Convergence and Divergence of Language Models under Different Random Seeds

Finlay Fehlauer,[ETH] **Kyle Mahowald,** **Tiago Pimentel** [ETH]

[ETH] ETH Zürich University of Texas at Austin
ffehlauer@ethz.ch, mahowald@utexas.edu, tiago.pimentel@inf.ethz.ch
Tr1ple-F/convergence-and-divergence-of-llms

Abstract

In this paper, we investigate the convergence of language models (LMs) trained under different random seeds, measuring convergence as the expected per-token Kullback–Leibler (KL) divergence across seeds. By comparing LM convergence as a function of model size and training checkpoint, we identify a four-phase convergence pattern: (i) an initial **uniform phase**, (ii) a **sharp-convergence phase**, (iii) a **sharp-divergence phase**, and (iv) a **slow-reconvergence phase**. Further, we observe that larger models reconverge faster in later training stages, while smaller models never actually reconverge; these results suggest that a certain model size may be necessary to learn stable distributions. Restricting our analysis to specific token frequencies or part-of-speech (PoS) tags further reveals that convergence is uneven across linguistic categories: frequent tokens and function words converge faster and more reliably than their counterparts (infrequent tokens and content words). Overall, our findings highlight factors that influence the stability of the learned distributions in model training.

1 Introduction

At their core, **language models** (LMs) are distributions over strings, $p_\theta(\mathbf{s})$, trained to approximate a **data-generating distribution** $p(\mathbf{s})$. Their massive improvements in recent years—typically attributed to increasing data, compute, and architecture size (Kaplan et al., 2020; Henighan et al., 2020)—suggests that LMs are getting ever more similar to this data-generating distribution. Notably, if LMs could perfectly fit this data-generating distribution p, they would all converge to the same p_θ.[1]

In practice, however, this convergence might: (i) not happen uniformly for all contexts; (ii) not happen at all for some contexts. This is the focus of our study: the convergence (and potential divergence) of LMs across scales, training, and contexts.

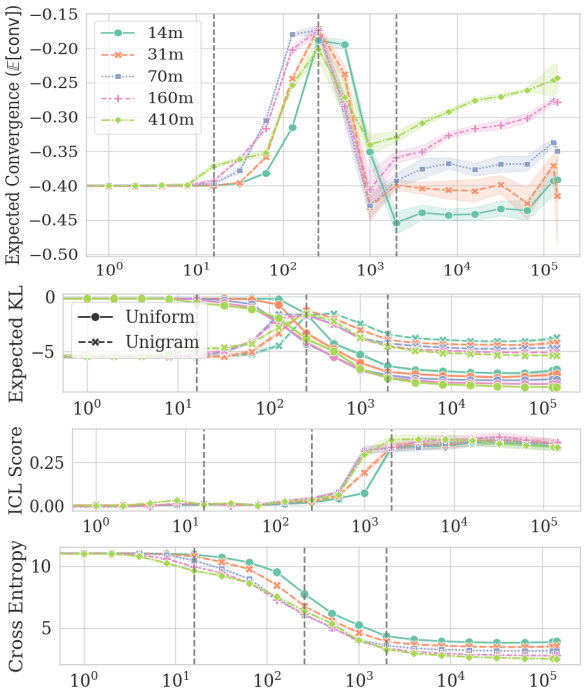

Figure 1: Estimated $\mathbb{E}[\texttt{conv}]$ across training steps (x-axis). Shaded areas represent 1σ confidence intervals.[2]

Given the scientific and engineering import of language models' training dynamics, a large body of work has examined it (Saphra and Lopez, 2019; Wei et al., 2022; Chen et al., 2024; van der Wal et al., 2025, *inter alia*). We highlight two important previous findings here. First, early in training, LMs reach a **unigram-output stage**, outputting a mostly context-agnostic distribution which matches word frequencies; only afterward, they start leveraging context (Chang and Bergen, 2022; Chang et al., 2024; Belrose et al., 2024). Second, and also early in training, transformers go through **induction-head formation**, which enables in-context learning (Olsson et al., 2022; Tigges et al., 2024).

[1] See Huh et al. (2024) for an even stronger claim: models not only converge in distribution, but all models across modalities will converge to true "platonic representations".

[2] ICL scores were measured as the expected difference in surprisal of a sentence's 500$^{\text{th}}$ token when: conditioned on the preceding 50 tokens of context vs. on 400 tokens.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32971–32980

Prior work has also shown that different aspects of input are learned at different rates. Lexical learning studies, for instance, show that words with different PoS are acquired at different rates (Chang and Bergen, 2022; Ficarra et al., 2025), suggesting token-specific convergence dynamics. Relatedly, Evanson et al. (2023) show that sentences with more complex structures are learned slower.

As mentioned above, we wish to analyse the convergence of language models here. To this end, we first define LMs' **convergence** as the negative expected Kullback-Leibler divergence of a model when trained under different seeds. Relying on this metric, we empirically find that larger models do not simply achieve stronger final convergence, but that convergence happens faster on them. However, we see that convergence is not monotonic throughout training (see Fig. 1, top). After a short initial **uniform phase**, there is a **sharp-convergence phase**; interestingly, this convergence phase coincides with the unigram-output stage found by prior work (Fig. 1, mid-top). Afterwards, models follow a **sharp-divergence phase**, where they start learning to use context. Finally, we see a **slow-reconvergence phase**, in which model predictions seem to stabilise and (at least for larger models) slowly reconverge to a unified solution; interestingly, the transition to this final phase seems to coincide with induction-head formation (Fig. 1, mid-bottom). Notably, these four phases happen while the models monotonically improve (Fig. 1, bottom); multiple seeds of the same model p_θ may thus get less similar to each other while simultaneously becoming more similar to the target distribution p.

Additionally, as in Chang et al. (2024), we study how convergence differs depending on the frequency, part of speech, or final surprisal of a predicted token. To this end, we define **LM conditional convergence** similar to LM convergence, but conditioning the expectation on a feature of the text (e.g., the target word being a noun). Using this metric, our analyses show that, while models' outputs seem to converge when predicting frequent or function words, their final-step convergence on other tokens may be worse than at initialisation.

2 Convergence and Divergence

In our study, we will measure convergence by analysing whether different models output similar probability distributions. To that end, we first assume there exists a distribution over model parameters $p(\boldsymbol{\theta})$, induced by a choice of architecture and the optimisation process. In other words, $p(\boldsymbol{\theta})$ represents a distribution over models trained under different random seeds. Given this distribution, we define convergence as:

Definition 1. *We quantify **convergence** in context $\mathbf{s}_{<t}$ as the negative expected divergence between two models $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ sampled from this distribution:*

$$\texttt{conv}(\mathbf{s}_{<t}) = \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{\theta}'}\Big[-d_{\mathbf{s}_{<t}}(p_{\boldsymbol{\theta}}, p_{\boldsymbol{\theta}'})\Big] \tag{1}$$

In theory, we could use any divergence function as d. Here, we will measure it as the Kullback-Leibler (KL) divergence:[3]

$$\begin{aligned} d_{\mathbf{s}_{<t}}(p_{\boldsymbol{\theta}}, p_{\boldsymbol{\theta}'}) &= \mathrm{KL}\Big(p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s}_{<t}) \mid\mid p_{\boldsymbol{\theta}'}(\cdot \mid \mathbf{s}_{<t})\Big) \\ &= \sum_{s \in \mathcal{S}} p_{\boldsymbol{\theta}}(s \mid \mathbf{s}_{<t}) \log \frac{p_{\boldsymbol{\theta}}(s \mid \mathbf{s}_{<t})}{p_{\boldsymbol{\theta}'}(s \mid \mathbf{s}_{<t})} \end{aligned} \tag{2}$$

An increase in $\texttt{conv}(\mathbf{s}_{<t})$ thus indicates convergence, while a decrease in this value indicates divergence. We chose the KL as it is a standard measure for comparing probability distributions. In practice, analysing LM convergence for each specific token–context pair can be challenging, and we thus define a global measure of convergence using its expectation.

Definition 2. *We quantify **expected convergence** as the expectation of convergence across contexts:*

$$\mathbb{E}[\texttt{conv}] = \mathop{\mathbb{E}}_{\mathbf{s}_{<t}}\Big[\texttt{conv}(\mathbf{s}_{<t})\Big] \tag{3}$$

Notably, while expected convergence gives an overall notion of how convergence behaves across a dataset, it can hide variations in convergence depending on the context and target token. To address this, we take inspiration from Chang et al.'s (2024) analyses, defining conditional convergence. Conditional convergence measures a model's expected convergence conditioned on a specific property.

[3] Prior work quantifies the convergence of LMs by computing the correlation across seeds in models' predictions (operationalised as, e.g., per-token surprisal, or a downstream task's outputs; Chang et al., 2024; van der Wal et al., 2025). We believe these correlations may hide nuances which $\mathbb{E}[\texttt{conv}]$ captures. E.g., two randomly initialised models (which output noisy uniform distributions), output per-token surprisals with near-zero correlations, independently of how close to uniform both their distributions are. Relatedly, two unigram language models which differ only in their temperature would output surprisals with near-one correlations.

Definition 3. *Let $\mathcal{S}_t$ be a set of tokens which have a specific target property to be conditioned on. We quantify a model's* ***conditional convergence*** *as the expectation of convergence across tokens and contexts which have this property:*

$$\mathop{\mathbb{E}}_{\mathcal{S}_t}[\texttt{conv}] = \mathop{\mathbb{E}}_{\mathbf{s}_{\leq t}}\Big[\texttt{conv}(\mathbf{s}_{<t}) \mid s_t \in \mathcal{S}_t\Big] \quad (4)$$

By computing conditional convergence for different token categories (e.g., nouns, verbs, function words), we can analyse how convergence varies across different tokens.

3 Experimental Setup

We now present the main choices made for our experiments. See App. A for more details.

Model. We analyse language models from the (Poly)Pythia suite here (Biderman et al., 2023; van der Wal et al., 2025).[4] For each model size, these LMs' architecture and optimisation procedure induce a distribution $p_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ over parameters, which we use in our definition of convergence (in Definition 1). This suite contains a set of 10 independently trained models per model size, simulating the set of samples $\boldsymbol{\theta}, \boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$ we need to estimate conv (in Eq. (1)). Furthermore, we analyse models with $\{14m, 31m, 70m, 160m, 410m\}$ parameters—the sizes available in PolyPythia—at logarithmically-spaced training steps: $\{0, 1, 2, 4, 8, ..., 512, 1k, 2k, 4k, ..., 128k\}$. Due to computational restrictions, we selected seeds $1, 3, 5, 7, 9$ for our analyses (ignoring other seeds).

Data. For our analyses, we use a subset of the Pile's validation set (Gao et al., 2020) covering 4,662 tokens; these tokens' contexts form a dataset: $\mathcal{D} = \{\mathbf{s}_{<t}^{(n)}\}_{n=1}^{N}$. We assume this data is sampled from the data-generating distribution $\mathbf{s}_{<t}^{(n)} \sim p(\mathbf{s}_{<t})$, allowing us to compute an unbiased estimate of expected convergence (in Definition 2).

Conditioning Properties. Finally, we also analyse how models converge while controlling for three properties: a token s_t's frequency, its part-of-speech (PoS), and its final-surprisal. We estimate tokens' frequencies by counting them on the Pile's validation set. We estimate PoS using the NLTK part-of-speech tagger (Bird and Loper, 2004).[5] We compute final-surprisal by—for a specific model size—using its last checkpoint to compute each token's surprisal: $-\log p_{\boldsymbol{\theta}}(s_t \mid \mathbf{s}_{<t})$. To estimate conditional convergence (Definition 3), we then define $\mathcal{S}_t$ using either log-spaced frequency or final-surprisals bins, or PoS classes.

4 Four Phases of Expected Convergence

Fig. 1 (top) presents our estimates of the expected convergence, i.e., $\mathbb{E}[\texttt{conv}]$, for models of different sizes and across training steps. In this figure, we see that convergence progresses across training in four clearly distinct phases.

Uniform Phase. This initial phase is roughly observed until step 16 and reflects a shared starting point of training, with models of different sizes presenting similar convergences. This has a simple explanation. As shown in Fig. 1 (mid-top), all LMs' outputs start similar to the uniform distribution, which is enforced by their parameters' initialisation. Interestingly, there is almost no change in convergence during this phase, which may be explained by the small learning rates used at these steps. (For convenience, we present Pythias' learning rates across training in Fig. 6 in App. B).

Sharp-convergence Phase. This second phase is roughly observed between steps 16 and 256, being characterised by a sharp increase in model similarity. Interestingly, as can be seen in Fig. 1 (mid-top), this phase corresponds quite clearly to a shift in LMs from mimicking a uniform to a unigram distribution.[6] As shown by prior work, these are also the training steps at which models' predictions (either in terms of surprisal, or downstream tasks outputs) seem to have maximal correlations with one another (Chang et al., 2024; van der Wal et al., 2025).

Sharp-divergence Phase. This third phase of training occurs between steps 256 and 2k, being characterised by a sharp decrease in model similarity. Interestingly, this phase coincides with the moment LMs start diverging from the unigram distribution, implying that, at least initially, LMs use of context differs significantly across seeds. Notably, the cross-entropy of Pythia models decreases monotonically throughout training (see Fig. 1, bottom), making such a sharp-divergence phase surprising: these LMs seem to monotonically approximate p, but each does so in a different way.

[4] We also present similar results for MultiBERT in §4.1.

[5] We detail how we convert from word- (output by NLTK) to subword-level tags (Pythia's tokens) in App. A.

[6] As discussed in §1, Chang and Bergen (2022) originally reported this unigram-output stage of LM learning.

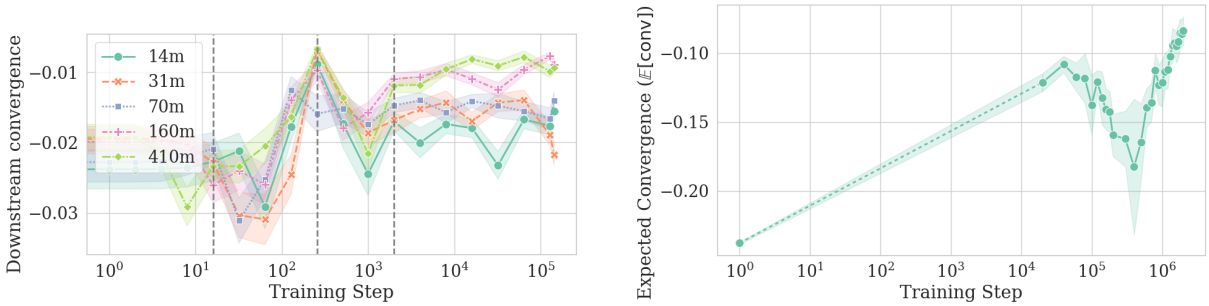

Figure 2: Estimated $\mathbb{E}[\texttt{conv}]$ across training steps: (left) on the Pythia model suite on BLiMP with $1\bar{\sigma}$ confidence intervals, (right) on the MultiBERT model suite on masked language modelling with 1σ confidence intervals.

Slow-reconvergence Phase. This final phase starts around step 2k, and is characterised by a slow increase in model similarity. Interestingly, as can be seen in Fig. 1 (mid-bottom), the step at which this phase starts coincides with an increase in in-context learning (ICL) scores (Olsson et al., 2022) and thus to induction-heads formation. This suggests that induction heads may not only enable in-context learning in large models, but also stabilise the training of transformer-based LMs. Further, the steepness of this final downward trend depends on model size, with larger models converging faster than smaller ones. In fact, for the smallest models, model convergence seems to mostly stabilise at this point, and they end training with higher $\mathbb{E}[\texttt{conv}]$ than they begin with, implying that these models do not in fact converge to a shared solution.

4.1 Convergence on Other Tasks and Models

We now assess whether model convergence dynamics are similar in: (i) downstream tasks, conducting experiments on BLiMP (Warstadt et al., 2020); (ii) other models, conducting experiments with the MultiBERT model suite (Sellam et al., 2022).

Convergence on Downstream Tasks. BLiMP is a dataset composed of pairs of grammatical and ungrammatical sentences, $\mathcal{D} = \{\mathbf{s}_{\checkmark}^{(n)}, \mathbf{s}_{\text{✗}}^{(n)}\}_{n=1}^{N}$, and whose task is to identify the grammatical one. For each of these pairs, models are then evaluated on whether they place more probability on $\mathbf{s}_{\checkmark}$ than on $\mathbf{s}_{\text{✗}}$. In this task, we thus restrict the support of our models' distribution to these two sentences:

$$\widehat{p_\theta}(\mathbf{s}_{\checkmark}) = \frac{p_\theta(\mathbf{s}_{\checkmark})}{p_\theta(\mathbf{s}_{\checkmark}) + p_\theta(\mathbf{s}_{\text{✗}})}, \quad \widehat{p_\theta}(\mathbf{s}_{\text{✗}}) = \frac{p_\theta(\mathbf{s}_{\text{✗}})}{p_\theta(\mathbf{s}_{\checkmark}) + p_\theta(\mathbf{s}_{\text{✗}})}$$

We then use this limited-support distribution to compute models' convergence as in §2, but with:

$$d_n(p_\theta, p_{\theta'}) = \sum_{\mathbf{s} \in \{\mathbf{s}_{\checkmark}^{(n)}, \mathbf{s}_{\text{✗}}^{(n)}\}} \widehat{p_\theta}(\mathbf{s}) \log \frac{\widehat{p_\theta}(\mathbf{s})}{\widehat{p_\theta}'(\mathbf{s})} \quad (5)$$

We present these downstream convergence results in Fig. 2 (left). This figure reveals that the downstream convergence pattern broadly mirrors the phases observed before, supporting the hypothesis that these training dynamics manifest at the task-level as well.

Convergence on Masked LMs. The MultiBERT model suite is composed of bidirectional transformers—as opposed to Pythia's autoregressive models—and allows us to evaluate whether convergence dynamics are similar in such masked language models. Unfortunately, the available checkpoints do not include steps between 0 and 20k, which prevents us from observing (i) an initial **uniform phase** or (ii) a **sharp-convergence phase**. However, the remaining dynamics appear consistent with the results on Pythia, including: (iii) a **sharp-divergence phase** followed by (iv) a **slow-reconvergence phase** (see Fig. 2, right).

5 Conditional Convergence

We now analyse conditional convergences: i.e., how convergence changes as a function of different contextual properties of a token. These results are presented in Fig. 3, where conditional convergences are presented for tokens based on either frequency, PoS, or final-surprisal.[7] Notably, for all these conditioning properties, the two initial phases of training (the uniform and sharp-convergence phases) present similar trends. This is likely because until the third phase of training (the sharp-divergence phase), LMs are not using context to make predictions. We will thus focus on the third and fourth convergence phases here.

Frequency. Fig. 3 (left) presents the conditional convergence $\mathbb{E}_{\mathcal{S}_t}[\texttt{conv}]$ for tokens in varying fre-

[7] We note that, within any of these categories—similarly to the general case– LMs present (almost) monotonically decreasing cross-entropy curves (see Fig. 9 in App. E).

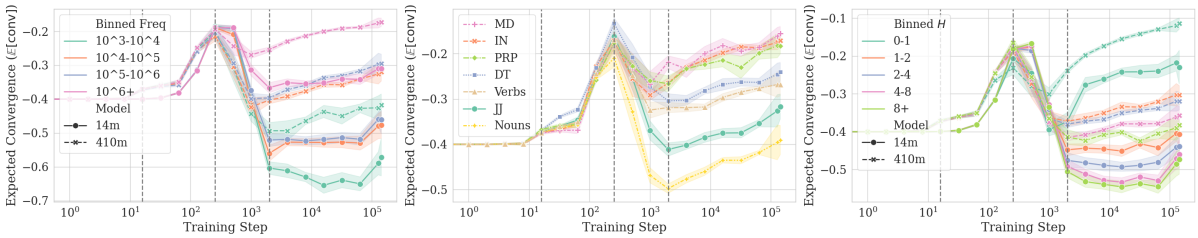

Figure 3: $\mathbb{E}_{\mathcal{S}_t}[\texttt{conv}]$ of selected models with 1σ confidence intervals. Conditioning property: (left) frequency; (center) parts of speech; (right) final surprisal.

quency bins. During the third training phase, frequent tokens' convergence stabilises at a relatively small value. In contrast, infrequent tokens fully suffer from a sharp-divergence, highlighting their greater variability and sensitivity to random seed. Interestingly, final convergence in infrequent tokens is smaller than initial convergence, suggesting LMs diverge on these tokens across training.

PoS Tags. Fig. 3 (center) presents the conditional convergence $\mathbb{E}_{\mathcal{S}_t}[\texttt{conv}]$ for tokens with varying PoS tags. This figure shows that content words (nouns, adjectives (JJ) and verbs) diverge more once entering the sharp-divergence phase than function words (determiners (DT), personal pronouns (PRP), prepositions or subordinating conjunctions (IN), and modal auxiliary words (MD)). Furthermore, function words achieve higher final convergence, whereas content words are more divergent.[8]

Final Surprisal. Fig. 3 (right) presents the conditional convergence $\mathbb{E}_{\mathcal{S}_t}[\texttt{conv}]$ for tokens within varying final-surprisal bins. This figure reveals that a token's convergence may be fairly different depending on how predictable it is. In particular, this figure shows that tokens with very low final-surprisal show strong convergence by the end of training. Convergence behaviour across tokens with higher final-surprisals (from 1 to 8+ bits), however, seems quite similar, with final-surprisal thus not greatly impacting convergence for these tokens.

5.1 Variance in Convergence across Tokens

Finally, we also analyse the variance of $\texttt{conv}(\mathbf{s}_{<t})$ across contexts $\mathbf{s}_{<t}$ throughout training. Fig. 4 presents these results. Interestingly, this figure shows that the initial two phases have very little variance in convergence across contexts $\mathbf{s}_{<t}$. The final two phases, however, present a significant increase in this variance. This again highlights the non-contextual nature of the two initial convergence phases.

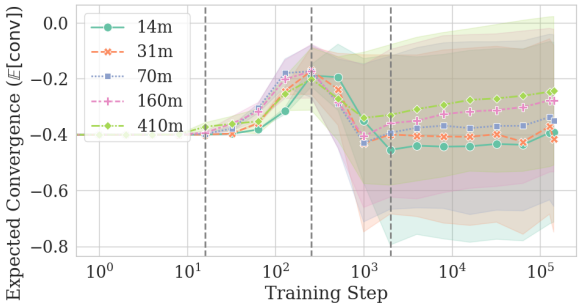

Figure 4: $\texttt{conv}(\mathbf{s}_{<t})$ across training, with shaded areas representing its standard across contexts $\mathbf{s}_{<t}$.

6 Conclusion

Our analyses reveal that convergence in language models is far from uniform. While global metrics like cross-entropy steadily improve, they conceal substantial variation in how individual tokens and contexts behave across training. We find that, across training, LMs' convergence goes through four phases, each with distinct characteristics. Additionally, we find that token frequency plays a dominant role in convergence: frequent tokens converge quickly and consistently, whereas rare tokens often diverge. Similarly, convergence is strongly shaped by linguistic features: function words exhibit stable predictions, while content words remain volatile (a result reminiscent of Chang and Bergen, 2022). Finally, we find that larger models tend to converge more consistently.

Limitations

Our analysis is limited in several ways. First, our analysis of a $\texttt{conv}(\mathbf{s}_{<t})$ measure based on the token-level KL prevents us from comparing different model families that rely on different tokenisers, as different distribution supports in $p_\theta(s_t \mid \mathbf{s}_{<t})$

[8] See App. C for a detailed analysis of how convergence changes across subclasses of nouns and verbs. Further, as different conditioning properties may be correlated (function words are typically frequent), we present a linear regression analysis of $\texttt{conv}(\mathbf{s}_{<t})$ in App. D to jointly analyse these properties impact; this analysis supported our main results here.

would prevent us from calculating the KL divergence. This could potentially be mitigated by converting these distributions to the byte- or word-level (Pimentel and Meister, 2024; Phan et al., 2025), but we leave that for future work. Second, due to computational constraints, our experiments were conducted on a small subset of the Pile's validation set. Third, our analysis is restricted to English-language data, leaving open questions about whether similar convergence dynamics occur in other languages and in multilingual settings. Finally, the presence of learning rate warm-up phases during early training, where we observe the most rapid shifts in model behaviour, may introduce artifacts that affect our interpretation of convergence and divergence.

Acknowledgements

We would like to thank Pietro Lesci for generously providing us with helpful references and feedback on earlier versions of this work. The authors also thank Prof. Yoon Kim for his valuable input and for providing computational resources. We thank Prof. Thomas Hofmann for supporting this research in his lab and for financing the conference participation, as well as for his broader guidance during the thesis work. Kyle Mahowald acknowledges funding from NSF CAREER grant 2339729.

References

Nora Belrose, Quintin Pope, Lucia Quirke, Alex Mallen, and Xiaoli Fern. 2024. Neural networks learn statistics of increasing complexity. *Preprint*, arXiv:2402.04362.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2024. Characterizing learning curves during language model pre-training: Learning, forgetting, and stability. *Transactions of the Association for Computational Linguistics*, 12:1346–1362.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*.

Linnea Evanson, Yair Lakretz, and Jean Rémi King. 2023. Language acquisition: do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218, Toronto, Canada. Association for Computational Linguistics.

Filippo Ficarra, Ryan Cotterell, and Alex Warstadt. 2025. A distributional perspective on word learning in neural language models. *Preprint*, arXiv:2502.05892.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. 2020. Scaling laws for autoregressive generative modeling. *Preprint*, arXiv:2010.14701.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. *Preprint*, arXiv:2405.07987.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Buu Phan, Brandon Amos, Itai Gat, Marton Havasi, Matthew J. Muckley, and Karen Ullrich. 2025. Exact byte-level probabilities from tokenized language models for FIM-tasks and model ensembles. In *The Thirteenth International Conference on Learning Representations*.

Tiago Pimentel and Clara Meister. 2024. How to compute the probability of a word. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18358–18375, Miami, Florida, USA. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. LLM circuit analyses are consistent across training and scale. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

A Detailed Experimental Setup

In this section, we expand on experimental choices used to estimate convergence.

Choice of $p(\boldsymbol{\theta})$ (Or, Analysed Model Architecture and Optimisation Process). Our definition of convergence (in Definition 1) relies on a distribution over models $p(\boldsymbol{\theta})$, which is itself induced by a choice of model architecture and optimisation process. Here, we will use the distribution $p(\boldsymbol{\theta})$ induced by Pythia's training process (Biderman et al., 2023); see their paper for details. The Pythia model suite includes model architectures of different sizes, and we analyse here models with $\{14m, 31m, 70m, 160m, 410m\}$ parameters.[9] Further, 154 checkpoints were released for each of these model sizes, allowing us to analyse how convergence evolves across training. Here, we analyse training steps at logarithmically-spaced intervals; specifically, we analyse checkpoints: $\{0, 1, 2, 4, 8, ..., 512, 1k, 2k, 4k, ..., 128k\}$.

Estimating Convergence (Or, Analysed Model). To compute $\texttt{conv}(\mathbf{s}_{<t})$ we need not only to choose a distribution $p(\boldsymbol{\theta})$, but to take an expectation over it. This is infeasible for large language models. We can, however, estimate $\texttt{conv}(\mathbf{s}_{<t})$ using pairs of independently sampled models $\boldsymbol{\theta}, \boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$. Luckily, van der Wal et al. (2025) recently presented the PolyPythia model suite, an extension of the original Pythia model suite with multiple trained models—using different randomisation seeds—for each model size. We treat each pair of models in the PolyPythia suite as a sample $\boldsymbol{\theta}, \boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$, which we use to estimate convergence. Due to computational restrictions, we selected seeds $1, 3, 5, 7, 9$ for our analyses (ignoring the other 5 seeds).

Estimating Expected Convergence (Or, Analysed Data). To compute $\mathbb{E}[\texttt{conv}]$, we must take an expectation over contexts $\mathbf{s}_{<t} \sim p(\mathbf{s}_{<t})$, which is again computationally infeasible. To avoid this issue, we use a data set $\mathcal{D} = \{\mathbf{s}_{<t}^{(n)}\}_{n=1}^{N}$ of samples which we assume to be drawn from the true distribution $p(\mathbf{s}_{<t})$; this allows us to compute an unbiased estimate of expected convergence. More specifically, we used samples from the Pile validation set (Gao et al., 2020) that covered the 4662 tokens.

[9] We restrict our analyses to these model sizes, as only those are covered by the PolyPythia model suite, whose relevance we expand on in the next paragraph.

Choice of $\mathcal{S}_t$ (Or, Analysed Token Property). Our definition of conditional convergence (Definition 3) relies on a choice of token property we which to condition on. Here, we will consider three such properties: a token s_t's frequency, its part-of-speech (PoS), and its final surprisal. We estimate a tokens' frequency by counting the number of times it appears on our dataset $\mathcal{D}$; we then define log-spaced bins $\mathcal{S}_t$ which we use to analyse tokens within those frequencies. We estimate a token's PoS tag using the NLTK part-of-speech tagger (Bird and Loper, 2004). Since our dataset is primarily in English, we use the standard PoS tags for this language. However, because NLTK and Pythia employ different tokenisation methods, we implement a mapping procedure to align PoS tags with the tokenised outputs. First, PoS tags are assigned at the word level using NLTK. These tags are then mapped to individual characters in the raw text. Finally, after tokenisation, each token inherits the tag that corresponds to the majority of its characters. Tokens without a majority label or with an "UNK" (unknown) label are excluded from our analysis. This process is illustrated in Fig. 5:

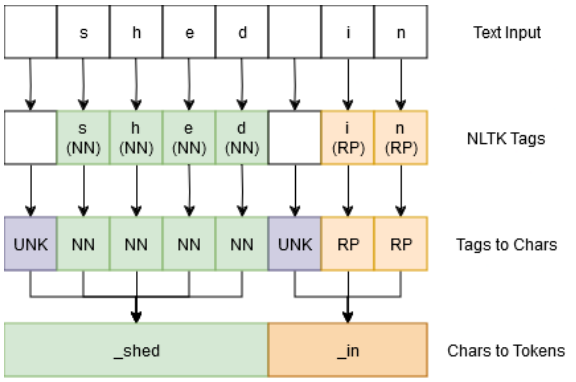

Figure 5: Illustration of the mapping of part of speech tags to tokens.

Finally, we estimate a tokens' final surprisal by, for each analysed model, using the model's final checkpoint to estimate the token's surprisal; similarly to our frequency analysis, we then define log-spaced bins $\mathcal{S}_t$ which we use to analyse tokens within those surprisal ranges.

B Pythias' Learning Rates

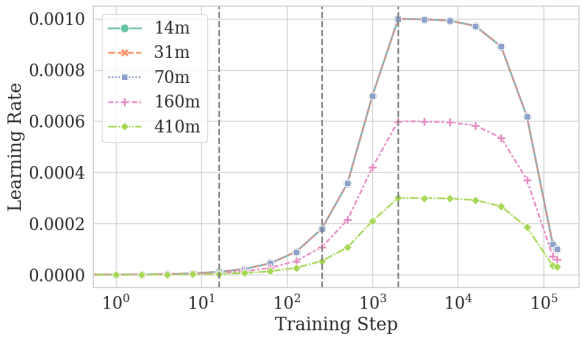

Figure 6: Learning rates for the different Pythia models.

C Conditional Convergence on Nouns and Verbs PoS

Differences in Nouns. Fig. 7 (left) shows $\mathbb{E}_{\mathcal{S}_{\text{noun}}}[\texttt{conv}]$ for the $410m$ parameter model. All three noun types—regular singular (NN), regular plural (NNS) and proper singular (NNP)—are roughly equally challenging for the model and their final convergence values are either similar or lower than the starting point at training step 0.

Differences in Verbs. Fig. 7 (right) shows the conditional convergence for the tokens where a verb is being predicted. Unlike nouns, verbs generally converge more throughout training. However, gerund or present participle (VBG) and past participle verbs (VBN) converge less than 3rd person singular present (VBZ), past tense (VBD) and base form verbs (VB). The latter might be simpler forms that could be more easily derived from context.

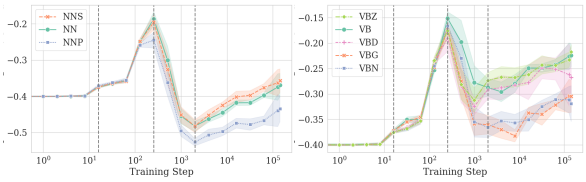

Figure 7: $\mathbb{E}_{\mathcal{S}_t}[\texttt{conv}]$ of selected models with 1σ confidence intervals. Conditioning property: (left) Nouns' PoS; (right) Verbs' PoS.

D Predicting Convergence with a Linear Regression

To explore which factors influence convergence the most, we follow (Chang et al., 2024) in performing a linear regression analysis. Our model consists of fitting the equation:

$$\begin{aligned}\texttt{conv}(\mathbf{s}_{<t}) \approx \alpha \cdot \log(\text{freq}(s_t)) \\ + \sum_{p \in \text{P.o.S. tags}} \beta_p \cdot \mathbf{1}_{\text{tag}(s_t)=p} \\ + \sum_{p \in \text{P.o.S. tags}} \gamma_p \cdot \mathbf{1}_{\text{tag}(s_{t-1})=p} \\ + \sum_{m \in \text{Model Sizes}} \delta_m \cdot \mathbf{1}_{m'=m}\end{aligned} \tag{6}$$

across contexts $\mathbf{s}_{<t} \in \mathcal{D}$. We fit one such model per training step for all analysed model sizes m'. This allows us to analyse the influence of our different conditional parameters across the training process.

Frequency (measured by the fitted parameter α) emerges as a significant factor influencing KL divergence, exhibiting a distinct pattern over the course of training (see Fig. 8, left). Initially, its influence is negligible, remaining close to zero. Around step 256, frequency begins to play a stronger role, positively influencing conv ($\alpha \approx 0.05$), indicating that frequent tokens stabilise earlier and exhibit lower KL divergence across random seeds. Beyond this point, the correlation slightly increases again, suggesting that while frequent tokens converge more quickly, their stabilisation process becomes less distinct as training progresses. This trajectory aligns with our observations of token convergence by frequency in Fig. 3.

The influence of model size on conv initially shows an unstable pattern, but later a trend emerges where larger models increase conv, while smaller models decrease conv relatively (see Fig. 8, right). It should be noted that while the learning rate is the same for the 3 smallest models, their influence on convergence is not. This suggests that the learning rate warm-up (see Fig. 6) is not the only determinant of these model convergence patterns.

E Conditional Cross-entropies

Fig. 9 presents conditional cross-entropies for our analysed models and conditioning properties.

F Tokens ranked by KL

Tab. 1 provides a list of tokens with high, low and medium KL respectively. We sampled the tokens exclusively from the natural language text and ignored the code snippet in our sample.

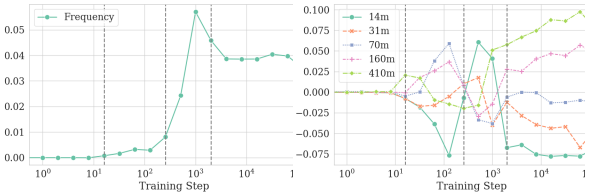

Figure 8: Linear regression's coefficients for frequency (left) and model size (right) when predicting convergence: conv($\mathbf{s}_{<t}$).

G Results Conditioning on Context Token

Fig. 10 shows the $\mathbb{E}_{\mathcal{S}_{t-1}}[\texttt{conv}]$ of the $410m$ models conditioned on properties of the last token in the context (s_{t-1}); similarly to Fig. 3 for the predicted token. Similarly, Fig. 11 shows the conditional cross-entropy, conditioned on properties of the last token in context (s_{t-1}), akin to Fig. 9.

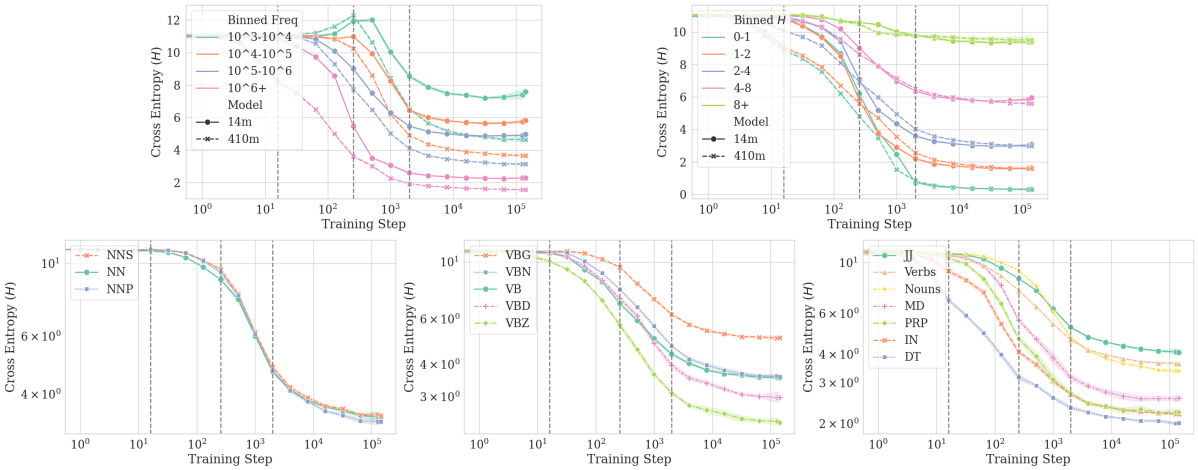

Figure 9: Conditional cross-entropy (H) of models with 1σ confidence intervals. Conditioning property: (top-left) Frequency; (top-right) Surprisal at end of training; (bottom-left) Nouns' PoS; (bottom-center) Verbs' PoS; (bottom-right) Other PoS.

Token (Low)	KL (Low)	Token (Mid)	KL (Mid)	Token (High)	KL (High)
ptions	-0.012750	)	-0.411134	OH	-1.317826
ied	-0.038621	_conv	-0.411211	ley	-1.332273
oration	-0.052642	_known	-0.411488	agn	-1.472763
isexual	-0.058059	_visual	-0.411673	hor	-1.537567
ll	-0.060047	ify	-0.411950	Ed	-1.572812
't	-0.060089	en	-0.412472	_=	-1.690981
N	-0.060991	]	-0.412764	var	-1.712604
_About	-0.065953	_founded	-0.413144	OR	-1.760247
_am	-0.066206	virtual	-0.414356	äll	-2.286940
ters	-0.067074	_whether	-0.414530	LEY	-2.874433

Table 1: Representative tokens with the lowest, highest and medium KL divergence. '_' represents a whitespace.

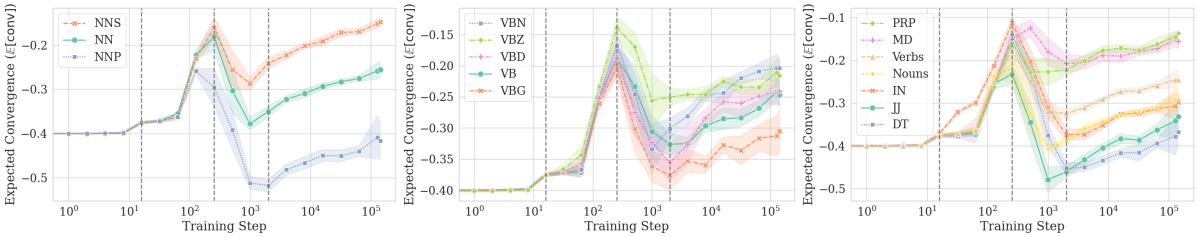

Figure 10: $\mathbb{E}_{s_{t-1}}[\texttt{conv}]$ of $410m$ models with 1σ confidence intervals. Conditioning property on context token (s_{t-1}): (left) Nouns' PoS; (center) Verbs' PoS; (right) Other PoS.

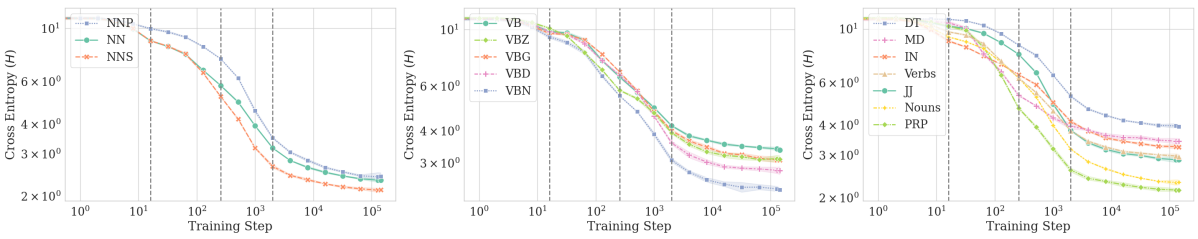

Figure 11: Conditional cross-entropy (H) of $410m$ models with 1σ confidence intervals. Conditioning property on the final context token (s_{t-1}): (left) Nouns' PoS; (center) Verbs' PoS; (right) Other PoS.