TWO TIMESCALES OF FORGETTING IN FINE-TUNING OF LANGUAGE MODELS: FAST BEHAVIORAL SUPPRESSION AND SLOW REPRESENTATIONAL DRIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Today, a core pipeline of artificial intelligence systems is pretraining on a broad range of text corpora and then fine-tuning on specific tasks or desired behaviors. Yet we still have limited understanding of how fine-tuning on a restricted data distribution modifies representations acquired during pretraining. In particular, it remains unclear whether surface behavioral suppression happens together with the erasure of latent representations that support it. In this work, we study supervised fine-tuning (SFT) by pretraining transformers on a controlled context-free grammar with hierarchical latent structure, the Random Hierarchy Model (RHM), and fine-tuning with a restricted subset of its generative rules. While during pretraining the model's ability to generate with the correct grammar rules coincides with building the internal representations of the corresponding latent, we find that SFT exhibits a very different dynamics: (i) the generation of deleted rules changes on a fast time scale inversely proportional to the magnitude of the token-latent correlation; meanwhile, their corresponding representations stay intact; (ii) the hidden representations of the deleted rules become forgotten on a much longer time scale, associated with a slow drift of the network parameters. Finally, we observe analogous results in a realistic setup with natural language, suggesting that this timescale separation in forgetting is a more general phenomenon of fine-tuning language models.

1 INTRODUCTION

Large language models (LLMs) are pretrained on broad text corpora and then adapted through supervised fine-tuning (SFT) for downstream tasks, instruction following, or safety alignment. Fine-tuning typically uses a narrower data distribution and can impair capabilities acquired during pretraining, a phenomenon known as *forgetting* (McCloskey & Cohen, 1989; Ramasesh et al., 2022; Shi et al., 2025). Yet a drop in an evaluation metric of the model based on its behavioral outputs does not imply that the information supporting it has been erased from its internal representations (Davari et al., 2022; Zheng et al., 2025; Kotha et al., 2024; Lynch et al., 2024). Understanding this distinction is essential for determining what fine-tuning changes and what evaluations of model behavior can establish.

Despite these observations, a quantitative account of the relationship between behavioral and representational forgetting remains limited. How quickly does fine-tuning suppress previously learned behavior? How long do the underlying representations persist, and what determines their eventual change? Addressing these questions requires a setting in which the relevant latent structure is known and its role in generation can be distinguished from its use in interpreting context.

We study these questions using the Random Hierarchy Model (RHM), a probabilistic context-free grammar with a known hierarchy of latent variables (Cagnetta et al., 2024). We pretrain decoder-only transformers by next-token prediction and then fine-tune them on sequences generated after deleting a subset of the grammar's production rules. This restriction changes which sequences the model should generate without introducing new rules to learn. The known generative structure lets us separately track the use of deleted rules in model outputs and the ability to use deleted rules in context to predict allowed continuations.

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

Our central finding is a separation between two timescales of forgetting. Generation using deleted rules is rapidly suppressed, while the model retains representations that allow it to interpret those rules in context and use them for prediction. This representational and predictive ability deteriorates only after much longer fine-tuning. Our analysis supports an explanation in which early behavioral adaptation occurs in a lazy regime, with little change to learned features, whereas continued noisy parameter drift gradually alters the representations that support the suppressed behavior.

Our contributions

- We characterize rapid behavioral suppression across the grammar's hierarchical levels. The suppression timescales increase with the rules' hierarchical level and the retained grammar size, while representations of the corresponding latent features remain largely preserved.
- We identify a much slower timescale for representational forgetting, accompanied by a loss of the ability to use deleted rules in context for prediction.
- We analyze parameter motion during fine-tuning and find evidence consistent with early lazy adaptation followed by noisy drift. Weight decay selectively accelerates representational forgetting while leaving the initial suppression of generation unchanged.
- We show that the first fine-tuning step for a minimal linear model is proportional to the difference in token-latent correlations between the fine-tuning and the pretraining grammar. We observe that the inverse of this signal magnitude is proportional to the timescales of hierarchical behavioral suppression.
- We observe an analogous timescale separation between behavioral and representational forgetting in a transformer pretrained on bilingual English–French text and fine-tuned on one language, extending the phenomenon beyond the synthetic grammar setting.

These findings show why evaluating forgetting requires examining both generated outputs and latent representations of previously learned structure. They also motivate a distinction relevant to safety fine-tuning: suppressing an unwanted behavior alone does not ensure that the representations supporting it have been removed, and the two phenomena occur through distinct dynamics.

2 Related work

Continual learning. A bulk of continual learning literature studied *catastrophic forgetting*, where the performance of neural networks deteriorates as new tasks are learned (Goodfellow et al., 2014; Lee et al., 2021; Ramasesh et al., 2021; Braun et al., 2022). It has also been shown that large weights after the initial learning period can be one of the causes for a loss of plasticity (Dohare et al., 2024; Nikishin et al., 2022; Lyle et al., 2023). Yet a decrease in task accuracy does not imply a comparable loss of information from hidden representations, and substantial retrieval of the information is plausible even after the model seems to forget in its output (Davari et al., 2022; Lynch et al., 2024; Kotha et al., 2024). In fact, selective-unlearning research has distinguished suppression of observable behavior from removal of information retained in network weights (Golatkar et al., 2020a;b). Malladi et al. (2023) characterize fine-tuning of pretrained language models with the Neural Tangent Kernel (NTK). We similarly observe that the faster timescale of behavioral change is supported by lazy dynamics. Blanc et al. (2020); Li et al. (2022) show that noisy gradient updates near the minimizer manifold drive a slow drift, consistent with noise-induced drift of hidden representations (Pashakhanloo & Koulakov, 2023). Our work connects these dynamics to forgetting in restricted fine-tuning, where such drift erodes the representation of deleted rules after their generation has been suppressed.

Learnability of the Random Hierarchy Model. The RHM is a synthetic generative grammar whose hierarchical latent structure allows the study of representation learning and its sample complexity (Cagnetta et al., 2024). For autoregressive models, Cagnetta & Wyart (2024); Cagnetta et al. (2025) relate the stepwise acquisition of successive rule levels in RHM to the amount of data needed to resolve correlations over longer contexts. Favero et al. (2025) study the acquisition of hierarchical compositional structure in training diffusion models, finding an analogous result. These works describe how hierarchical structure is acquired from randomly initialized networks and its corre-

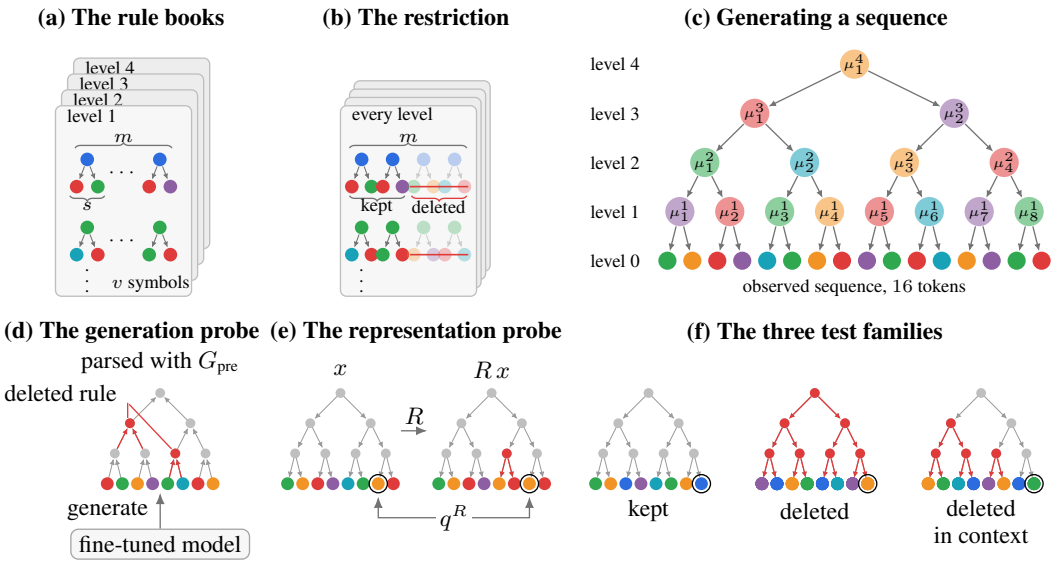


Figure 1: The Random Hierarchy Model and our setup. **(a)** The rule books, one per level, prescribe how each symbol can be expanded in m lower-level synonymic representations. They define the pretraining grammar G_{pre}. **(b)** The restriction that defines the grammar used in fine-tuning, G_{ft}: for every symbol on every level, only m_{keep} rules stay valid while the others get deleted. **(c)** An example of a sequence with its latent tree structure generated by an RHM with depth $L = 4$ and branching factor $s = 2$. **(d)** to **(f)** The schematic diagram of the generation probe, the representation probe, and how the sequences in the three test families are generated to measure prediction loss. The gray nodes use the kept rules to generate their children, while the red nodes use the deleted rules.

sponding sample complexities. In this work, we focus on how the previously learned hierarchical latent structure changes throughout the fine-tuning with restricted data distribution.

3 Random Hierarchy Model

3.1 Pretraining and fine-tuning setup

Data model. The Random Hierarchy Model (RHM) is a probabilistic context-free grammar that generates sequences through a hierarchy of production rules (Cagnetta et al., 2024). A sequence is obtained by starting from a root node and expanding a regular tree from it, with constant branching factor s. The generation proceeds for L levels, until obtaining $s^L \equiv D$ leaf nodes which correspond to the observed tokens. We call this observable level $\ell = 0$, while the levels $\ell = 1, \dots, L$ are unobserved latent levels (Figure 1(a) and (c)). Each node carries one of v symbols. One instance of the RHM corresponds to drawing at each level vm distinct ordered s-tuples uniformly without replacement from the v^s possible tuples and assigning m to each parent symbol. These alternative expansions are called synonyms. The resulting rule book is shared by all nodes at the same level. No s-tuple is assigned to two different parent symbols, so each valid sequence has a unique latent tree. Once the rule books are drawn, they define a grammar G and remain fixed. The grammar generates data by drawing one root symbol uniformly and iteratively sampling one of the synonymic lower-level representations with equal probability. The number of grammatical sequences is $vm^{\frac{D-1}{s-1}}$, therefore exponential in the sequence length D.

We write μ_i^ℓ for the ith node from the left at level ℓ. Negative indices count from the right at the same level: μ_{-1}^ℓ is the rightmost node and μ_{-2}^ℓ the second from the right. For example, $\mu_{-2}^2 = \mu_3^2$ in the $L = 4$, $s = 2$ tree. The grammar generates the training sequences. The leaf tokens are one-hot encoded in the possible v symbols and used to train a transformer.

Pretraining. For a fixed grammar G_{pre}, we train decoder-only transformers (Karpathy, 2022) by next-token prediction. Each batch contains 128 freshly sampled sequences, corresponding therefore to an online learning regime. Training minimizes the mean cross-entropy over the sequence positions with a successor, using Adam (Kingma & Ba, 2015) at a constant learning rate of 10^{-4}. Architecture, training durations and other settings are given in Appendix A.1.

In our experiments, we use $s = 2$ and vary combinations of other values like $L = 4$, $v = 24$, $m \in \{6, 8, 10\}$ and $L = 3$, $v = 16$, $m = 8$.

Restricted fine-tuning. For the fine-tuning data, we construct G_{ft} by retaining $m_{\text{keep}} < m$ synonymic representations of every symbol in every rule book and deleting the other rules from G_{pre} (Figure 1(b)). The retained rule subset is fixed, and the resulting sequences are generated by sampling from the retained synonyms uniformly. Thus, the fine-tuning sequences from G_{ft} form a narrower distribution of G_{pre}, while each node still takes one of the possible v values (although its marginal distribution is modified by G_{ft}). Training uses no explicit penalty on deleted rules and no replay of G_{pre} data. Starting from the final pretrained weights, we continue next-token prediction on freshly sampled G_{ft} sequences, with the same optimizer settings and reset Adam moments.

3.2 Metrics for forgetting

Prediction loss. We measure the last-token test cross-entropy $\mathcal{L}_{\text{last}}$, conditioning on all preceding tokens. For pretraining, we compare it with the Bayes-optimal losses $\mathcal{L}_k$, $k = 0, \cdots, L$, obtained when only the previous $s^k - 1$ context tokens are available. These references are computed by belief propagation on G_{pre} (Cagnetta et al., 2025; Sclocchi et al., 2025); $\mathcal{L}_0$ uses no context and $\mathcal{L}_L$ uses the full sequence context.

For fine-tuning, we evaluate $\mathcal{L}_{\text{last}}$ on three fixed test families, all valid under G_{pre} (Figure 1(f)). In the *kept-grammar* family, every internal node uses a kept rule. In the *deleted-grammar* family, every internal node uses a deleted rule. In the *deleted-grammar-in-context* family, the L ancestors of the last token use kept rules and all other internal nodes use deleted rules. The last family tests whether the model can interpret deleted rules in the context to predict a kept continuation.

Generation probe. We adapt the rule-based evaluation of generated sequences in Favero et al. (2025). At each checkpoint, we sample the first token from the G_{ft} marginal and generate the remaining tokens autoregressively at temperature 1 (Figure 1(d)). We parse the sequences with G_{pre} and measure the fraction of all nodes at each level that use a deleted rule. For a given m_{keep}, the reference deleted-rule rate under G_{pre} is $1 - \frac{m_{\text{keep}}}{m}$.

Representation probes. Following Cagnetta et al. (2025), we measure the similarity of the transformer hidden representations from input x and a transformed input Rx.

- First, we sample x from unmodified G_{ft}.
- Then, transformation R replaces the rule from G_{ft} at the chosen latent node (μ_{-2}^{ℓ}, $\ell = 1, 2, 3$) with a *deleted synonym* while keeping all other nodes to generate the same rules and symbols as x (Figure 1(e)).
- For x and Rx, we take $\hat{h}$, a coordinate-wise standardized d-dimensional hidden representation after the final normalization layer at the second-to-last token position.
- We measure the representation similarity as

$$q^R = \frac{1}{d}\,\mathbb{E}_{x,R}\left[\hat{h}(x)\cdot\hat{h}(Rx)\right], \tag{1}$$

  at each checkpoint throughout fine-tuning.

Thus, the metric measures the difference in hidden representations between kept rules and deleted rules when they are supposed to predict the same next token. In fact, for a transformer trained on G_{pre}, $q^R \simeq 1$ since latent synonyms have the same hidden representations. A decrease in q^R indicates that a deleted rule acquired a different hidden representation with respect to its original group of synonyms.

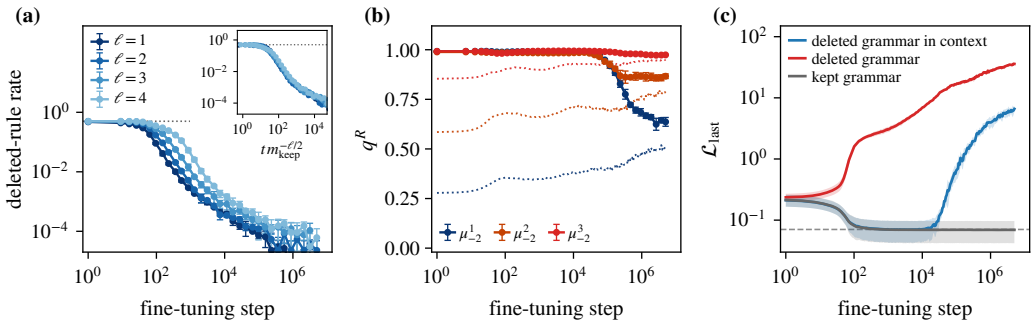


Figure 2: The two timescales of forgetting in the restriction at $m = 6$ ($m_{\text{keep}} = 3$); mean $\pm$ std over 4 seeds. We reproduce the same results for $m = 8$ and $m = 10$ (Figure 5). **(a)** The deleted-rule rate at levels $\ell = 1$ to 4 against the fine-tuning step t; dotted line: the reference value $1/2$. Inset: the same curves without error bars, against the rescaled step $tm_{\text{keep}}^{-\ell/2}$ of Section 4.2. **(b)** The deleted-rule overlap q^R at μ^1_{-2}, μ^2_{-2} and μ^3_{-2}, with each node's symbol reference (dotted). **(c)** $\mathcal{L}_{\text{last}}$ on the three test families; dashed line: the Bayes-optimal loss of G_{ft}, seed mean.

To set the baseline of q^R, we evaluate the same overlap after a symbol replacement: the transformation replaces the symbol at the chosen node by a uniformly drawn different symbol and regenerates its subtree with kept rules. We call this value the *symbol reference*. It is the overlap between inputs whose latents differ, so a decrease of q^R toward it indicates that the representation separates a deleted rule from its kept synonyms as it separates different symbols.

4 Results

4.1 Hierarchical learning in next-token prediction pretraining

Cagnetta & Wyart (2024); Cagnetta et al. (2025) studied pretraining on the RHM for the last-token prediction task, and Favero et al. (2025) studied it for discrete diffusion models. They showed that deep generative models learn to generate sequences consistent with the ℓ-level grammar rules when the amount of training data reaches the level $P_\ell \sim vm^{\ell+1}$ (or $P_\ell \sim vm^{2\ell-1}$ depending on specific architecture and task). Specifically, the value of P_ℓ is obtained from a signal-to-noise argument, where the signal strength is given by the correlation strength between $\ell - 1$-level latent tuples and visible tokens, while the sampling noise strength is given by the law of large numbers. Since we consider online learning, the number of pretraining time steps $\tau^{(pre)}$ is proportional to seen training samples, and the corresponding training timescales $\tau_\ell^{(pre)}$ to learn level ℓ rules follow $\tau_\ell^{(pre)} \propto P_\ell$.

In Figure 9 of Appendix A.3, we confirm that the next-token prediction task follows the stage-wise learning shown in Cagnetta et al. (2025); Favero et al. (2025). The transformer's last-token loss decreases through the plateaus $\mathcal{L}_0, \dots, \mathcal{L}_4$, with transitions happening at learning times $\tau_1^{(pre)}, \dots, \tau_L^{(pre)}$ whose scaling is consistent with

$$\tau_\ell^{(pre)} \propto m^{\ell+1}. \tag{2}$$

4.2 Fine-tuning: two timescales of forgetting

Restriction fine-tuning rapidly suppresses the generation of deleted rules, while their representations and their predictive use in context persist for much longer.

Rapid suppression of generation. Figure 2(a) shows that the deleted-rule rate falls from near $1/2 = 1 - m_{\text{keep}}/m$ to near zero at every level and remains low throughout fine-tuning. Suppression proceeds from the bottom rule book ($\ell = 1$) to the root ($\ell = 4$), with the same ordering across seeds. Unparseable sequences due to grammatical errors occur during the initial fine-tuning but become rare later, so the decrease in deleted-rule use is accompanied by grammatically valid generations.

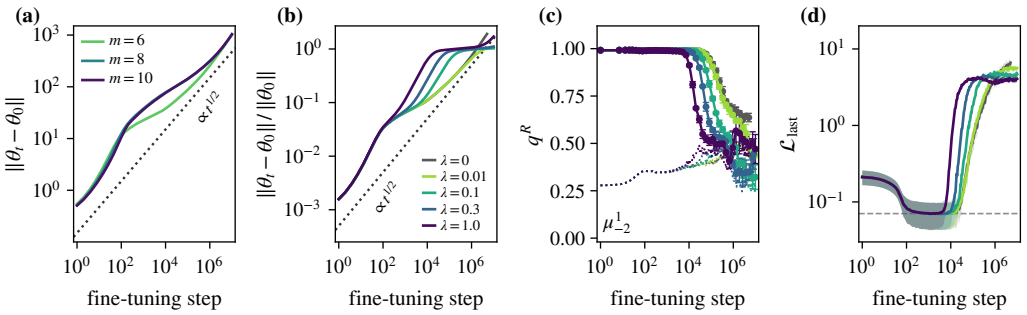


Figure 3: Parameter motion and weight decay in the restriction fine-tuning with $m_{\text{keep}} = m/2$; mean $\pm$ std over 4 seeds. **(a)** Distance $\|\theta_t - \theta_0\|$ from the pretrained parameters at $m = 6$, 8 and 10 without weight decay; dotted line: $t^{1/2}$ guide. **(b)** to **(d)** The $m = 6$ runs with weight decay λ (legend in (b)): **(b)** the distance relative to the pretrained norm, $\|\theta_t - \theta_0\|/\|\theta_0\|$, with a dotted $t^{1/2}$ guide; **(c)** q^R at μ^1_{-2}, with the symbol references dotted; **(d)** $\mathcal{L}_{\text{last}}$ on the deleted-grammar-in-context family, with the Bayes-optimal loss of G_{ft} dashed.

In the inset of Figure 2(a) and in Figure 6, we plot the generation curves against $tm_{\text{keep}}^{-\ell/2}$, where t counts fine-tuning steps. The collapse shows that the fine-tuning timescale for the generation drop is consistent with

$$t_\ell^{(ft)} \sim m_{\text{keep}}^{\ell/2}, \tag{3}$$

which is inversely proportional to the magnitude of the correlations between a token and its level-ℓ latent ancestor for an RHM with $m = m_{\text{keep}}$ (Cagnetta & Wyart, 2024). In Section 4.4 and Appendix B, we show that these correlations set the magnitude of the first fine-tuning gradient update, therefore suggesting that this signal strength determines the timescale for generation change during fine-tuning.

Delayed change in representations. The representation probe q^R remains close to 1 after generation has been suppressed at every level (Figure 2(b)). During this period, replacing a kept rule in context with a deleted synonym has little effect on the representation used for prediction. The model therefore retains approximate invariance to deleted expressions of the same latent despite rarely generating them.

Only after orders of magnitude more fine-tuning does q^R decrease toward the symbol reference. The probe q^R at $\ell = 1, 2, 3$ starts to decrease at similar times, in contrast to the clear sequential ordering of learning each hierarchical latent level during pretraining (Figure 9(b)).

Predictive use of deleted rules in context. In Figure 2(c), the three test families distinguish suppression of deleted continuations from the loss of the ability to interpret deleted rules in context. The kept-grammar loss decreases smoothly toward its Bayes-optimal value. The deleted-grammar loss rises during the initial suppression of generation and continues to increase.

The deleted-grammar-in-context sequences require the model to interpret deleted rules in the context while predicting a continuation in kept rules. Their loss initially decreases together with the kept-grammar loss and remains close to the G_{ft} Bayes-optimal loss for a much longer period after the rapid increase of the deleted-grammar loss. This indicates that the deleted rules remain useful for prediction after the behavioral suppression, and that their predictive use deteriorates much more slowly.

4.3 Weight motion and the slow timescale

The delayed change in representations occurs while the parameters continue to move after the behavioral adaptation.

Parameter motion without weight decay. Let θ_0 denote the pretrained parameters and θ_t the parameters after t fine-tuning steps. In Figure 3(a), the distance $\|\theta_t - \theta_0\|$ depends on m at early

and intermediate times, but at late times the weights for three grammar sizes approach a common asymptotic growth approximately proportional to $t^{1/2}$. The generation is suppressed at fine-tuning times $t \sim 10^2$–10^3 while parameters have moved by a small fraction of their pretraining norm (Figure 3(b), $\lambda = 0$) and q^R stays close to 1 (Figure 2(b)). The output thus adapts while the parameters and the internal representations probed by q^R change little, suggesting lazy fine-tuning dynamics that leave internal representations unchanged (Jacot et al., 2018; Chizat et al., 2019).

The motion continues long after the kept-grammar loss has reached its Bayes-optimal value and the mean gradient norm on fresh G_{ft} batches has decreased by more than an order of magnitude since the beginning of the fine-tuning (Figure 7). By the time q^R decreases, the weight-variation norm is a non-negligible fraction (i.e., $> 20\%$) of the pretraining weight norm.

Weight motion and representation change under weight decay. To test the connection between late weight motion and the slow timescale of representation change, we repeat the $m = 6$ restriction runs with decoupled weight decay (Loshchilov & Hutter, 2019) of strength $\lambda \in \{0.01, 0.1, 0.3, 1\}$, starting from the same pretrained networks and keeping all other settings unchanged. At learning rate $\eta = 10^{-4}$, decay alone multiplies each decayed weight by $1 - \eta\lambda$ per step.

At early times, the relative weight displacement follows the same trajectory for all decay strengths (Figure 3(b)). The deleted-rule generation rate also follows the same trajectory across all tested decay strengths and hierarchical levels (Figure 8). At later times, increasing λ causes the weight displacement to depart earlier from the trajectory without decay and become comparable to the pretrained weight norm sooner (Figure 3(b)). Correspondingly, q^R begins to decrease toward the symbol reference earlier (Figure 3(c)), and similarly the loss on the deleted grammar in context starts rising earlier (Figure 3(d)).

Thus, faster late weight motion is accompanied by earlier changes in the representations of deleted rules, while the measured adaptation of generation to the fine-tuning grammar remains unchanged.

4.4 Theoretical considerations on the generation adaptation

In this section, we propose an origin for the observed fine-tuning timescales of suppressed generation (Section 4.2), summarized in Equation 3. In line with existing theories of RHM learnability, we relate the strength of the learning signal to the magnitude of token-latent correlations, observing that they are consistent with the learning timescales of the restricted grammar.

Building on the observation in Section 4.2 that the internal representations stay unchanged in the early fine-tuning dynamics, in Appendix B.1 we consider a linear model on fixed features that learned to encode the latent data structure during pretraining. In this simplified setting, the first fine-tuning gradient step has size

$$\Delta p(y \mid x) = p_{\text{ft}}(y \mid x) - p_{\text{pre}}(y \mid x),$$

where $p_{\text{pre}}(y \mid x)$ and $p_{\text{ft}}(y \mid x)$ are respectively the pretraining and fine-tuning distributions of token y conditioned on context x.

Hierarchical scaling of the fine-tuning signal. Let $h(x)$ denote the hidden latent identified by the context x. Contexts corresponding to the same latent are synonymous and carry the same information about the target: $p_\nu(y \mid x) = p_\nu(y \mid h(x))$ with $\nu \in \{\text{pre}, \text{ft}\}$. We use ℓ to denote the depth of the lowest common ancestor connecting the context subtree to y.

Define the connected context-latent–token correlation

$$C_\nu^{(\ell)}(h, y) = p_\nu(h, y) - p_\nu(h)p_\nu(y).$$

The context-dependent part of the prediction is therefore

$$g_\nu^{(\ell)}(y \mid x) := p_\nu(y \mid x) - p_\nu(y) = \frac{C_\nu^{(\ell)}(h(x), y)}{p_\nu(h(x))}. \tag{4}$$

Marginal probabilities at RHM nodes are random quantities of typical size $1/v$ (Cagnetta et al., 2024). Therefore, in terms of RHM parameter scaling magnitude, we can write

$$g_\nu^{(\ell)}(y \mid x) \sim v\, C_\nu^{(\ell)}(h(x), y).$$

The target fine-tuning correction separates into a marginal shift and a context-dependent contribution: $\Delta p(y \mid x) = p_{\rm ft}(y) - p_{\rm pre}(y) + \Delta g^{(\ell)}(y \mid x)$. We assume that the network biases quickly adapt to the variation in the marginal distribution, so that we are left with the analysis of the context-dependent part:

$$\Delta p(y \mid x) \simeq \Delta g^{(\ell)}(y \mid x) \sim v \left[C_{\rm ft}^{(\ell)}(h(x), y) - C_{\rm pre}^{(\ell)}(h(x), y) \right]. \tag{5}$$

Crucially, the pretrained and fine-tuning correlations are not independent: the restricted grammar retains a subset of the original rules. Consequently, the magnitude of their difference depends on their shared contribution, and its typical value can be computed through the average $\mathbb{E}$ over the choice of RHM rules: $\mathbb{E}\left[\left(C_{\rm ft}^{(\ell)} - C_{\rm pre}^{(\ell)} \right)^2 \right] = \mathbb{E}\left[(C_{\rm ft}^{(\ell)})^2 \right] + \mathbb{E}\left[(C_{\rm pre}^{(\ell)})^2 \right] - 2\mathbb{E}\left[C_{\rm ft}^{(\ell)} C_{\rm pre}^{(\ell)} \right]$.

In Appendix B.2, we estimate $\Delta p(y \mid x)$ from the RHM rule structure. In terms of magnitude scaling, we get

$$\Delta p(y \mid x) \sim \frac{\sqrt{1 - (m_{\rm keep}/m)^\ell}}{\sqrt{v}\, m_{\rm keep}^{\ell/2}}. \tag{6}$$

Difference between pretraining and fine-tuning sample complexity argument. Existing estimates of RHM pretraining sample complexity compare the magnitude of latent–token correlations with the sampling noise in their empirical estimates (Cagnetta & Wyart, 2024; Favero et al., 2025). This comparison identifies the number of samples needed to resolve the statistical structure used to learn latent representations. In online training with fixed batch size, the number of observed samples is proportional to the number of training steps.

For restriction fine-tuning, we instead find that the measured timescales for suppressing deleted-rule generation (Equation 3) are consistent with the inverse magnitude of the conditional correction in Equation 6, which gives the magnitude of the first gradient update. A possible explanation for the difference in pretraining and fine-tuning dynamics is that fine-tuning reuses the latent representations acquired during pretraining. Adaptation can therefore proceed by changing the predictions associated with existing features, without first reconstructing the latent hierarchy. A more precise understanding of the observed different timescale behaviors requires a deeper analysis of the learning dynamics under Adam, which remains an open problem.

5 Experiments with natural language data

We primarily use RHM as a controlled synthetic setup to discover and understand the dissociation between the timescales of behavioral and latent forgetting. To connect these findings to more naturalistic data, we devise experiments with a decoder-only transformer trained with a bilingual natural language corpus.

For pretraining, we train an 8-layer decoder-only transformer on a corpus in which each topically coherent passage is written in either English (EN) or French (FR) sentences, chosen at random. This is a naturalistic reflection of RHM where the latent contains the grammatical rules for generating English or French text. We then fine-tune on a monolingual (EN or FR) corpus, mirroring the fine-tuning on restricted generative rules in RHM. Finally, we evaluate the same measures of behavioral and latent forgetting in this model. Details of the pretraining and fine-tuning corpus generation, training setup, and evaluation are in Appendix C.

We focus on the model fine-tuned on EN-only text here, but the results are consistent with the FR-only fine-tuned model (Figure 12). After fine-tuning on EN-only text, we replicate the measurement setting used for RHM in Figure 2. In Figure 4(a), we find that the generation rate of FR drops rapidly long before 10^2 steps. To test whether the latent representations of one language in the model are intact, we measure the representational similarity between coherent sentences and incoherent versions with randomly shuffled tokens. High similarity indicates that the model lost capacity to encode coherent meaning of the language, a sign of representational forgetting (Figure 4(b)), which happens only after 10^2 steps. In Figure 4(c), we replicate the measurement of Figure 1(f) in the bilingual model, again marking the transition in generation suppression and representational forgetting at 10^2

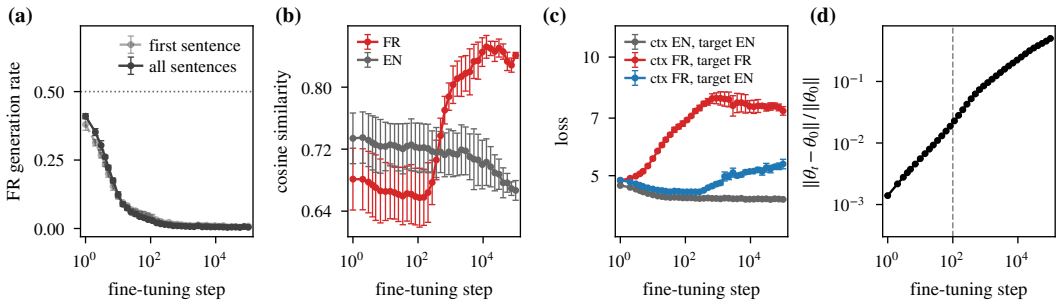


Figure 4: Bilingual pretraining followed by EN-only fine-tuning. **(a)** FR generation rate drops rapidly before the 10^2 fine-tuning steps, both for the first sentence and all sentences in the context window. **(b)** Cosine similarity between representations of coherent and token-shuffled sentences in FR and EN; higher similarity indicates loss of latent linguistic capability. Rapid rise of FR happens after 10^2 steps. **(c)** Reproducing the three test families (Figure 1(f)). Prediction losses for EN targets with EN context (gray), FR targets with FR context (red), and EN targets with FR context (blue) show a similar trend, characterized by a further drop of the loss for gray (kept rules), a faster rise of the loss in red (deleted rules) and a slower timescale of blue (deleted rules in context). **(d)** Parameter distance relative to the pretrained norm; the vertical line marks 10^2 steps coinciding with the transition in the behavioral and representational forgetting.

steps. Finally, we also observe a slight crossover in the increase of the norm of relative weight change (Figure 4(d)), showing that also in this setting the model weights keep drifting even when the output has adapted to the fine-tuning distribution. Overall, these experiments show that the separation of timescales between behavioral and representational forgetting is not restricted to the synthetic RHM but is a general phenomenon in fine-tuning with natural language.

6 Discussion

Our results show that the outcome of fine-tuning on restricted data is governed by two timescales: a fast timescale of behavioral forgetting in outputs, due to a lazy dynamics on the pretrained features and consistent with the latent-token correlations, and a slow timescale of representational forgetting, stemming from the accumulating fine-tuning gradient updates. This provides a plausible mechanism behind the widely reported observations in LLMs of a discrepancy between the drop in performance of older tasks and the actual erasure of representations. Our findings highlight that, beyond the lazy-regime dynamics in the early phase of fine-tuning (Malladi et al., 2023) which modifies model behavior without changing learned features, the slower drift (Blanc et al., 2020; Li et al., 2022) can cause latent erasure on much longer timescales.

In this work, the considered setting is constrained to a specific form of fine-tuning, that is a narrower distribution of the training data. In Appendix A.4, we show that fine-tuning with out-of-distribution data generated from new sets of rules shows forgetting-relearning dynamics. We leave a thorough analysis of this case for future work.

While here we focus exclusively on SFT, Reinforcement Learning (RL) is another important and widely used fine-tuning method in modern training pipelines. Studying how knowledge and skills from pretraining can continue to be learned, composed, and reused throughout RL post-training (Lee et al., 2024; Zhao et al., 2025) is an exciting future direction.

AI Usage Statement

We use LLM assistance during manuscript preparation to improve clarity and consistency. In particular, LLMs were used to help reorganize mathematical notation and improve the manuscript for grammar, style, and readability. We use LLM assistance for coding implementation of the experiments and styling figures, but ideation, design of the experiments and theoretical models, and interpretation of the results are from the authors.

Ethics Statement

While our work aims for a fundamental understanding of language models and their fine-tuning process, we do not anticipate any immediate societal impact from this research.

Reproducibility Statement

We report detailed experimental setup to reproduce all the results in Appendix. The code base to reproduce all results will be open-sourced upon acceptance.

References

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process. In *Conference on Learning Theory*, pp. 483–513. PMLR, 2020.

Lukas Braun, Clémentine Dominé, James Fitzgerald, and Andrew Saxe. Exact learning dynamics of deep linear networks with prior knowledge. *Advances in Neural Information Processing Systems*, 35:6615–6629, 2022.

Francesco Cagnetta and Matthieu Wyart. Towards a theory of how the structure of language is acquired by deep neural networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 83119–83163. Curran Associates, Inc., 2024. doi: 10.52202/079017-2645. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9740da1c07c7b451af14e11523f95271-Paper-Conference.pdf.

Francesco Cagnetta, Leonardo Petrini, Umberto M. Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The Random Hierarchy Model. *Phys. Rev. X*, 14:031001, Jul 2024. doi: 10.1103/PhysRevX.14.031001. URL https://link.aps.org/doi/10.1103/PhysRevX.14.031001.

Francesco Cagnetta, Alessandro Favero, Antonio Sclocchi, and Matthieu Wyart. Scaling laws and representation learning in simple hierarchical languages: Transformers versus convolutional architectures. *Phys. Rev. E*, 112:065312, Dec 2025. doi: 10.1103/qtd6-nl8p. URL https://link.aps.org/doi/10.1103/qtd6-nl8p.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16712–16721, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Davari_Probing_Representation_Forgetting_in_Supervised_and_Unsupervised_Continual_Learning_CVPR_2022_paper.html.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024.

Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16286–16306, 2025.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309. IEEE, 2020a.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, pp. 383–398. Springer, 2020b.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations*, 2014.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Andrej Karpathy. nanoGPT: The simplest, fastest repository for training/finetuning medium-sized GPTs. `https://github.com/karpathy/nanoGPT`, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Suhas Kotha, Jacob Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations*, volume 2024, pp. 24110–24139, 2024.

Jin Hwa Lee, Stefano Sarao Mannelli, and Andrew Saxe. Why do animals need shaping? A theory of task composition and curriculum learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26837–26855, 2024.

Sebastian Lee, Sebastian Goldt, and Andrew Saxe. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, pp. 6109–6119. PMLR, 2021.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? A mathematical framework. In *International Conference on Learning Representations*, 2022.

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

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in LLMs. *arXiv preprint arXiv:2402.16835*, 2024. URL `https://arxiv.org/abs/2402.16835`.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pp. 23610–23641. PMLR, 2023.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 4226–4237, Torino, Italia, May 2024. ELRA and ICCL. URL `https://aclanthology.org/2024.lrec-main.377`.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 16828–16847. PMLR, 2022.

Farhad Pashakhanloo and Alexei Koulakov. Stochastic gradient descent-induced drift of representation in a two-layer neural network. In *International Conference on Machine Learning*, pp. 27401–27419. PMLR, 2023.

Vinay Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of model and pretraining scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2022.

Vinay V. Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LhY8QdUGSuw.

Sebastian Raschka. Naive Bayes and text classification I – Introduction and theory. *arXiv preprint arXiv:1410.5329*, 2014.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1): e2408799121, 2025.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025.

Jörg Tiedemann. The Tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pp. 1174–1182, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.139.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pp. 479–480, Lisboa, Portugal, November 2020. European Association for Machine Translation. URL https://aclanthology.org/2020.eamt-1.61.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=dp4KWuSDzj.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations*, volume 2025, pp. 67081–67146, 2025.

A Additional results and experimental details on the Random Hierarchy Model

A.1 Experimental details

The settings below apply to the restriction experiments with $L = 4$, $s = 2$, $v = 24$, $m \in \{6, 8, 10\}$ and $m_{\text{keep}} = m/2$. We retain the last m_{keep} entries of each symbol's rule list. Since the entries are exchangeable under the grammar draw, this is equivalent to selecting a uniformly random subset. Table 1 supplements the training setup in Section 3.1.

For the representation probes of Section 3.2, both inputs are standardized using the coordinate-wise mean and standard deviation of the untransformed samples at the same checkpoint. After replacing a rule, we regenerate its descendants using kept rules; tokens outside the selected subtree are unchanged. The symbol reference uses the same original samples as q^R. In the generation probe, unparseable nodes remain in the denominator and count as neither kept nor deleted.

Table 1: Network, training and evaluation settings supplementing Section 3.1 for the $L = 4$ restriction experiments.

blocks	4, with layer normalization before each attention and feed-forward operation
hidden width d, heads	512, 16
feed-forward layers	hidden width $4d$, Gaussian-error linear unit (GELU)
positional embedding	learned, absolute
readout	linear, tied to the token embedding, after a final layer normalization
biases, dropout	none
parameters n_θ	12.6×10^6
Adam coefficients	$\beta_1 = 0.9$, $\beta_2 = 0.999$
weight decay	none in baseline runs; decay in Section 4.3 applies to matrices and embeddings, not layer-normalization gains
gradient clipping	none
test set	2^{15} held-out sequences
pretraining steps	10^6 at $m = 6$; 10^7 at $m = 8$ and 10
fine-tuning steps	up to 10^7, continuing beyond the initial convergence of the kept-grammar loss
fine-tuning checkpoints	nominal 280-point logarithmic grid, rounded to integer steps with duplicates removed; weights only
seeds	4 per m, each with its own grammar and network initialization
generation probe	8192 sequences per checkpoint
representation probe	4096 original sequences per node and checkpoint
test families	8192 sequences per family and seed

A.2 Additional results

Figure 5 extends Figure 2 to $m = 8$ and $m = 10$. Representational change occurs earlier at $m = 6$ than at these two grammar sizes, whose timescales are similar. However, their base networks were pretrained for longer (Table 1) and have larger parameter norms. This comparison therefore does not isolate the dependence of the slow timescale on m.

The mean gradient in Figure 7 is estimated from $N = 256$ independent batches of 128 sequences per checkpoint. For their gradients g_b,

$$\widehat{S} = \frac{\left\| \sum_{b=1}^{N} g_b \right\|^2 - \sum_{b=1}^{N} \|g_b\|^2}{N(N-1)} \tag{7}$$

is an unbiased estimate of $\|\mathbb{E}[g]\|^2$. We plot $\sqrt{\max(\widehat{S}, 0)}$, evaluated separately for each seed before averaging. The estimator removes the batch-variance contribution that biases the squared norm of a finite-batch average upwards.

Figure 8 compares the generation probe in the runs with and without weight decay of Section 4.3. At every level, the deleted-rule rate falls along the same curve for all λ. At late times, strong decay leaves a higher residual rate, which remains orders of magnitude below its initial value.

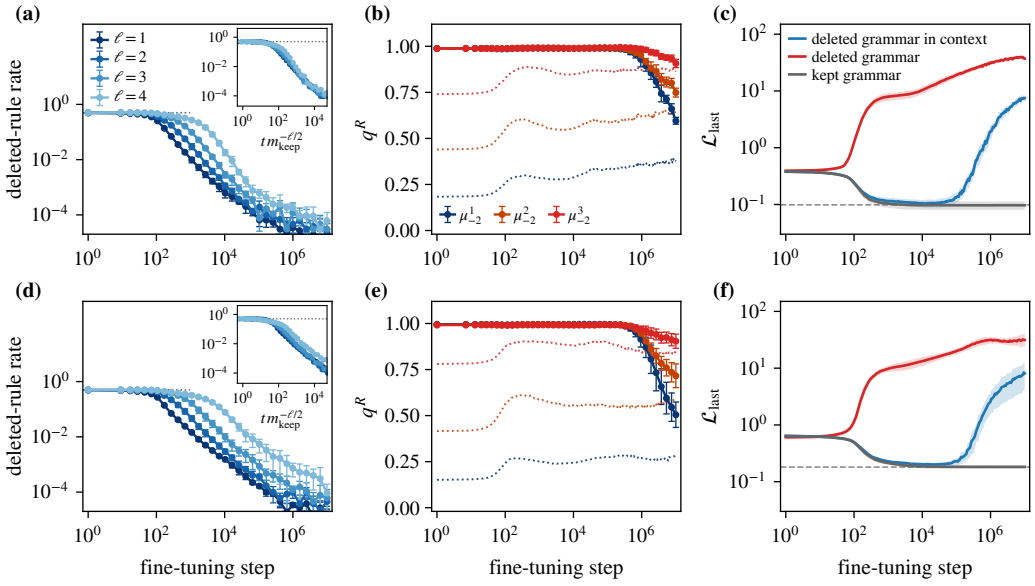


Figure 5: Restriction fine-tuning at $m = 8$, $m_{\text{keep}} = 4$ (top row), and $m = 10$, $m_{\text{keep}} = 5$ (bottom row); mean $\pm$ std over 4 seeds. **(a), (d)** Deleted-rule generation rates at levels $\ell = 1, \ldots, 4$ against fine-tuning step t; dotted line: $1/2$. Insets: the same seed means without error bars against $t\, m_{\text{keep}}^{-\ell/2}$, as in Figure 2(a). **(b), (e)** q^R at μ^1_{-2}, μ^2_{-2} and μ^3_{-2}, with each node's symbol reference (dotted). **(c), (f)** $\mathcal{L}_{\text{last}}$ on the three test families; gray dashed line: the seed-averaged Bayes-optimal loss of G_{ft}. Colors follow Figure 2.

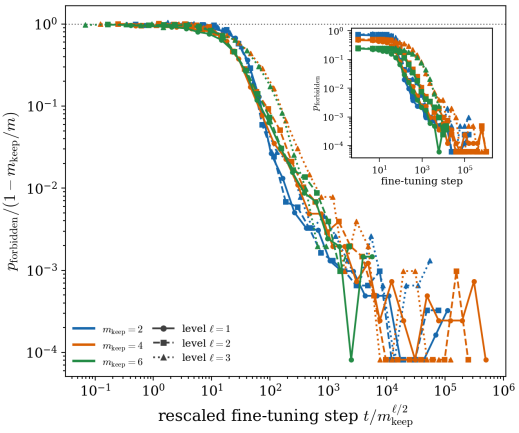


Figure 6: Restriction fine-tuning for RHM grammar with $L = 3$, $v = 16$, $m = 8$ and varying values of m_{keep}. The probability of generating deleted rules starts decreasing at different timescales depending on m_{keep} and rule level ℓ (inset). Rescaling the fine-tuning time step by $m_{\text{keep}}^{\ell/2}$ approximately collapses the point where the curves start decreasing (main).

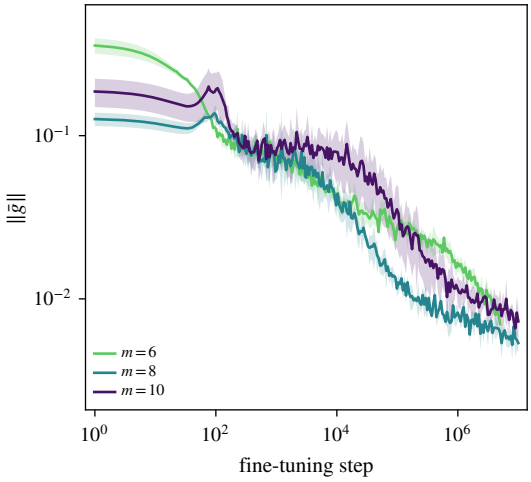


Figure 7: Norm of the mean gradient on fresh G_{ft} batches, estimated with equation 7, in the restriction runs at $m = 6$, 8 and 10 without weight decay; mean $\pm$ std over 4 seeds.

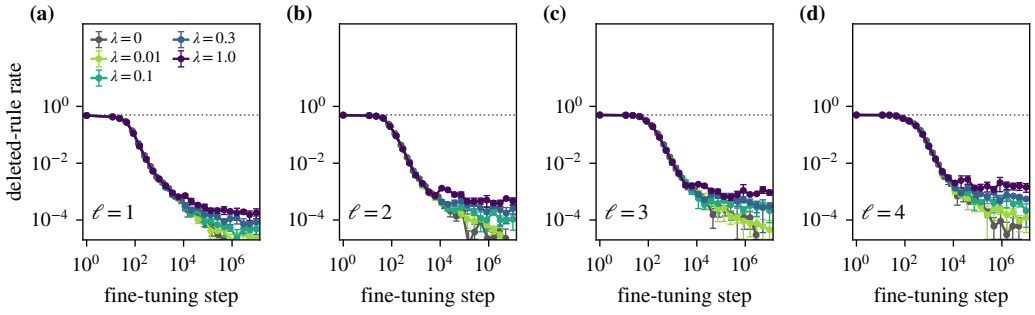


Figure 8: Deleted-rule rate at levels $\ell = 1$ to 4, panels **(a)** to **(d)**, in the $m = 6$ restriction runs with weight decay λ (legend in (a)), on the axes of Figure 2(a); dotted line: $1/2$; mean $\pm$ std over 4 seeds.

A.3 PRETRAINING: NETWORK DEPTH AND SCALING ACROSS GRAMMAR SIZES

We supplement Section 4.1 with a network-depth comparison and a test across grammar sizes. The loss curves show $\mathcal{L}_{\text{last}}$ against the number of sampled training sequences P. For the plateau references defined in Section 3.2, belief propagation marginalizes all leaves outside the specified context, following Eq. 22 of Cagnetta et al. (2025). Each reference is evaluated using the seed's own grammar and 8192 held-out sequences.

The figures mark the sample scales of Section 4.1 with their prefactor,

$$P_1 \propto (1-f)^{-1} v\, m, \qquad P_\ell \propto (1-f)^{-1} v\, m^{\ell+1} \quad (\ell \geq 2), \qquad f = \frac{m}{v^{s-1}}. \tag{8}$$

The bottom book is the exception to the form $m^{\ell+1}$. No latent below it has to be inferred, and learning it only requires observing its vm admissible s-tuples (Favero et al., 2025); P_1 is the first scale of Eq. 12 in Cagnetta & Wyart (2024).

During pretraining, we evaluate the overlap defined in equation 1 under the symbol replacement that defines the symbol reference in Section 3.2, sampling x from G_{pre} and regenerating the replaced subtree with G_{pre}. We denote this probe by q^S. It is close to 1 when the representation does not depend on the latent at that node and decreases as this dependence develops. At the end of pretraining at $m = 6$, its values are close to those of the symbol reference at the start of fine-tuning in Figure 2(b).

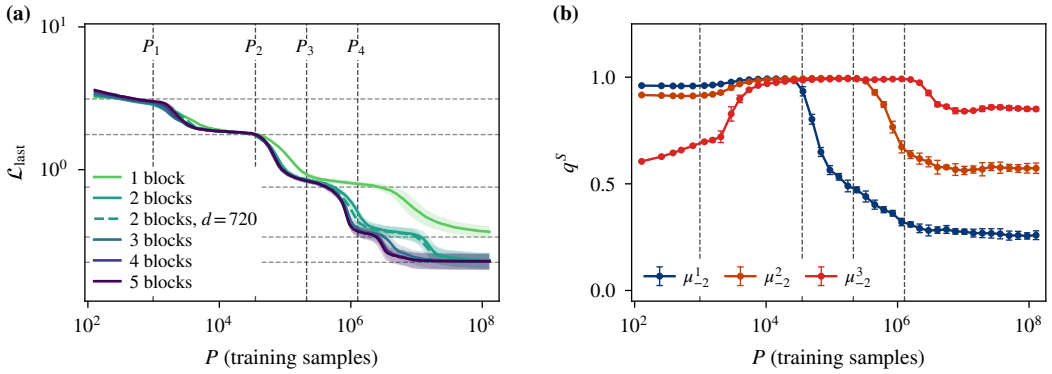


Figure 9: Pretraining at $m = 6$. **(a)** $\mathcal{L}_{\text{last}}$ for networks with 1 to 5 blocks, and a two-block network with hidden width 720 (dashed, labeled $d = 720$ in the legend), approximately matching the parameter count of the four-block network; mean $\pm$ std over 4 seeds. Horizontal dashed lines: the seed-averaged plateau losses $\mathcal{L}_0, \ldots, \mathcal{L}_4$. Vertical dashed lines: the sample scales $P_1, \ldots, P_4$ of equation 8. Their common prefactor is fixed by the geometric mean, over the five network depths, of the sample count where the mean loss crosses $\mathcal{L}_1$. **(b)** q^S for the four-block network at μ^1_{-2}, μ^2_{-2} and μ^3_{-2}; mean $\pm$ std over 8 seeds, with the same vertical lines.

The depth comparison uses the same grammars and training settings, with 10^6 steps for each network. Reducing the number of blocks delays the later loss decreases, and the one-block network does not reach the full-context reference within the training duration. Increasing the width of the two-block network leaves its learning curve close to that of the narrower two-block network. Thus, the delay in these runs cannot be attributed solely to their smaller parameter count.

The comparison across grammar sizes uses 10^6 pretraining steps at $m = 6, 7, 9$ and 10^7 at $m = 8, 10$. Under the rescaling in Figure 10, the departures from successive plateaus approximately align across m. The agreement concerns the transition locations; the losses reached after each transition remain grammar-dependent.

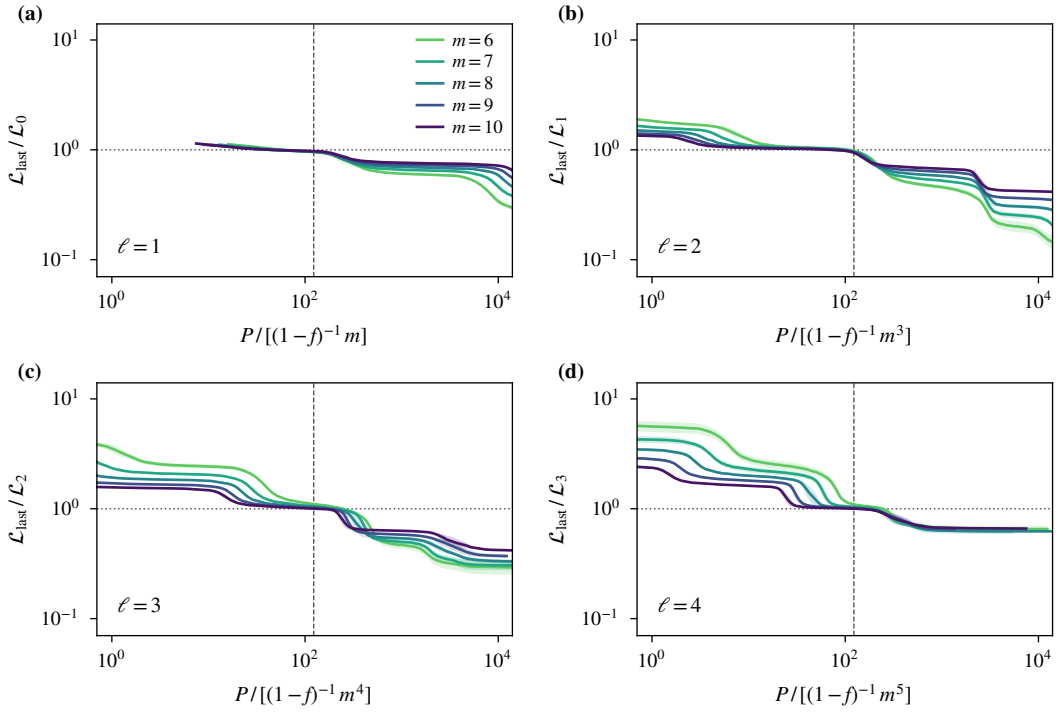


Figure 10: Last-token learning curves of the four-block network at $m = 6, \dots, 10$; mean $\pm$ std over 8 seeds. Panels **(a)** to **(d)** correspond to the transitions $\ell = 1, \dots, 4$. Each plots $\mathcal{L}_{\text{last}}/\mathcal{L}_{\ell-1}$, using the seed's own plateau reference, against P divided by the m-dependent factor of P_ℓ in equation 8. Dotted horizontal line: 1. Dashed vertical line: the common prefactor fixed in Figure 9, expressed on the rescaled axis.

A.4 Fine-tuning after resampling one rule book

We redraw the rule book at one level ℓ independently from the same RHM ensemble, leaving the other books unchanged. Some tuples or rules can therefore recur. Unlike the restriction in Section 3.1, this intervention introduces new parent–child associations. We consider $\ell = 1, \dots, 4$ at $m = 6$ and fine-tune for 1.25×10^6 steps, with four seeds per level and the optimization settings of Section 3.1. Here G_{ft} denotes the resampled grammar. We evaluate the loss and plateau references defined in Section 3.2 and the symbol-replacement probe q^S of Appendix A.3, using this grammar for both the original inputs and the transformed subtrees. The plateau references are averaged over 10^5 held-out sequences per seed.

In Figure 11(a) to (d), after the initial adjustment, the loss approaches the reference $\mathcal{L}_{\ell-1}$ associated with the resampled level: $\mathcal{L}_3$ for the root book, $\mathcal{L}_2$ for level 3, $\mathcal{L}_1$ for level 2, and approximately the random-guess loss for the bottom book. The unchanged books below the edit retain their conditional expansions, allowing prediction from local structure. Their symbol marginals can nevertheless change, so resampling need not preserve the old local prediction probabilities.

For the probed nodes $\mu_{-2}^{\ell'}$ with $\ell' \geq \ell - 1$, q^S rises toward 1 (Figure 11(e) to (h)), indicating reduced dependence on these latents of the new grammar. Lower nodes show smaller, transient changes. Since the probe uses G_{ft}, it measures adaptation to the new associations and does not establish complete loss of the old representations.

The subsequent loss decreases pass through successive plateau references, accompanied by decreases in the corresponding probes. Resampling a lower book requires relearning more levels and takes longer. This relearning is much slower than the initial adjustment, whose duration also varies with the resampled level.

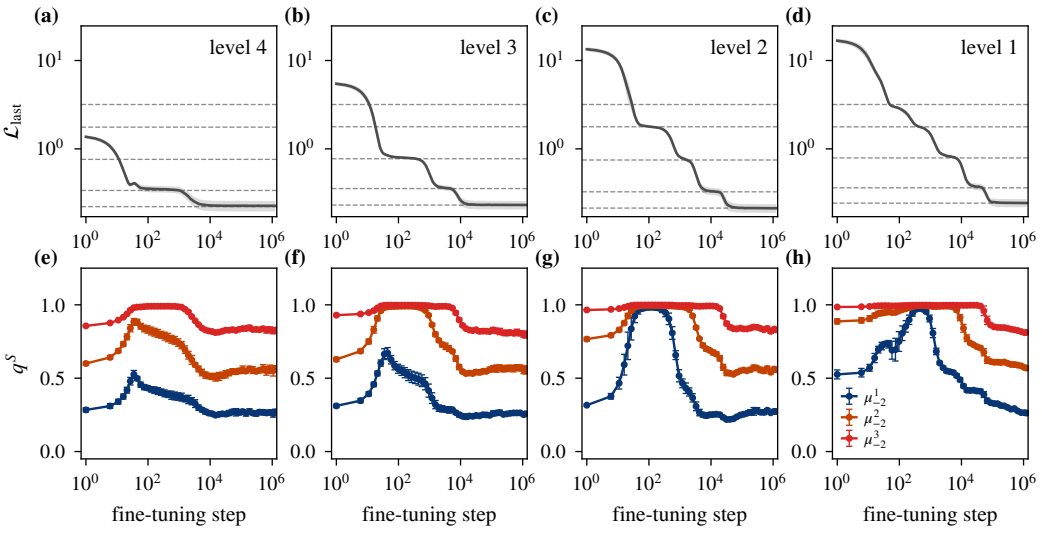


Figure 11: Fine-tuning after resampling one rule book at $m = 6$; columns run from the root book (level 4) to the bottom book (level 1). Curves and uncertainties show mean $\pm$ std over 4 seeds. **(a)** to **(d)** $\mathcal{L}_{\text{last}}$ on sequences from the resampled grammar G_{ft}; gray dashed lines: $\ln v$ and the seed-averaged plateau losses $\mathcal{L}_1, \ldots, \mathcal{L}_4$ of G_{ft}. **(e)** to **(h)** q^S at μ^1_{-2}, μ^2_{-2} and μ^3_{-2}, using G_{ft}. All panels share the fine-tuning-step axis.

B Generation fine-tuning and correlations

B.1 Fine-tuning with fixed features

Consider predicting a token $y \in [v]$ from a context x in an RHM sequence. Let h denote the hidden latent at the lowest common ancestor connecting the context to the target.

We consider an idealized d-dimensional fixed encoder learned during pretraining which recovers this latent from the context:

$$\phi(x) = \phi_h, \qquad \frac{\phi_h^\top \phi_{h'}}{d} = \kappa\, \delta_{hh'}.$$

The predictor is a trainable linear readout,

$$f_W(x) = W\phi(x), \qquad p_W(\cdot \mid x) = \text{softmax}(f_W(x)),$$

with $W \in \mathbb{R}^{v \times d}$. Its predictions depend on the context only through the encoded ancestor h. Training minimizes cross-entropy, with ϕ held fixed.

Pretraining. We initialize the readout at pretrained weights W^* that approximately fit the pretraining distribution:

$$p^*_h := \text{softmax}(W^*\phi_h) \simeq p_{\text{pre}}(\cdot \mid h).$$

Fine-tuning. Following the restriction procedure in Section 3.1, only $m_{\text{keep}} < m$ production rules are retained per restricted parent. This changes the ancestor frequencies and conditional targets to

$$\pi_h = p_{\text{ft}}(h), \qquad q_h = p_{\text{ft}}(\cdot \mid h).$$

Writing

$$p_h = \text{softmax}(W\phi_h),$$

the population fine-tuning loss becomes

$$\mathcal{L}(W) = -\mathbb{E}_{(x,y)\sim p_{\text{ft}}}\left[\log p_W(y \mid x)\right] = -\sum_h \pi_h \sum_y q_h(y) \log p_h(y).$$

Initial readout update. The first population gradient update with learning rate η reads

$$W_1 - W_0 = \eta \sum_h \pi_h (q_h - p_h) \phi_h^\top .$$

At the pretrained initialization,

$$W_1 - W^* = \eta \sum_h \pi_h (q_h - p_h^*) \phi_h^\top .$$

By orthogonality of the features, the corresponding logit change for each ancestor is exactly

$$f_{1,h} - f_h^* = (W_1 - W^*)\phi_h = \eta d \kappa \pi_h (q_h - p_h^*). \tag{9}$$

The ancestor marginals π_h are random quantities of typical size $1/v$. Approximating $\pi_h \simeq 1/v$, we get

$$f_{1,h} - f_h^* \simeq \frac{\eta d \kappa}{v} \left[p_{\mathrm{ft}}(\cdot \mid h) - p_{\mathrm{pre}}(\cdot \mid h) \right],$$

showing that the initial readout update is proportional to the change in the ancestor-conditioned target distribution.

B.2 Computation of the correlation difference

We estimate the RMS change in the conditional prediction,

$$\delta_\ell^2 := \mathbb{E}\Big[\big(p_{\mathrm{ft}}^{(\ell)}(y \mid h) - p_{\mathrm{pre}}^{(\ell)}(y \mid h) \big)^2 \Big], \tag{10}$$

where $\mathbb{E}$ averages over RHM rules. As in the main text (Section 4.4), h denotes the context node and ℓ the depth of its lowest common ancestor with the target. For simplicity, we consider the case of a binary RHM tree ($s = 2$) and retain m_{keep} of the original m rules per parent at every relevant level. We use a balanced, large-vocabulary approximation: node marginals have typical size $1/v$, and their fluctuations are neglected in the leading estimate.

Case $\ell = 1$. For $\ell = 1$, the context is a visible token c and the target y is its sibling. Let $\mathcal{S}_c$ be the set of originally allowed successors, and let $\mathcal{S}_c^{\mathrm{keep}} \subset \mathcal{S}_c$ be the retained set. Unambiguity ensures that each pair (c, y) occurs in at most one production. Under uniform parent frequencies, the conditional is therefore uniform on the corresponding set of successors. There are on average m such successors before restriction and m_{keep} afterward. Replacing the counts by these values gives

$$p_{\mathrm{pre}}(y \mid c) \simeq \frac{\mathbf{1}_{\{y \in \mathcal{S}_c\}}}{m}, \qquad p_{\mathrm{ft}}(y \mid c) \simeq \frac{\mathbf{1}_{\{y \in \mathcal{S}_c^{\mathrm{keep}}\}}}{m_{\mathrm{keep}}}.$$

Thus a retained successor changes by $1/m_{\mathrm{keep}} - 1/m$, whereas a deleted successor changes by $-1/m$. Summing the squared changes gives

$$\sum_y \big[p_{\mathrm{ft}}(y \mid c) - p_{\mathrm{pre}}(y \mid c) \big]^2 \simeq m_{\mathrm{keep}} \left(\frac{1}{m_{\mathrm{keep}}} - \frac{1}{m} \right)^2 + \frac{m - m_{\mathrm{keep}}}{m^2} = \frac{1}{m_{\mathrm{keep}}} - \frac{1}{m}. \tag{11}$$

Dividing by v yields the mean squared change per output token:

$$\delta_1^2 \simeq \frac{1}{v} \left(\frac{1}{m_{\mathrm{keep}}} - \frac{1}{m} \right), \qquad \delta_1 \simeq \frac{\sqrt{1 - m_{\mathrm{keep}}/m}}{\sqrt{v\, m_{\mathrm{keep}}}}. \tag{12}$$

General depth ℓ. For $\ell > 1$, the context identifies a latent h on one side of the LCA, and the target descends from its other child through $\ell - 1$ further levels. There are therefore approximately m^ℓ predictive paths before restriction and m_{keep}^ℓ afterward, where each path counts sequences of rule choices.

Let $\mathcal{P}_{\mathrm{pre}}$ and $\mathcal{P}_{\mathrm{ft}} \subset \mathcal{P}_{\mathrm{pre}}$ denote these path sets. Let $I_\alpha(y)$ be one if path α ends at y and zero otherwise. This indicator has disorder mean $1/v$ and variance $(1/v)(1 - 1/v) \simeq 1/v$. Suppressing the fixed arguments $(y \mid h)$, the conditional probabilities are approximated by

$$p_{\mathrm{pre}}^{(\ell)} \simeq \frac{1}{m^\ell} \sum_{\alpha \in \mathcal{P}_{\mathrm{pre}}} I_\alpha, \qquad p_{\mathrm{ft}}^{(\ell)} \simeq \frac{1}{m_{\mathrm{keep}}^\ell} \sum_{\alpha \in \mathcal{P}_{\mathrm{ft}}} I_\alpha.$$

Neglecting correlations between distinct paths, each additional level reduces the variance by a factor $1/m$ or $1/m_{\text{keep}}$, respectively:

$$\text{Var}\Big(p_{\text{pre}}^{(\ell)}\Big) \simeq \frac{1}{vm^\ell}, \qquad \text{Var}\Big(p_{\text{ft}}^{(\ell)}\Big) \simeq \frac{1}{vm_{\text{keep}}^\ell}. \tag{13}$$

The two averages are not independent: all retained paths also contribute to the pretrained prediction. Keeping their shared contribution gives

$$\text{Cov}\Big(p_{\text{ft}}^{(\ell)}, p_{\text{pre}}^{(\ell)}\Big) \simeq \frac{|\mathcal{P}_{\text{ft}}|}{m_{\text{keep}}^\ell m^\ell}\frac{1}{v} \simeq \frac{1}{vm^\ell}. \tag{14}$$

Both conditional probabilities have disorder mean $1/v$. Their mean squared difference therefore equals the sum of their variances minus twice this covariance, giving

$$\boxed{\delta_\ell^2 \simeq \frac{1}{v}\left(\frac{1}{m_{\text{keep}}^\ell} - \frac{1}{m^\ell}\right), \qquad \delta_\ell \simeq \frac{\sqrt{1-(m_{\text{keep}}/m)^\ell}}{\sqrt{v}\, m_{\text{keep}}^{\ell/2}}.} \tag{15}$$

This recovers the one-level calculation and vanishes when no rules are deleted. For a fixed retained fraction, the leading dependence on m_{keep} is $m_{\text{keep}}^{-\ell/2}$. Using $C_\nu^{(\ell)}(h,y) = p_\nu(h)[p_\nu^{(\ell)}(y \mid h) - p_\nu(y)]$, the balanced approximation relates the correlation difference to the conditional difference by an additional factor $1/v$. Hence

$$\mathbb{E}\Big[\big(C_{\text{ft}}^{(\ell)} - C_{\text{pre}}^{(\ell)}\big)^2\Big] \simeq \frac{1}{v^3}\left(\frac{1}{m_{\text{keep}}^\ell} - \frac{1}{m^\ell}\right). \tag{16}$$

C Additional results and experimental details on bilingual small language model experiment

C.1 Pretraining and fine-tuning of bilingual small language model

Translation. We primarily built the training corpus from *uonlp/CulturaX* (Nguyen et al., 2024), a widely used multilingual web text collection after cleaning and deduplication. Among 2.21M respective documents of English and French from the CulturaX corpora, we randomly sample 450,000 documents per language. Each document is split into sentences. For simplicity, we assume new sentences when followed by line breaks, then terminal punctuation followed by whitespace and an upper-case letter, digit or opening quote, with an abbreviation list. We keep a document if it consists of at least 3 sentences and at least half of its sentences survive the criteria (at least 60% letters are among non-space characters, no URL or e-mail address, at most 256 tokens, and a token-unigram language classifier that agrees with the document language), resulting in 427,631 English and 419,171 French documents. Every kept sentence is machine-translated into the other language pair, from English to French and from French to English with *Marian Transformer-big* (Tiedemann, 2020; Tiedemann & Thottingal, 2020), sentence by sentence without cross-sentence context. After the translation, we apply the same criteria again to each sentence, which keeps $\sim$97% of the sentences.

Bilingual text generation for pretraining. After the translation of each document, we rebuild the same document by randomly sampling each sentence's language. In other words, the contents of the document are the same, but the bilingual document switches each sentence's language between English and French randomly. Therefore, language is half native and half translated. We train our own byte-level BPE tokenizer using English and French corpora of CulturaX, with a vocabulary of 16,384 tokens that yields comparable counts for English and French (0.248 tokens/char for English and 0.259/char for French). The bilingual document has median 548 tokens and 18 sentences.

Fine-tuning data. For fine-tuning, we use an unseen fraction of the data from CulturaX. After the pretraining with bilingual documents, we fine-tune the model with monolingual documents, reflecting the fine-tuning with data generated from restricted grammar in RHM.

Training setup. The model is a decoder-only nanoGPT architecture (Karpathy, 2022) with 8 pre-LayerNorm blocks, 8 heads, width 512, MLP width 2048 with GELU activation, rotary position embeddings, no biases, no dropout, tied input/output embeddings and context length of 512. This architecture resulted in ∼34M parameters. We train with the next-token prediction objective with batch size 64, for 10^5 steps, corresponding to 3.28B tokens. For optimization, we used AdamW with β = (0.9, 0.95), weight decay of $\{0, 0.1\}$, peak learning rate 6×10^{-4}, 1,000 warm-up steps, cosine decay to 6×10^{-5}, gradient clipping at 1.0. For fine-tuning, we take the final checkpoints from the pretraining and continued next-token prediction training for 10^5 steps at batch size of 64, with the constant learning rate 6×10^{-5} and same optimizer setting. We perform 3 seeds of pretraining and applied a single seed of English monolingual fine-tuning and French monolingual fine-tuning.

C.2 Evaluation of bilingual pretrained–monolingual fine-tuned language models

Free generation. We give a single end-of-text token in-context and let the model autoregressively generate 160 tokens at single context in total 1024 samples. For each passage, we sample 160 tokens with temperature 1.0. We count each sentence by splitting at newline tokens. To classify each sentence's language, we use a naive Bayes classifier over the vocabularies (Raschka, 2014). We take 50M tokens from EN and FR each from the CulturaX corpus and count unigram frequency of each token in each language's corpus to estimate a categorical distribution over the vocabulary. Then we sum log-likelihood ratio between FR and EN over the tokens in a sentence. A positive value means the sentence is likely French and vice versa. We test that held-out monolingual sentences show $\sim 99\%$ accuracy with a 10-token span using the classifier. We report the generation rate of the removed language during fine-tuning on both 1) the first sentence from generation and 2) the entire generated sentences.

Representational similarity. We subsample $6k$ single sentences from $6k$ unseen test passages. For the shuffled incoherent sentences, we keep the first and the last words of the sentence but shuffle the order of the remaining words, to minimize the effect due to a difference in the last word. We take the hidden representation after the final layer norm at the last word token of the sentence, center the hidden representation coordinate-wise and compute the absolute cosine similarity of the representations from coherent and shuffled sentences in each language.

Prediction loss. We used $6k$ unseen test passages in each language that consist of more than 3 sentences. We feed in all preceding sentences of each passage as a context either in EN monolingual or FR monolingual, but not the last target sentence. Then, we measure the cross-entropy loss on the first token of the target sentence either in EN or FR. This metric is a reflection of the three test families of the prediction loss (Figure 1(f)). Assuming we fine-tune with EN: *kept rule* is equivalent to measuring loss on EN target given EN context, *deleted rule* is equivalent to measuring loss on FR target given FR context, and *deleted in-context* measures loss on EN target given FR context.

C.3 Additional results on bilingual pretrained–monolingual fine-tuned language models

In the main text (Section 5), we only show the results from the EN-fine-tuned model. We report the FR-fine-tuned model (Figure 12) and also the models trained without weight decay (Figures 13 and 14), which show a consistent trend.

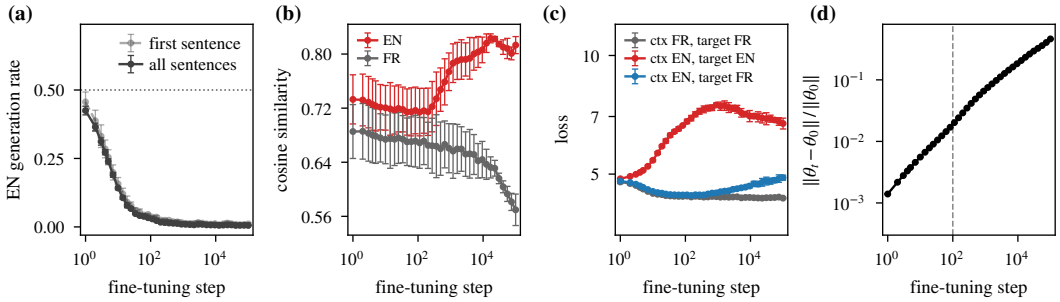


Figure 12: The measurements of Figure 4 for French-only fine-tuning. **(a)** English generation rate. **(b)** Coherent–shuffled representation similarity for English (red) and French (gray). **(c)** Prediction loss for French targets with French context (gray), English targets with English context (red), and French targets with English context (blue). **(d)** Relative parameter distance, as in Figure 4(d).

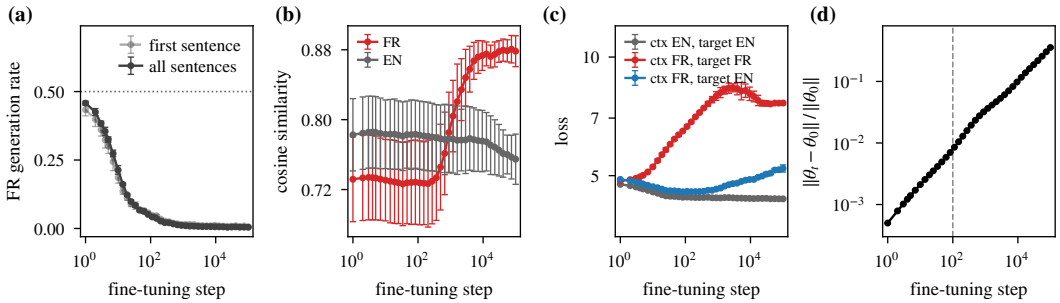


Figure 13: Bilingual pretraining and English-only fine-tuning without weight decay. Panels (a) to (d), colors and reference line as in Figure 4: French generation, coherent–shuffled representation similarity, prediction loss and relative parameter distance.

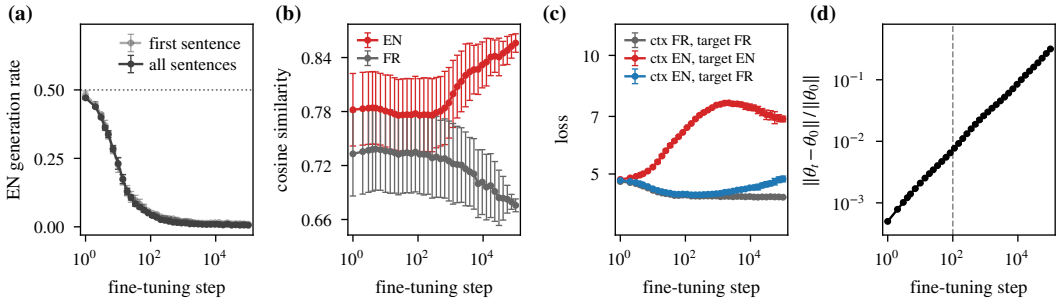


Figure 14: Bilingual pretraining and French-only fine-tuning without weight decay. Panels (a) to (d), colors and reference line as in Figure 12: English generation, coherent–shuffled representation similarity, prediction loss and relative parameter distance.