The Learning Trajectories of Individual Tokens in Language Model Training

Anonymous ACL submission

Abstract

This study investigates token learning in language models by analyzing the token-level training trajectories of three English-trained models of the OLMo architecture. Building on initial observations of training trajectories, we proceed to quantifying the relative contributions of data-driven characteristics, such as frequency, token length, and token nature to the learning process. We find that frequency is the primary factor in determining the learning time of a token, although the nature of the token also has a non-negligible effect. Conversely, surprisal at learning time is driven predominantly by token nature, with frequency playing a secondary but significant role, particularly in smaller architectures. Our findings suggest that we might be able to reduce the time a model needs to learn a token. But they also suggest that there is a limit to the learning of a token, that might be fundamentally constrained by the token's nature.

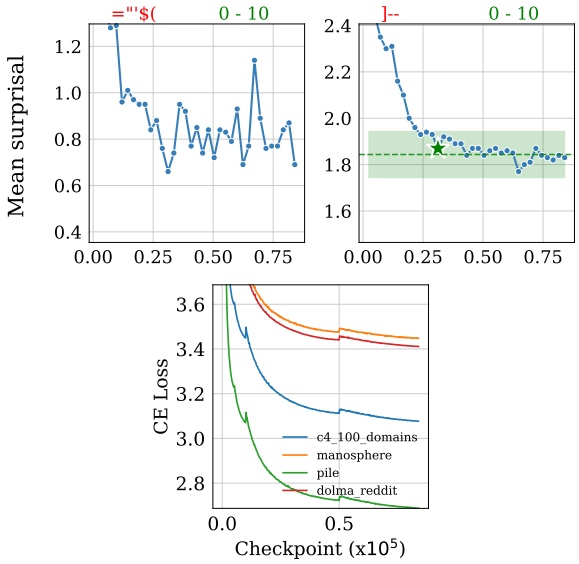

Figure 1: **Top.** Mean surprisal trajectory for a non-converging (left) and a converging (right) token during OLMo-150M training. **Bottom.** Cross-entropy loss trajectory on various validation datasets for the same model.

1 Introduction

Language models (LMs) are learners of tokens in context. However, what "learning a token" might mean for a language model remains underexplored. Previous studies show that the form and number of tokens impact model learning. Tokenization might benefit from pruning infrequent tokens (Chizhov et al., 2024), while identifying underperforming tokens (Land and Bartolo, 2024) or token training dynamics (Lin et al., 2025) might improve training. While these studies are interested in tokens primarily as means of optimizing training, they do not offer a comprehensive perspective on the variety of tokens that compose a language model's vocabulary, their characteristics, and their training dynamics.

There are two primary factors that enter into play when estimating the surprisal of natural text with a model: the model capacity and the entropy of the text, with the tokenizer acting as an intermediary. Studies on scaling laws (Kaplan et al., 2020a) aim to predict the final loss of a model based primarily on model size, data size and computational budget. In this study, we have a complementary objective: to predict per-token final loss from characteristics derived from the data, such as token frequency, length and nature, that are invariant to the model.

While the overall validation loss captures aggregate learning trends, it provides insufficient insight on how each token is learned, which is our objective. As shown in Figure 1, the losses computed over various validation datasets follow well-defined power-laws, but the individual token losses exhibit diverse behaviors. Although in the same frequency range, one token might converge before the end of training, while another might not.

We make the following hypothesis: the charac-

teristics of a token, including both intrinsic properties, such as its length, and distributional properties, such as its frequency, play a part in how a token is learned by a language model. If we can determine when the learning trajectory for a token has converged, given a token and an evaluation dataset composed of a set of contexts of that token, then training at the same frequency might be redundant for that specific token beyond the point of convergence and can be optimized.

We introduce our methodology in Section 3. To better anchor our research questions, we first perform an examination of training dynamics across a token sample in Section 3, and we then formally define our research questions in Section 4. After establishing a method to detect token convergence (Section 5), we measure the impact of frequency, token length, and token nature in Section 6. [1]

2 Related Work

Training dynamics. Interest in studying training dynamics is not restricted to LMs. Toneva et al. (2019) investigate the learning dynamics of neural network image classifiers, in order to identify which examples are forgotten over the course of learning. They draw inspiration from studies on catastrophic forgetting (Hu et al., 2019), and find that some examples seen during training are unforgettable, while others, especially those with noisy labels, are easily forgotten. Building on their work, Paul et al. (2021) propose a different method to identify difficult examples in vision model learning, which has the advantage of being applicable early in training. In LMs this time, Swayamdipta et al. (2020) examine training dynamics to identify easy, difficult and ambiguous examples in popular downstream benchmarks and aim to uncover the role of each in the overall model performance.

Memorization. Interest in memorization comes from two main areas: security and factual knowledge. Loosely, a string is memorized if the model generates it exactly as it appeared in the training data when given the same context. Carlini et al. (2023) find that larger models memorize more, and that repeated examples and longer sequences facilitate extraction of memorized data. Tirumala et al. (2022) show that larger models memorize faster and forget less. Biderman et al. (2023a) show that memorization in language models cannot be predicted from smaller or partial models. Finally, (Biderman et al., 2023b) show that training order has little impact on memorization. While these studies focus exclusively on training data, we evaluate models on in-distribution unseen contexts.

Language acquisition. Chang and Bergen (2022) inspect training trajectories in order to establish LM age of acquisition for 600 tokens. They find that frequency plays a greater role in language models than in children, and that language models seem to learn unigram and then bigram probabilities in a systematic way during training. Motivated by the language acquisition comparison, and in contrast with the present study, they restrict selection to tokens that are English words and define token learning by drawing inspiration from theories of word learning in children.

Trajectories of examples. Related studies look at token-context pairs in order to determine their trajectories during training. Xia et al. (2023) find that, early in training, the same pairs have similar trends independently of model scale, while larger models unlock new phenomena later. Chang et al. (2024) extend their study and find that more frequent token-context pairs reach lower surprisals and are less likely to be forgotten. While in these studies the focus is on token-context pairs, we look at tokens trajectories by abstracting over their contexts.

3 Methodology

Models. We analyzed the training trajectories of three models of the OLMo family: the OLMo-1B model trained and provided by Groeneveld et al. (2024), and two models that we trained using the same framework, of size 150M and 40M parameters, respectively. The OLMo-1B model was trained on a subsample of 2T tokens of the English corpus Dolma (Soldaini et al., 2024), in one pass over the (shuffled) training data. The two smaller models were trained on a prefix of the same data, scaled by their size. We have used the *Chinchilla* linear relation between training data and number of parameters (Hoffmann et al., 2022) to scale the size of the training data for the smaller models using OLMo-1B as a reference. The 150M model, approximately $8\times$ smaller than OLMo-1B, was therefore trained on $8\times$ less data. Similarly, the OLMo-40M model was trained on $12\times$ less data (details in Appendix A). Following current pre-training practices (Grattafiori et al., 2024), all

[1] Code and data can be found at https://anonymous.4open.science/r/lm-memgen-24DC/.

models were trained beyond the ∼20 token per parameter scaling law recommendation. We kept the same batch size (2048) and context size (2048) across models, to ensure training step equivalence. We scaled the architectural parameters and embedding dimensions relative to model size (1B, 150M, 40M) as follows: layers (16, 12, 8), heads (16, 12, 8) and embedding size (2048, 768, and 384). We adjusted the learning rate for the smaller models to 2×10^{-4} (from 4×10^{-4} for OLMo-1B) and the warmup (2000, 1000 and 5000, respectively) to maintain optimization stability. Validation loss curves showed minimal peaks and stable convergence plateaus across model sizes. While these configuration parameters are not necessarily optimal for each individual size, this study does not attempt to construct strict scaling laws for token convergence. Instead, our objective is to demonstrate that the mechanisms by which data-derived characteristics influence training dynamics holds across different model scales.

Tokens. The three models share a common vocabulary of 50,280 tokens. Given the significant computational cost required to evaluate the full vocabulary (1,000 contexts $\times$ 50,000 tokens $\times$ 70 checkpoints $\times$ 3 models $\approx$ 10.5T inferences), we perform our analysis on a subset of $N = 510$ tokens. This subset was constructed using a hybrid sampling strategy: a uniform random sample of $n = 255$ tokens to ensure broad coverage, supplemented by two targeted samples: one based on frequency ($n = 88$) and another based on token nature ($n = 167$). To sample by frequency, we first needed to assess the frequency of each token in a training batch. Throughout the paper "batch" refers to the global batch, consisting of 2,048 contexts of size 2,048 tokens. While we initially considered computing the frequency from the training data, an estimation we performed showed that the token frequency variation across 500 batches was low (Appendix B): 90% of tokens have a confidence interval width inferior to 5 (occurrences per batch), and for the median token, the true value of the frequency is within ± 6.5% of the estimate in 95% of cases. Given these results and the size of the training data of OLMo-1B, we determined that the cost in resources did not justify the relatively low gain in precision.

We were then able to assign the tokens to 10 balanced frequency ranges, between 0 and 1,000,000 and sample 10 tokens from each of the 10 frequency ranges, if possible, amounting to a total of 88 tokens. To sample by token nature, we defined 12 "token nature" categories and sampled from the vocabulary accordingly. These 12 categories are: *Delimiter*, *Digit*, *Letter*, *Subword*, *Punctuation*, *Punctuation prefix*, *Punctuation suffix*, *Number prefix*, *Number suffix*, *Word*, *Prefix* and *Suffix*. A description of the categories and examples are given in Appendix C. We sample up to 14 tokens for each category, amounting to 167 additional tokens. This approach aims to capture the linguistic and distributional diversity of the vocabulary while maintaining a manageable evaluation scale. Frequency, token nature and token length distributions of the token population and the sample are provided in Appendix D.

Evaluation contexts. We follow a similar methodology as Chang and Bergen (2022) to obtain the training curves: we evaluate different model checkpoints on validation datasets of 1,000 contexts per token. To obtain evaluation contexts, we use a held-out portion of the training data, consisting of 38,020 batches unseen by the selected checkpoints. From this data, we extract the prefix of the first sequence in each batch that contains the target token, thus selecting 1,000 contexts per target token. Following previous studies (Chang and Bergen, 2022; Carlini et al., 2023) we impose a minimum context length of 20 tokens, resulting in context lengths ranging from 20 to 2047. This choice is also motivated by preliminary experiments in which we observed a high variability in surprisal for contexts shorter than 20 tokens. As contexts are cut and shuffled before training, no position bias is expected (analysis in Appendix E).

Checkpoints. To create training trajectories we measure token surprisal at different checkpoints in model training. A checkpoint N is a snapshot of the model after it has seen N batches of 4M tokens. For OLMo-1B we select a subset of 49 checkpoints, uniformly spaced (every 10,000) between checkpoint 10,000 and checkpoint 700,000, except between checkpoints 117,850 and 330,000, where no checkpoints are available. The final OLMo-1B model corresponds to checkpoint 738,020, but we limit our study to checkpoint 700,000, as the training data seen after this point is used for evaluation. For the 150M model we select the 8$\times$ smaller equivalents, that is, the checkpoints between 1,200 and 84,000, every 1200 steps (70 checkpoints in total). Similarly, for the 40M model, we select the 12$\times$ smaller equivalents, that is, 70 checkpoints between 300 and 21,000, every 300 steps.

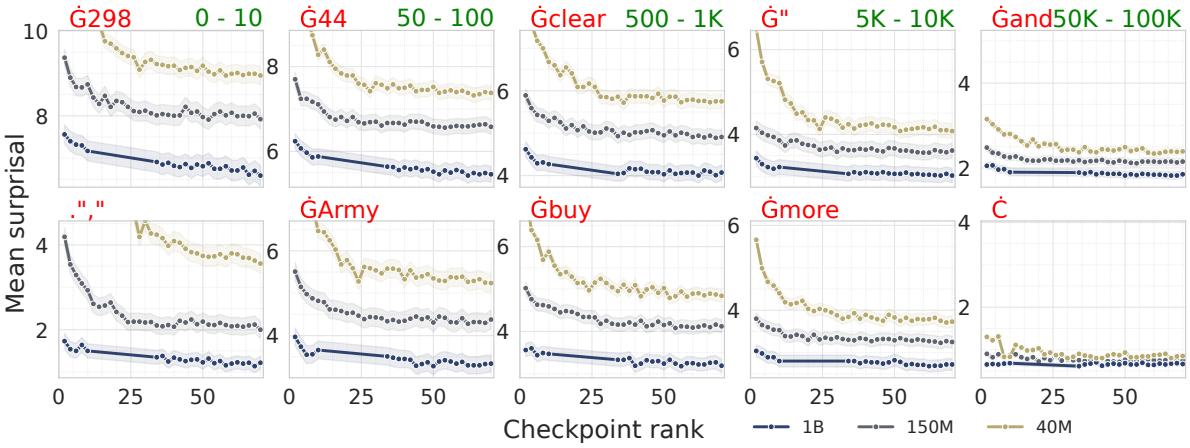

Figure 2: Training trajectories for OLMo-1B, OLMo-150M and OLMo-40M models, for two tokens (in red) per frequency range (in green). The "Ġ" character denotes a leading space in BPE tokenization. We keep Unicode placeholder characters such as "Ġ" to ensure reproducibility.

4 Token Trajectory Experiments

Examples of the learning curves obtained using the procedure described in the previous section are presented in Figure 2. We present the trajectories for 2 randomly-selected tokens in each of 5 frequency ranges (0-10 to 50,000-100,000). To compare models of different capacities, we map training checkpoints for each model onto a scale of 1 to 70, which we term *learning timepoints*. This alignment ensures that models are compared at stages of data exposure that were adjusted by capacity (number of parameters). Under this mapping, the x-axis represents the i-th timepoint rather than checkpoints. Furthermore, we reduce the visual scale of the training curves to highlight trajectory dynamics, resulting in comparable y-axis lengths and scales but differing lower and upper y-axis bounds.

The shape of the trajectories is determined by architectural configurations that we vary, such as training data and the number of model parameters, but also by optimization hyperparameters that we tune for stability, such as the learning rate and warmup schedule. While the latter certainly contribute to shaping the overall loss curve, it remains unclear to what extent these optimization choices systematically alter the loss at the individual token level. In this study we limit ourselves to the analysis of token-derived properties and leave the quantification of hyperparameter influence for future work.

High-frequency tokens converge to similar surprisals independently of model size. In terms of overall loss, smaller models generally converge to higher losses than their larger counterparts. However, on an individual token level, it is unclear whether all tokens converge to higher losses in smaller models. The tendency in Figure 2 seems to partially support this hypothesis, as token trajectories in smaller models generally start, evolve and converge at higher values of surprisal. However, very frequent tokens seem to have similar per-token loss independently of model size. We also observe that per-token surprisal at convergence does not seem completely determined by frequency. Infrequent tokens can also converge at low surprisals. This is even more visible if we consider a larger range of tokens (Appendix F), where we find that the lower bound of surprisal appears independent of frequency, while the upper bound decreases as frequency increases. This suggests that while high frequency is not a requirement for low surprisal, it acts as a constraint that limits the average surprisal a token can elicit.

The learning trajectory of a particular token does not seem consistent across model scales. We might expect that a difficult token whose trajectory does not converge in a larger model, would also fail to converge in smaller ones. However, for infrequent tokens, we observe significant idiosyncratic behavior. In some cases smaller models converge earlier than the larger one, albeit at higher surprisals, while in others, such as for very frequent tokens, the learning of a particular token does indeed seem consistent across scale.

Token curves seem to follow different power laws. Given that the overall loss follows a power curve, we might expect the same from the individual token losses. While for some tokens this seems

to be the case, for others, such as the very frequent tokens, the power-law relation is less clear. Furthermore, not all tokens that seem to follow power laws seem to follow the same one.

Building upon these preliminary observations, we address the following research questions:

- How can the time of acquisition (or convergence) of a specific token be formally defined?
- To what extent does a token's frequency and nature influence its timepoint of convergence?
- To what extent does a token's frequency and nature influence its surprisal at convergence?
- How does model scale influence token convergence time and surprisal value?

5 Token Convergence Definition

In neural network training, convergence refers to the state where an optimization process has found a stable solution. For language models, however, phenomena such as double descent (Belkin et al., 2019; Nakkiran et al., 2020) and grokking (Power et al., 2022) make convergence difficult to define. Standard criteria such as early-stopping (i.e whereby convergence is achieved when validation loss stops improving during training) are rarely used in language model training. Precursors of modern language models such as GPT-1 (Radford et al., 2018) and BERT (Devlin et al., 2019) used a fixed training budget, and while GPT-2 (Radford et al., 2019) monitored validation loss, the model remained underfit. In subsequent work, in the context of scaling laws (Kaplan et al., 2020b; Hoffmann et al., 2022), convergence was defined as an optimal relation between model parameters, training data and compute. However, modern language models were trained beyond scaling laws recommendations, as downstream task performance was observed to continue to increase (Grattafiori et al., 2024) despite overall validation loss stagnation. Therefore, current LM pre-training is not stopped by convergence criteria of the overall loss but is bounded by the amount of training data available (Villalobos et al., 2024).

We base our per-token convergence criterion on a validation loss. However, in contrast to early-stopping, the loss is not aggregated over tokens in a dataset, but computed for each token, over specific contexts of that token. Another difference from early-stopping is that we do not aim to stop model training, but rather inform it, by analyzing token trajectories and ultimately adapting training to their convergence patterns.

The overall loss trajectory is shaped among other hyperparameters by the cosine decay schedule, which progressively reduces the learning rate to its minimum threshold, allowing each model to finish training in a stable final plateau. However, as this hyperparameter is applied on the overall loss, it does not account for the differences in the trajectory of individual tokens, and cannot be used directly to determine convergence.

We define **token convergence** as follows:

The training curve of a token converges if a sequence of checkpoints can be identified such that it includes the last checkpoint and that there are no significant differences between the mean surprisal values at these checkpoints. We define the ***convergence checkpoint*** *as the first checkpoint in the sequence and the* ***convergence surprisal*** *as the mean surprisal at the convergence checkpoint.*

To identify the longest stable sequence, a backwards pairwise comparison is conducted starting from the final checkpoint. The algorithm allows for minor fluctuations as long as they are preceded by a stable plateau. To ensure that these small fluctuations do not add up to a significant gradual shift, we perform a final verification comparing the beginning to the end of the sequence to confirm they remain equivalent. A description of the algorithm is provided in Appendix G.

To establish the equivalence of the mean surprisals of two checkpoints we used a two one-sided t-test (Lakens, 2017). Contrary to a classic t-test that cannot assert equivalence, this procedure permits to establish it (within specified confidence intervals) by showing that the difference between two means is inferior to a provided threshold. This brings about the question of determining the value of the threshold that would best define practical equivalence.

We find that for all models, increasing this equivalence threshold (hereafter named SESOI for smallest effect size of interest) from 0.01 to 0.4 results in a shift from none to all tokens converging. This indicates that during late training, mean surprisal fluctuations are confined to a narrow band, which interestingly, is of the same size across the three models. In the absence of a standardized criterion for selecting a surprisal threshold, the choice of a specific value, such as 0.1 rather than 0.2, depends on the resources available and the purpose

Model	40M		150M		1B	
SESOI	0.1	0.2	0.1	0.2	0.1	0.2
λ^*	1.7	1.4	0.6	0.7	1.5	1.3

Table 1: Optimal transformations of the response variable **convergence checkpoint**.

of the model. To account for this, we adopt the notation model-x to denote token convergence for a given model under a SESOI threshold of x.

We find that the proposed algorithm is largely robust to variations in hyperparameters, specifically sequence length (L_{seq}), ending sequence (L_{end}), and granularity (Appendix H). Consequently, we set L_{end} and L_{seq} at 3 and 5, respectively, while adopting the finest available granularity corresponding to the selected checkpoints.

6 Explaining Convergence

Having identified the tokens that converge under our definition and their time of convergence, we proceed to explain convergence based on three independent variables: token estimated frequency, token length, and token nature. We perform two linear regressions, where we explain convergence checkpoint and convergence surprisal separately, as two distinct response variables. The independent variables frequency and length are treated as numerical features. The categorical feature, token nature, is modeled using dummy variables, with the category *Word* designated as the reference.

6.1 Convergence Checkpoint

Initial data inspection reveals that the relation between frequency and convergence checkpoint seems to follow a power-law, while that between token length and convergence checkpoint seems more noisy and difficult to assess visually. Given these observations, the data does not fulfill the assumptions of linear regression without transformation. To determine the transformations that would best linearize the data, we perform a grid search of transformations on the independent variables and use the Box-Cox method (Montgomery et al., 2012) to find the best transformation of the dependent variable (Appendix I).

Across all model scales and parameters, the optimal transformation for frequency is found to be the logarithm, which aligns with the broader implications of Zipf's Law (Zipf, 1949) regarding the logarithmic nature of word frequency distribution. Unlike frequency, the optimal transformation for token length is sensitive to model scale and convergence threshold. Since using non-logarithmic transformations amounted to marginal gains for the 40M and 1B models, and to ensure a consistent cross-model comparison, we adopt a unified model with log transformations for both frequency and length.

The optimal transformations for convergence checkpoint vary with model scale (Table 1). While the 40M and 1B models exhibit diminishing returns in convergence checkpoint with an increase in features ($\lambda > 1$), the 150M model seems to shows accelerating returns ($\lambda \approx 0.5$). This would suggest that convergence behavior does not follow a linear path as models grow larger, but rather oscillates with model capacity. However, the 1B model results may be less stable due to missing checkpoints and the results might be influenced by the choice of hyperparameters.

We model the relation between convergence checkpoint and the independent variables with the linear regression model defined in the equation below.

$$y_i^{\lambda} = \beta_0 + \beta_1 \ln(f_i) + \beta_2 \ln(l_i) + \sum_{k=1}^{11} \gamma_k N_{ik} + \varepsilon_i$$

The term y_i represents the convergence checkpoint of token i. The power, λ, represents the model-specific optimal power transformation of the checkpoint determined via the Box-Cox method. The f_i and l_i terms denote the frequency and length of the token i. The term N_{ik} represents indicator variables corresponding to the token nature categorical feature. The coefficients γ_k capture category-specific changes in the convergence checkpoint relative to the reference category "Word", and ε_i denotes the error term.

The resulting coefficients of determination (Total R^2) are presented in Table 2. The R^2 values are very similar across models when fixing the convergence threshold to 0.1 (SESOI), suggesting that the features consistently explain between 35% and 38% of the variability in convergence checkpoints, regardless of model size. Frequency has the highest explanatory power across models (Table 2). For the 40M-0.1 model, frequency uniquely accounts for 80% of the total R^2, while for the 150M model it accounts for 83%. For the 1B model, it accounts for 71% of the total R^2 when the threshold is 0.1,

Model	Partial R^2 (Predictors)			Total R^2
	f	n	l	
40M-0.1	0.3032	0.1136	0.0275	0.380
40M-0.2	0.3730	0.1685	0.0312	0.468
150M-0.1	0.3233	0.1172	0.0315	0.385
150M-0.2	0.3796	0.1499	0.0529	0.445
1B-0.1	0.2592	0.1473	0.0296	0.353
1B-0.2	0.3438	0.1339	0.0251	0.446

Table 2: Feature importance in explaining convergence checkpoint. Partial R^2 for a feature x represents the proportion of the variation in the convergence checkpoint that is explained by x, which was not already explained by the other variables in the model.

Model	y	f		l	
		$\%y$	Δ_y	$\%y$	Δ_y
40M	12.3k	-6.05	-744	-7.68	-944
	14.1k	-4.80	-677	-6.09	-859
	19.8k	-2.69	-534	-3.42	-677
150M	31.2k	-11.63	-3,630	-14.90	-4,650
	39.6k	-10.08	-3,993	-12.92	-5,115
	79.2k	-6.65	-5,268	-8.52	-6,750
1B	340k	-9.01	-30,619	-14.08	-47,884
	420k	-6.56	-27,549	-10.26	-43,083
	660k	-3.33	-21,977	-5.21	-34,368

Table 3: Percentage change in convergence checkpoint ($\%y$) for 100% feature (frequency f and length l) increase. Step deltas (Δ_y) represent the absolute reduction in checkpoints at convergence, when doubling frequency or length. The SESOI threshold is of 0.1.

and for slightly less when using the more permissive threshold of 0.2. The second most important feature is token nature, which has a consistent impact across models and thresholds, independently explaining between 28% and 39% of the total R^2 variability in the convergence checkpoint. Token length exhibits limited explanatory power across all models.

Given the non-linear relationship between the features and the target variable, specifically the power-law transformation y^λ, the regression coefficients cannot be interpreted directly. To quantify the precise impact of frequency and length on the convergence checkpoint, we calculate the percentage change in the convergence checkpoint relative to a 100% change in the independent variables (details in Appendix J). We evaluate their impact on three types of tokens: early-converging (25th percentile), median-converging, and late-converging (maximum checkpoint). We note that the results are

Model	Stage (Checkpoints)	$\Delta\%$	Steps Δ
40M	Early (12.3k)	-11.58	-1424
	Mid (14.1k)	-9.10	-1282
	Late (19.8k)	-5.03	-996
150M	Early (31.2k)	-27.57	-8602
	Mid (39.6k)	-24.10	-9545
	Late (79.2k)	-16.20	-12,831
1B	Early (110k)	-33.13	-112,651
	Mid (330k)	-23.44	-98,444
	Late (550k)	-11.51	-75,961

Table 4: Relative impact of token nature *Suffix* with respect to reference category *Word* on convergence checkpoints across models. The SESOI threshold is of 0.1.

robust to the SESOI threshold, therefore we report only the 0.1 threshold results in Table 3 (complete results in Appendix K).

The lowest linear regression model F-statistic obtained is of 13.6 ($p = 3.02 \times 10^{-24}$) for OLMo-1B with the 0.1 SESOI threshold. The other models and settings yielded even higher F-statistics, and lower corresponding p-values. This significant F-statistic shows that our sample of $N = 510$ tokens provides a robust explanation of convergence that is highly unlikely to be an artifact of random chance.

The percentage gain in checkpoint convergence ($\%y$) from doubling token frequency is between 3% and 12% across model scales. In terms of percentages, it is the early-convergent tokens that seem to have the highest reduction in number of checkpoints. However, in absolute numbers, the values are comparable among tokens converging at different stages. In other words, having a $2\times$ higher frequency leads to approximately a 10% reduction in the number of checkpoints needed for convergence.

Token nature also has a definitive impact on convergence checkpoint. To illustrate this aspect, we choose the category *Suffix*, for which we present the results in Table 4 (the complete results are provided in Appendix C). This category does not refer to the morphological suffix, but more broadly to alphanumeric tokens that do not start by a space. *Suffix* tokens converge sooner than *Word* tokens of mean frequency and length. While this effect is present across all stages, it is most pronounced for early-converging tokens and appears to scale with model capacity, with larger models exhibiting a greater sensitivity to token nature than the 40M variant. As shown in Appendix C, the coefficients are negative for all categories that are significant, indicating that they converge earlier than *Word* tokens,

Model	40M		150M		1B	
SESOI	0.1	0.2	0.1	0.2	0.1	0.2
λ^*	0.9	0.9	0.8	0.8	0.6	0.7

Table 5: Optimal transformations of the response variable **convergence surprisal**.

	Partial R^2 (Predictors)			
Model	f	n	l	**Total** R^2
40M-0.1	0.4529	0.4270	0.0229	0.658
40M-0.2	0.4173	0.4209	0.0189	0.603
150M-0.1	0.3427	0.4488	0.0088	0.599
150M-0.2	0.3287	0.4302	0.0065	0.576
1B-0.1	0.3270	0.4500	0.0181	0.573
1B-0.2	0.2785	0.4141	0.0032	0.542

Table 6: Feature importance in explaining **convergence surprisal**. Partial R^2 for a feature x represents the proportion of the variation in the convergence checkpoint that is explained by x, which was not already explained by the other variables in the model.

when all else is at the mean. This would suggest that *Word* tokens represent the most difficult category to learn. Other high-impact categories include single-character classes such as *Digit*, *Letter*, and *Subword*. Additionally, we observe distinct scaling trends: *Punctuation* only becomes a significant predictor in larger models, while *Numbers* lose significance as model size increases. Finally, lexical roles appear to supersede morphological ones, as the distinction between prefixes and suffixes yields no consistent cross-category effect.

6.2 Convergence Surprisal

While the optimal feature transformations when explaining convergence surprisal are logarithm for frequency and inverse for token length, we log-transformed both frequency and length to be consistent with the previous analysis. The optimal lambda values indicate a quasi-linear relation between the log-transformed features and convergence surprisal, consistent across LMs (Table 5). Furthermore, lambda seems to decrease by 0.05 every time the model doubles in size, indicating that the optimal lambda parameter might decrease log-linearly with size. In other words, in the larger model of size 1B, the gap in surprisal values between frequent and infrequent tokens is less pronounced than in smaller models.

The features explain a high proportion of variance in convergence surprisal, with R^2 values ranging from 54% to 65% (Table 6). Frequency is a consistent predictor, but its relative contribution decreases as the model grows in size. Token nature, on the other hand, gains in importance, becoming the primary predictor of convergence surprisal in larger models, suggesting that larger architectures are less dependent on raw frequency and more reliant on the intrinsic nature of the token. Interestingly, a consistent set of categories, namely *Digit*, *Letter*, *Suffix*, *Number suffix*, *Delimiter* and *Subword*, are significant and show similar standardized coefficients across model scales and convergence thresholds (Appendix L). This might suggest that convergence surprisal is to an extent token-inherent, independently of the effect of model size.

While the convergence checkpoint can naturally be associated to the moment in training when the token was sufficiently learned, convergence surprisal is difficult to interpret. We might be tempted to interpret it as a measure of performance, of how "well" the token was learned by the model, and the effect of model size supports this interpretation. However, next-word prediction is not deterministic: in many cases multiple tokens are equally valid at a specific position, and it is expected that an appropriate model would reflect this behavior. In other words, even supposing the model was perfect, a context might have two equally valid continuations. At the beginning of training, we cannot know if surprisal is high due to insufficient data or due to some intrinsic characteristics of the token. At convergence, we can suppose that what was to be learned by the model was learned, and that token surprisal rather reflects a token-inherent property, namely the extent to which the token is substitutable for another.

7 Conclusions

In this study we have shown that token-level training trajectories are useful tools for understanding how models learn. We showed that these trajectories follow structured patterns, partly governed by frequency and token nature. Although scaling the model improves both convergence speed and surprisal value, the two behave differently: convergence checkpoint seems to follow a frequency-based 'recipe', while the surprisal reached at convergence seems more inherent to the token's nature. In future work we aim to leverage these findings to develop an efficient learning method that modifies training data to reach token convergence faster.

Limitations

Our analysis focuses on English-trained models to establish token convergence patterns. We restricted our scope to the OLMo model family to maintain a consistent tokenization strategy. Since token boundaries and therefore their characteristics vary significantly across segmentations, holding the tokenizer constant was a methodological choice meant to ensure that our observations of convergence timepoint and surprisal were functions of model scale rather than confounding segmentation strategies. While this ensures high internal validity, the absolute values of convergence time and surprisal may shift under different tokenization strategies.

Ethical considerations

B2. The OLMo-1B model is released under the Apache license 2.0, which permits use of the models. The license for the code and data that we release are present in the repository url we provide in Section 1.

C1. The approximate number of GPU hours was of 20,000 hours on a V100 80GB GPU.

C4. We used `spacy` version 3.7.2 to determine if a token belongs to the English vocabulary.

E1. Copilot was used as a coding assistant and ChatGPT and Gemini for brainstorming and rephrasing.

References

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Gregory Anthony, Shivanshu Purohit, and Edward Raff. 2023a. Emergent and predictable memorization in large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023b. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

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.

Pavel Chizhov, Catherine Arnett, Elizaveta Korotkova, and Ivan P. Yamshchikov. 2024. Bpe gets picky: Efficient vocabulary refinement during tokenizer training. *Preprint*, arXiv:2409.04599.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. Olmo: Accelerating the science of language models. *Preprint*, arXiv:2402.00838.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2019. Overcoming catastrophic forgetting

via model adaptation. *International Conference on Learning Representations*.

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020a. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

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

Daniël Lakens. 2017. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8:194855061769717.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. *Preprint*, arXiv:2405.05417.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2025. Rho-1: Not all tokens are what you need. *Preprint*, arXiv:2404.07965.

Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. 2012. *Introduction to Linear Regression Analysis*, 5th edition. Wiley, Hoboken, New Jersey.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2020. Deep double descent: Where bigger models and more data hurt. In *ICLR 2020*. Also appeared in Journal of Statistical Mechanics, 2021.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: finding important examples early in training. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *CoRR*, abs/2201.02177.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*. Accessed: 2024-11-15.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. *International Conference on Learning Representations*.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? limits of llm scaling based on human-generated data. *Preprint*, arXiv:2211.04325.

Larry Wasserman. 2004. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer, New York.

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Veselin Stoyanov. 2023. Training trajectories of language models across scales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13711–13738, Toronto, Canada. Association for Computational Linguistics.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley.

A Training details of OLMo models

The complete configurations are provided in the Github repository.

Model	# param.	Train (# tokens)	Train (# steps)	Batch size	Context size	Layers	Attn. Heads
OLMo-40M	38M	0.10×10^{12}	24,000	2048	2048	16	16
OLMo-150M	151M	0.35×10^{12}	84,000	2048	2048	12	12
OLMo-1B	1.12B	2.93×10^{12}	700,000	2048	2048	8	8

Table 7: Details on the training of new OLMo models (and existing model OLMo-1B).

B Frequency estimation

We estimated the frequency of a token in the training data as the average frequency of the token over 500 batches sampled randomly from the first 700,000 batches in the training data of the OLMo-1B model.

To validate these estimates, a pivotal intervals bootstrap (Wasserman, 2004) ($n = 1000$ resamplings, size 250) was performed. Across the 49,955 tokens with non-null frequency, the median 95% confidence interval (CI) width was 1.70 (IQR: 1.05–2.89), with 90% of tokens maintaining CI width inferior to 5. In other words, for a median token, its true frequency falls within a window of size 1.70 centered around the estimate in 95% of cases. We report CI widths statistics in Figure 3 and Table 8. Given that high-frequency tokens skew absolute CI distributions (higher frequencies have higher CI intervals), we also report CI widths relative to the mean in Figure 4 and Table 8. This relative measure shows that for the median token, the true value of the frequency is within $\pm$ 6.5% of the estimate in 95% of cases.

Statistic	95% CI Width	Relative Width
Mean (Std. Dev)	2.78 (6.54)	0.20 (0.29)
Median	1.70	0.13
25th Percentile	1.05	0.08
75th Percentile	2.89	0.22
90th Percentile	5.06	0.37
Maximum	468.23	5.00

Table 8: Descriptive statistics for absolute and relative 95% confidence interval widths across 49,955 tokens.

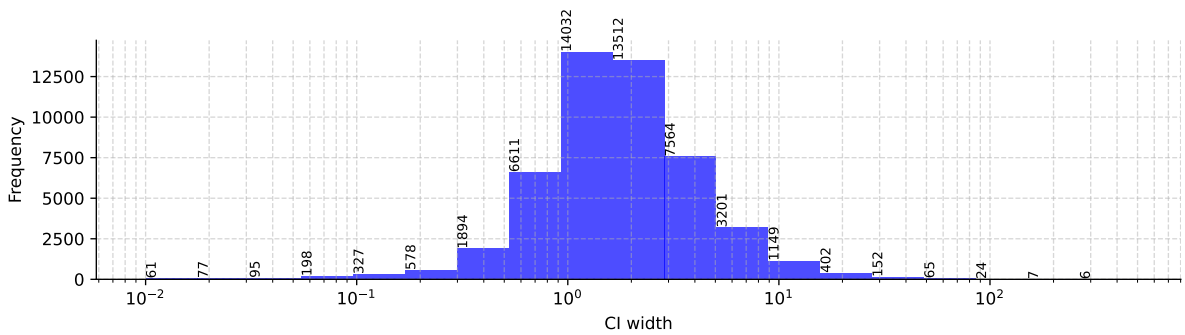

Figure 3: Histogram of CI widths.

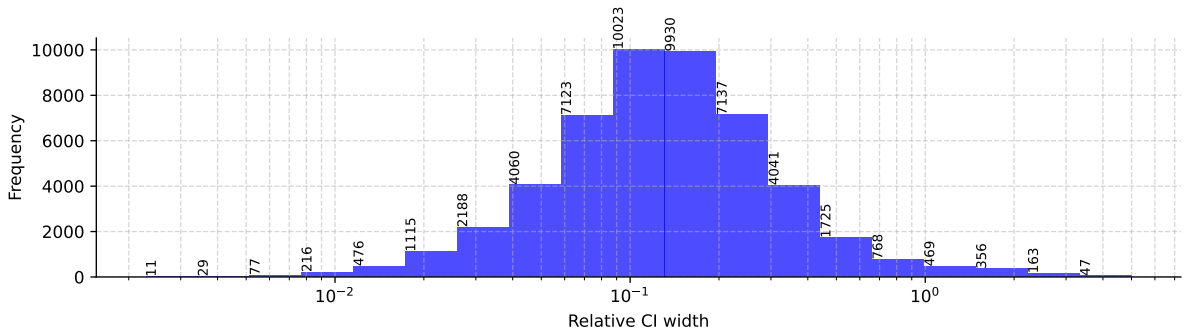

Figure 4: Histogram of CI widths relative to mean frequency.

C Description of token categories.

We define the following 12 token categories. To identify to which category a token belongs, the conditions are verified in the order given below until one is satisfied. For example, a token of nature *suffix* is an alphabetic string that did not validate any previous category. The tokens in the last category, *subword*, are tokens that did not validate any of the previous 11 categories. The examples are given in the format used in the vocabulary file and might contain placeholders, such as *Ġ*, the placeholder for the space character " ". We chose this format so that strings can be searched in the vocabulary file of the model.

We considered that a token belongs to the category *Word* if it can be assigned a POS tag. We used the POS tagger provided by `spacy` version 3.7.2. This approach had better coverage than the list of headwords in `nltk` version 3.9.1 which we used as comparison. Numerical tokens or punctuations being tagged as words was unlikely to be an issue, given that these categories were checked before. We present the heuristics we used for extraction and examples of tokens from each category in Table 9. To simplify, we indicate whether or not a token is a word as belonging or not to **V**.

Token nature	**Conditions**	**Examples from sample**
digit	isdigit, len 1	*0*, *1*, *2*
number suffix	isdigit	*04*, *181*, *1981*, *830*, *76*
number prefix	starts with space, isdigit	*Ġ2002*, *Ġ122*,*Ġ349*
delimiter	isspace or contains tab or newline	(23 consecutive spaces), *Ċĉ* (\n and \t)
punctuation	$\in$ string.punctuation, len 1	\ or *)* or / or *Ġ,*
punctuation suffix	$\in$ string.punctuation	+\ or .\″,\″ or *#{$* or — (3 dashes)
punctuation prefix	starts with space, $\in$ string.punctuation	*Ġ{*\ or *Ġ=* or *Ġ-* or *Ġ[ ],*
letter	isalpha, len 1	*æīĢ* or *åĪ¶* (Chinese characters)
prefix	starts with space, isalpha, $\notin$ V	*ĠFe*, *Ġvitro*, *ĠRS*, *ĠÃ§*, *Ġfacto*, *ĠR*
word	starts with space, isalpha, $\in$ V	*ĠHar*, *Ġsolved*, *ĠCycle*, *ĠVideo*
suffix	isalpha	*correct*, *amic*, *rangian*, *inject*, *aving*
subword	OTHER	*à®* (sequence of bytes)

Table 9: Description and examples of **token nature** categories.

D Additional information on tokens distribution.

Comparative histograms of the 50,280 tokens in the OLMo vocabulary and the 510 token sample on frequency range, token nature and token length are given in Figures 5, 6 and 7.

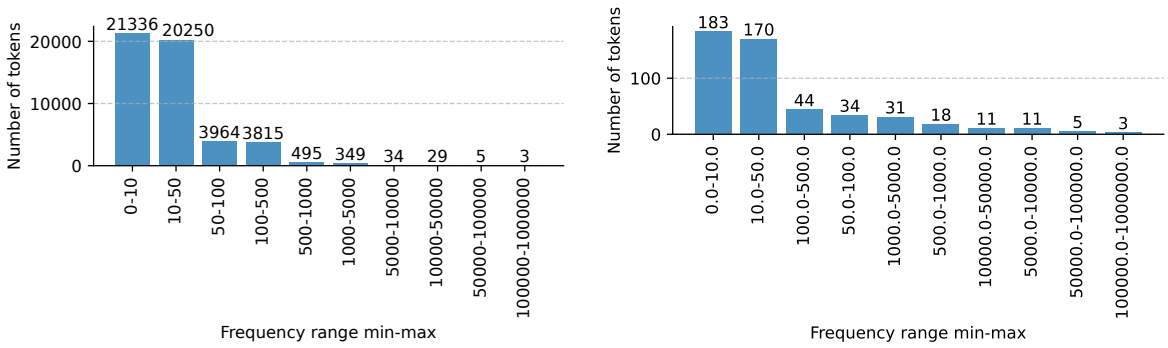

Figure 5: **Left:** Histogram of token counts by frequency range for all tokens in the vocabulary. **Right:** for the 510 token sample.

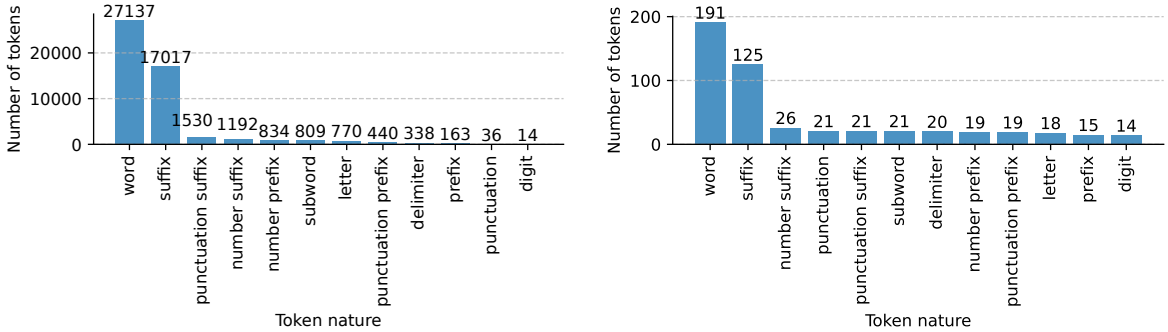

Figure 6: **Left:** Histogram of token counts by token nature for all tokens in the vocabulary. **Right:** for the 510 token sample.

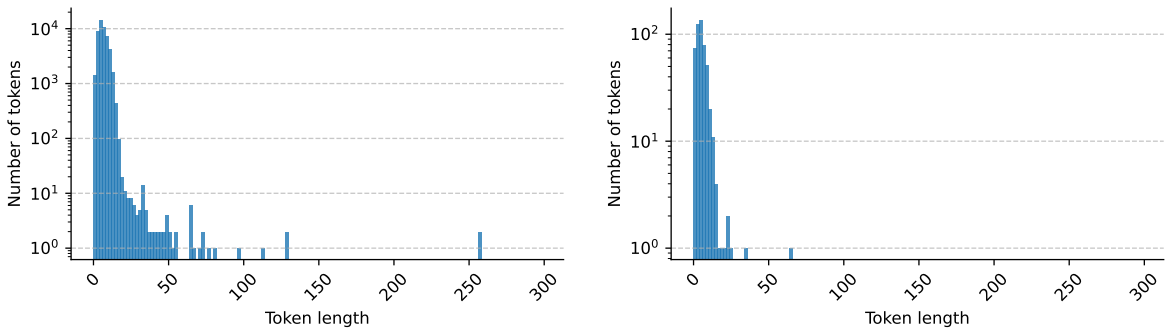

Figure 7: **Left:** Histogram of token counts by token length for all tokens in the vocabulary. **Right:** for the 510 token sample.

E Average token position in context

The contexts are sampled from held-out training data, that has already been arranged into batches of random contexts of length 2048 tokens. The position of the target token in context ranges therefore between 20 (the minimum context size we imposed) and 2047, the context window minus the target token. If the documents composing the training data were split according to sentence or paragraph boundaries, we might expect a position bias to emerge, that is, some tokens to appear primarily in a specific absolute position. For example, a pronoun would appear primarily in the first position in a sentence, leading to a smaller average than less syntactically-constrained tokens. However, as the data comes from training batches, it is not split according to sentence boundaries, but it is split in contiguous chunks of 2048 tokens and shuffled randomly, making contexts extend beyond sentence, paragraph and document boundaries. We thus expect the position bias to be mitigated.

We analyze the mean position per token by computing the mean length of the 1000 contexts for each token. As shown in Figure 8, the average positions of tokens form a tight cluster around the mean cross-token position of 1034.4, closely aligning with the theoretical midpoint of 1033.5 = (20 + 2047)/2, suggesting that tokens are uniformly distributed throughout the context window. While individual token occurrences are broadly distributed ($\sigma = 594$), the mean positions across the sample are highly consistent, varying by only 1.8% ($SD_{\bar{x}} = 18.55$).

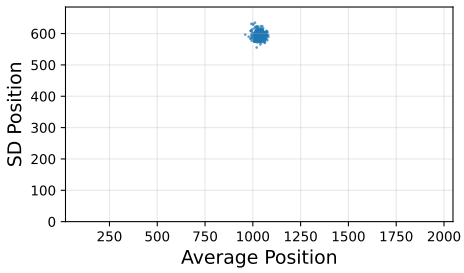

Figure 8: Scatter plot of the average position vs. standard deviation of the position for the 510 tokens.

F Trajectories for a larger range of tokens

In Figures 9, 10 and 11, we plot 8 tokens per frequency range for each of the 3 models (in order OLMo-40M, OLMo-150M and OLMo-1B), as well as the average surprisal and standard deviation among the total 510 tokens.

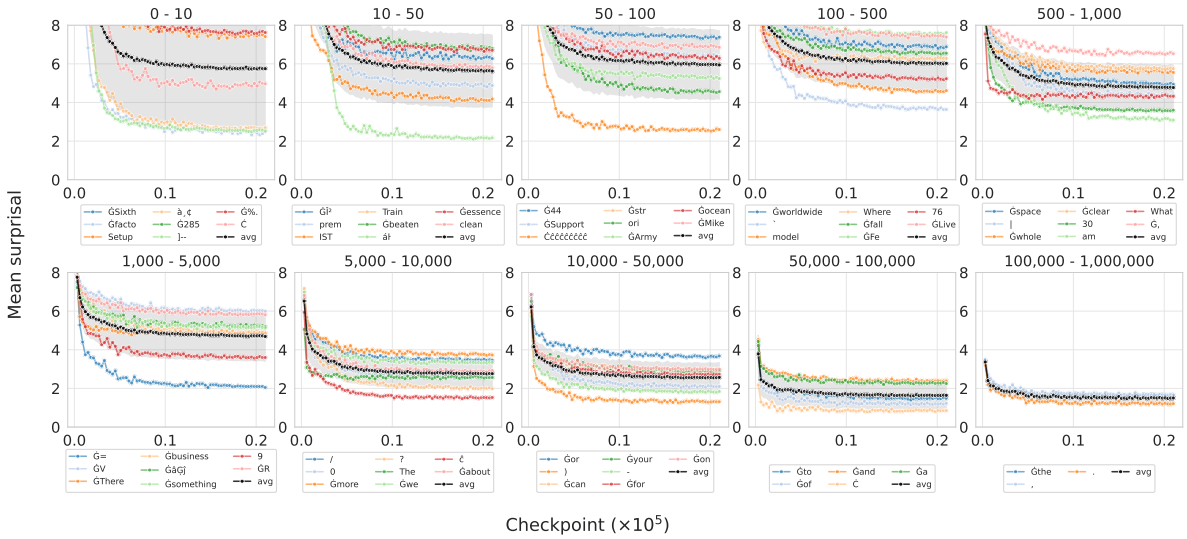

Figure 9: Training trajectories of different tokens and the per-frequency average curve and standard deviation for model **OLMo-40M**.

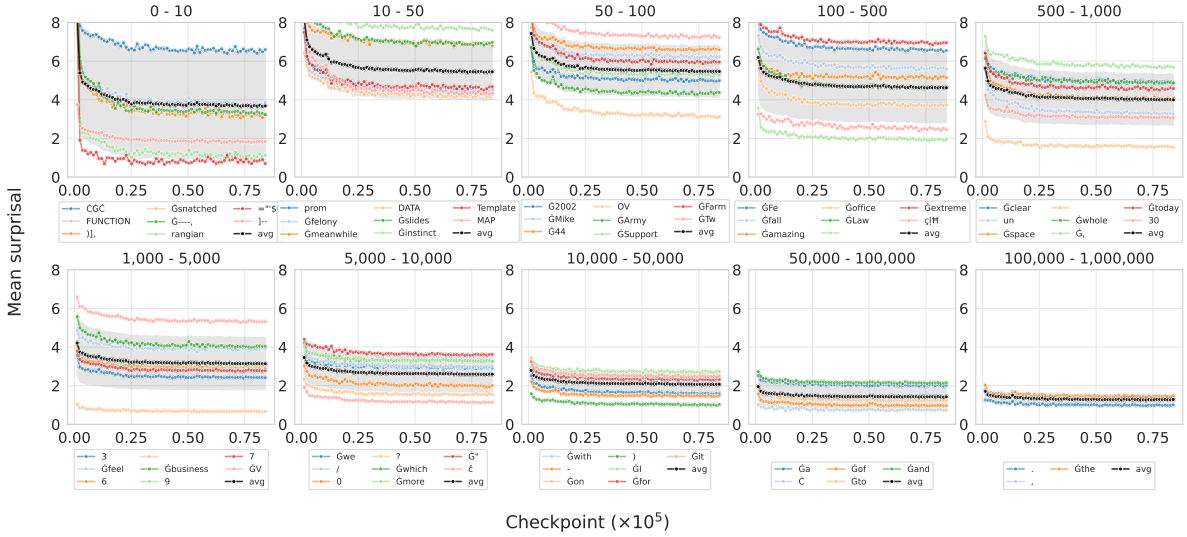

Figure 10: Training trajectories of different tokens and the per-frequency average curve and standard deviation for model **OLMo-150M**.

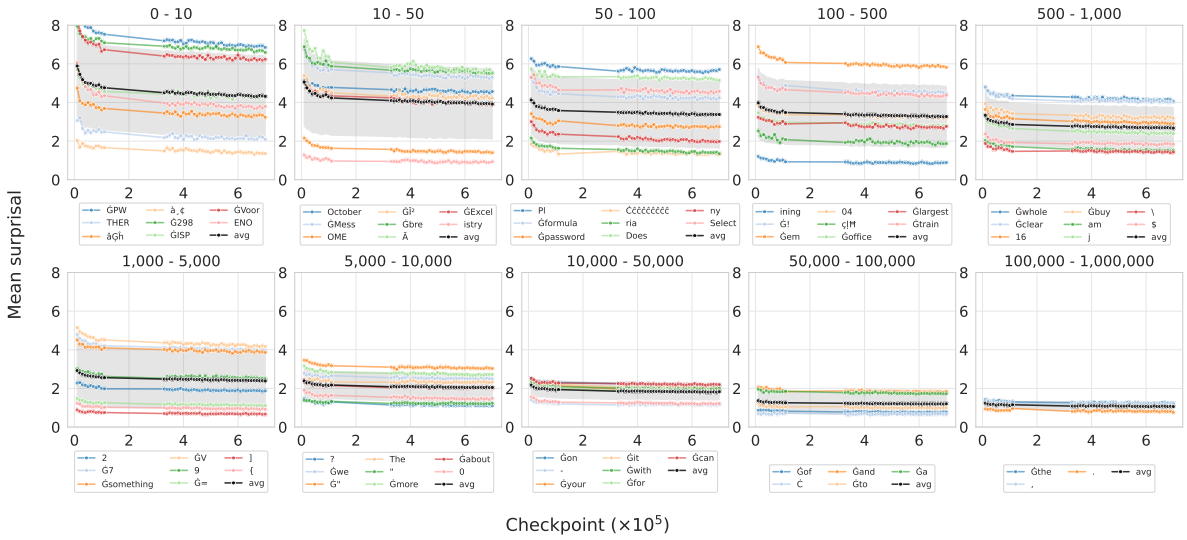

Figure 11: Training trajectories of different tokens and the per-frequency average curve and standard deviation for model **OLMo-1B**.

G Convergence Algorithm

Algorithm 1: Token-wise Convergence Detection

Input : tokens $\mathcal{T} = \{t_1, \ldots, t_n\}$, checkpoints $\mathcal{C} = \{c_1, \ldots, c_m\}$, surprisals $\mathcal{S} = \{s_1, \ldots, s_m\}$
Output : convergence status, convergence checkpoint c^*, convergence surprisal s^*
Parameters : L_{end} (Min. ending plateau length), L_{seq} (Min. total sequence length)
Constants : L_{min} (Min. plateau length) = 2

foreach *token* $t \in \mathcal{T}$ **do**
  $\mathcal{P}_t \leftarrow \emptyset$; `// List of plateaus for token` t
  $S_{curr} \leftarrow [c_1]$; `// Current sequence`
  `// Identify plateaus`
  for $i \leftarrow 2$ **to** m **do**
    if `is_equivalent`(c_{i-1}, c_i) **then**
      Append c_i to S_{curr};
    else
      if $|S_{curr}| \geq L_{min}$ **then**
        Add S_{curr} to $\mathcal{P}_t$;
      $S_{curr} \leftarrow [c_i]$; `// Reset sequence to current checkpoint`
  `// Check the final sequence`
  if $|S_{curr}| \geq L_{min}$ **then**
    Add S_{curr} to $\mathcal{P}_t$;
  `// Evaluate Convergence`
  $\mathcal{P}_{end} \leftarrow$ last plateau of $\mathcal{P}_t$;
  $\mathcal{S}_{res} \leftarrow$ concatenate all $c \in \mathcal{P}_t$;
  if $|\mathcal{P}_{end}| \geq L_{end}$ ***and*** $|\mathcal{S}_{res}| \geq L_{seq}$ ***and*** $c_m \in \mathcal{S}_{res}$ **then**
    $c^* \leftarrow$ first element of $\mathcal{S}_{res}$;
    $s^* \leftarrow \frac{1}{|\mathcal{S}_{res}|} \sum s \in \mathcal{S}_{res}$; `// Mean surprisal of resulting sequence`
    return true, c^*, s^*;
return false, none, none;

Algorithm 2: Checkpoint Equivalence with TOST for Paired Samples
Used as `is_equivalent`(c_i, c_j)

Input : paired surprisals $\mathcal{X}_i = \{x_1, \ldots, x_n\}$, $\mathcal{Y}_j = \{y_1, \ldots, y_n\}$(the surprisal values of checkpoints c_i and c_j on n evaluation items), smallest effect size of interest (SESOI) δ, significance level α

Output : p-values (p_1, p_2)

```
// Compute paired differences
for i ← 1 to n do
    d_i ← y_i − x_i;
n ← |{d_i}|;
// Compute summary statistics
d̄ ← (1/n) Σ_{i=1}^{n} d_i ;                                   // Mean difference
s_d ← sqrt( (1/(n−1)) Σ_{i=1}^{n} (d_i − d̄)^2 ) ;              // Sample std. dev.
SE ← s_d / sqrt(n) ;                                          // Standard error
df ← n − 1;
;                                                             // Degrees of freedom
// Compute TOST statistics
t_1 ← (d̄ + δ)/SE ;                                            // Test μ_d > −δ
p_1 ← 1 − CDF(t_1, df)
t_2 ← (d̄ − δ)/SE ;                                            // Test μ_d < δ
p_2 ← CDF(t_2, df)
// Decide equivalence
if p_1 < α and p_2 < α then
    return true
return false
```

H Convergence parameters

We study how the parameters of the convergence algorithm, ending plateau length L_{end}, sequence length L_{seq}, effect size SESOI and α, as well as the input checkpoints granularity affect the outcome number of converged tokens.

H.1 SESOI, L_{min}, L_{seq}

To identify the impact of these parameters on the number of convergent tokens, we conducted a sensitivity analysis using the OLMo-1B model at the 700,000-step checkpoint. While holding the model architecture and final checkpoint constant, we varied the following variables:

- checkpoint granularity: 1) every 10,000 checkpoints (49 checkpoints), 2) every 20,000 checkpoints (24) and 3) every 50,000 checkpoints (10)
- $L_{end} \in \{2, 3, 4\}$
- $L_{seq} \in \{3, 4, 5\}$
- SESOI $\in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4\}$

We plot the resulting number of converged tokens for each configuration in Figure 12. As expected, the effect size (SESOI) parameter is the primary determinant of the output. Increasing the parameter from 0.01 to 0.4 results in a shift from none to all tokens converging in all configurations. Additionally, the results seem robust with respect to the checkpoints granularity. Halving the number of checkpoints, by

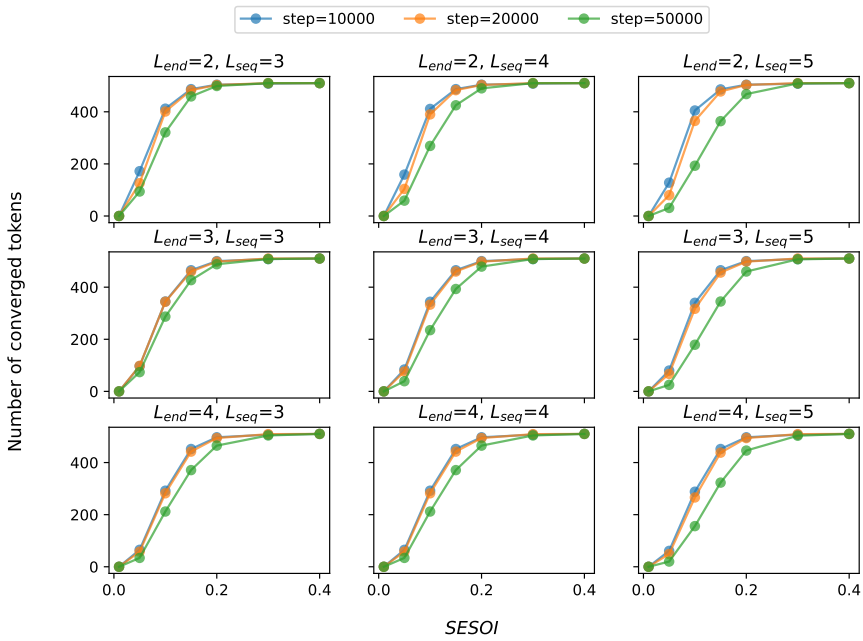

Figure 12: Number of converged tokens depending on convergence parameters SESOI, L_{end} and L_{seq} and input checkpoints granularity (step) for the OLMo-1B model.

increasing the granularity from 10,000 (49 checkpoints) to 20,000 (24 checkpoints), results in overlapping curves. At 50,000, the number of converging tokens is consistently lower, but the small number of resulting checkpoints (10) may affect the results. Finally, the parameters L_{end} and L_{seq} seem to have little influence on the curves. As a result, we fix henceforth L_{end} and L_{seq} to 3 and 5, and the granularity to 10,000, for more precision. There is no absolute way to decide on the SESOI value. Whether the threshold for convergence should be a mean surprisal value of 0.1 instead of 0.2, for example, depends on the resources available and the purpose of the model.

H.2 Final checkpoint

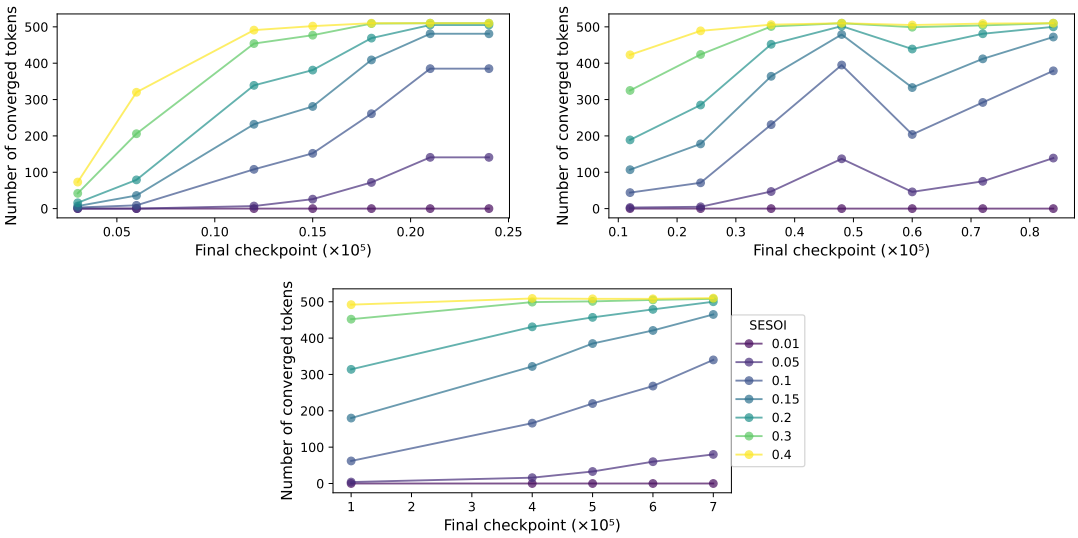

Figure 13: Number of converged tokens at different checkpoints, depending on effect size SESOI for OLMo-40M, 150M and 1B, respectively.

The convergence detection algorithm operates by evaluating each intermediate checkpoint against a 'final' checkpoint of the model. Because this approach defines convergence relative to the last provided

checkpoint, the selection of the final training step may affect the output. Following the parameters established in Subsection H.1, we utilized a sequence length (L_{seq}) of 5 and a threshold (L_{end}) of 3 and a granularity of 10,000 for OLMo-1B. To account for varying training scales, we set the corresponding granularity to 1,200 for OLMo-150M, and 300 for OLMo-40M.

To determine how the choice of the final checkpoint affects convergence detection, we varied the final checkpoint for each model. The checkpoints were sampled at seven equal intervals based on model-specific granularity: $300i$ for OLMo-40M, $1,200i$ for OLMo-150M, and $10,000i$ for OLMo-1B, where $i \in \{1, 2, \ldots, 7\}$. We also varied the SESOI value across the following values $\{0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4\}$.

The results are reported in Figure 13. Convergent tokens counts generally exhibit an upward trajectory, following the intuition that more and more tokens converge as training progresses. One exception is the model OLMo-150M, for which we observe that the final number of converged tokens seems to be attained early, around checkpoint 40,000-50,000. Across all three models, the lower boundary ($\text{SESOI} = 0.01$) consistently yields zero converged tokens, confirming that this threshold is too restrictive for practical convergence detection. In contrast, the higher boundary changes during training and across models, suggesting a refinement process is taking place. For model OLMo-1B, a permissive 0.4 threshold leads to 96% token convergence at checkpoint 100,000, however, by checkpoint 700,000, a significantly stricter 0.2 threshold leads to a similar proportion of 98% of tokens converging.

I Transformations for predicting checkpoint

In this section we detail the methodology used to find the optimal transformations that linearize the data in order to apply linear regression. We perform a grid-search of transformations on the independent variables and use the Box-Cox method (Montgomery et al., 2012) to find the best transformation of the dependent variable. We test several candidate transformations t of the independent variables: square root, logarithmic, multiplicative inverse and identity. For the dependent variable, we cannot directly apply transformations, as the error metrics, which are computed on the dependent variable, will no longer be comparable due to the resulting changes in scale. To counter this issue, the Box-Cox method first normalizes the transformed dependent variable by the geometric mean and then computes the error metrics (Eq. 1 and 2). This procedure ensures that the residual sum of squares error metric (RSS) values are scale-invariant and allows for a valid comparison of model fit across the range of transformations. We tested 41 power values for the dependent variable (λ) between -2 and 2, corresponding to multiplicative inverse ($\lambda = -1$, $\frac{1}{y}$), log ($\lambda = 0$), linear ($\lambda = 1$, y^1), square root ($\lambda = 0.5$, $\sqrt{y}$), quadratic ($\lambda = 2$, y^2) and other transformations corresponding to power values in between.

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda \cdot \dot{y}^{\lambda-1}} & \text{if } \lambda \neq 0 \\ \dot{y} \ln(y_i) & \text{if } \lambda = 0 \end{cases} \tag{1}$$

where $\dot{y}$ represents the geometric mean of the dependent variable y, defined as:

$$\dot{y} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln(y_i)\right) \tag{2}$$

For each model scale and SESOI parameter, we apply each of the four transformations t to frequency and length and each of the 41 Box-Cox transformations (given by λ values $\in \{-2, 2\}$) to the checkpoint variable while normalizing by the geometric mean. We compute the standardized RSS values for each of the 656 transformations and select the frequency and length transformations as well as the λ value for the convergence checkpoint that minimizes the RSS.

J Interpreting results as elasticity

Given the non-linear relationship between the features and the target variable, specifically the power-law transformation y^{λ}, the regression coefficients cannot be interpreted directly. To provide a more intuitive

measure of feature importance, we calculate the point elasticity ($\epsilon = \frac{\%\Delta y}{\%\Delta x}$). This represents the predicted percentage change in the convergence checkpoint y resulting from a 1% change in a given feature x. By applying the partial derivative with respect to either one of the features (denoted as x) to both sides of the equation in (6.1), we obtain the equation in (3). We replace (3) in the elasticity equation and obtain (4). Given a $\%\Delta x$ change in x, the resulting $\%\Delta y$ is the elasticity ϵ multiplied by the change in x. The elasticity for the numeric features does not depend on the value of x, but does depend on the value of y, that is, tokens that converge early have a different behavior than tokens that converge late.

$$\frac{\partial y}{\partial x} = \frac{\beta}{x\lambda y^{\lambda-1}} \tag{3}$$

$$\epsilon_x = \frac{\%\Delta y}{\%\Delta x} = \frac{\partial y}{\partial x}\frac{x}{y} = \frac{\beta}{x\lambda y^{\lambda-1}}\frac{x}{y} = \frac{\beta}{\lambda y^{\lambda}} \tag{4}$$

Given that token nature is not a numeric variable, we cannot compute elasticity. We will compute instead the percentage change in convergence checkpoint when switching from the reference category (the category 'Word') to one of the other token natures (such as 'Suffix', 'Digit' etc.). We define relative change of the category k w.r.t the reference category as shown in (5). We replace y_k using the regression formula (6.1) and obtain (6), where we note the other features with their coefficients and the intercept as Base, to simplify. We replace back Base as y_{ref}^{λ} (the reference category has a γ of 0) and obtain (7).

$$\Delta y_{\text{rel}} = \frac{y_k - y_{\text{ref}}}{y_{\text{ref}}} \tag{5}$$

$$\Delta y_{\text{rel}} = \frac{y_k}{y_{\text{ref}}} - 1 = \frac{(\text{Base} + \gamma_k)^{1/\lambda}}{(\text{Base})^{1/\lambda}} - 1 = \left(1 + \frac{\gamma_k}{\text{Base}}\right)^{1/\lambda} - 1 \tag{6}$$

$$\Delta y_{\text{rel}} = \left(1 + \frac{\gamma_k}{y_{\text{ref}}^{\lambda}}\right)^{1/\lambda} - 1 \tag{7}$$

K Checkpoint further details

We report the complete results on the impact of frequency and token length in Table 3. Additionally, we report the results of the impact of different token natures (*Digit*, *Letter*) in Tables 11-15.

Model	**Token**	**Ckpt** (y)	$\%\Delta y_f$	**Steps** Δ_f	$\%\Delta y_l$	**Steps** Δ_l
40M-0.1	9	12,300	-6.05	-744	-7.68	-944
40M-0.1	IST	14,100	-4.80	-677	-6.09	-859
40M-0.1	isle	19,800	-2.69	-534	-3.42	-677
40M-0.2	OME	9,300	-7.50	-697	-7.86	-731
40M-0.2	Template	10,800	-6.08	-657	-6.37	-688
40M-0.2	Ġ$("#	17,700	-3.04	-539	-3.19	-565
150M-0.1	*[9spaces]*	31,200	-11.63	-3,630	-14.90	-4,650
150M-0.1	1	39,600	-10.08	-3,993	-12.92	-5,115
150M-0.1	Ġlibr	79,200	-6.65	-5,268	-8.52	-6,750
150M-0.2	4	20,400	-12.24	-2,497	-17.30	-3,529
150M-0.2	Bal	26,400	-10.22	-2,698	-14.44	-3,813
150M-0.2	Ġ}\	72,000	-5.06	-3,645	-7.16	-5,152
1B-0.1	1	340,000	-9.01	-30,619	-14.08	-47,884
1B-0.1	ÑĮ	420,000	-6.56	-27,549	-10.26	-43,083
1B-0.1	åıÍ	660,000	-3.33	-21,977	-5.21	-34,368
1B-0.2[2]	Ġgrow	330,000	-9.15%	-30,199	-8.85%	-29,217
1B-0.2	Ġgenotypes	550,000	-4.71%	-25,909	-4.56%	-25,065

Table 10: Shift in checkpoint (y) for 100% feature increase. Step deltas represent the absolute reduction in training steps at convergence, when doubling frequency ($\%\Delta y_f$) or length ($\%\Delta y_l$).

Model	Stage (Steps)	Δ%	Steps Δ
40M-0.1	Early (12.3k)	-35.77	-4400
	Mid (14.1k)	-27.36	-3858
	Late (19.8k)	-14.61	-2892
150M-0.1	Early (31.2k)	-37.77	-11783
	Mid (39.6k)	-33.15	-13127
	Late (79.2k)	-22.47	-17796

Table 11: Impact of **Letter** token nature on convergence checkpoints relative to Word tokens across model scales.

Model	Stage (Steps)	Δ%	Steps Δ
40M-0.1	Early (12.3k)	-21.96	-2701
	Mid (14.1k)	-17.08	-2408
	Late (19.8k)	-9.31	-1844
150M-0.1	Early (31.2k)	-30.52	-9522
	Mid (39.6k)	-26.71	-10578
	Late (79.2k)	-18.00	-14253

Table 12: Impact of **Digit** token nature on convergence checkpoints relative to Word tokens different model scales.

Model	Stage (Steps)	Δ%	Steps Δ
40M-0.1	Early (12.3k)	-11.57	-1423
	Mid (14.1k)	-9.09	-1281
	Late (19.8k)	-5.03	-995
150M-0.1	Early (31.2k)	-21.51	-6711
	Mid (39.6k)	-18.76	-7430
	Late (79.2k)	-12.55	-9943

Table 13: Impact of **Number Prefix** token nature on convergence checkpoints across models and thresholds.

Model	Stage (Steps)	Δ%	Steps Δ
40M-0.1	Early (12.3k)	-15.75	-1937
	Mid (14.1k)	-12.32	-1738
	Late (19.8k)	-6.78	-1342
150M-0.1	Early (31.2k)	-37.91	-11827
	Mid (39.6k)	-33.27	-13177
	Late (79.2k)	-22.56	-17866
1B-0.1	Early (110k)	-27.66	-94033
	Mid (330k)	-19.68	-82664
	Late (550k)	-9.72	-64170

Table 14: Impact of **Number Suffix** token nature on convergence checkpoints across models and thresholds.

Model	Stage (Steps)	Δ%	Steps Δ
40M-0.1	Early (12.3k)	-15.21	-1871
	Mid (14.1k)	-11.91	-1680
	Late (19.8k)	-6.56	-1298
150M-0.1	Early (31.2k)	-20.50	-6397
	Mid (39.6k)	-17.88	-7080
	Late (79.2k)	-11.95	-9467

Table 15: Impact of **Delimiter** token nature on convergence checkpoints across models.

L Surprisal further details

We report the standardized coefficients for the features involved in predicting convergence surprisal in Table 16.

Predictor	**40M-0.1**	**40M-0.2**	**150M-0.1**	**150M-0.2**	**1B-0.1**	**1B-0.2**
Obs.	385	505	379	500	340	500
R^2	0.641	0.603	0.599	0.576	0.573	0.542
Adjusted R^2	0.629	0.593	0.585	0.565	0.556	0.529
Numeric Features						
Frequency	-0.682	-0.643	-0.566	-0.557	-0.581	-0.507
Token Length	-0.166	-0.150	—	—	-0.164	—
Character Types						
Digit	-0.186	-0.169	-0.151	-0.148	-0.163	-0.126
Letter	-0.261	-0.318	-0.303	-0.285	-0.339	-0.273
Punctuation	—	—	—	—	—	—
Affixes						
Prefix	—	-0.061	—	—	—	—
Suffix	-0.532	-0.561	-0.561	-0.583	-0.616	-0.584
Num. Pref.	—	—	—	—	—	—
Num. Suff.	-0.152	-0.144	-0.114	-0.106	-0.085	—
Punct. Pref.	—	—	—	—	—	—
Punct. Suff.	—	—	—	—	—	—
Other						
Delimiter	-0.214	-0.233	-0.268	-0.249	-0.241	-0.201
Subword	-0.353	-0.316	-0.279	-0.275	-0.308	-0.247

Table 16: Standardized regression coefficients (β^*) across model scales and thresholds. Negative values indicate lower convergence surprisal.