learning (Table 2, Figure 1). E3**: the added hard rank persists through continual learning. **E4**: it stays transversal, rather than being absorbed into the old span. **E5**: it persists even when the old-domain function is forgotten or deliberately destroyed, the decoupling counterexample. **E6**: the preserved subspace is re-tasked across domains, not frozen.
- **Generality test (Table 3, top). E7**: the same persistence holds under a second, mechanistically opposite FP operator, Net2Net.
- **Effective rank (Table 3, bottom). E8**: at fixed parameter count, effective rank separates redundant from efficient growth, which hard rank and parameter count cannot.

4.2 Growth adds exactly K functional directions

We first describe how a grown model is turned into a rank measurement, then verify Theorem 1 at the moment of growth.

Measuring functional rank. On a probe input x, the functional Jacobian is $J_\theta(x) = \partial f_\theta(x)/\partial\theta$. For the added directions, we take the K gate columns $J_\alpha = [\partial f/\partial\alpha_1 \mid \cdots \mid \partial f/\partial\alpha_K]$ (Lemma 1), stacked over a probe set. For Net2Net the gate is fixed at one, but $\partial f/\partial\alpha_i$ still equals the block's output, zero at initialization and nonzero after training (the dormant-to-active signature, Section 4.4).

Matrix-free computation. On one probe batch of 2×128 tokens over a $50{,}257$-way vocabulary the output has 12.9M dimensions, so at 857M parameters the full Jacobian ($\approx 10^{16}$ entries) is far too large to form. We only need its K gate columns and the 64 sampled old-direction columns of Section 4.2, each an output-sized vector obtained by a single forward-mode Jacobian–vector product with a one-hot tangent; together they occupy about 5 GB per probe batch, and a complete measurement takes a few minutes on one GPU.

Two rank operators. From the stacked columns we report two measures: the **hard rank**, an integer count of singular values above $10^{-3}\,\sigma_{\max}$ (probe-independent, our headline measure), and the soft **effective rank** (Roy & Vetterli, 2007), which depends on the probe. Appendix B gives definitions and probe settings.

Transversal residual. Rank additivity (Theorem 1) requires the added directions to be independent of the old parameter directions. We test this by projecting the gate columns off the span of the old-parameter Jacobian: because that Jacobian is far too large to form, we sample 64 random directions in the old parameter space, compute their Jacobian columns by the same forward-mode procedure, orthonormalize them into a projector P_{old}, and report the hard rank of $(I - P_{\text{old}})\,J_\alpha$. A residual rank of K means every added direction carries functional content outside the old span; a residual rank below K means some lie inside it. Appendix B gives the per-checkpoint projector construction and the thresholding convention.

Subspace rotation across domains. We compare the gate subspace after training on domain B with the subspace after subsequently training on domain C, via the principal angles between the two K-dimensional column spaces, to see whether the preserved subspace is *re-tasked* across domains. A frozen subspace would show principal angles near zero; a preserved but re-oriented one shows angles near the random baseline of two independent K-dimensional subspaces (near $90°$).

Estimator validation. We validate the pipeline with an exact-rank self-test on a synthetic problem of known structure (Appendix B); the degeneracy stress test at scale is experiment E2 below, and it shows that the transversality condition of Theorem 1 is not a tautology at this scale. Appendix B additionally reports a cautionary case in which a coarser proxy manufactured a collapse that the functional-Jacobian estimator does not show.

Table 1: Growth adds exactly K functional directions (gate-zero, $g = 4$, $K = 36$, at the moment of growth).

New blocks made degenerate	0% (E1)	25%	50%	75%
Live new blocks	36	27	18	9
Hard rank of added directions (E2)	36	27	18	9

Table 2: Capacity survives continual learning and is decoupled from retained function (gate-zero, clean-FP unless noted). The scale-check rows report rank and residual only. Per-seed old-domain perplexities behind the ranges are in Appendix C.

Continual-learning condition	Seeds	Hard rank (E3)	Transv. residual (E4)	Old-domain function (E5)
2-domain isolation (A→B)	42, 43, 44	$36 \to 36$	$36 \to 36$	A degraded, PPL $25.9 \to$ 92–93
3-domain sequence (A→B→C)	42, 43, 44	$36 \to 36$	$36 \to 36$	B forgotten, PPL $28 \to 481$
Replay onto conflict	42, 43, 44	$36 \to 36$	$36 \to 36$	A destroyed, PPL $25.9 \to$ 125–127
Scale check (2-domain, seed 42)				
$g = 2$, $K = 12$	42	$12 \to 12$	$12 \to 12$	–
$g = 3$, $K = 24$	42	$24 \to 24$	$24 \to 24$	–

Results. Theorem 1 predicts that under transversality growth adds exactly K independent directions. We verify this at the moment of growth.

The $K = 36$ gate columns have hard rank 36 and are transversal to the old parameters (residual 36): growth adds exactly K independent directions, each carrying functional content outside what the base already spans (E1). The rank counts live functional directions, not blocks: making a fraction of the new blocks degenerate by zeroing their output projections drops the hard rank to exactly the number of live blocks ($36/27/18/9$ for $0/25/50/75\%$ zeroed; Table 1), so the transversality condition of Theorem 1 has real bite at this scale (E2).

4.3 The added capacity survives continual learning, and is decoupled from retained function

We track the added subspace through continual learning under conditions that progressively destroy the represented function (Table 2).

Across every condition, the dimension of the added K-dimensional subspace is preserved. Its hard rank and its transversal residual both stay at $36 \to 36$ (grown $\to$ after CL), with zero variance across seeds and probe batches, while the effective rank drifts only mildly and depends on the probe (E3, E4). What moves is the function, which Table 2 traces from merely degraded, to catastrophically forgotten (domain B, perplexity $28 \to 481$), to deliberately destroyed by replay onto a conflicting objective (domain A, $25.9 \to 127$), with the rank unchanged at every step (E5). The destroyed case is decisive: the geometric capacity is fully intact in a model whose learned function has been overwritten. The *residual* stays at 36, not merely the rank, so the added directions remain independent of the old parameter span rather than collapsing into it.

The subspace persists, but it is not frozen. The directions specialized to domain B and to domain C are nearly orthogonal (median principal angle $\approx 86^\circ$, against a random baseline whose angles all lie in $[85, 90]^\circ$; about 5 of the 36 directions are shared), so the dimension is conserved while its orientation is repurposed from one domain to the next (E6; Figure 1, with the full distribution in Appendix D). Function-preserving growth thus adds a K-dimensional budget that continual learning neither exhausts nor protects. Forgetting coincides with substantial reorientation of a dimension-preserved subspace, not a loss of functional capacity.

4.4 Generality of the property across FP-Growth Operator

Why two operators suffice. An FP-at-init depth construction must make a new residual branch output zero. For a branch with a linear output map, this leaves exactly two constructions, either zeroing the gate that scales the branch or zeroing the branch's output weights (Section 2); cancelling

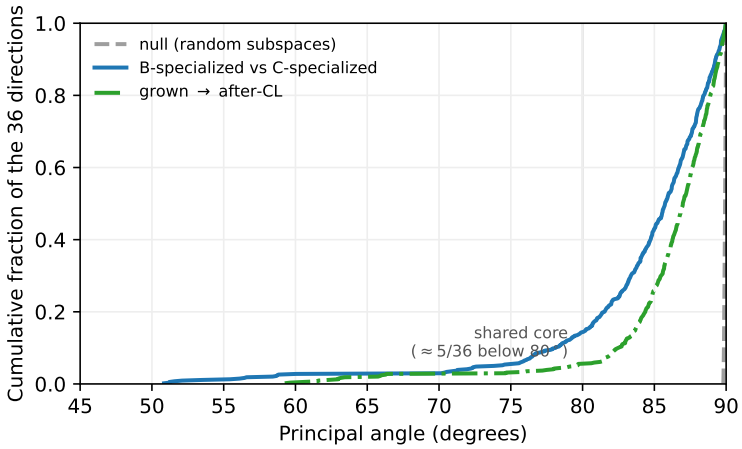

Figure 1: **The added subspace is re-tasked, not frozen (E6).** Empirical CDF of the $K = 36$ principal angles between gate subspaces, for domain B versus domain C and for grown→after-CL. Both track the near-90° random baseline rather than the near-0° a frozen subspace would give, so the dimension rotates rather than staying fixed. Full distribution in Appendix D.

Table 3: Generality test and effective-rank separation of redundant from efficient growth ($g = 4$, $K = 36$, clean-FP). Effective rank uses the random-token probe; Appendix E extends E8 to $g = 2, 3$.

Mechanism	Class	Seeds	Hard rank	Transversal residual	Eff. rank after CL
E7: persistence across the two FP mechanisms (grown → after CL)					
Gate-zero (zero gate)	1	42, 43, 44	$36 \to 36$	$36 \to 36$	30.9 / 30.8 / 31.0
Net2Net (zero output weights)	2	42, 43, 44	$0 \to 36$	$0 \to 36$	33.4 / 33.6 / 33.6
E8: at the moment of growth, $g = 4$, 857M parameters					
New-block initialization		Seed	Hard rank		Eff. rank at growth
Clone (round-robin)		42, 43, 44	36		28.9 / 28.7 / 28.8
Random		42, 43, 44	36		35.7 / 35.6 / 35.7

duplication, the remaining FP mechanism, is width-native and outside our depth scope. Gate-zero and Net2Net are one representative of each, so within this class they exhaust the FP-at-init depth constructions.

Table 3 shows both mechanisms reaching the same endpoint by opposite routes. Gate-zero's directions are live at initialization and *preserved* (hard rank $36 \to 36$); Net2Net's output weights are zeroed, so its directions are dormant at initialization and *reached* ($0 \to 36$), activating through training. After continual learning, both have hard rank 36 and transversal residual 36 across all three seeds (E7). The residual is what makes this non-trivial. A trained 48-layer network trivially has full rank in its last blocks, but residual 36 means those directions are independent of the old parameter span (and this holds as we sample far more of that span; Appendix B), i.e. genuinely *added* capacity rather than a re-spanning of what the base already had. Net2Net's higher after-CL effective rank (33.4–33.6 against gate-zero's 30.8–31.0 on the same probe) is consistent with its carrying no cloning redundancy. We keep the verb precise. Net2Net supports the survival and generality claims, not the at-initialization $+K$ additivity of Section 4.2, which is a live-gate result.

Effective rank separates redundant from efficient growth. Effective rank can separate two operators that share the same K, a distinction hard rank cannot make because it is fixed at K for any transversal growth. At fixed $g = 4$ and identical parameter count, clone-initialized new blocks have mean effective rank 28.8 against 35.7 for random initialization (Table 3, lower panel), showing that cloning adds redundant directions that hard rank and parameter count miss. That clones are redundant is intuitive; the point is that the measure sees it (E8).

5 LIMITATIONS

Scope of the growth studied. Our claims are about function-preserving depth growth. We test one representative of each of the two FP-at-init constructions available to a linearly-gated depth branch, so members within a class that differ only in input initialization (GradMax, LEMON-T1) are not run, and two cases lie outside the paper: width growth (Class 3, canceling duplication) and growth that is not function-preserving at initialization. The generality result should therefore be read as holding across these two FP depth-growth constructions, not across growth in general (Section 6).

Scale. All grown models descend from a single 253M base checkpoint, grown to at most 857M parameters under our compute budget. The scale check varies the growth factor, and hence K, but not the base, so we have no evidence from a second, independently trained base.

Continual-learning regimes. We do not sweep continual-learning methods; regularization-based methods such as elastic weight consolidation (Kirkpatrick et al., 2017), distillation toward the previous model (Li & Hoiem, 2018), other replay fractions, and unprotected fine-tuning are untested. This bounds only the continual-learning results. Rank additivity at growth is a property of the initialization, independent of the training method, and the mechanism comparison uses one regime for both operators. How other continual-learning constraints re-task the preserved subspace remains open.

6 FUTURE WORK

Our results are scoped to function-preserving depth growth, and the most natural extensions follow from what that scope leaves out.

The width axis and the third FP mechanism. Function-preserving growth admits a third mechanism, canceling duplication (Net2WiderNet (Chen et al., 2016), bert2BERT (Chen et al., 2022), LEMON-T2 (Wang et al., 2024)), which is width-native and so outside this paper's depth scope. Extending the rank analysis to width growth would make Class 3 measurable, and would require re-deriving rank additivity for width, where the added directions are not gate columns.

Growth that is not function-preserving. Whether persistence extends to non-FP growth is open. Plain stacking of trained blocks (G_{stack} (Du et al., 2024)) and depth up-scaling (Kim et al., 2023) start far from the base function, and LiGO (Wang et al., 2023) learns a small-to-large linear map rather than a closed-form function-preserving initialization; neither is function-preserving by construction.

Rank-aware growth. Functional rank is a diagnostic today; making it a control signal first requires showing it predicts downstream performance. Our redundancy result relates effective rank to how growth was initialized (Section 4.4), not to what a grown model can subsequently learn. Controlled runs that vary the added rank at fixed parameter count would test whether rank *causes* downstream differences, and whether a rank-aware policy could add capacity efficiently rather than repair what was lost.

7 CONCLUSION

For zero-gated function-preserving growth, the *geometric* capacity a growth operator adds, the rank increment of its functional Jacobian, is exactly K under a transversality condition, and a matrix-free estimator recovers it at the scale of an 857M-parameter Transformer without ever forming the Jacobian.

Our main finding is that this geometric capacity is decoupled from retained function. Through continual learning, the added K-dimensional subspace is preserved and stays transversal to the old parameters, even when the old domain is catastrophically forgotten or its function is deliberately destroyed, while its orientation is re-tasked from one domain to the next. The same behavior holds for both function-preserving depth-growth constructions we study, a zero-initialized gate and zeroed output weights. The consequence is precise: forgetting coincides with reorientation of a dimension-preserved subspace rather than a shortage of functional capacity, and keeping dimensionality high cannot on its own preserve the learned function.

REPRODUCIBILITY STATEMENT

The theoretical claims are stated with their full assumptions in Section 3 and proved in Appendix A (Lemma 1, Theorems 1 and 2, Corollary 1). The measurement pipeline is specified in Appendix B: the hard- and effective-rank operators, the exact-rank self-test on a synthetic problem of known structure, the per-checkpoint old-parameter projector, the thresholding convention, and the transversality sampling sweep. The experimental setup—the single shared 253M base checkpoint, the two function-preserving growth operators, the three-domain continual-learning protocol, and the seeds—is given in Section 4.1. All persistence and generality results are reported across seeds $42/43/44$, with the per-seed continual-learning perplexities in Appendix C, the full subspace-rotation distribution in Appendix D, and the effective-rank sweep over growth factors in Appendix E. The one exception is the scale check at $g = 2$ and $g = 3$ (Table 2), which we ran at seed 42 only because of time and compute constraints; it remains reproducible from that seed. Determinism is per-machine (GPU architecture and library versions), so we report multi-seed results rather than claiming bit-exact reproduction. Code and configurations to reproduce the measurements, tables, and figures are included in the supplementary material.

USE OF LARGE LANGUAGE MODELS

We used LLM-based coding and writing assistants at several stages of this work, and we verified all resulting outputs. Concretely:

- **Retrieval and discovery.** LLMs helped search for and summarize related work. Every cited paper was subsequently read by the authors, and all attributions and bibliographic details were checked against the original sources.
- **Research ideation and execution.** LLMs helped implement the experiments, the rank estimator, and the analysis and plotting code, and helped locate and fix bugs. All generated code was reviewed, executed, and tested, and every reported number was produced by these verified pipelines.
- **Drafting and polishing.** LLMs helped draft and revise sections of the paper and edit prose for clarity. The authors wrote the technical content, checked every claim and proof, and take full responsibility for the final text.

No theoretical result, experimental measurement, or conclusion was accepted from an LLM without independent verification by the authors.

REFERENCES

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence (UAI)*, 2021.

Supriyo Chakraborty and Krishnan Raghavan. On understanding of the dynamics of model capacity in continual learning. *arXiv preprint arXiv:2508.08052*, 2025.

Cheng Chen, Yichun Yin, Lifeng Shang, Xin Jiang, Yujia Qin, Fengyu Wang, Zhi Wang, Xiao Chen, Zhiyuan Liu, and Qun Liu. bert2BERT: Towards reusable pretrained language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations (ICLR)*, 2016.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632, 2024.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning (ICML)*, 2021.

Wenyu Du, Tongxu Luo, Zihan Qiu, Zeyu Huang, Yikang Shen, Reynold Cheng, Yike Guo, and Jie Fu. Stacking your transformers: A closer look at model growth for efficient LLM pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Utku Evci, Bart van Merriënboer, Thomas Unterthiner, Fabian Pedregosa, and Max Vladymyrov. GradMax: Growing neural networks using gradient information. In *International Conference on Learning Representations (ICLR)*, 2022.

Katie Everett. Transforming rank: How architecture navigates the spectral pathologies of depth. *arXiv preprint arXiv:2607.14018*, 2026.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Xue Han, Yitong Wang, Junlan Feng, Wenchun Gao, Qian Hu, and Chao Deng. LOIRE: Lifelong learning on incremental data via pre-trained language model growth efficiently. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (ICLR)*, 2022.

Minyoung Huh et al. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research (TMLR)*, 2023.

Adam Ibrahim et al. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. SOLAR 10.7b: Scaling large language models with simple yet effective depth up-scaling. *arXiv preprint arXiv:2312.15166*, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Jiashun Liu, Johan Obando-Ceron, Aaron Courville, and Ling Pan. Neuroplastic expansion in deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Wesley J. Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.

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

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Peihao Wang et al. Learning to grow pretrained models for efficient transformer training. In *International Conference on Learning Representations (ICLR)*, 2023.

Yite Wang, Jiahao Su, Hanlin Lu, Cong Xie, Tianyi Liu, Jianbo Yuan, Haibin Lin, Ruoyu Sun, and Hongxia Yang. LEMON: Lossless model expansion. In *International Conference on Learning Representations (ICLR)*, 2024.

Tao Wei, Changhu Wang, Yong Rui, and Chang Wen Chen. Network morphism. In *International Conference on Machine Learning (ICML)*, 2016.

Yiqun Yao, Zheng Zhang, Jing Li, and Yequan Wang. Masked structural growth for 2x faster language model pre-training. In *International Conference on Learning Representations (ICLR)*, 2024.

Liyu Zerihun. Estimating the effective rank of vision transformers via low-rank factorization. *arXiv preprint arXiv:2512.00792*, 2025.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *International Conference on Computer Vision (ICCV)*, 2023.

A PROOFS

A.1 PROOF OF LEMMA 1

Proof.

(i) Each new block contributes $\alpha_i B_{\phi_i}(\cdot)$ to its residual stream; at $\alpha_i = 0$, the block is the identity, so the composed map equals the base map.

(ii) By the chain rule, $\partial f/\partial \phi_i = (\partial f/\partial \mathrm{out}_i)\, \alpha_i\, (\partial B_{\phi_i}/\partial \phi_i) = 0$ since $\alpha_i = 0$.

(iii) $\partial f/\partial \alpha_i = (\partial f/\partial \mathrm{out}_i)\, B_{\phi_i}(h_i(x))$ carries no factor of α_i and is therefore generally nonzero.

□

A.2 PROOF OF THEOREM 1

Proof. Write $J = [\, J_{\mathrm{old}} \mid J_\alpha \,]$ and let P be the orthogonal projector onto $\mathrm{col}(J_{\mathrm{old}})$. The column space of J is the sum of $\mathrm{col}(J_{\mathrm{old}})$ and $\mathrm{col}(J_\alpha)$, and its part outside $\mathrm{col}(J_{\mathrm{old}})$ is spanned by the columns of $(I-P)J_\alpha$; hence $\mathrm{rank}\, J = \mathrm{rank}\, J_{\mathrm{old}} + \mathrm{rank}\big((I-P)J_\alpha\big)$. *Transversality* means that no nonzero combination of the gate columns lies in $\mathrm{col}(J_{\mathrm{old}})$, i.e. the gate columns are linearly independent of the old column space; equivalently, $(I-P)J_\alpha$ has full column rank K.

- If transversality holds, $(I-P)J_\alpha$ has full column rank K, so $\mathrm{rank}\, J = r + K$.
- If transversality fails, some nonzero combination of the gate columns lies in $\mathrm{col}(J_{\mathrm{old}})$, so $(I-P)J_\alpha$ has a nontrivial kernel and rank strictly less than K, giving $\mathrm{rank}\, J < r + K$.

□

A.3 PROOF OF COROLLARY 1

Proof.

(i) At the growth point the only live new directions are the K gate columns (Lemma 1(iii)) and the new-weight columns vanish (Lemma 1(ii)), so the added rank is at most K; $K \le d_{\mathrm{out}}$ since rank is bounded by the number of rows, and $K \le P_{\mathrm{new}}$ since each gate is one of the new parameters. Equality $\Delta\,\mathrm{rank} = K$ under transversality is Theorem 1.

(ii) Each gate is a single parameter that scales a block B_{ϕ_i} of many weights, so P_{new} counts the K gates together with all block weights ϕ; hence $K \ll P_{\mathrm{new}}$ whenever each block carries many weights.

(iii) The remaining $P_{\mathrm{new}} - K$ new-weight directions have zero Jacobian columns at the growth point by Lemma 1(ii), so they lie in the Jacobian null space; this vanishing carries the factor α_i and disappears once $\alpha_i \neq 0$, which is why the statement is first-order at initialization only.

□

A.4 PROOF OF THEOREM 2

Proof.

(i) The constant-rank theorem: about any $\theta_0 \in U$ there are C^1 coordinates in which $F(u) = (u_1, \ldots, u_r, 0, \ldots, 0)$, so $F(U)$ is locally an embedded r-dimensional submanifold and $\mathrm{im}\, DF_\theta = T_{F(\theta)}M$ has dimension r.

(ii) If F were constant on the open set U, then $DF \equiv 0$ on U, contradicting $r > 0$; so F takes at least two values on U, at points that share rank r by hypothesis. If L_T is non-constant on $F(U)$, choose $\theta_1, \theta_2 \in U$ with $L_T(F(\theta_1)) \neq L_T(F(\theta_2))$; both have rank r, and $F(\theta_1) \neq F(\theta_2)$ since a non-constant function of F separates them.

□

Remark (When the hypotheses hold). Theorem 2 is local and generic: same-rank, different-function pairs exist wherever the rank is constant and positive. This holds on the open set where the rank attains its maximum, exactly where our measured checkpoints sit ($\mathrm{rank} = K = 36$), and our model, with smooth activations, is C^1 as the theorem requires.

B ESTIMATOR DETAILS

Rank operators and probes. From the singular values σ of the stacked Jacobian columns, we report two rank measures. The *hard rank* at tolerance τ is

$$r_\tau(J) = \#\{\, i : \sigma_i > \tau\, \sigma_{\max} \,\}, \qquad \tau = 10^{-3},$$

an integer, probe-independent in practice, and our headline measure. The *effective rank* (Roy & Vetterli, 2007) is

$$\mathrm{erank}(J) = \exp\big(H(\sigma)\big), \qquad H(\sigma) = -\sum_i \bar\sigma_i \log \bar\sigma_i, \qquad \bar\sigma_i = \frac{\sigma_i}{\sum_j \sigma_j},$$

the exponential Shannon entropy of the L_1-normalized singular values: a soft, real-valued measure that depends on the probe distribution, reported as secondary. Unless stated otherwise, each measurement uses 3 probe batches of 2 sequences of length 128, drawn as uniformly random tokens or as held-out WikiText text; the hard rank agrees across the two probe types, and we state which was used when reporting effective rank.

Exact-rank self-test. On a synthetic problem with known structure (ambient dimension 400, $K = 6$ planted directions), the matrix-free pipeline recovers the exact planted rank, recovers residual rank K when the planted directions are orthogonal to a planted old span, and recovers residual rank zero when they lie inside it, exercising both the hard-rank threshold and the transversal-residual computation end to end.

Per-checkpoint projector. The old-parameter projector P_{old} used for the transversal residual (Section 4.2) is recomputed from the current model at each checkpoint; it is not a fixed reference. Under isolation, the old parameters are frozen, but the downstream map from the old representation to the output changes as the appended blocks train, so $\partial f/\partial\theta_{\mathrm{old}} = (\partial f/\partial h_A)(\partial h_A/\partial\theta_{\mathrm{old}})$ changes through its first factor even though the second is fixed.

Thresholding. Hard rank counts singular values above $\tau\,\sigma_{\max}$ with $\tau = 10^{-3}$. For the residual $(I - P_{\mathrm{old}})J_\alpha$, we threshold relative to $\sigma_{\max}$ of J_α itself, not of the residual, so that a vanishing residual (added directions lying inside the old span) reads as rank zero rather than spuriously full.

Coarse proxies can manufacture collapse. An earlier coarse proxy, the effective rank of the gradient covariance projected to a low dimension, reported an apparent rank collapse under continual learning. The functional-Jacobian estimator does not reproduce it, and the drop did not survive multi-seed replication: it was an artifact of the low-dimensional projection and a single early checkpoint, not a functional-rank collapse. This is part of why we measure the functional Jacobian directly rather than a gradient-covariance surrogate.

Sampling the old span (transversality stress test). The transversal residual projects J_α off a subspace spanned by m randomly sampled old-parameter directions, so independence from $m = 64$ samples is a priori weaker than independence from the full old span. We therefore vary the sampling budget m and the probe length, which sets the output dimension $D = 2 \times \mathrm{seq} \times 50{,}257$, and recompute the residual. Every grid point uses a single probe batch of two sequences at seed 42 in fp32. Table 4 shows the outcome: the residual hard rank is $36 = K$ in every cell of a grid over probe lengths $\mathrm{seq} \in \{16, 32, 64, 128\}$ and budgets $m \in \{64, \ldots, 1024\}$. Sampling more of the old span does move its closest direction nearer to J_α (the minimum principal angle falls, for example $64.1^\circ \to 42.5^\circ$ as $m : 64 \to 1024$ at $\mathrm{seq} = 16$), but never enough to remove a dimension: the added directions stay *independent* of the sampled old span at every budget, while not being *orthogonal* to it. The projector is QR throughout; the points that exceed a single GPU's memory use a D-sharded tall-skinny QR across GPUs, validated to reproduce the single-GPU result exactly (rank 36; principal angles to 10^{-2} degrees) at $\mathrm{seq} = 128, m = 256$.

Table 4: Transversality stress test: minimum principal angle (degrees) between the $K = 36$ added gate directions and the span of m sampled old-parameter directions, across probe length (seq) and sampling budget m. **The residual hard rank is** $36 = K$ **in every cell**, so independence from the sampled old span is invariant to how much of it we sample, even as the minimum angle shrinks with m.

seq$\backslash m$	64	128	256	512	1024
16	64.1	57.2	52.3	47.3	42.5
32	71.7	67.4	62.0	56.4	51.0
64	77.5	73.0	67.7	62.7	56.8
128	79.6	74.8	69.8	64.8	59.2

C PER-SEED CONTINUAL-LEARNING DETAIL

Table 2 reports the old-domain function as a range over seeds $42/43/44$; the per-seed values are in Table 5. In every run, the added subspace is preserved (hard rank and transversal residual $36 \to 36$) while the old-domain function degrades, is forgotten, or is destroyed, so the decoupling holds seed by seed, not only on average. Perplexity is the old-domain value before continual learning versus after it (PPL_A for the two-domain and replay conditions; PPL_B after training on the third domain for the three-domain sequence).

Table 5: Per-seed detail behind Table 2 (E5). Old-domain perplexity before $\to$ after continual learning, with the added-subspace hard rank and transversal residual (grown $\to$ after-CL). Rank and residual are $36 = K$ in every run; only the retained function varies.

Seed	Old-domain PPL (pre $\to$ post)	Hard rank	Transv. residual
2-domain isolation A→B, A degraded (PPL_A)			
42	$25.9 \to 93.0$	$36 \to 36$	$36 \to 36$
43	$25.9 \to 92.2$	$36 \to 36$	$36 \to 36$
44	$25.9 \to 93.3$	$36 \to 36$	$36 \to 36$
3-domain sequence A→B→C, B forgotten (PPL_B)			
42	$28.3 \to 480.7$	$36 \to 36$	$36 \to 36$
43	$28.4 \to 459.1$	$36 \to 36$	$36 \to 36$
44	$28.4 \to 437.0$	$36 \to 36$	$36 \to 36$
Replay onto conflict, A destroyed (PPL_A)			
42	$25.9 \to 127.7$	$36 \to 36$	$36 \to 36$
43	$25.9 \to 127.4$	$36 \to 36$	$36 \to 36$
44	$25.9 \to 125.5$	$36 \to 36$	$36 \to 36$

D SUBSPACE ROTATION OF THE ADDED DIRECTIONS

For the $K = 36$ new-block gate directions, we form the JVP columns $J_\alpha = \partial f / \partial \alpha$ (Appendix B) and compute the principal angles between two gate subspaces; the bin distribution (counts out of 36) shows how many directions are shared (small angle) versus re-tasked (near $90°$). Angles are averaged over 5 probe batches, and for B-vs-C over seeds $42/43/44$ (agreement to std ≤ 1.4 per bin). The random baseline is two independent 36-dimensional subspaces in $\mathbb{R}^D$ ($D = 2 \times 128 \times 50{,}257$; 5 draws), which in this dimension are essentially orthogonal, with all 36 angles in $[85, 90]°$. This is the same measurement as Figure 1.

Table 6: Principal-angle distribution: mean count of the 36 directions per angular bin (degrees). The random baseline concentrates every direction in $[85, 90]°$; B-vs-C has a small sub-$80°$ core and is otherwise near-orthogonal.

Comparison	$[0, 45)$	$[45, 60)$	$[60, 70)$	$[70, 80)$	$[80, 85)$	$[85, 90]$
Random baseline	0.0	0.0	0.0	0.0	0.0	36.0
B-vs-C (mean over seeds)	0.0	0.9	0.1	4.1	10.3	20.5
Grown $\rightarrow$ after-CL (gate-zero)	0.0	0.1	0.9	0.9	7.4	26.7

Against the random baseline, which places all 36 angles at $\geq 85°$, the B-vs-C distribution has about $15/36$ below $85°$ and about $5/36$ below $80°$, but only about one direction below $60°$: a small shared core, with the rest rotated to near-orthogonal. The dimension is conserved throughout (hard rank $36/36$; main text), so forgetting is capacity reuse rather than loss.

Caveat. The B-specialized reference is a proxy: the two-domain (after-B) state was checkpointed only at seed 42, so the fixed B reference is the exp1 seed-42 checkpoint, a same-recipe but $\sim 3\%$-off stand-in (PPL_B 29.2 versus 28.3); the C seed is varied over $42/43/44$.

E EFFECTIVE RANK ACROSS GROWTH FACTORS

Table 7 extends the E8 comparison of Table 3 from $g = 4$ to $g \in \{2, 3, 4\}$ ($K = 12, 24, 36$). For each growth factor, the new blocks are either cloned round-robin from the base (the gate-zero operator) or freshly randomly initialized; the gate is zero in both cases, so the grown model is function-preserving. We measure the gate Jacobian at the moment of growth with the random-token probe (5 probe batches of 2×128 tokens) and repeat over seeds $42/43/44$, where the seed sets the clone noise and the random initialization.

Table 7: Effective rank of the K added directions at the moment of growth, by growth factor and new-block initialization (mean $\pm$ std over seeds $42/43/44$). Hard rank equals K in every cell and for every seed.

g	K	New-block initialization	Hard rank	Eff. rank at growth	Eff. rank$/K$
2	12	Clone (round-robin)	12	11.39 ± 0.02	0.949
		Random	12	11.96 ± 0.02	0.996
3	24	Clone (round-robin)	24	20.39 ± 0.04	0.850
		Random	24	23.81 ± 0.04	0.992
4	36	Clone (round-robin)	36	28.81 ± 0.05	0.800
		Random	36	35.67 ± 0.04	0.991

Hard rank and parameter count cannot distinguish the two initializations, since both add exactly K directions at every g. Effective rank can. Random initialization stays near full at every growth factor (effective rank$/K \geq 0.99$), while cloning falls from 0.95 at $g = 2$ to 0.80 at $g = 4$, consistent with round-robin cloning reusing each base block $g - 1$ times. The effective rank varies by at most 0.05 across seeds.