# THE MATERNAL PHASE OF LANGUAGE MODEL TRAINING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

What does initialization contribute to the training of a language model, and when and how does training take over from it? Early training is dominated by the initial weights, yet runs that differ only in their seed end with nearly the same loss, so control must pass from what a network is given to what it learns, at a point that has not been located. Embryos make the same handover at the maternal-to-zygotic transition, from materials deposited in the egg to the embryo's own genome, and embryologists locate it by following gene expression through the stages of development. We do the same with a developmental atlas of Pythia that follows the expression of every MLP unit across fifty runs at five sizes and four larger models, and we retrain small models to intervene where the atlas can only correlate. We find that (i) training begins with a maternal phase, in which expression is a copy of initialization set by each unit's projection onto the mean input of its layer; (ii) the phase ends not because the units' weights change, since they keep the initial pattern, but because the mean input rotates, the deepest layers first; (iii) its duration is set by the accumulated update measured against the initial weight scale; and (iv) little of it remains afterwards, and a small network damaged once training is under way restores its loss and its repertoire of unit behaviours but reassigns roles among its units. More broadly, the atlas suggests that the toolkit of developmental biology, from staging to transplantation and lesion, can be turned on training, provided that what a network inherits is first separated from what it learns.

## 1 INTRODUCTION

A fertilized egg does not begin from nothing. Before the embryo's own genes are switched on, its first cell divisions are driven by the RNAs and proteins that the mother deposited in the egg. Control then passes, at the maternal-to-zygotic transition, from these inherited materials to the programme of the embryo's own genome, and embryologists know both when this transition happens and what sets its timing (Schier, 2007; Tadros & Lipshitz, 2009; Newport & Kirschner, 1982a); the top of Figure 1 sketches this handover. Knowing where it lies has practical as well as conceptual value, because a measurement taken before the transition describes what the egg was given rather than what the embryo is doing, and the two are easy to confuse.

The training of a language model also begins from inherited material, namely the random weights drawn before any data are seen, and it too is dominated by that material at first: early training stays close to initialization, which is one reason it is run under a learning-rate warmup (Chizat et al., 2019; Kosson et al., 2024b; Kalra & Barkeshli, 2024). Yet the network that training finally produces depends on its initialization in some respects and hardly at all in others. Runs that differ only in their random seed reach nearly the same loss and, once the arbitrary ordering of their units is allowed for, often closely related solutions (Sellam et al., 2022; van der Wal et al., 2025; Fehlauer et al., 2025; Entezari et al., 2022; Ainsworth et al., 2023). On the other hand, the initial weights decide which sparse subnetworks can be trained (Frankle & Carbin, 2019), disturbances during an early phase have lasting effects (Achille et al., 2019; Frankle et al., 2020), and the trajectory of training is highly sensitive to its initial conditions (Altıntaş et al., 2025). Somewhere in training, therefore, control must pass from what the network was given to what it learns, as it does in the embryo. Unlike the embryo's transition, however, this one has not been located. We do not know what a network inherits from its initialization, when and by what route training takes over, or what remains of it afterwards. Figure 1 sets the two handovers side by side and previews the answers that the rest of the paper gives.



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

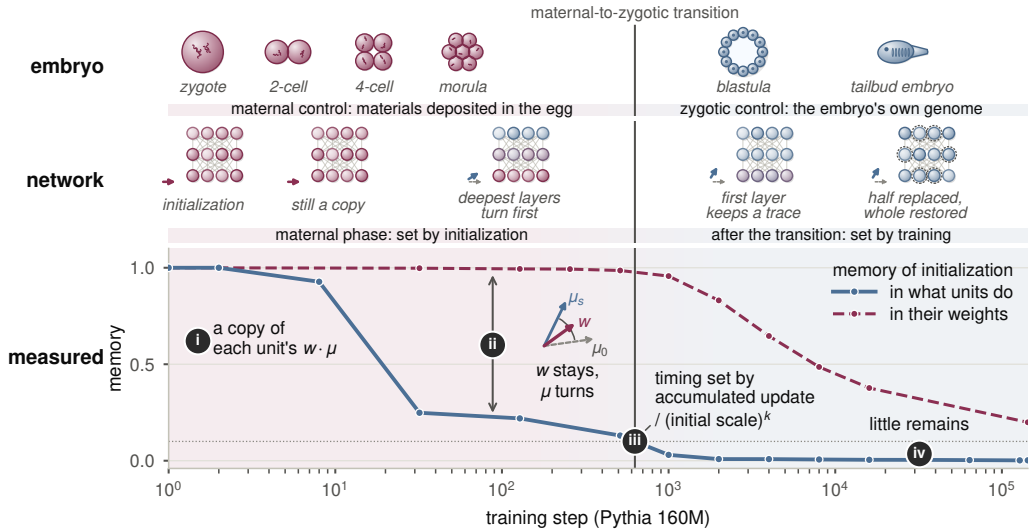


Figure 1: **A network hands control from its initialization to training, as an embryo hands it from maternal materials to its own genome.** Top, an embryo, aligned with the network only at the transition. Middle, three MLP layers (the first at the bottom), units shaded by expression and the mean input drawn as an arrow. Bottom, memory of initialization in Pythia 160M, averaged over ten runs (solid) and recomputed in one run with the current weights read against the initial mean input (dashed); the vertical line marks clearance. Numerals mark the four findings listed below.

The gap matters in practice because developmental readings of training have become common, from an explicit "embryology" of language models (Chiang et al., 2020; Wang et al., 2025) to analyses of how features and circuits appear across checkpoints (Ge et al., 2026; Tigges et al., 2024), and many of them rest on the early checkpoints in which such a transition would take place. If those checkpoints still carry the inheritance, a reading of them may describe what the network was given rather than what it has learned, which is exactly the confusion that embryologists avoid by locating the transition.

That training has an early phase of this kind, which we call the maternal phase, is expected from warmup and from the work cited above, and we do not claim it as a discovery. What we ask is what the phase consists of at the level of single units, how it ends and what sets its length. Our developmental atlas of the Pythia suite (Biderman et al., 2023; van der Wal et al., 2025) follows the expression of every MLP unit through training, as an embryologist follows a gene through the stages of an embryo, reading each run as an individual and each checkpoint as a developmental stage, across fifty runs at five sizes and single runs of four larger models (Section 2). Where the atlas can only correlate, we retrain small models from the released initial weights and intervene directly.

The answers follow the questions in order (Figure 1). (i) What a unit inherits is a state that the untrained network defines for itself, namely the unit's projection onto the mean input of its layer; the first checkpoints are copies of this state, and which unit inherits which value does not affect what the network computes (Section 3). (ii) Training takes over not because the units' own input weights change but because learning in the blocks below each layer rotates the mean input against which the layer's units are read, so that the inherited state leaves behaviour layer by layer, the deepest layers first (Section 4.1). (iii) How long the handover takes is set by the accumulated update measured against the initial weight scale, a single ratio through which both the learning rate and the initialization scale act, consistent with the later clearance of the two largest Pythia models, which are initialized at about twice the usual scale (Section 4.2). (iv) Little of the inheritance remains afterwards, and a small network damaged once training is under way restores its loss and its repertoire of unit behaviours while reassigning roles among its units, which is regulation in the embryologist's sense (Section 5).

## 2 A DEVELOPMENTAL ATLAS OF PYTHIA

**The released Pythia runs are read as individuals observed at developmental stages, and retrained copies of the two smallest sizes are used wherever we need to intervene.** Pythia trains decoder-only transformers of many sizes on the same data in the same order, and PolyPythias adds nine further runs

at each of the five smallest sizes that differ from the original only in the random seed (Biderman et al., 2023; van der Wal et al., 2025). Since that gives repeated individuals of one species alongside a series of species of different size, we read each run as an individual and each checkpoint as a developmental stage. The atlas has two panels: a population panel of the ten individuals of each size from 14M to 410M, and a scale panel of one run each from 70M to 12B, read more densely at the start of training, at every power of two up to step 512; the released 2.8B model is excluded for a reason of data integrity (Appendix B.1). All sizes train for 143,000 steps with a linear learning-rate warmup over the first 1,430, and everything this paper describes begins inside that warmup. Where the released checkpoints can only be correlated, we retrain 14M and 70M networks from their released initializations and intervene on them. They train at a much smaller batch than Pythia does, so we treat them as a model of the maternal phase rather than a reproduction of it (Appendix B.2).

**One measured quantity, the expression of a unit, and two derived from it, its memory of initialization and the step at which that memory clears, carry the argument of the paper.** At each checkpoint we run the same set $C$ of 46 sentences (Appendix F) through the model and record, for every unit $i$ of every MLP layer, the mean absolute value of its activation over tokens and sentences. We call this quantity the expression of the unit,

$$E_i(s) \;=\; \frac{1}{|C|}\sum_{c\in C}\frac{1}{|T_c|}\sum_{\tau\in T_c}\big|\,h_i(\tau,c;s)\,\big|, \tag{1}$$

where $T_c$ are the valid tokens of sentence $c$ and $h_i$ is the GELU output of unit $i$ at training step $s$. Two quantities derived from it carry the argument. The first is the memory of initialization, the Pearson correlation taken across units between expression at step 0 and expression at step $s$,

$$M_{\mathcal{U}}(s) \;=\; \mathrm{corr}_{i\,\in\,\mathcal{U}}\big(E_i(0),\,E_i(s)\big), \tag{2}$$

where $\mathcal{U}$ is every MLP unit of the network, or the units of a single layer when we speak of that layer's memory. The second summarizes the decay of $M$ by a single number, the clearance step,

$$s^* \;=\; \min\{\, s \;:\; M_{\mathcal{U}}(s) < 0.10 \,\}, \tag{3}$$

located by linear interpolation against $\log s$ between the two stages that bracket the crossing. The threshold of 0.10 in equation 3 is a convention. Where a conclusion depends on it we say so, and Section 4 tests the central claim about timing without any threshold at all.

**The atlas resolves differences only above floors that it sets itself, and it does not depend on the sentences it is read with, although it reads the transition least reliably.** Two floors bound what the atlas can resolve (Appendix C). The first is set by repeat runs that differ only through the nondeterminism of execution (Summers & Dinneen, 2021), and a difference between two networks smaller than theirs is not a difference between networks at all. The second is set by the distance between a unit and its nearest neighbour inside one network (Li et al., 2016), and a difference smaller than that does not separate a unit from its neighbours, which makes it the finest resolution the atlas has. Neither floor is a constant of the method, and the first in particular rises with the size of the model and when a run is executed on mixed hardware, so each comparison in Section 5 is read against the floor that belongs to it. The atlas does not depend on the particular 46 sentences either: read with half of them, it moves the clearance step less than the choice of network does (Figure 8). The two halves agree least at the transition itself, however, so the stage at which the inherited pattern is being replaced is also the stage at which the instrument reads it least reliably.

## 3 What a unit inherits

**The first checkpoints are copies of the initial network, and what they copy is one number per unit: its projection on the mean input of its layer.** The first checkpoints of a training run are copies of the initial network: taken across units, the expression at step 2 correlates with the expression at initialization at 1.000 at every size, and the copy then decays the faster the larger the model (Figure 2a). That early training changes a network so little is expected, since the learning rate is still tiny and warmup exists precisely to keep the first updates small (Kosson et al., 2024b; Kalra & Barkeshli, 2024; Fehlauer et al., 2025; Ge et al., 2026); what a unit-level atlas adds is what the copied state consists of. It consists of one number per unit, and that number comes from the network rather than from the data, since reading the step-0 network with uniform random token identifiers instead

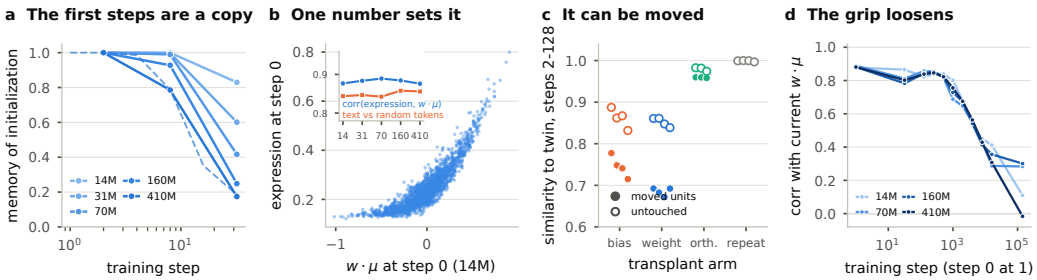


Figure 2: **The first checkpoints copy one number per unit, which can be moved and whose hold loosens with training.** (a) Memory of initialization, mean of ten runs per size; dashed, the 410M run of the scale panel. (b) Step-0 expression against $w \cdot \mu$ in one 14M run; inset, both correlations by size. (c) Similarity to the untransplanted twin over steps 2 to 128, one point per run; orth., a weight change orthogonal to $\mu$; repeat, no transplant. (d) Correlation of expression with the current $w \cdot \mu$.

of sentences leaves each unit's expression correlated with its expression on real text at 0.84 to 0.86 across the five sizes. The reason is a property of an untrained unit. Its pre-activation is $z = w^\top x + b$, where $x$ is the normalized input to its MLP layer, and at initialization every bias is zero, so the unit's mean pre-activation is $w \cdot \mu$, its projection onto the mean input $\mu$ of the layer (Francazi et al., 2024; Simon et al., 2026). That operating point is what an untrained unit's expression reflects: the two correlate at 0.88 at every size (Figure 2b). The mean input is a direction the untrained network defines for itself, and each unit's position relative to it is the maternal material the unit starts training with.

**Moving that number moves what a unit does but not what the network computes, and its hold on the unit loosens as training proceeds.** Because this material is a property of the weights, it can be moved from one unit to another, which tests whether it is causal rather than merely correlated with what units do. In retrained 14M and 70M networks we exchanged the operating points of a random half of the MLP units in every layer and trained each modified network beside an untransplanted twin with the same initial weights and data order (Appendix B.3). Over the first hundred steps the moved units depart from the twin while the untouched units stay with it (Figure 2c), so the inherited state is causal for the maternal phase. It is not, however, a lever on the network as a whole. A permutation within a layer leaves that layer's distribution of operating points unchanged, and an untrained network has no predictions that could be spoiled, so the loss never moved by more than 0.008 nats over steps 50 to 200. The hold that the operating point has on a unit is tightest at the beginning and loosens as training proceeds: in 14M the correlation between expression and the unit's own current $w \cdot \mu$ falls from 0.88 at initialization to 0.54 at step 4,000 and to 0.11 at the end of training (Figure 2d).

**What a unit inherits governs the maternal phase and not the development that follows, though the obvious way of testing this gives the wrong answer.** A fate map for a network defines each unit's fate from the shape of its expression trajectory and asks what at initialization predicts it. Read that way, initialization looks decisive, explaining a fifth to a third of the variance of step-0 expression. The signal is an artefact of the copied checkpoints, which carry the initial value into the label, and it survives shuffling the units' identities, so it requires no development at all: once the trajectory starts after those checkpoints the same quantity falls to a few thousandths, and only a small residue of the inheritance survives into later development. Appendix D gives the analysis, the controls that identify the artefact, and the diagnostic we draw from it.

## 4 The Transition

### 4.1 How the inherited state leaves behaviour

**The memory does not vanish at once but layer by layer, the deepest layers first, so the number for a whole network is an average over very different schedules.** The memory of initialization falls away over the first hundreds to thousands of steps, sooner the larger the model (Figure 3a), and within a network it does not vanish all at once but decays layer by layer, the deepest layers going first (Figure 3b). In 410M the last six layers cross the 0.10 threshold by step 32, the sixteen layers above the second cross it between steps 260 and 1,489 in order of depth, the second layer crosses it at step 4,831, and the first layer never crosses it at all. The same ordering holds at every size. The memory

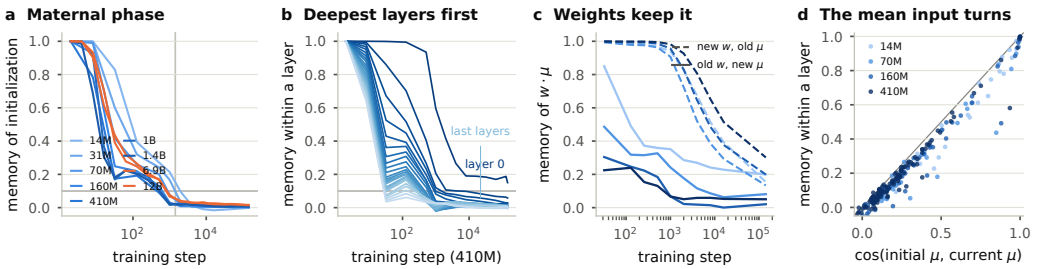


Figure 3: **The inherited state leaves behaviour layer by layer, the deepest layers first, because the mean input turns while the weights keep the initial pattern.** (a) Memory of initialization; mean of ten runs at 14M to 410M, single runs at 1B to 12B; orange, the two models initialized at about twice the usual scale; vertical line, end of warmup. (b) Memory within each layer of 410M, mean of ten runs. Horizontal lines in (a, b) mark the threshold of 0.10. (c) Memory of $w \cdot \mu$ recomputed with either the weights or the mean input at its initial value, one run per size. (d) Layer memory against the cosine of the turn of its mean input, runs of (c) at steps 32 to 1,000.

of a whole network is therefore an average over layers that clear at very different times, which is one reason its crossing step is sensitive to the threshold at which the crossing is defined.

**What removes it is not a change in the units' own weights, which still hold the initial pattern, but a turn of the direction those weights are read against.** The operating point $w \cdot \mu$ can change either because $w$ changes or because $\mu$ changes, and the two can be separated by recomputing the quantity with the current weights and the initial mean input, or the other way round (Appendix B.4). At step 128 of 14M training, the version computed with the current weights and the initial mean input still correlates with the initial operating points at 0.99, whereas the version computed with the initial weights and the current mean input has fallen to 0.49, which is close to the memory of expression itself at 0.38 (Figure 3c). The input weights at that point differ from their initial values by about 5% of their norm. The mean input, by contrast, has rotated.

**Behaviour follows that turn almost exactly, and the intervention we designed to prove the point changed the measurement instead of the dynamics.** For weight vectors drawn independently from an isotropic distribution, the correlation across units between two projections is the cosine between the two directions (Cho & Saul, 2009), so a layer whose mean input has turned by a given angle should retain a memory equal to the cosine of that angle (Appendix B.4). The behaviour of the units follows this prediction. Over all layers of the four sizes at steps 32 to 1,000, the memory of expression tracks the cosine between the initial and the current mean input (Figure 3d): the two quantities correlate at 0.98, with the memory lying on average between 0.04 and 0.09 below the cosine, and among layers read at the same step it is still between 0.95 and 0.998, so the agreement is not a shared trend over time. At 410M, at step 512, the weights still reproduce the initial operating points at 0.995 while the memory of expression has fallen to 0.09.

To move from this description to a cause, we replaced each layer's current mean input by its step-0 value with a hook on every MLP input projection (Appendix B.5). Applied at measurement alone, in an ordinary run, the hook recovers most of the lost memory, raising it to 0.85 at step 2,000 and 0.66 at step 4,000, so the information is still held in the weights; later in training even this fails, because the weights themselves have moved (Figure 2d). Applied during training, the same hook keeps the memory high, but only because expression is then read against the very direction being held, so it changes the measurement rather than the dynamics. The rotation account therefore remains a description rather than a demonstrated cause, and an intervention that alters the rotation without altering the direction against which expression is read is left to future work.

**The turn is written by the blocks below each layer, which is why the first layer keeps its memory and the deepest lose theirs first.** At step 1,000 the cosine between a layer's initial and current mean input falls with depth in 14M, 70M and 160M (Figure 4a): the first layer has barely turned, at a cosine between 0.85 and 0.90, while the deepest layers have turned the most, the same order in which the layers clear. Splitting the token-mean residual stream into the contributions of the blocks that write to it shows where the turn comes from (Figure 4b). In 14M, 70M and 160M at steps 32 to 1,000, the change in the mean entering every layer above the first is shared between the earlier MLP blocks and the attention blocks, which account for 45 to 95% and 5 to 55% of it respectively when projected on

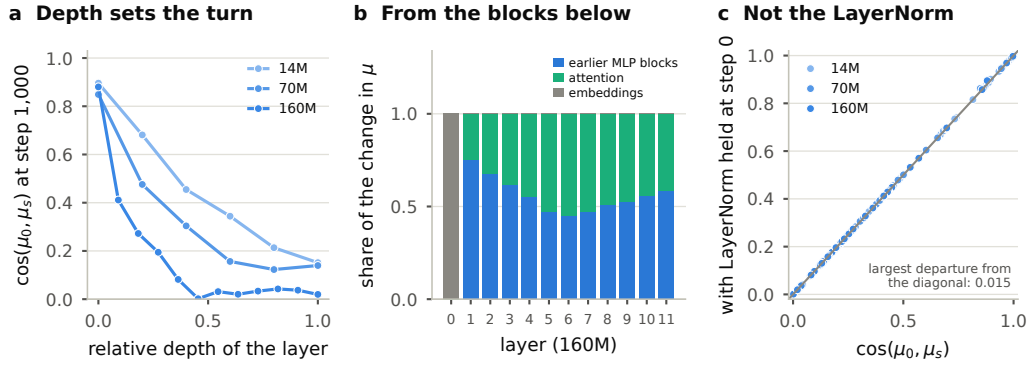


Figure 4: **The turn of the mean input is written by the blocks below each layer, so it grows with depth and owes nothing to the LayerNorm.** (a) Cosine between each layer's initial and current mean input at step 1,000, one run per size. (b) Share of the change in the mean input of each layer of 160M that comes from each kind of block below, along the direction of change. (c) The same cosine with the LayerNorm frozen at step 0, against its ordinary value, runs of (a) at steps 32 to 1,000.

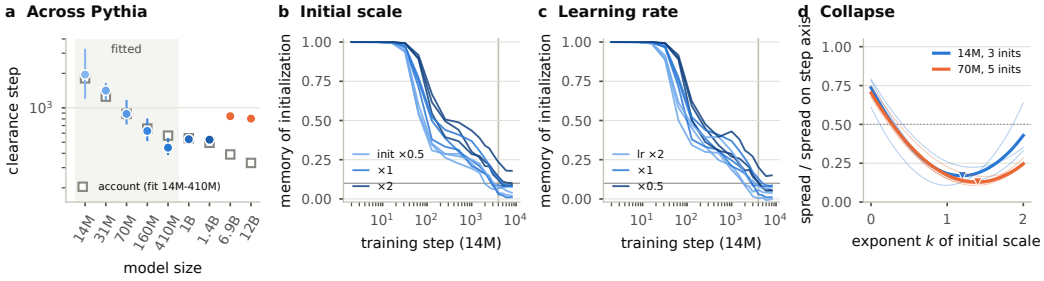


Figure 5: **Clearance comes sooner in larger models except the two with a larger initialization, and is set by the accumulated update measured against the initial scale.** (a) Clearance step; at 14M to 410M, circles give the mean of ten runs and bars the range of single runs; orange, the two models initialized at twice the usual scale. (b, c) Retrained 14M networks with the initial scale (b) or the learning rate (c) changed, three initializations each. (d) Spread between the curves of (b, c) on the axis $U_k$, relative to the step axis; thin lines, single initializations; triangles, the best $k$.

the direction of change, while the embeddings contribute nothing measurable. The normalization is not the source either, since holding the LayerNorm gain and bias at their initial values changes the cosine by at most 0.015 (Figure 4c). The first layer reads only the embeddings, so its mean input barely turns, which is why it keeps its memory when every other layer has lost it. The deepest layers clear first because their mean input accumulates the changes made by every layer below them, an explanation consistent with the fall of the cosine with depth in Figure 4a.

### 4.2 What sets the schedule

**Larger models clear sooner, but the two largest break the trend, and the released models cannot say why.** The clearance step of the population-mean memory falls from 1,952 at 14M to 448 at 410M (Figure 5a), an ordering that survives a change of threshold or a switch to rank correlations (Appendix E). A simple account predicts this trend. Under Adam each weight moves by roughly the learning rate at each step, so a pre-activation summed over $d$ inputs and $L$ preceding layers drifts in proportion to the learning rate times $dL$, and during warmup that drift accumulates with the square of the step, which makes larger models clear sooner despite their smaller learning rates. Fitted as a single threshold on the five population sizes (Appendix B.6), the account extends correctly to 1B and 1.4B but fails at the two largest models, in direction as well as in size: it places 6.9B and 12B at steps 391 and 330, whereas they clear at steps 845 and 805.

What the account leaves out is the initialization scale. Through a documented configuration error, 6.9B and 12B draw their input-side weights with a fixed standard deviation of 0.02, about twice the width-scaled value used up to 1.4B (EleutherAI, 2025), and a state written at a larger scale may well

take longer to erase (Kosson et al., 2024b; Chizat et al., 2019). The released panel, however, cannot say how much of the delay this explains, both because the size of the delay depends on the threshold at which clearance is read and because, among the six width-scaled models, the initial scale is a function of the width, so that no fit can separate the two (Appendix E). We therefore turn to networks in which the initialization scale can be changed while everything else is held fixed.

**In networks we control, a larger initialization delays clearance and a larger learning rate hastens it.** We retrained 14M under five conditions, with every block weight matrix scaled at initialization by 0.5, 1 or 2, or with the learning rate scaled by 2 or 0.5, each from three initializations (Appendix B.8). Both directions are plain in the memory curves: doubling the initial scale moves the fall of memory to later steps (Figure 5b), and doubling the learning rate moves it to earlier ones (Figure 5c). Their size is harder to pin down with a threshold. A pilot on a single initialization, read at the 0.10 threshold, gave a clearance step that grew roughly as the square root of the initialization scale and fell roughly as the square root of the learning rate, but these exponents drifted with the threshold, reaching $+1.16$ at 0.25, so we compared whole curves rather than crossings.

**The initialization scale and the learning rate act on clearance only through the accumulated update measured against the initial weight scale, at 14M and at 70M alike.** The quantity on which the curves are compared is the accumulated learning rate divided by a power of the initial scale, $U_k(s) = \alpha^{-k} \sum_{t=1}^{s} \eta_t$, where $\eta_t$ is the learning rate at step $t$, $\alpha$ is the factor by which the initial block weights are scaled, and the exponent $k$ is chosen so that the memory curves of the five conditions coincide, an invariance that analyses of Adam predict with $k = 1$ (Kosson et al., 2024b) and whose learning-rate part is the clock proposed by Das et al. (2026). On this axis the curves collapse (Figure 5d). At 14M the spread between them falls to 17% of its value on the raw step axis at the best exponent, $k = 1.20$, against 74% when the learning rate alone is accumulated ($k = 0$), and a separate exponent fitted to the learning rate comes out at 1.05, so the learning rate enters only through its sum. The same holds at 70M, where the curves collapse to 13% of the raw spread at $k = 1.40$. Clearance is therefore set by how far the accumulated update has gone relative to the initial scale, with a power between about 1 and 1.4 whose exact value may differ between sizes (Appendix B.8).

The clock has a precedent in embryos. In the frog *Xenopus*, the midblastula transition that starts zygotic transcription is timed neither by a fixed number of divisions nor by a fixed time, but by the ratio of nucleus to cytoplasm reaching a critical value (Newport & Kirschner, 1982a;b). The end of the maternal phase in a network is likewise set by a ratio, of accumulated change to inherited scale, and changing either term shifts its timing, which biologists call heterochrony (Gould, 1977).

## 5 What the transition leaves behind

### 5.1 What remains of the maternal state

**Clearance is a fall below a threshold rather than an erasure, and the first layer and the late trajectory classes still carry a trace of initialization.** Clearance marks the step at which the memory of initialization falls below 0.10, not the step at which it vanishes, and two residues outlast it, as trained networks are known to keep traces of their initialization (Belrose & Scherlis, 2024; Zhu et al., 2025). The first is the first layer, whose mean input barely turns, and which still ends training with a memory of 0.04 at 14M rising to 0.15 at 410M (Figure 6a), so that this residue grows with size even as the clearance step of the whole network shrinks. The second is a trace in the trajectory classes of Section 3 and Appendix D. When those classes are built only from trajectories that start after each size's clearance, class membership still explains up to ten times more of step-0 expression than a permutation null, significantly so in every case at 410M (Figure 6b). The effect is at least thirty times smaller than the one obtained when the copied stages are included, but it is there.

**A disturbance written into the maternal state is erased along with it.** These residues are faint traces of the initial pattern, and whether a change written into the maternal state can likewise outlast it is a separate question. The transplant of Section 3, which exchanged the operating points among a random half of the units in every layer, answers it, because it can be followed through the rest of training (Figure 6c). Its effect is read as the distance between the log expression trajectory of each unit and that of the same unit in the untransplanted twin (Appendix B.10). That distance shrinks from 0.11–0.13 to 0.04–0.06 by step 512, as the maternal state that carries it is erased. It grows again after step 1,000, reaching 0.25–0.32 by step 8,000, but this late divergence is not a memory of

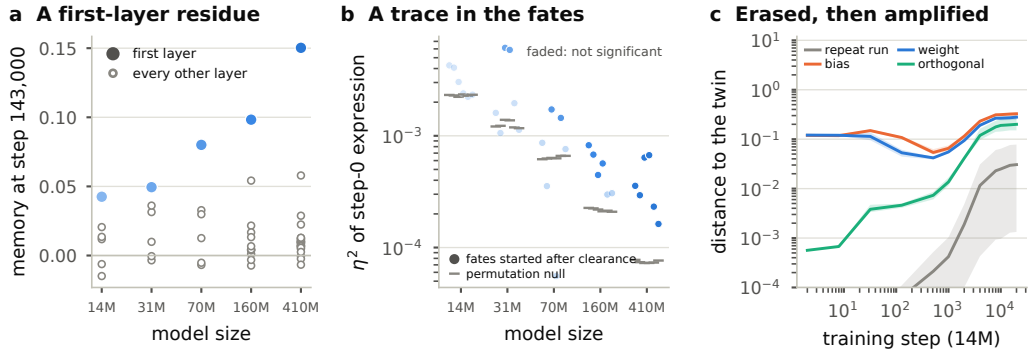


Figure 6: **A faint trace of initialization outlasts clearance, but a disturbance written into the maternal state is erased along with it.** (a) Final memory, mean of ten runs per size. (b) Variance of step-0 expression explained by trajectory classes started after clearance, three runs per size. (c) Distance to the twin after the transplant of Figure 2c, mean and range over three initializations.

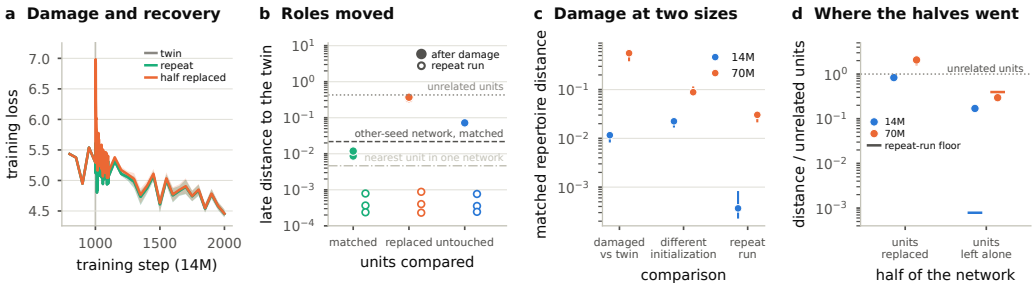


Figure 7: **After a lesion, a 14M network restores its loss and its repertoire while reassigning roles, whereas at 70M the repertoire does not return.** (a) Training loss of 14M networks with half the MLP units of every layer replaced at step 1,000; mean and range over three initializations. (b) Late distance to the twin at 14M. (c) Matched repertoire distance after damage, beside its two baselines. (d) Distance of each half of the network relative to unrelated units of its size. In (c, d), points are medians and bars ranges over three initializations.

the transplant: the orthogonal control, which perturbs the same units by the same amount without moving their operating points, starts two hundred times closer and ends in the same range, so training amplifies any small difference rather than remembering this particular one (Altıntaş et al., 2025; Summers & Dinneen, 2021).

### 5.2 How the network answers damage

**Replacing half the units at step 1,000 does real damage, and the loss recovers in full.** When Driesch separated the first two cells of a sea-urchin embryo, each formed a complete though smaller larva, so that the whole was restored, not by returning each cell to its original course but by having a cell that would have built half of the larva build all of it (Driesch, 1892). Embryos of this kind are called regulative, in contrast to mosaic embryos, whose cells follow fates fixed so early that a lost part stays lost (Gilbert & Barresi, 2016). Asking the same of a network requires a disturbance that does damage, which the transplant does not. At step 1,000 of 14M training we therefore replaced a random half of the MLP units in every layer with fresh draws from the initialization, with a twin and a repeat run for each of three initializations (Appendix B.9). Over the next fifty steps the training loss exceeded the twin's by 0.36 to 0.38 nats (Figure 7a), and it then recovered, as work on relearning after damage and on re-initialization leads one to expect (Plaut, 1996; Taha et al., 2021; Zhou et al., 2022): the gap fell below 0.05 nats within 250 to 350 steps, and the damaged networks ended training within 0.006 nats of their twins.

**The repertoire of unit behaviours returns but the roles are partly reassigned, so the 14M network is regulative at the level of the whole.** Repertoires are compared over steps 2,000 to 10,000 by matching the units of each layer one to one with those of the twin (Appendix B.10). The match

leaves a mean distance of 0.009 to 0.012, which is closer than the 0.018 to 0.025 between twins grown from different initializations, in every run (Figure 7b). The roles, however, moved. A reinitialized unit ends 0.34 to 0.38 away from the unit it replaced, close to the 0.41 to 0.45 that separates unrelated units, so it does not return to its predecessor's course, in line with the relocation of removed functions reported by Lo et al. (2024). The untouched half moves as well, far beyond the drift between repeat runs, ending 16 to 17% of the way to the unrelated-unit level, and on either scale of distance it stays closer to its own course than to an unrelated unit (Appendix B.10). What the network computes and the set of behaviours its units display are therefore restored while roles are partly reassigned, so development at 14M is regulative in Driesch's sense at the level of the whole.

**At 70M the loss recovers but the repertoire does not, and the two sizes differ in how the halves of the network respond.** In an exploratory repeat at 70M, again with three initializations, the same lesion raised the loss by only 0.19 nats, and by the end of training the gap was within 0.003 nats. The repertoire, however, did not come back: one-to-one matching leaves a distance of 0.40 to 0.61, against 0.079 to 0.110 between twins of different initializations (Figure 7c). The replaced units end beyond the unrelated-unit level of their size, although they have not fallen silent, while the units left alone do not move measurably beyond a repeat-run floor that at 70M already lies more than a third of the way to that level, so any re-tuning of the survivors is below what the measure can resolve (Figure 7d; Appendix B.9). The lesion at 70M was the weaker of the two, and the contrast rests on two sizes with three initializations each, so what we report is regulation at one size, under one kind of damage, at one moment and for 10,000 steps; larger models are left to future work.

## 6 Related work

**Reading training as development.** Language-model pretraining has been described as an embryology (Chiang et al., 2020; Wang et al., 2025), and stagewise development (Hoogland et al., 2025) and critical periods (Achille et al., 2019) are established ideas. What we add is not the analogy but its use: specific concepts of embryology, namely a maternal phase, the timing of its end and the distinction between regulative and mosaic development, become measurements and interventions on single units.

**Initialization and early training.** That early training stays close to initialization, and that traces of initialization survive it, has been shown for the features and the outputs of the same Pythia runs (Ge et al., 2026; Fehlauer et al., 2025), and the token-shared direction of the residual stream is well documented (Biś et al., 2021; Sun et al., 2024). That the pace of early training is set by the update measured against the weights is also known: under Adam it predicts that the initial scale acts with an exponent of one (Kosson et al., 2024b), and accumulated learning rate has been proposed as the clock by which adaptive optimizers forget their initialization (Das et al., 2026). What is new here is what the inherited state is for a single unit, why it is lost, through a rotation of the mean input rather than a change in the weights, and the measured exponent, which lies somewhat above one.

**Damage and recovery.** Networks relearn quickly after damage (Plaut, 1996), re-initializing part of a network during training is followed by recovery (Taha et al., 2021; Zhou et al., 2022), and removed functions move to other neurons (Lo et al., 2024). What we add is a unit-level account of how the whole is restored while roles are reassigned among its units, with an exploratory contrast at 70M. Appendix A reviews this work in full, and Table 1 classifies each result as known, extended or new.

## 7 Discussion

**Summary.** Initialization hands training a transient state that the untrained network defines for itself, each unit's projection on the mean input of its layer (Section 3). This state governs units for the first hundreds to thousands of steps and then leaves behaviour, not because it is erased, since the weights keep it, but because the mean input it is read against turns (Section 4.1). How long it lasts is set by the accumulated update measured against the initial weight scale (Section 4.2). Little of it remains afterwards (Section 5.1), and a 14M network damaged once development is under way restores the whole while reassigning the roles of its units (Section 5.2).

**Contributions.** An early phase is expected from warmup and is not our discovery. We add a unit-level account of what is inherited, how it leaves behaviour, and the exponent with which the initial scale sets its duration, a clock that analyses of Adam anticipate (Kosson et al., 2024b; Das et al., 2026).

### AI USE STATEMENT

In this work, we used generative AI tools only for language polishing, improving the grammar, wording, and clarity of text drafted by the authors. We have not used generative AI tools for research ideation, for the design of the proposed method or the experimental protocol, for implementation, for the generation or analysis of results, for the creation of figures and tables, or for literature discovery and citation; all of these are the original work of the authors. We have reviewed all AI-assisted work, checking every suggested revision against the authors' original text and the underlying results, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

All analyses read the publicly released Pythia and PolyPythias checkpoints (Biderman et al., 2023; van der Wal et al., 2025), and the measured quantities are defined in Section 2, with the sentences from which expression is read listed in Appendix F. The retrained networks, the transplants, the controlled scaling runs and the lesions are specified in Appendix B, and the floors and the dependence on the sentence set that bound what the measurements resolve are reported in Appendix C. The key numbers in the text are checked against the stored results by a script. We will release the code, the per-unit expression data for every checkpoint and the results of all retrained runs upon publication.

## REFERENCES

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *International Conference on Learning Representations*, 2019.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 2018.

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

Gül Sena Altıntaş, Devin Kwok, Colin Raffel, and David Rolnick. The butterfly effect: Neural network training trajectories are highly sensitive to initial conditions. In *International Conference on Machine Learning*, 2025.

Nora Belrose and Adam Scherlis. Understanding gradient descent through the training Jacobian. *arXiv preprint arXiv:2412.07003*, 2024.

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

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Nils Blomqvist. On the relation between change and initial value. *Journal of the American Statistical Association*, 1977.

Colin Bredenberg, Cristina Savin, and Roozbeh Kiani. Recurrent neural circuits overcome partial inactivation by compensation and re-learning. *Journal of Neuroscience*, 2024.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. Pretrained language model embryology: The birth of ALBERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, 2019.

Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, 2009.

Mohua Das, Pierfrancesco Beneventano, Shibshankar Dey, Gareth H. McKinley, and Tomaso Poggio. Do deep networks forget initialization? A forgetting-time view of practical inductive bias. *arXiv preprint arXiv:2605.29152*, 2026.

Richard Diehl Martinez, Pietro Lesci, and Paula Buttery. Tending towards stability: Convergence challenges in small language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Hans Driesch. Entwicklungsmechanische Studien I. Der Werth der beiden ersten Furchungszellen in der Echinodermenentwicklung. *Zeitschrift für wissenschaftliche Zoologie*, 1892.

EleutherAI. Pythia: README errata. GitHub repository `https://github.com/EleutherAI/pythia`, 2025.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022.

Katie E. Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. In *International Conference on Machine Learning*, 2024.

Finlay Fehlauer, Kyle Mahowald, and Tiago Pimentel. Convergence and divergence of language models under different random seeds. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Emanuele Francazi, Aurelien Lucchi, and Marco Baity-Jesi. Initial guessing bias: How untrained networks favor some classes. In *International Conference on Machine Learning*, 2024.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020.

Jorge Gallego-Feliciano, S. Aaron McClendon, Juan Morinelli, Stavros Zervoudakis, and Antonios Saravanos. Hidden dynamics of massive activations in transformer training. *arXiv preprint arXiv:2508.03616*, 2025.

Xuyang Ge, Wentao Shu, Jiaxing Wu, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. Evolution of concepts in language model pre-training. In *International Conference on Learning Representations*, 2026.

Scott F. Gilbert and Michael J. F. Barresi. *Developmental Biology*. Sinauer Associates, 2016.

Stephen Jay Gould. *Ontogeny and Phylogeny*. Harvard University Press, 1977.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *Transactions on Machine Learning Research*, 2024.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated interpretability metrics do not distinguish trained and random transformers. In *International Conference on Learning Representations*, 2026.

Jesse Hoogland, George Wang, Matthew Farrugia-Roberts, Liam Carroll, Susan Wei, and Daniel Murfet. Loss landscape degeneracy and stagewise development in transformers. *Transactions on Machine Learning Research*, 2025.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. In *Advances in Neural Information Processing Systems*, 2024.

Eamonn Keogh and Jessica Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 2005.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *International Conference on Machine Learning*, 2024a.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Analyzing & reducing the need for learning rate warmup in GPT training. In *Advances in Neural Information Processing Systems*, 2024b.

Nikolaus Kriegeskorte, W. Kyle Simmons, Patrick S. F. Bellgowan, and Chris I. Baker. Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience*, 2009.

Guangqi Li and Yongxin Li. Pre-carved niches: The formation dynamics of modular task partitions in early LLM training. *arXiv preprint arXiv:2609.01170*, 2026.

Siquan Li, Yao Tong, Haonan Wang, and Tianyang Hu. Transformers are born biased: Structural inductive biases at random initialization and their practical consequences. *arXiv preprint arXiv:2602.05927*, 2026.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *International Conference on Learning Representations*, 2016.

Michelle Lo, Fazl Barez, and Shay B. Cohen. Large language models relearn removed concepts. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The Hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Zhengqing Miao and Katharina Dobs. Competitive retraining reveals the resilience and reallocation of functional specialization in deep neural networks. In *Conference on Cognitive Computational Neuroscience*, 2025.

John Newport and Marc Kirschner. A major developmental transition in early Xenopus embryos: I. characterization and timing of cellular changes at the midblastula stage. *Cell*, 1982a.

John Newport and Marc Kirschner. A major developmental transition in early Xenopus embryos: II. control of the onset of transcription. *Cell*, 1982b.

P. D. Oldham. A note on the analysis of repeated measurements of the same subjects. *Journal of Chronic Diseases*, 1962.

David C. Plaut. Relearning after damage in connectionist networks: Toward a theory of rehabilitation. *Brain and Language*, 1996.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, 2017.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models. In *Findings of the Association for Computational Linguistics: EACL 2024*, 2024.

Alexander F. Schier. The maternal-zygotic transition: death and birth of RNAs. *Science*, 2007.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. The MultiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*, 2022.

Elana Simon, Etowah Adams, and James Zou. On the relationship between activation outliers and feature death in sparse autoencoders. In *International Conference on Machine Learning*, 2026.

Zachary Speck and Asa Shepard. Learned, then lost: A measured single-example counterfactual in pre-training. *arXiv preprint arXiv:2608.19168*, 2026.

Cecilia Summers and Michael J. Dinneen. Nondeterminism and instability in neural network optimization. In *International Conference on Machine Learning*, 2021.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling*, 2024.

Wael Tadros and Howard D. Lipshitz. The maternal-to-zygotic transition: a play in two acts. *Development*, 2009.

Ahmed Taha, Abhinav Shrivastava, and Larry S. Davis. Knowledge evolution in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

N. Joseph Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. In *Advances in Neural Information Processing Systems*, 2020.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. Llm circuit analyses are consistent across training and scale. In *Advances in Neural Information Processing Systems*, 2024.

Howe Tissue, Venus Wang, and Lu Wang. Scaling law with learning rate annealing. In *Advances in Neural Information Processing Systems*, 2025.

Yu-Kang Tu and Mark S. Gilthorpe. Revisiting the relation between change and initial value: a review and evaluation. *Statistics in Medicine*, 2007.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. PolyPythias: Stability and outliers across fifty language model pre-training runs. In *International Conference on Learning Representations*, 2025.

George Wang and Daniel Murfet. Patterning: The dual of interpretability. *arXiv preprint arXiv:2601.13548*, 2026.

George Wang, Garrett Baker, Andrew Gordon, and Daniel Murfet. Embryology of a language model. *arXiv preprint arXiv:2508.00331*, 2025.

Yang Xu, Yi Wang, Hengguan Huang, and Hao Wang. Tracking the feature dynamics in LLM training: A mechanistic study. *arXiv preprint arXiv:2412.17626*, 2024.

Greg Yang, James B. Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023.

Mengxia Yu, De Wang, Qi Shan, Colorado J. Reed, and Alvin Wan. The super weight in large language models. *arXiv preprint arXiv:2411.07191*, 2024.

Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron Courville. Fortuitous forgetting in connectionist networks. In *International Conference on Learning Representations*, 2022.

Sally Zhu, Ahmed M. Ahmed, Rohith Kuditipudi, and Percy Liang. Independence tests for language models. In *International Conference on Machine Learning*, 2025.

## A Related work in full

This appendix states, for each result of the paper, what earlier work had already shown and what we add. A result is known when earlier work reports essentially the same thing, an extension when earlier work reports the general phenomenon and we add a unit-level measurement, a mechanism, a quantity or a setting, and new when we found no earlier report. Table 1 collects the verdicts.

### A.1 Reading training as development

Pretraining has been called an embryology before, for ALBERT's checkpoints (Chiang et al., 2020) and for the token-level susceptibilities of a small transformer, whose body plan can then be steered through the data (Wang et al., 2025; Wang & Murfet, 2026). Stagewise development (Hoogland et al., 2025), critical periods (Achille et al., 2019) and a short, decisive early phase (Frankle et al., 2020) are established. To our knowledge a maternal phase, heterochrony and the distinction between regulative and mosaic development have not been applied to neural networks, and the paper uses them as measurements and interventions on single units rather than as a new analogy.

### A.2 The inherited state

Because Pythia's learning rate warms up over the first 1% of training, its first checkpoints are expected to lie close to initialization, and features inherited from initialization have been seen in the same checkpoints (Ge et al., 2026; Xu et al., 2024). Trained networks keep detectable traces of their initialization (Belrose & Scherlis, 2024; Zhu et al., 2025; Li et al., 2026), and a unit's response at initialization is set by its alignment with the mean of its input (Francazi et al., 2024; Simon et al., 2026). We confirm the copied checkpoints at the level of units, and we add the strength of the operating point for the MLP units of a language model, the transplant that moves it, the loosening of its grip over training, and the location of the surviving residue in the first layer. The fate-map artefact of Appendix D is a new instance of a known family, in which structure from an untrained model passes for learned structure (Adebayo et al., 2018; Heap et al., 2026) or a selection criterion leaks into its own test (Kriegeskorte et al., 2009); Li & Li (2026) report structure before learning in Pythia-410M, and we have not tested whether the check applies to it.

### A.3 How the inherited state is lost

That a layer's input changes because earlier layers learn is internal covariate shift (Ioffe & Szegedy, 2015), and the change in a layer's features splits into a weight term and an input term (Yang et al., 2023). The token-shared component of transformer representations is well documented (Biś et al., 2021; Razzhigaev et al., 2024), as are the early MLP blocks that write its largest part (Sun et al., 2024; Yu et al., 2024) and its growth over Pythia's checkpoints (Gallego-Feliciano et al., 2025), and the identity relating random projections to the angle between two vectors is standard (Cho & Saul, 2009). Networks are also known to converge to their final state from the bottom up (Raghu et al., 2017), which concerns arrival at the end of training rather than departure from its start. What we add is that the loss of unit-level memory is carried by the input term, as a rotation of each layer's mean input measured against step 0, that the memory tracks the cosine of that rotation, and that the rotation is written by the MLP and attention blocks below, a decomposition we did not find in earlier work. The causal test of Appendix B.5 failed, so this account rests on decomposition and correlation.

### A.4 What sets the timing

Under Adam, changing the initial magnitude of the weights acts like changing the learning rate, with an exponent of one (Kosson et al., 2024b;a; Everett et al., 2024), and Das et al. (2026) propose accumulated learning rate as the clock by which adaptive optimizers forget their initialization. The direction of our timing results and the role of accumulated learning rate are therefore expected. What we believe is new is the measured exponent of the initial scale, about 1.2 at 14M and 1.4 at 70M, somewhat above the predicted value of one. The larger initialization of Pythia 6.9B and 12B is a documented configuration error (EleutherAI, 2025); what we add is its consequence for when those two models clear, although the released panel cannot separate initialization scale from width.

### A.5 Comparing units across runs

Matching the units of two networks one to one is standard (Li et al., 2016; Tatro et al., 2020), as is comparison with networks grown from other seeds (Gurnee et al., 2024). Execution nondeterminism alone separates runs (Summers & Dinneen, 2021), and early training amplifies any small perturbation whatever its content (Altıntaş et al., 2025; Speck & Shepard, 2026), which our orthogonal control reproduces for single units. What we add is the combination of a same-seed repeat as a floor and a

different-seed twin as a ceiling for units matched on their late expression trajectories; that the floor is high at 70M is consistent with this literature but confounded with a mixture of hardware.

### A.6 Damage and recovery

Driesch's experiment on the sea urchin (Driesch, 1892) founded the distinction between regulative and mosaic development. Networks relearn quickly after damage (Plaut, 1996), re-initializing part of a network during training is followed by recovery (Taha et al., 2021; Zhou et al., 2022), and after retraining, removed functions move to other neurons (Lo et al., 2024) while networks recover with the units that remain (Bredenberg et al., 2024; Miao & Dobs, 2025), unlike the inference-time self-repair of McGrath et al. (2023). Recovery of the loss after our lesion is therefore expected, and we do not claim the slightly lower final loss as an improvement. What we add is a unit-level test in language-model pretraining of whether the repertoire returns and whether replaced units inherit their predecessors' roles; that they do not is expected but not guaranteed, since silenced units that are not redrawn can return to their old roles (Miao & Dobs, 2025). The contrast at 70M has no counterpart we could find, and we report it only as exploratory.

Table 1: **What is known, what is extended and what is new.** Each result of the paper, the status we assign it after reviewing earlier work (Appendix A), and the closest earlier work. Known: reported before in essentially the same form; we confirm it on our readout. Extension: the general phenomenon is known; we add a unit-level measurement, a mechanism, a quantity or a setting. New: no earlier report found.

| Result | Status | Closest earlier work |
|---|---|---|
| *The maternal phase (Section 3)* | | |
| The earliest checkpoints are copies of step 0 at the unit level | Known | warmup schedule; Ge et al., 2026; Fehlauer et al., 2025 |
| Step-0 expression is set by the operating point $w \cdot \mu$; a transplant moves it; its grip loosens over training | Extension | Francazi et al., 2024; Simon et al., 2026 |
| Copied checkpoints produce a spurious fate map; a check against it | Extension | Kriegeskorte et al., 2009; Adebayo et al., 2018; Heap et al., 2026 |
| *The transition (Section 4)* | | |
| Memory clears layer by layer, the deepest first; the first layer never clears | Extension | Raghu et al., 2017; Ge et al., 2026 |
| The weights keep their initial pattern while the mean input rotates | Extension | Ioffe & Szegedy, 2015; Yang et al., 2023; Kosson et al., 2024b |
| Memory tracks the cosine of that rotation | New | the identity itself is standard; Cho & Saul, 2009 |
| The rotation is written by earlier MLP and attention blocks | Extension | Sun et al., 2024; Yu et al., 2024; Gallego-Feliciano et al., 2025 |
| Larger models clear sooner, from 14M to 1.4B | Extension | Diehl Martinez et al., 2024 |
| A larger initialization delays clearance and a larger learning rate hastens it | Known | Kosson et al., 2024b;a; Das et al., 2026 |
| The learning rate acts only through its sum | Known | Das et al., 2026; Tissue et al., 2025 |
| The exponent of the initial scale is about 1.2 at 14M and 1.4 at 70M | New | exponent 1 predicted by Kosson et al., 2024b; Everett et al., 2024 |
| Pythia 6.9B and 12B clear late because of their larger initialization | Extension | error documented by EleutherAI, 2025 |
| *What the transition leaves behind (Section 5)* | | |
| A residue of initialization survives training, mostly in the first layer | Extension | Belrose & Scherlis, 2024; Zhu et al., 2025; Li et al., 2026 |
| A transplant written into the initial state is erased by step 512 | Extension | Fehlauer et al., 2025 |
| Late divergence amplifies any small difference, whatever its content | Known | Summers & Dinneen, 2021; Altıntaş et al., 2025; Speck & Shepard, 2026 |
| The loss recovers after half the MLP units are redrawn | Known | Plaut, 1996; Taha et al., 2021; Zhou et al., 2022 |
| The repertoire returns, replaced units take new roles, and survivors shift partially | Extension | Lo et al., 2024; Bredenberg et al., 2024; Miao & Dobs, 2025 |
| At 70M the loss recovers but the repertoire does not (exploratory) | New | none found |
| *Framing and method* | | |
| Embryological concepts as unit-level measurements and interventions | Extension | Chiang et al., 2020; Wang et al., 2025 |
| Repeat-run floor and different-seed ceiling for units matched on trajectories | Extension | Li et al., 2016; Summers & Dinneen, 2021 |

# B EXPERIMENTAL DETAILS

## B.1 THE ATLAS PANELS AND THEIR CHECKPOINTS

The atlas has two panels. The population panel covers the ten individuals of each size from 14M to 410M, that is the released EleutherAI/pythia-14m, 31m, 70m, 160m and 410m together with their PolyPythias seeds 1–9, and it reads each of them at steps 0, 2, 8, 32, 128, 512, 1,000, 2,000, 4,000, 8,000, 16,000, 32,000, 64,000, 128,000 and 143,000. The scale panel covers one run each of pythia-70m, 160m, 410m, 1b, 1.4b, 6.9b and 12b at twenty-six stages, among them every power of two up to step 512, so that the early steps in which the maternal phase unfolds are sampled densely. The 14M model was read at those same twenty-six stages as well, and it joins the seven models of the scale panel in the alignment test of Appendix E. The released 2.8B model, which falls between 1.4B and 6.9B, is absent from the panel for a reason of data integrity rather than of design. We audited the weight files of every revision we hold, and in the 2.8B repository nineteen of the twenty-three revisions we could audit serve byte-identical weights, so that its early development cannot be read from the release. Where the two panels overlap they agree: the scale-panel runs of 70M, 160M and 410M clear at steps 903, 595 and 436, while the memory curve averaged over the ten individuals of the same sizes clears at steps 885, 628 and 448. Throughout the atlas, expression is computed with the weights held in bfloat16. At step 8 the memory of initialization stands at 0.999, 0.997, 0.990, 0.927 and 0.786 from 14M to 410M, while in the 410M run of the scale panel it has already fallen to 0.35 there (Figure 2a). Expression at step 0 correlates with the operating point $w \cdot \mu$ with a rank correlation 0.91 to 0.95, and the operating points computed with the mean input of text and of random tokens agree at 0.84 to 0.88 (Figure 2b).

## B.2 RETRAINING

Where the atlas can only correlate, we retrain small networks from their released initializations and intervene on them. These runs start from the released step-0 weights of pythia-14m, using seeds 0 to 2, or of pythia-70m, using seeds 0 to 4. They train on consecutive sequences of 2,049 tokens taken in order from the beginning of Pythia's own preshuffled training stream, at 32 sequences per step, with AdamW at a learning rate of $10^{-3}$ unless stated otherwise, $\beta = (0.9, 0.95)$, weight decay of 0.01 on matrices, gradient clipping at 1 and bf16 autocast. Expression is logged in the training loop using Equation equation 1, so that the retrained networks and the atlas are read with the same instrument. The transplant runs of Section 3 last 20,000 steps with a 200-step warmup and cosine decay to 10% of the peak rate; the damage runs of Section 5.2 last 10,000 steps under the same schedule; and the controlled clearance runs of Section 4.2 follow an 8,000-step schedule with a 4,000-step warmup, which the 14M runs complete and the 70M runs stop at step 4,000, the end of warmup. The clamp runs of Section 4.1 retrain 14M from three initializations under the same schedule and also stop at step 4,000. A batch of 32 sequences is far from Pythia's own batch of 1,024, and we therefore regard these networks as a model of the maternal phase rather than as a reproduction of it. The transplant runs and the pilot controlled runs were executed in resumable segments on A40, A100 and H200 GPUs, whereas the later controlled runs, the clamp runs and the damage runs were confined to A100 GPUs. The difference matters, because the repeat runs that we use as a nondeterminism baseline are only as tight as the hardware they ran on.

## B.3 THE TRANSPLANT AND ITS CONTROL

The transplant exchanges the operating points of a random half of the MLP units in every layer at step 0. In every layer that half $T$ and a random permutation of $T$ are drawn with a fixed seed shared between a run and its twin, an unmodified copy trained from the same initialization. The layers are then modified in order, and each layer's mean input is measured again on the partly modified network before that layer's own units are touched, so that the transplant is defined against the network as it actually is. The shift needed to give a unit the original operating point of the unit that the permutation assigns to it is applied either to the bias or to the weight vector along $\mu$, which defines the bias and weight arms of the experiment. Each transplant in the three 14M runs is accompanied by a control that leaves the operating points alone: the same units receive a weight change in a random direction orthogonal to $\mu$, with an average norm of 0.0605 against 0.0610 in the weight transplant and a per-unit correlation of 0.87 between the two, measured at step 0 of the first 14M run. The single 70M run was trained without that control, so wherever the comparison with an orthogonal perturbation matters, the

evidence comes from 14M alone. The correlation between intended and realized operating points is 0.9998. Over steps 2 to 128 the moved units keep a similarity of 0.67 to 0.78 to the same units in the twin, while the units left alone keep a similarity of 0.83 to 0.89 (Figure 2c).

The transplant also carried a test of its own, which asked whether the transplanted units return to their late trajectories, and that test failed. We looked for three signs of such a return over the late window (Appendix B.10): that the similarity of the transplanted units to their twins comes within 0.02 of that of the untouched units, that it reaches 90% of the similarity between repeat runs, and that their distance to the twin is no more than 1.25 times the distance between repeat runs. The first appeared in five of eight run–arm combinations and the other two in none. The nondeterminism baseline of this test was also weak, because the repeat runs were executed in segments on mixed GPU types. The one repeat run that stayed on matched hardware ended at a late distance of 0.0007, whereas the others reached 0.006 to 0.05 at 14M and 0.70 at 70M, a size at which nondeterminism alone moves units nearly as far as the transplant, which reaches 0.93 to 1.15.

### B.4 Separating the weights from the mean input

A forward hook on each MLP input projection records, for every unit, its mean pre-activation over valid tokens and sentences together with its operating point $w \cdot \mu$, where $\mu$ is the mean input of the layer over the same tokens, computed once on the 46 sentences and once on uniform random token identifiers of matched lengths. For the decomposition itself we load the seed-0 checkpoints of 14M, 70M, 160M and 410M in float32 and compute, layer by layer, $W_s\mu_s$, $W_s\mu_0$ and $W_0\mu_s$ (Figure 3c). The identity behind the cosine account of Section 4.1 is the following. If the rows $w_i$ of $W$ are independent draws from an isotropic distribution with covariance $\sigma^2 I$, then across units $\mathrm{Cov}(w_i \cdot a, w_i \cdot b) = \sigma^2 a \cdot b$ and $\mathrm{Var}(w_i \cdot a) = \sigma^2 \|a\|^2$, so that the correlation between $w \cdot a$ and $w \cdot b$ is the cosine between $a$ and $b$. A layer whose mean input has turned by a given angle should therefore retain, on this account, a memory equal to the cosine of that angle. For the initial weights the identity holds almost by construction, and the measured correlations of $W_0\mu_0$ with $W_0\mu_s$ do follow the cosine closely, within 0.01 to 0.03 on average and 0.07 at most over all layers of 14M, 70M, 160M and 410M at steps 32 to 1,000. We report this only as a check on the arithmetic.

### B.5 The mean-input clamp

The clamp asks whether holding each layer's mean input fixed during training keeps the memory of initialization. It uses a forward pre-hook on every MLP input projection that replaces the token-mean $m$ of the current batch, counting valid tokens only where an attention mask is given, by the mean $\mu_0$ measured on the 46 sentences at step 0. Two versions were run, one that holds the direction and the length, $x \mapsto x - m + \mu_0$, and one that holds the length alone, $x \mapsto x - m + m\,\|\mu_0\|/\|m\|$. Both are identities on the atlas sentences at step 0, where the recomputed expression matches the atlas at $r = 0.99999$. Applied at measurement alone to an ordinary run, the same hook reads each unit against the step-0 mean input without altering training, which is how the recovered memory of Section 4.1 is obtained.

### B.6 The warmup-accumulation account

The warmup-accumulation account of Section 4.2 needs the width, depth and peak learning rate of each released configuration. In order from 14M to 12B, the widths are 128, 256, 512, 768, 1024, 2048, 2048, 4096 and 5120; the depths are 6, 6, 6, 12, 24, 16, 24, 32 and 36; and the peak learning rates are $10^{-3}$, $10^{-3}$, $10^{-3}$, $6 \times 10^{-4}$, $3 \times 10^{-4}$, $2.5 \times 10^{-4}$, $2 \times 10^{-4}$, $1.2 \times 10^{-4}$ and $1.2 \times 10^{-4}$. Written out, with peak learning rate $\eta$, width $d$, depth $L$ and a linear warmup of $s_{\mathrm{w}}$ steps, the accumulated drift after $s$ steps is $C(s) = \eta d L\, s^2/(2s_{\mathrm{w}})$ for $s \le s_{\mathrm{w}}$ and $\eta d L\,(s_{\mathrm{w}}/2 + s - s_{\mathrm{w}})$ afterwards. We fitted a fixed drift threshold as the median of $C$ at the observed clearance steps of the five population sizes and predicted from it the clearance steps of the scale panel: 542, 495, 391 and 330 steps for 1B, 1.4B, 6.9B and 12B. The observed steps were 531, 526, 845 and 805.

### B.7 Reading the initial weights

We read the initial weights of the released checkpoints by HTTP range requests into the first, middle and last blocks of each step-0 checkpoint. Every model up to 1.4B uses a width-scaled initialization

for all of its input-side matrices and embeddings, with standard deviation $\sqrt{2/(5d)}$, which is 0.01397 at 1B. The two largest models instead draw them from a normal distribution with standard deviation 0.0200, which is 2.02 and 2.26 times the width-scaled value for their widths, and their output projections, drawn with standard deviations of 0.00250 and 0.00236 for 6.9B and 12B, are 2.6 and 3.0 times larger than the width-and-depth-scaled value $2/(L\sqrt{d})$ used at the other sizes.

### B.8 THE CONTROLLED SCALING RUNS

Scaling the initialization of a controlled run means scaling the attention input and output projections and both MLP matrices of every block, but not the embeddings; scaling the learning rate scales the peak of the schedule and leaves the 4,000-step warmup and the cosine decay in place. The controlled experiments were run in two rounds. In a pilot, retrained 14M networks from a single initialization had all of their block weight matrices scaled at initialization by 0.5 or 2 or else had the learning rate scaled by 0.5 or 2, under a 4,000-step warmup. Clearance came at step 2,789 in the unmodified network, at 1,979 and 4,639 with half and twice the initial scale, and at 4,089 and 1,890 with half and twice the learning rate. The fitted exponents, $+0.61$ for the initialization scale and $-0.56$ for the learning rate, are close to the $+1/2$ and $-1/2$ that follow if clearance requires the accumulated update to reach a fixed multiple of the initial scale while that update still grows with the square of the step inside warmup. They were not stable, however: at thresholds of 0.15, 0.20 and 0.25 the first exponent became $+0.49$, $+0.78$ and $+1.16$, which is a reminder that a threshold crossing is a fragile summary of a curve that flattens as it approaches the threshold. The second round repeated the five conditions from three initializations each, all trained on a single GPU type.

In the second round the five 14M curves differed on the raw step axis by a mean variance across conditions of 0.0048, and with $k = 2$ the spread on the rescaled axis was 43% of that. The best exponent was 1.30, 1.00 and 1.35 for the three initializations, and the spread is flat near its minimum, staying within 25% of it for $k$ between 0.90 and 1.50. A bootstrap interval over the three initializations, 1.00 to 1.35, contains 1 and excludes 0, but with three initializations that interval is essentially their range, and a different percentile convention would place its lower edge at 1.06, so we do not claim that the exponent is exactly 1. Inside warmup an exponent of 1.2 makes the clearance step grow roughly as the 0.6 power of the initialization scale and the $-0.5$ power of the learning rate, which is what the pilot found. Threshold crossings remained fragile in this round as well: at 0.10 only one of the three initializations crossed in all five conditions within 8,000 steps, and at 0.20 the exponent for the initialization scale ranged from $+0.57$ to $+0.80$ across initializations.

The 70M experiment started from five released initializations. Its best exponent, 1.40, is 0.20 above the 14M value, and its bootstrap interval, 1.35 to 1.45, with per-initialization optima from 1.30 to 1.50, excludes 1; with $k$ fixed at 1, however, the 70M curves still collapse to 17.4% of the raw spread, as comfortably as at 14M, where the corresponding figure is 18.4%, and the spread stays within 25% of its minimum for $k$ between 1.10 and 1.70. Threshold exponents are steadier at 70M than at 14M, ranging from $+0.69$ to $+0.84$ for the initialization scale and from $-0.71$ to $-0.54$ for the learning rate over five initializations, which is what an exponent of 1.4 implies for a crossing inside warmup. Whether the power is universal we cannot say. The per-initialization optima are 1.00 to 1.35 at 14M and 1.30 to 1.50 at 70M, a difference of 0.17 in the mean, which an exact permutation test over the eight values makes $p = 0.09$ one-sided; if the difference is real, it points in the direction that 6.9B and 12B suggest, namely a stronger dependence at larger sizes.

### B.9 THE DAMAGE RUNS

At step 1,000 a random half of the MLP units in every layer is replaced, and which units are replaced is fixed by a plan seed shared between a run and its twin. Each replaced unit has its input row redrawn from a normal distribution with the standard deviation that the matrix had at step 0, its bias set to zero, and its column of the MLP output projection redrawn in the same way, and the Adam moments of all those entries are cleared. At 14M the training loss had fallen from 9.7 at step 50 to 5.3 when the lesion was applied. At the end of training the damaged networks sat below their twins by $-0.0061$, $-0.0006$ and $-0.0024$ nats in the three initializations, where repeat runs differ by $+0.0015$, $+0.0001$ and $-0.0002$. At 70M the loss recovered faster than at 14M, reaching within 0.05 nats of the twin in 54 to 67 steps, although two 70M runs that differ only through nondeterminism already differ by as much as 0.07 nats over those steps, so that time lies inside the noise. At 70M repeat runs are

0.022 to 0.034 apart after matching, and the replaced units have not fallen silent, since their median expression at step 10,000 is 0.119 to 0.122 against 0.125 to 0.131 in the twins. Read against each size's unrelated-unit level, at 14M the replaced units end at 0.82 to 0.87 of it, and the units left alone move to 0.17 of the unrelated level against a floor of 0.0006 to 0.002. At 70M the replaced units end at 1.7 to 2.4 times the unrelated level, while the units left alone sit at 0.28 to 0.37 of that level against a floor of 0.34 to 0.42. Whether this reversal reflects a real dependence on size, or a limit of the measure at 70M, where repeat runs alone differ by 0.49 to 0.69 on the same-unit distance, three runs cannot say.

### B.10 Distances, matching and baselines

For a unit $j$ and two runs, $D_j$ is the mean over the stages of a window of the squared difference of log expression. Similarity between two runs is the correlation, across the stages of a window, between a unit's standardized expression in the two; the early window, used in Section 3, consists of steps 2, 8, 32 and 128, and the late window of the transplant runs, used for the transplant's own test (Appendix B.3), runs from step 1,000 to step 20,000. After the lesion we compare repertoires over the late window of the damage runs, steps 2,000 to 10,000, and the matched distance between two networks is the mean $D$ after the units of each layer are paired one to one by the Hungarian algorithm on the matrix of $D$ between the two runs, rather than by giving each unit its nearest neighbour, so that no unit of the twin can be used twice. Three baselines accompany the comparison: a repeat run, a twin grown from a different initialization, and, inside a single network, each unit's nearest other unit of the same layer. The fraction of the distance to the unrelated-unit level that the untouched units have covered is $\rho = (D_U(\text{damaged}) - D_U(\text{repeat}))/(D_{\text{null}} - D_U(\text{repeat}))$, where $D_U$ averages $D_j$ over the units that were left alone and $D_{\text{null}}$ is the same average with each unit of the twin paired with a random unit of the same layer.

The transplant distances of Section 5.1 are read against unrelated units of the same layer and run, which are 0.47 to 0.48 apart at 14M and 1.42 apart at 70M, and against twins grown from different initializations, which are 0.47 to 0.49 apart and exist at 14M only; on the square root of the distance the orthogonal control starts fourteen times closer than the transplant rather than two hundred. After the lesion, the untouched units end at a distance of 0.070 to 0.072 from where they would otherwise have been. On the mean squared difference of log expression that movement is 92 to 295 times the drift between repeat runs, and on its square root it is 10 to 17 times the repeat runs and 38 to 40% of the way to the unrelated-unit level. Read on either scale the movement is partial: the untouched units end well short of the unrelated-unit level on the squared scale and still short of halfway on its square root.

## C What the instrument resolves, and what it depends on

Two floors fix what these measurements are able to resolve at 14M, the size at which most of our interventions are made, and both are needed to interpret the distances between networks in Section 5. Both are measured on the matched distance of Appendix B.10, over the late window of the damage runs, all of which were executed on one GPU type (Appendix B.2). Two 14M runs that differ only through the nondeterminism of execution end 0.0002 to 0.0008 apart, and a unit and its nearest neighbour inside a single 14M network are 0.0044 to 0.0052 apart. A difference smaller than the first of these is not a difference between networks at all, and a difference smaller than the second does not separate a unit from its neighbours within one network. Neither figure is a constant of the method. At 70M, and in runs that were executed in segments on mixed hardware, repeat runs drift considerably further (Appendices B.3 and B.9), and each comparison is read against the floor that belongs to it.

A further property concerns the instrument rather than the networks, namely how much the atlas depends on the particular sentences it is read with. The sentences are short, topic-diverse English sentences, listed in Appendix F. Because each unit's expression is recorded on every sentence separately, the whole atlas can be read again with only half of the sentences, and the two halves can then be compared. Over two hundred random splits of the 46 sentences, the clearance step computed from 23 sentences agrees with the full-set value to within 4.0% in every one of the fifty runs, and the spread that the choice of sentences induces is smaller at every size than the spread between the ten runs of that size, so which sentences one reads with matters less than which network one reads (Figure 8a). The two halves agree closely at initialization and late in training, and least at the

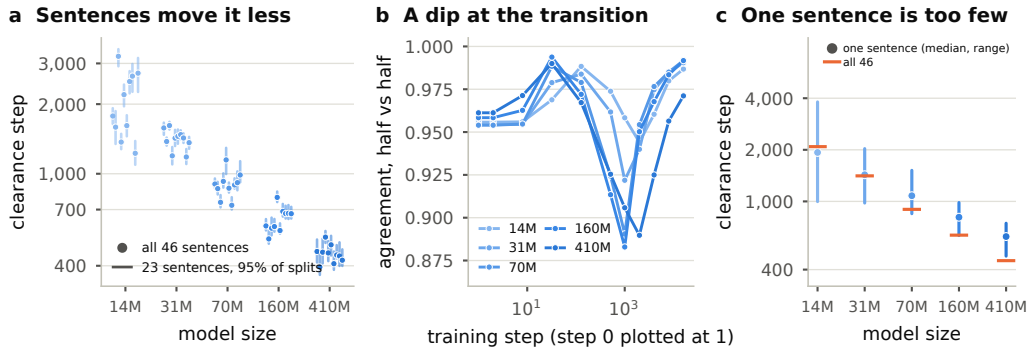


Figure 8: **The atlas depends less on its sentences than on the network it reads, although half-splits agree least at the transition and a single sentence is not enough.** (a) Clearance step of every run at each size, read with all 46 sentences (points) and with 23 sentences over two hundred random half-splits (bars, the central 95% of splits). (b) Correlation across units between the expression read from the two halves of a split, averaged over ten runs; step 0 is drawn at 1. (c) Clearance step read from a single sentence, as the median and range over the 46 sentences, against the value read from all of them (orange).

transition itself, where their agreement falls to between 0.88 and 0.94 around each size's clearance step before recovering (Figure 8b), so the moment at which the inherited pattern is being replaced is also the moment at which the instrument reads it least reliably. A single sentence is not enough for any of this: in a 14M network the clearance step obtained from one sentence ranges over a factor of 3.6, the median over the ten runs (Figure 8c). We should be precise about what this control establishes and what it does not. Only three sentence counts were compared, namely 46, 23 and one, so the halves show that the answer does not depend on which half it is read with, rather than identifying the number of sentences at which the measure would begin to drift. Two complementary halves of one set of sentences are also not an independent corpus, since they share genre and length, so the control bounds the sampling of sentences and says nothing about a change of domain.

## D A FAILED PRELIMINARY ANALYSIS: THE FATE MAP AND ITS ARTEFACT

This appendix reports a preliminary analysis that failed. Early in this project we built a fate map of the atlas, took it for a developmental result, and only later found it to be an artefact of the copied checkpoints; we report it because the way it failed is instructive for anyone who reads training developmentally. Developmental biologists build fate maps by marking cells early and asking what those cells become. The analogous analysis for a network defines each unit's fate from the shape of its expression trajectory and asks whether anything present at initialization predicts that fate. We took each unit's expression over the stages of the population panel from a chosen starting step onwards, standardized it across those stages so that only its shape and not its level could count, and clustered the units of each run into $K = 8$ classes by $k$-means. How much of the unit's step-0 expression the class explains is measured by $\eta^2$, the share of its variance that lies between classes, whose chance level for $n$ units is $(K-1)/(n-1)$, and its significance by a test that shuffles the class labels 500 times. Leaving step 0 itself out of the trajectory appears to be enough to keep the predictor out of the fate. It is not. With trajectories that start at step 2, class membership explains between a fifth and a third of the variance of step-0 expression, rising from 0.21 at 14M to 0.35 at 410M (Figure 9a). The effect reaches the smallest $p$-value the test can give in every run, it is of similar size with four or sixteen classes instead of eight, and it follows transplanted operating points. We first took it for a developmental determinant.

The explanation is that step 2, and at the smaller sizes step 8 as well, are copies of step 0. Standardizing a trajectory measures each stage against the unit's own mean, so a unit whose initial expression is high begins high whatever training does afterwards, and the initial value therefore enters the class label through the copied stages. This is the coupling between change and initial value that biostatisticians described decades ago (Oldham, 1962; Blomqvist, 1977; Tu & Gilthorpe, 2007), in a form familiar

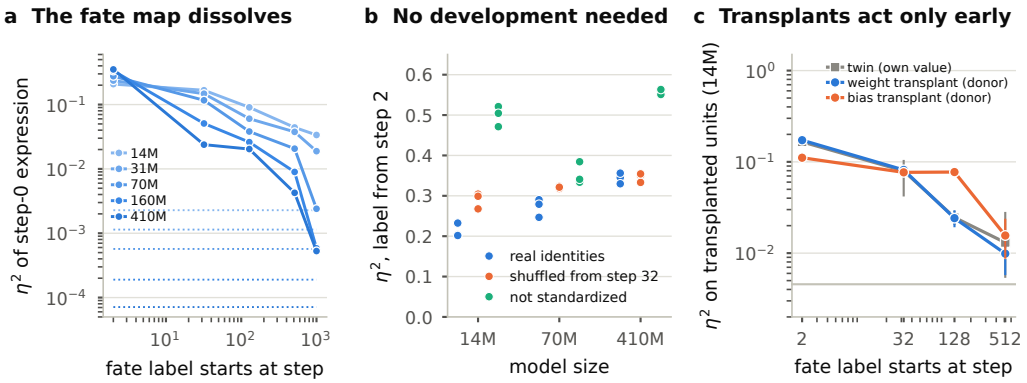


Figure 9: **A fate map read from trajectories that include the copied checkpoints is predicted by initialization, and both it and the transplant's effect on it vanish once those checkpoints are left out.** Units are clustered into eight classes by the shape of their standardized expression trajectory from a given starting step, and $\eta^2$ is the share of the variance of step-0 expression explained by the classes. (a) $\eta^2$ against the starting step, averaged over ten runs of each size; dotted lines mark chance. (b) $\eta^2$ for trajectories starting at step 2, with the real unit identities, with identities shuffled within layers from step 32 on, and without standardization, for three runs each of 14M, 70M and 410M. (c) $\eta^2$ of the operating point that transplanted units received, or in the twin of their own, in retrained 14M networks; mean and range over three runs, with chance in grey.

from circular analysis in neuroscience and from time-series clustering (Kriegeskorte et al., 2009; Keogh & Lin, 2005). What is specific to networks is that the copied stages are produced by warmup and are invisible unless one looks for them. Two controls make the point directly (Figure 9b). When unit identities are shuffled within each layer from step 32 onwards, so that no unit's later development is connected to its own start, class membership explains as much of step-0 expression as it does with the real trajectories, and at 14M more. When the trajectories are left unstandardized, it explains more still. The signal needs no development at all.

Moving the start of the trajectory past the copied stages reduces the effect by one to two orders of magnitude by step 1,000 (Figure 9a). Starting after each size's clearance, at step 4,000 or 8,000, class membership still explains at most 0.006 of step-0 expression in three runs of each size, which is up to ten times a permutation null, with the excess significant at $p < 0.05$ in 14 of 30 cases and in all six cases at 410M (Figure 6b). The transplant, which had seemed to confirm the fate map, fares no better under the same treatment (Figure 9c). In the three retrained 14M runs, the operating point that a unit received explains between 0.165 and 0.177 of its class when trajectories start at step 2, but only between 0.006 and 0.017 when they start at step 512, and a unit's own operating point in the untransplanted twins falls in the same way. Moving a unit's operating point therefore changes its maternal phase, which the copied stages record, and not the classes that the unit joins afterwards. The practical lesson is a small one and easy to apply. Before interpreting early developmental structure, one should measure how similar each checkpoint still is to initialization, and then show that the conclusion survives when those checkpoints are removed. In Pythia this matters more than it might seem, because the dense early checkpoints at steps 1, 2, 4 and onwards to 512 all fall inside warmup.

## E What the released panel can settle about the timing, and how robust the clearance steps are

How much the released panel could settle about the role of the initialization scale can itself be answered with a measurement. For each model of the scale panel, together with 14M read at the same stages, we locate the step at which its memory first falls through each of six levels between 0.8 and 0.1 and take the logarithm of that step as the model's position at that level (Figure 10a). A model's offset is its position minus the mean position of all models at the same level, averaged over levels, and the misalignment of the panel is the variance of the positions across models, averaged

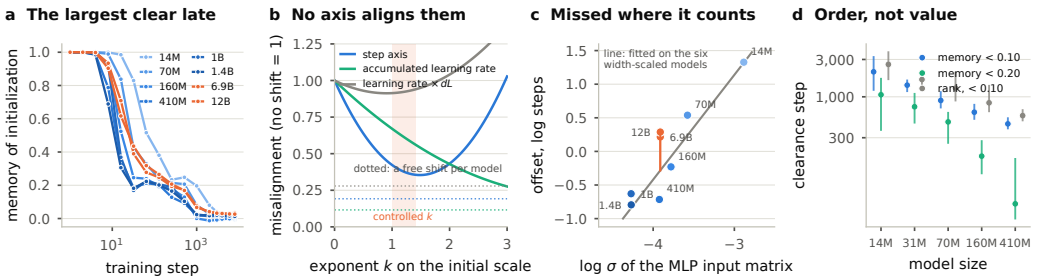


Figure 10: **The released panel cannot fix how strongly the initialization scale matters, although the order of clearance across sizes is robust.** (a) Memory of initialization for 14M and the seven models of the scale panel; orange marks 6.9B and 12B, whose input-side initialization is about twice the width-scaled value. (b) Misalignment of the models' memory curves after each is shifted by its initial scale to the power $k$, relative to no shift, on three axes; dotted lines mark what a free shift per model removes, and the band marks the exponent found in the controlled runs. (c) The shift each model needs on the step axis against the logarithm of its initialization scale; the line is fitted on the six width-scaled models, and orange marks how far the two largest are missed. 1B and 1.4B, like 6.9B and 12B, share a scale. (d) Clearance step of the population panel at thresholds of 0.10 and 0.20 and with a rank correlation, as mean and range over ten runs.

over levels. Dividing the step axis by the initial scale raised to a power $k$ moves every position of a model by the same amount, $k$ times the logarithm of its initial scale, so the question is whether shifts of this form bring the curves together nearly as well as a free shift for each model does (Figure 10b). They do not, and the panel cannot tell how strongly the initial scale matters, for three reasons. First, among the six models that use width scaling the initial scale is a function of the width, the two being collinear at $-1.000$ by construction, so no fit can separate them. Second, the only models whose initialization breaks that rule are 6.9B and 12B, and a line fitted to the offsets of the six smaller models against the logarithm of their initial scale predicts the two largest worse, missing them by 0.50 and 0.57 in log steps, than simply assigning them the mean offset of the six, which misses them by 0.30 and 0.37 (Figure 10c). Third, the exponent that best aligns the panel depends on the axis it is read on: it is 1.48 on the step axis, close to the value of the controlled experiments, and 4.24 once the accumulated learning rate is divided out. What the panel does show clearly is how far the drift account of Section 4.2 stood from the data, since on that account's own axis, the learning rate times the width times the depth, no shift proportional to the initial scale removes even a tenth of the misalignment.

The two panels also show how robust the clearance steps themselves are. Here we use the mean of the ten runs' own clearance steps at each size, which sits slightly later than the crossing of their mean memory curve used in Section 4.2. It falls from 2,087 at 14M to 451 at 410M, and the order of the sizes survives a threshold of 0.20 and a rank correlation, which places every size later, at 2,594 at 14M and 575 at 410M (Figure 10d). Neighbouring sizes overlap, however, since the ten runs of 14M range from 1,226 to 3,221 and those of 410M from 395 to 532. The delay of the two largest models is far less robust. At the 0.10 threshold they clear 1.6 times later than 1B and 1.4B, but at the 0.20 threshold eight to ten times later. The curves show why (Figure 10a): 1B and 1.4B rest on a plateau between about 0.1 and 0.2 from step 32 to step 512, where a small change of threshold moves the crossing by hundreds of steps, whereas the two largest models are still at 0.39 and 0.44 at step 32 and reach that band only by step 512, at 0.17 or below, and every crossing there is placed by interpolating across the long gap between checkpoints 512 and 1,000. The direction of the effect holds at every threshold; its size does not.

## F THE SENTENCES FROM WHICH EXPRESSION IS READ

Every checkpoint of the atlas and every retrained network is read with the 46 sentences listed in Table 2, and a unit's expression is its mean absolute activation over their valid tokens, as defined in

Equation equation 1. The sentences are short, topic-diverse English sentences. Appendix C shows that the atlas does not depend on which half of them it is read with.

Table 2: **The 46 sentences from which expression is read.** Every checkpoint of the atlas and every retrained network is read with this set.

| | | | |
|---|---|---|---|
| 1 | The river flowed quietly through the valley before reaching the sea. | 24 | Engineers tested the new alloy under extreme pressure. |
| 2 | She adjusted the telescope and waited for the clouds to clear. | 25 | The library extended its hours during final exam week. |
| 3 | Interest rates were raised again to slow down inflation. | 26 | Snow began to fall just as the parade started. |
| 4 | The committee will meet on Thursday to review the proposal. | 27 | The company announced layoffs affecting several departments. |
| 5 | A thin layer of frost covered the windshield this morning. | 28 | A small boat drifted near the edge of the harbor. |
| 6 | The recipe calls for two cups of flour and a pinch of salt. | 29 | The professor explained the theorem using a simple diagram. |
| 7 | Researchers published new findings on coral reef degradation. | 30 | Tourists gathered to watch the sunset over the cliffs. |
| 8 | He tightened the last bolt and stepped back from the engine. | 31 | The factory switched to a four-day work week last month. |
| 9 | The museum's new exhibit features pottery from the Bronze Age. | 32 | Her essay argued that the policy had unintended consequences. |
| 10 | Traffic was unusually light on the highway this afternoon. | 33 | The garden was overrun with weeds after the long rain. |
| 11 | The novel's protagonist struggles with memory loss after the accident. | 34 | Pilots reported turbulence over the mountain range. |
| 12 | Solar panels were installed on the roof last spring. | 35 | The city council debated the new zoning regulations for hours. |
| 13 | The orchestra rehearsed the symphony's second movement twice. | 36 | A stray cat wandered into the bakery and fell asleep by the oven. |
| 14 | Local farmers reported a smaller wheat harvest this year. | 37 | The satellite entered orbit precisely on schedule. |
| 15 | The judge postponed the hearing until further notice. | 38 | Workers paved the old dirt road leading to the farmhouse. |
| 16 | A gentle breeze carried the smell of rain across the field. | 39 | The negotiations stalled over disagreements about tariffs. |
| 17 | The startup raised a new round of funding last week. | 40 | Fireflies lit up the backyard as the evening grew dark. |
| 18 | Children lined up outside the classroom before the bell rang. | 41 | The surgeon reviewed the scans before the operation. |
| 19 | The bridge was closed for repairs after the inspection. | 42 | A new coffee shop opened on the corner of Fifth Street. |
| 20 | Astronomers detected a faint signal from a distant galaxy. | 43 | The glacier has retreated significantly over the past decade. |
| 21 | The chef garnished the plate with a sprig of mint. | 44 | The choir practiced the hymn one final time before the service. |
| 22 | Voter turnout was higher than expected in the coastal districts. | 45 | Investors reacted cautiously to the earnings report. |
| 23 | The dog chased its tail in circles on the porch. | 46 | The hikers followed the trail markers back to the parking lot. |