Predicting the Formation of Induction Heads

Tatsuya Aoyama   Ethan Gotlieb Wilcox   Nathan Schneider
Department of Linguistics, Georgetown University
{ta571, ethan.wilcox, nathan.schneider}@georgetown.edu

Abstract

Arguably, specialized attention heads dubbed induction heads (IHs) underlie the remarkable in-context learning (ICL) capabilities of modern language models (LMs); yet, a precise characterization of their formation remains unclear. In this study, through a series of experiments using both natural and synthetic data, we show that (1) a simple equation combining batch size and context size predicts the point at which IHs form; (2) surface bigram repetition frequency and reliability strongly affect the formation of IHs, and we find a precise Pareto frontier in terms of these two values; and (3) local dependency with high bigram repetition frequency and reliability is sufficient for IH formation, but when the frequency and reliability are low, categoriality and the shape of the marginal distribution matter.

1 Introduction

The practical utility of modern LMs depends heavily on their ability to perform ICL, broadly construed as performing a task based on the input provided at inference time. Various accounts have been provided to explain the internal workings of this capability, and among them are studies that find certain attention heads affecting LMs' ICL capabilities [11, 6, 15]. The emergence of such heads during pretraining is sometimes referred to as phase change or phase transition [4, 1].

Factors that affect phase transition, particularly the formation of IHs, which are characterized by their (often long-range) in-context copying behavior, are not fully understood. Chan et al. [3] study the data properties that lead to different learning outcomes in a few-shot learning of image classification, where few-shot examples are image-label pairs. They find that (1) burstiness (similar things appearing in clusters), (2) within-class variation, and (3) dynamic class membership all promote ICL and demote in-weight learning (IWL). Interestingly, matching the marginal distribution of the labels to a Zipfian distribution was the only variable that led to high ICL and IWL simultaneously. Edelman et al. [6] proposes a synthetic task dubbed ICL Markov Chain (ICL-MC) using a Markov Process involving 2–8 symbols. Taking a Bayesian approach to ICL [14], Edelman et al. [6] randomly initialize the transition matrix at the beginning of each epoch, thereby making it impossible to learn the underlying distribution of the symbols. They find that the models go through phases where their predictions most closely match the uniform distribution, then in-context unigram counts, and lastly in-context bigram counts. They show that by forming IHs that attend to all of the bigram continuations of the current token in the context, models achieve the Bayes-optimal bigram solution.

In the current literature including the aforementioned studies, ICL is often studied as a few-shot learning (e.g., 3) with input-output pairs, or with an unrealistic setting that necessitates ICL as opposed to IWL (e.g., 6). As such, this raises the question of why LMs trained on natural language form IHs (and ICL capabilities). Furthermore, the point at which IH form is not fully understood. Some report a narrow range (1B–3B pretraining tokens; e.g., 11) while others find a much wider range (64M–2B pretraining tokens; 1). To fill these gaps, in this study, we analyze **how pretraining configuration and data properties affect whether IHs emerge and at what point they first appear**. Through a series of experiments using both natural and synthetic data, we show that (1) a simple

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: First Workshop on CogInterp: Interpreting Cognition in Deep Learning Models.

equation combining batch size and context size can predict the IH formation point (Section 3); (2) the surface bigram repetition frequency and reliability strongly affect the formation of IHs, and a precise Pareto frontier can be described in terms of these two values (Section 4); and (3) local dependency (non-independence between two consecutive tokens) with high frequency and reliability are sufficient for IH formation, but when the frequency and reliability are low, the presence of latent categories of symbols and the shape of the marginal distribution matter (Section 5).

2 Methods

2.1 Metrics

IHs are defined by the copying behavior, such that if the model has seen an $\langle \mathsf{A}, \mathsf{B} \rangle$ sequence *in-context* and the current token is A, then an IH is a head that promotes the prediction of B as the next token, completing the $\langle \mathsf{A}, \mathsf{B}, \dots, \mathsf{A}, \mathsf{B} \rangle$ sequence. Following Olsson et al. [11], we quantify this behavior using prefix-matching score (PS). Given a random sequence of tokens $\mathbf{x}$ repeated twice, PS of a head h at layer l is its average attention from the source token x_i to the next token of its previous occurrence: $\frac{1}{|\mathbf{x}|-1} \sum_{i=|\mathbf{x}|+1}^{2|\mathbf{x}|} \alpha^{(h,l)}(x_i, x_{i-(|\mathbf{x}|-1)})$. As to the size of the sequence $\mathbf{x}$, `TransformerLens` library [9] adopts $|\mathbf{x}| = 50$, which we also do; however, when analyzing models with smaller context sizes, we adjust $|\mathbf{x}|$ accordingly: $|\mathbf{x}| = \min(\frac{|\text{context}|}{2}, 50)$.

2.2 Models and Checkpoints

We use a toy version of GPT2 [12] with 2 layers and 8 attention heads per layer for all experiments. For experiments with natural language, we adopt a pretrained GPT2 tokenizer with a vocabulary size of 50,257, unless otherwise specified. All models analyzed in this study were trained from scratch for 1B pretraining tokens. We save intermediate checkpoints at 250K and 500K tokens, [1M, 10M) tokens at 1M increments, [10M, 100M) tokens at 10M increments, and [100M, 1B] tokens at 100M increments, resulting in 30 checkpoints per model. We additionally use pretrained Pythia models [2] for a follow-up analysis (e.g., Appendix D).

2.3 Data

For natural texts (Section 3), unless otherwise specified, we use the English subcorpus from the Common Crawl Corpus (`CC100`; 5, 13). We create a sample of 100M tokens from this corpus, using the pretrained GPT2 tokenizer. As mentioned earlier, all models are trained for 10 epochs on this sample, for the total of 1B tokens. For semi-synthetic texts (Section 4), we use a token-to-token transition matrix obtained from the same sample from the `CC100` corpus.

3 Experiment 1: Natural Data

3.1 Methods

As briefly introduced earlier, Aoyama and Wilcox [1] find that training an LM with different batch sizes results in different phase transition points. In this experiment, we change context size and batch size to investigate their effect on the formation of IHs. Specifically, we experiment with log-spaced batch and context sizes of $\{4, 8, 16, 32, 64, 128, 256, 512\}$, and $\{4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$, respectively. Since grid search is expensive, we fix the batch size at 16 while changing the context size, and fix the context size at 1024 while changing the batch size. We additionally test the effect of bigram repetitions alone by selecting subsets of the pretraining data and manipulating the proportion of bigram repetitions.

Batch Size. In Figure 1 (left), we find (1) the larger the batch size, the lower the eventual PS; in other words, a larger batch size leads to weaker IHs at the end of the pretraining, and (2) the smaller the batch size, the later the "spike" in PS; in other words, training an LM with a smaller batch size results in later emergence of IHs, as measured by the number of updates.

Context Size. In Figure 1 (center), we find that (1) the smaller the context size, the later the elbow of the curve is, meaning that the onset of the IH formation is later in pretraining, as measured by the

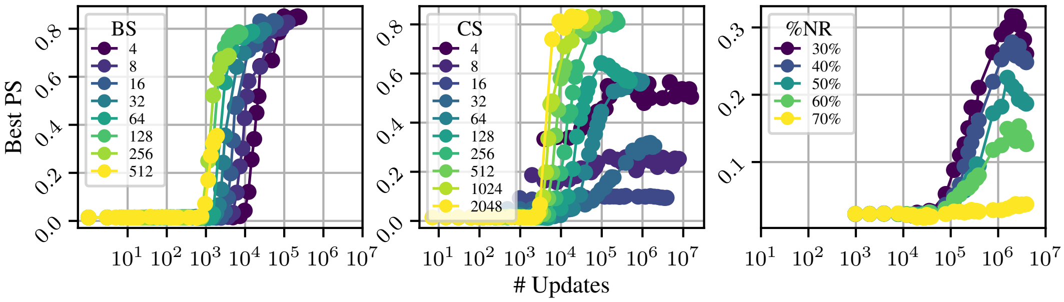

Figure 1: **Developmental trajectories of PS of LMs with various batch sizes (left), context sizes (center), and repetitions (right) over the course of 1B tokens of pretraining.** BS, CS, %NR stands for batch size, context size, and the proportion of chunks with no repetitions, respectively.

number of updates, and (2) the smaller the context size, the flatter the slope once IHs start forming, and the extreme case (context size $\leq$ 16) is a flat line, or the complete suppression of IHs (see Appendix B on how we determine "random" attention). For the shifting effect in (1), since this effect was observed both when changing batch size and context size, we suspect that this could be attributed to the number of tokens the model is exposed to at each update. For the slanting effect in (2), we suspect that, in natural texts, a larger context size will naturally contain more occurrences of $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B} \rangle$ patterns, which may have a threshold below which IHs will not form.

It is important to note that we observe an inverse shifting effect when plotting against the number of pretraining *tokens* in Figure 3 in Appendix A, thereby ruling out the possibility that the observed shifting effect is an artifact of each point on x-axis representing a different number of pretraining *tokens*.

Repetition. The number of bigram repetitions increases as context size grows (see Figure 4 in Appendix C), which we hypothesize to cause the slanting effect of the context size. To tease apart these two phenomena, we manipulate the occurrence rate of bigram repetitions within each chunk, while controlling for the batch size and context size. See Appendix C for more details on how we manipulate the repetition rate while controlling for the context size. In Figure 1 (right), the lower the proportion of chunks with no bigram repetitions, the higher the best PS a given model achieves. Second, there is no "shifting" of the curve, and we only observe the "slanting" effect. This confirms the observation earlier that the "shifting" is due to the number of tokens an LM sees per update, and "slanting" is due to the rate at which an LM encounters repeated bigrams.

Given these observations, we fit a regression model that predicts the IH formation point in training steps and obtain a fitted linear model:

$$U_{PT} = T\sqrt{BC}, \tag{1}$$

where T, B, C, and U_{PT} are the constant, batch size, context size, and the number of updates at which IHs form, respectively. We find a strong correlation between the predicted and actual emergence points ($r = 0.98, p < .0001$). See Appendix D for more details on the fitting procedure and correlation analysis.

4 Experiment 2: Semi-Natural Data Using Natural Bigram Statistics

4.1 Methods

In Figure 1 (right), we manipulated the repetition rate at the chunk level; in other words, we only manipulated the proportion of chunks with at least one bigram repetition. To more precisely manipulate the repetition rate at the token level, we define two metrics, *frequency* and *reliability* of repetition.

Frequency measures the relative frequency at which $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ is observed in a given data, expressed as $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$. It is simply the proportion of tokens in a given data that complete a $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ sequence as the second occurrence of A, where $\mathsf{A} \neq \mathsf{B}$.

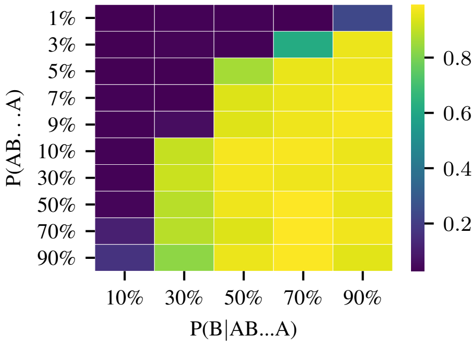

Figure 2: **Best PS across all heads at the end of the training for each frequency reliability combination.** Scores are represented in colors, with brighter colors representing higher scores.

	P(AB...A) P(B|AB...A)		
	10-10	10-30	90-90
$\text{Zipf}_{[+\text{D}+\text{C}]}$	✗	✓	✓
$\text{Zipf}_{[+\text{D}-\text{C}]}$	✗	✗	✓
$\text{Zipf}_{[-\text{D}-\text{C}]}$	✗	✗	✗
$\text{Unif}_{[+\text{D}+\text{C}]}$	✗	✗	✓
$\text{Unif}_{[+\text{D}-\text{C}]}$	✗	✗	✓
$\text{Unif}_{[-\text{D}-\text{C}]}$	✗	✗	✗
$\text{Gaus}_{[+\text{D}+\text{C}]}$	✗	✗	✓
$\text{Gaus}_{[+\text{D}-\text{C}]}$	✗	✗	✓
$\text{Gaus}_{[-\text{D}-\text{C}]}$	✗	✗	✗

Table 1: Inducion head formation in each of the pretraining data generated by the Markov Processes. Each column represents a frequency-reliability configuration.

Reliability measures the conditional probability with which B is observed given $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$, expressed as $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$. It is the proportion of tokens in a given data that complete a $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B} \rangle$ sequence as the second occurrence of B, where $\mathsf{A} \neq \mathsf{B}$, divided by the aforementioned *frequency*. See Appendix E for a detailed walkthrough of these two metrics, and Appendix F for how these two metrics change for various context sizes in natural data.

To create training data that resemble natural data and also satisfy desired values for these two metrics, we first generate a token-to-token transition matrix based on the bigram statistics from natural data, CC100. We then sample from this matrix, while imposing the specified frequency-reliability configuration, and train an LM for each configuration. See Appendix G for more details on the sampling procedure.

4.2 Results

Each cell of Figure 2 represents an LM trained in the configuration specified by the x and y axes. We conduct a grid search over a search space defined by all possible combinations of each metric ranging from {0.1, 0.3, 0.5, 0.7, 0.8}. We initially found that the formation of PS was insensitive to different values of $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ when $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}) \geq 0.1$, and hence conducted an additional grid search over $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}) \in \{0.01, 0.03, 0.05, 0.07, 0.09\}$. With the total of 50 models colored based on the best PS in Figure 2, we can clearly see a Pareto frontier, where a decrease in either value will result in the loss of IHs. Notably, LMs studied here seem to show stronger sensitivity to reliability than to frequency. In the bottom half of Figure 2, [i, j] and [j, i] do not always show the same result. For example, the 10% row always forms IHs except for the 10% column; however, no LMs under the 10% column form an IH. This is notable, given that [i, j] and [j, i] cells have identical numbers of bigram repetitions because $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B}) = P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}) P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$.

5 Experiment 3: Synthetic Data

The previous experiment relied on a Markov Process obtained from a naturally occurring text (i.e., `CC100`). The main goal of this last experiment is to describe the properties of the underlying Markov Process necessary and/or sufficient for IH formation.

5.1 Methods

For simplicity, and to allow for a more precise control over the data properties, we limit our scope to the second order Markov Process as the underlying generative process, which can be expressed as a token-to-token transition matrix $T \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. Once the desired properties (see below) are specified, we optimize the matrix using the Adam optimizer (see Appendix H for details).

We consider three properties that we hypothesize to affect the formation of IHs: (1) local dependency, (2) categoriality, and (3) the shape of the marginal distribution. For (1) local dependency (±D), it is construed as: $+\mathrm{D}$ iff $P(w_{t+1}|w_t) \neq P(w_{t+1})$. In other words, unless the random variable W is i.i.d. at each position t, we consider this variable as $+\mathrm{D}$. This simply means that a distribution is $+\mathrm{D}$ if a word affects what word comes next.

For (2) categoriality ($\pm$C), as IHs have been shown to copy abstract patterns, such as semantic categories (e.g., color-object sequences; 11), we suspect that the presence of categories promotes the formation of IHs. To make this property compatible with the optimization process, we define categoriality by inter-group and within-group similarity scores. We define the presence ($+$C) and absence ($-$C) of categoriality as having within-category similarity of 0.4 and 0.1, respectively. Between-category similarity was always set to 0.1. See Appendix J for the details on similarity score calculation.

Lastly, for (3) marginal token distribution shape, we consider 3 distribution shapes that are increasingly less uniform: Uniform, Gaussian, and Zipfian distributions. This is because natural language is uniquely characterized by a Zipfian distribution, an inverse power law that expresses the frequency of a given word as inversely correlated with its rank, and that it has been shown to affect the emergence of ICL capabilities (e.g., 3).

Taken together, we have 3 (shape of marginal distribution) × 2 (local dependency) × 2 (categoriality) = 12 unique data configurations. We note that a subset of these configurations, specifically the $-$D$+$C configurations, are not conceivable. This is because, since $-$D is defined as a matrix with identical rows (each token's transition distribution is identical), both within-category and between-category similarities are 1. Hence, we have the total of 9 combinations of features, each of which is a process that generates the pretraining data. See Table 3 in Appendix K for the summary of the configurations, as well as the empirical validations of each configuration.

5.2 Results

In Table 1, we first find that no property is by itself a *sufficient* condition for IH formation. For example, the highest bigram repetition condition (column under 90-90) still fails to produce IHs under the $-$D condition. Second, related to the first point, local dependency is a *necessary* condition; no configurations with $-$D form IHs. Intuitively, if each token is i.i.d., the model might not be learning to utilize past tokens in context to facilitate the next token prediction. Third, as we showed in Section 4, we reconfirm that some level of bigram repetition is a *necessary* condition (Section 4). No configurations under the 10-10 column promote the formation of IHs. Lastly, interestingly, marginal distribution and categoriality seem to matter only when the bigram repetition is near the Pareto frontier. Whereas $\pm$C and distribution shape do not affect the IH formation for 10-10 and 90-90 columns (always ✗ for 10-10, and always ✓ for 90-90), when the bigram repetition is at the Pareto frontier (under the 10-30 column), only Zipf$_{[+\mathrm{D}-\mathrm{C}]}$ results in the formation of IHs.

6 Conclusion

In this study, we showed that the emergence of IHs can be predicted by batch size and context size (Section 3). We also showed that the frequency and reliability of bigram repetitions can express a precise Pareto frontier below which IHs cease to form (Section 4). Lastly, we found that, among local dependency, categoriality, the shape of the marginal distribution, frequency, and reliability, none of them alone was a sufficient condition to ensure the formation of IHs (Section 5. However, we find that local dependency coupled with high frequency and reliability always result in IH formation, and that categoriality matters only when the frequency and reliability are close to the Pareto frontier. In this sense, natural language maintains a delicate balance of having sufficient bigram repetitions, while having both local dependency and categoriality, which together promote the formation of IHs in LMs.

References

[1] Tatsuya Aoyama and Ethan Wilcox. Language models grow less humanlike beyond phase transition. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24938–24958, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL `https://aclanthology.org/2025.acl-long.1214/`.

[2] 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. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*, 2023. URL https://arxiv.org/abs/2304.01373.

[3] Stephanie C.Y. Chan, Adam Santoro, Andrew Kyle Lampinen, Jane X Wang, Aaditya K Singh, Pierre Harvey Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=lHj-q9BSRjF.

[4] Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO5PiKHELW.

[5] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

[6] Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, eran malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qaRT6QTIqJ.

[7] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

[8] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

[9] Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

[10] Byung-Doh Oh and William Schuler. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350, 2023. doi: 10.1162/tacl_a_00548. URL https://aclanthology.org/2023.tacl-1.20/.

[11] 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. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[13] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.494/.

[14] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

[15] Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025. URL https://arxiv.org/abs/2502.14010.

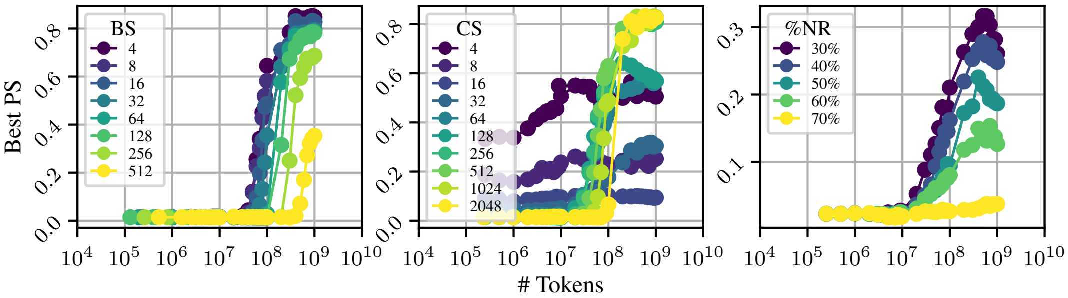

Figure 3: **Developmental trajectories of PS of LMs with various batch sizes (left), context sizes (center), and repetitions (right) over the course of 1B tokens of pretraining.** BS, CS, %NR stands for batch size, context size, and the proportion of chunks with no repetitions, respectively.

A Figure 1 plotted against the number of training tokens

In Figure 1, we plotted how batch size, context size, and bigram repetition rate affect the formation point of IHs, measured in the number of training steps. Trivially, models with different batch sizes and context sizes will have been exposed to different numbers of tokens at the same training step. This could raise the question of whether or not the observed shifting effect is just an artifact of the total number of training tokens being different on the same point on the x-axis. Hence, we show the same graph plotted against the number of total training tokens, instead of the number of training steps in Figure 3. Here agian, we observe the shifting effect, but in the opposite direction: the smaller the batch/context size, the earlier the phase transition point. Therefore, we can say that, even when measured in the number of training tokens, IHs form at different points when we change batch size and/or context size.

B Determining the threshold for random attention

In Figure 1 (center), the lines representing context sizes of 4, 8, and 16 seem to be somewhat flat, meaning that the model does not improve in its ability to attend back to the token necessary to complete the repeated bigram. The high PSs associated with these models are simply due to the higher attention preceding tokens can get by chance; with the context size of 4, for example, the model has at most 4 tokens to attend back to, with the random attention of 0.25.

To systematically determine what counts as "above random," we simulate the random attention over previous tokens via Markov Chain Monte-Carlo (MCMC) using a Dirichlet distribution with a uniform prior. We find that attention weights as strong as 0.72, 0.35, and 0.16 are necessary for models with context sizes of 4, 8, and 16, respectively, to be considered *above random* at the alpha level of 0.01. Hence, we conclude that these three context sizes do not promote the formation of IHs, and in the remaining analyses in this subsection, we will focus on the rest of the models.

C Distribution of bigram repetitions with various context sizes

Since adding or removing bigrams in naturally occurring texts introduces noise, such as broken syntax, we manipulate the frequency of repeated bigrams in natural language data by first putting tokenized texts into chunks of c, where c is the context size of the LM, and then selecting those natural chunks to ensure $p\%$ of the chunks of the resulting training data include no bigram repetition at all, where p is the parameter we can control. Modern LMs have a context size of at least 1024. However, in naturally occurring texts, sequences of 1024 tokens without any repeated bigrams $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B} \rangle$ in them are very rare, if not non-existent. In general, as shown in Figure 4, larger context size trivially tends to contain more repeated bigrams, and we find that the context size of 64 strikes the balance between including enough context and containing a good number of chunks with and without repeated bigrams. Hence, in Section 4 and Section 5, we use LMs with a context size of 64.

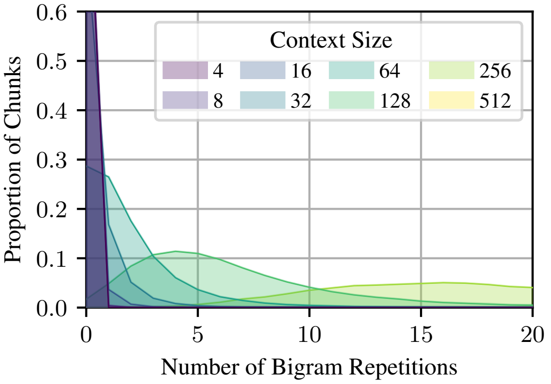

Figure 4: **Smoothed distribution of chunks with various numbers of bigram repetitions.** Context sizes 1024 and 2048 were rendered invisible, and hence removed from the plot. The plot is truncated at y=0.6 for readability, but context sizes of 4, 8, and 16 had >95% of chunks with no bigram repetitions.

D Predicting the emergence points

We have seen that context size and batch size affect the phase transition point (Section 3). Here, we ask: can we predict the phase transition point only using the training configuration as variables (i.e., before we train the model)? In existing studies, points at which a phenomenon of interest happens are often defined in terms of the number of pretraining tokens (e.g., 11, 10), or in floating-point operations (FLOPs; e.g., 8). It has been shown that LMs seem to go through a phase transition at different points in terms of the number of pretraining tokens [1] and the number of updates (Section 3), and hence it is not a good candidate to describe the phase transition point. We can also infer that FLOPs is not a reliable measure to capture this either, because (1) FLOPs is a function of input and model size, and (2) models that are orders of magnitude different in size seem to go through a phase transition at around the same time in terms of the number of pretraining tokens (e.g., 11).

Hence, we propose a simple model size-agnostic law that predicts the number of updates at which a given LM goes through the phase transition based on the context size and batch size, which takes the form:

$$U_{\text{PT}} = e^{\alpha} B^{\beta} C^{\gamma} \tag{2}$$

where U_{PT} is the number of updates at which a given model goes through phase transition. e^{α} serves as an intercept, as we will see below. Following Kaplan et al. [8], we estimate the parameters α, β, and γ using a simple ordinary least squares regression in log space:

$$\log U_{\text{PT}} = \alpha + \beta \log B + \gamma \log C \tag{3}$$

Fitting this model, we obtain $\alpha = 13.5$, $\beta = -0.51$, and $\gamma = -0.56$. We approximate these two parameters as $\beta = \gamma = -0.5$ as this promotes interpretability and facilitates readability. Now, a constant raised by a constant is a constant, so let us call $e^{\alpha} = 750000$ a constant T. Plugging these parameters back into Equation (2), we obtain:

$$\begin{aligned} U_{\text{PT}} &= \frac{T}{\sqrt{BC}} \\ T &= U_{\text{PT}}\sqrt{BC} \end{aligned} \tag{4}$$

The key intuition behind Equation (4) is that the model-agnostic constant T is a function of the *quantity* of training, or the number of updates U_{PT}, at which phase transition occurs. At the same time, the *quality* of each update matters, and it correlates with the number of tokens the model sees at each update, which is a function of context and batch sizes $\sqrt{BC}$. Let us call the generalized form of the right hand side (RHS) of this equation, $U\sqrt{BC}$, the number of token-weighted updates (TWUs), given that it is the number of updates scaled by batch size and context size to incorporate the number of tokens seen at each update. Equation (4) suggests that the number of TWUs at which phase transition occurs can be expressed as a constant T across model and training configurations.

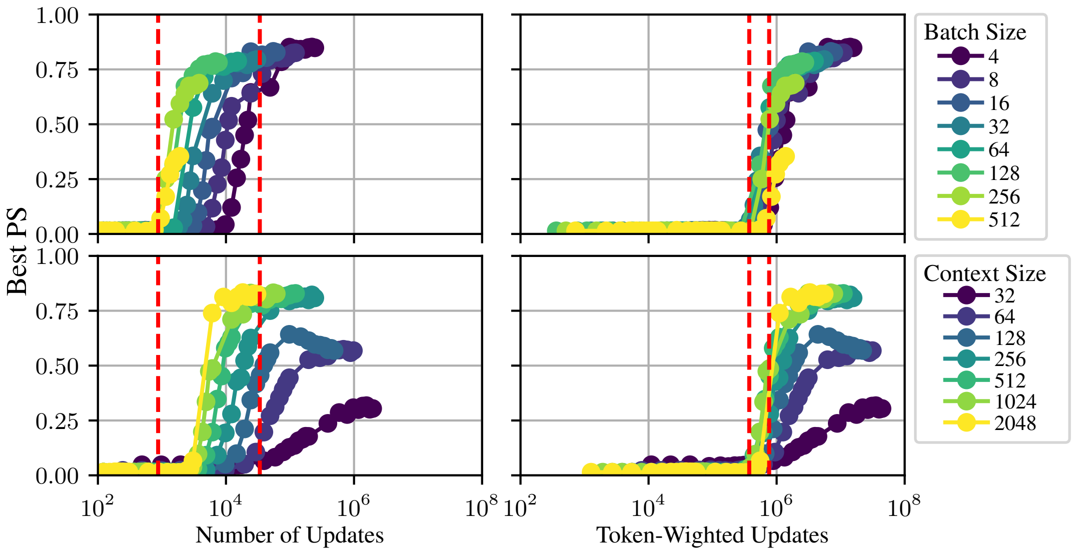

Figure 5: **Development of the highest PS score across all heads at each checkpoint plotted against the number of updates (left column) and TWUs (right column).** Different line colors represent different batch sizes (top row) and context sizes (bottom row). All four plots share the scales of the x/y-axes. The x-axis is in log-scale, since both batch and context sizes increase exponentially, and hence the number of updates decreases exponentially. The 2 red dotted lines per plot represent the column-wise range of the inflection points.

To visualize the invariance of phase transition points measured by the number of TWUs across various training configurations, we plot each model's (of different batch and context sizes) best PS over the course of pretraining against the number of updates (left) and against $U\sqrt{BC}$ (right) in Figure 5. For each curve, we fit a piece-wise liner function (PWLF) with 3 segments, as we observe the initial flat line, and a sharp increase, and then an eventual plateau. We take the "knot" between the first and second segments of each fitted PWLF as the transition point, and plot the column-wise range with red dotted lines. Because $T = U\sqrt{BC}$ and T remains constant across different models, we expect to see a single point at which the models go through a phase transition. Indeed, we can see that phase transition occurs across a wide range in terms of the number of updates (left), but the range becomes much smaller and the transition point seems to converge at a single point once it's measured in terms of the number of TWU (right).

To further verify that this simple law indeed predicts the phase transition point of LMs trained with various training configurations, we can reformulate Equation (4) to predict the number of tokens N:

$$\begin{aligned} T &= U_{\text{PT}}\sqrt{BC} \\ T\sqrt{BC} &= U_{\text{PT}}BC \end{aligned} \tag{5}$$

Because $N = UBC$ by definition, RHS is N, and we get:

$$N_{\text{PT}} = T\sqrt{BC} \tag{6}$$

The left hand side (LHS) of Equation (5) is the *observed* number of pretraining tokens at which phase transition occurs, and the RHS is the *predicted* point based on context size C and batch size B, as well as the empirically found constant $T = 10^{5.7}$. Now we can predict the number of tokens N at which phase transition occurs, based on a constant T and training configurations C and B. In Figure 6, x-axis and y-axis correspond to the LHS and RHS, or the predicted and observed number of pretraining tokens at which phase transition occurs, respectively. We can see a strong correlation of $r > .99$ before the log-transformation (for readability, as shown in Figure 6), and $r = .98$ after the log-transformation, where $p < .001$ for both. This law holds for more than 2 orders of magnitude, echoing the robust correlation between psychometric predictive power (PPP) tipping points and phase transition points that held across 64M to 2B pretraining tokens [1].

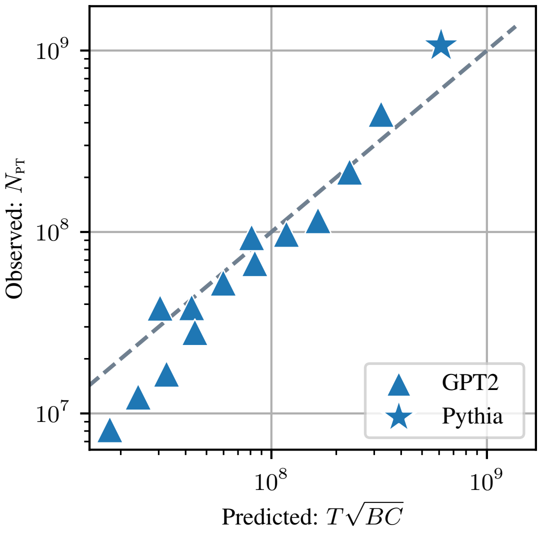

Figure 6: **Predicted and observed number of pretraining tokens at which phase transition occurs.** x-axis and y-axis represent *predicted* and *observed* points of phase transition, expressed in the number of pretraining tokens, respectively. A strong correlation of $r = .98$ ($p < .001$) is found.

E Frequency $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and Reliability $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$

Recall that Elhage et al. [7], Olsson et al. [11] define IHs as specific heads that complete the repeated $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B} \rangle$ sequence when seeing $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$. As a naive hypothesis, we speculate that the presence of repeated bigrams (separated by an arbitrarily long sequence of tokens within the LM's context size) is essential for IH formation. Consider the following sequence, which we will use as a running example: In this example, there are 4 tokens that occur more than once (i.e., A, B, E, F).

Word	A	B	C	D	E	F	A	B	E	F
R_U	0	0	0	0	0	0	1	1	1	1
R_B	0	0	0	0	0	0	1	0	1	0

Table 2: R_U and R_B of an example sequence. This sequence has a $P(R_U) = \frac{4}{10}$ and $P(R_B) = \frac{2}{10}$, hence **frequency** and **reliability** are 0.4 and 0.5, respectively.

For the first occurrence of these 4 tokens, each of them is followed by *something*; for example, A is followed by B, and B is followed by C. For the second occurrence of these 4 tokens, one can induce that a bigram tends to repeat *in-context*, if these 4 tokens are reliably followed by what followed them the first time they occurred; namely, if A, B, E, F, are followed by B, C, F, A, respectively. In Table 2, there exist 4 such opportunities for learning the bigram repetition, and 2 such opportunities actually reward such learning, since only the second occurrences of A and E are followed by the same bigram continuations of their first occurrences. Here, we have touched upon the two knobs we aim to define here. **Frequency** involves the tokens that are of the word type occurring for the n-th time where $n \geq 2$ (A, B, E, F in Table 2). **Reliability** measures, of all such tokens, how many of them actually complete the same bigram continuation (B, E in Table 2). We will now define each of these two measures formally.

Frequency: given a binary variable *unigram repetition* $R_U \in \{0, 1\}$, whose value is 1 if a given token is the second occurrence of A of the $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ sequence (i.e., a repeated unigram), and 0 otherwise, we define "frequency" as $P(R_U)$. Informally, in the induction term, this is $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$, or the rate at which $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ sequence is encountered. In practice, this is equivalent to the sum of all unigram counts in a given sequence, with the first occurrence removed:

$$P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}) = \frac{1}{|\mathbf{s}|} \sum_{w \in \mathcal{V}} \max(c(w, \mathbf{s}) - 1, 0) \qquad (7)$$

where $\mathcal{V}$ is the vocabulary, or the set of all possible unigrams, and $c(w, \mathbf{s})$ is a count of a word w in a sequence $\mathbf{s}$ of the same size as the model's context size. In Table 2, we saw 4 such tokens (tokens where $R_U = 1$: A, B, E, F) in the context of size 10, hence $P(R_U) = P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}) = \frac{4}{10}$. Technically, consecutive occurrences of a given token type should not be counted; however, this does not happen frequently in the real data, and no tokens were allowed to occur twice in a row in synthetic data, trivializing this problem (see Algorithm 1).

Reliability: given a binary variable *bigram repetition* $R_B \in \{0, 1\}$, whose value is 1 if a given token's *next token* is the second occurrence of B of the $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B} \rangle$ sequence (i.e., a repeated bigram), and 0 otherwise, I define "reliability" as $\frac{P(R_B)}{P(R_U)}$. Informally, in induction term, $P(R_B)$ is $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B})$, and reliability, or $\frac{P(R_B)}{P(R_U)}$, is equivalent via the chain rule to the conditional probability $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$: the rate at which an $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ sequence is followed by B:

$$\begin{aligned} & P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}) \\ &= \frac{P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B})}{P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})} \\ &= \frac{P(R_B)}{P(R_U)} \\ &= \frac{\frac{1}{|\mathbf{s}|} \sum_{b \in \mathcal{B}} \max(c(b, \mathbf{s}) - 1, 0)}{\frac{1}{|\mathbf{s}|} \sum_{w \in \mathcal{V}} \max(c(w, \mathbf{s}) - 1, 0)} \end{aligned} \tag{8}$$

where $\mathcal{B}$ denotes the set of all possible bigrams, and $c(b, \mathbf{s})$ the count of the bigram b in a given sequence $\mathbf{s}$. In Table 2, of all the 10 positions (tokens), 2 completed the $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B} \rangle$ sequence (tokens where $R_B = 1$: second occurrences of B and F), hence $P(R_B) = P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B}) = \frac{2}{10}$. Therefore, reliability is $\frac{2}{10} \div \frac{4}{10} = \frac{1}{2}$. It might make more intuitive sense to compute this directly without using the chain rule: of all the 4 tokens that complete the $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ sequence, 2 of them are followed by B, hence $\frac{2}{4} = \frac{1}{2}$. Equivalently, of the tokens where $R_U = 1$ in Table 2, half of them also have $R_B = 1$. However, for the computational purpose, the chain rule is much simpler, which is the reason we introduced the chain rule based calculation above. In this study, "frequency" and $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ are used interchangeably, and so are "reliability" and $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$.

Note that this is a more precise characterization of what Chan et al. [3] called "burstiness." In their formulation, where the task was to predict the label of an image given image-label pairs in-context, a "bursty" sequence contains certain image-label pairs more often than others, while controlling for the marginal distribution over all sequences. This is effectively equivalent to increasing both frequency and reliability in our terms. Two sequences of image-label pairs $(\mathsf{I_i}, \mathsf{L_i})$: $\langle \mathsf{I_1}, \mathsf{L_1}, \mathsf{I_2}, \mathsf{L_2}, \mathsf{I_3}, \mathsf{L_3}, \mathsf{I_4}, \mathsf{L_4} \rangle$ (non-bursty) and $\langle \mathsf{I_1}, \mathsf{L_1}, \mathsf{I_2}, \mathsf{L_2}, \mathsf{I_1}, \mathsf{L_1}, \mathsf{I_3}, \mathsf{L_3} \rangle$ (bursty) can be reformulated as $\langle \mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}, \mathsf{F}, \mathsf{G}, \mathsf{H} \rangle$ and $\langle \mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{A}, \mathsf{B}, \mathsf{E}, \mathsf{F} \rangle$, respectively, and the former has frequency and reliability of 0, whereas the latter has frequency and reliability of 1/8 and 1/2 ($\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A} \rangle$ is followed by B, but $\langle \mathsf{B}, \mathsf{C}, \ldots, \mathsf{B} \rangle$ is not followed by C), respectively.

F Frequency $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and Reliability $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ in Natural Text

We can verify that natural texts with various chunk sizes are not only different in terms of the number of repeated bigrams (as shown in Figure 4), but also in the two probabilities defined above. Figure 7 confirms this: both $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ increases as the context size increases. While $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ seems to increase log-linearly (x-axis is in log scale), the increase in $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ seems to slow down. Since IHs were not forming, or very weak at most, for context size of 32, we speculate that the threshold values of $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ to be somewhere between 0.1–0.2, and 0.1–0.15, respectively.

G Algorithm for Frequency- and Reliability-Constrained Training Data Generation

To fully control the two properties, frequency $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and reliability $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ in text data, we need a way to sample words from some distribution, while enforcing desired values

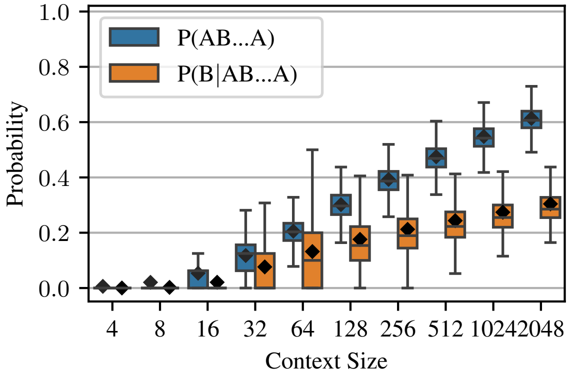

Figure 7: **Distribution of chunks with various $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ for each context size.** Only the quartile box, median (center line in each box), mean (diamond), and whiskers are shown, and Outliers are not shown for readability.

of these two knobs. To this end, we approximate natural language by first tokenizing texts from the English subcorpus of the Common Crawl Corpus (`CC100`; 5, 13) and collecting token bigram statistics. We then create a token-to-token transition matrix $T \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $\mathcal{V}$ is the vocabulary. We use off-the-shelf GPT2 tokenizer, but reduced the vocabulary size to 10,000. In Algorithm 1, we outline the semi-synthetic data generation algorithm, where $(\mathcal{A}\backslash\mathcal{B})$ denotes an asymmetric difference, or a set of elements in $\mathcal{A}$ but not in $\mathcal{B}$, $\{x \mid x \in \mathcal{A}$ and $x \notin \mathcal{B}\}$. $\mathcal{U}_{<t-1}$ is a set of unigrams attested before time step $t-1$, and $\mathcal{B}_x$ denotes a set of attested bigram continuations of token x. The idea is that we first generate the first half of the sequence by randomly walking through the Markov chain. This is to ensure that a sufficient number of unique tokens are present in the sequence before the restricted generation can take place; otherwise, Algorithm 1 often produces degenerate sequences especially with high values of $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$. For the second half, based on the condition (if the prefix constitutes $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}\rangle$) and constraints (to make a new $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}\rangle$ and/or $\langle \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A}, \mathsf{B}\rangle$), the word at each time step w_t is sampled from the distribution $\mathcal{D}$ restricted to some subset $\mathcal{S}$ that satisfies the conditions and constraints. For simplicity, no tokens were allowed to occur consecutively.

H Matrix Optimization

In Section 5, we defined the underlying Markov Processes as transition matrices, which were then used to generate synthetic data, via a set of desired properties. These transition matrices were optimized using the Adam optimizer to satisfy the desired properties as closely as possible.

$$\mathcal{L} = \lambda_1\mathcal{L}_D + \lambda_2\mathcal{L}_E + \lambda_3\mathcal{L}_P + \lambda_4\mathcal{L}_{WC} + \lambda_5\mathcal{L}_{WA} \tag{9}$$

where the 5 terms correspond to distribution loss, entropy loss, peakedness loss, within-category loss, and across-category loss, respectively. The 5 λs correspond to the weighting factors.

Distribution loss ($\mathcal{L}_D$). This term is to ensure the transition matrix $T \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ has a marginal distribution of the desired shape (Uniform, Gaussian, or Zipfian, as discussed in Section 5). We penalize the divergence from the desired shape by including KL divergence between the actual marginal distribution and the desired distribution:

$$\mathcal{L}_D = \sum_{i=1}^{|\mathcal{V}|} P(w_i)\frac{\log P(w_i)}{\log Q(w_i)}, \tag{10}$$

where P and Q are desired and actual marginal distributions, respectively.

Entropy loss ($\mathcal{L}_E$). This term is to ensure each transition matrix is comparable in predictability. Because a transition matrix sampled from natural language had the entropy of ≈6.2, we set the target entropy value to be 6.2, and included the squared difference as a loss term:

$$\mathcal{L}_E = (H_{\text{target}} - H(T))^2, \tag{11}$$

Algorithm 1 Corpus generation constrained on $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$

```
Sample a token sequence s from the distribution $\mathcal{D} : \{w \mapsto \mathcal{D}_w \in \mathbb{R}^{|\mathcal{V}|} \mid w \in \mathcal{V}\}$,
constrained on $\alpha = P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$ and $\beta = P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$
Input: cxt_size, $\mathcal{V}$, $\mathcal{D} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, $\alpha$, $\beta$
s ← []                                       ▷ init a token sequence
$\mathcal{U}_t \leftarrow \{\}$                                    ▷ init a unigram set at step $t$
$\mathcal{B}_t \leftarrow \{\}$          ▷ init a bigram continuation dict $\{w \mapsto \mathcal{B}_t(w) \mid w \in \mathcal{U}_t\}$ at step $t$
$m \leftarrow$ ctx_size//2
for $t$ from 1 to $m$ do                             ▷ first half
    $w_t \sim \mathcal{D}_{w_{t-1}}(\mathcal{V}\backslash\mathcal{U}_{<t})$
    add $w_t$ to s
    update $\mathcal{U}_t$ and $\mathcal{B}_t$
for $t$ from $m$ to cxt_size do                      ▷ second half
    is_aba ← $w_t$ in $\mathcal{U}_{t-1}$
    make_aba ← random.random() $\leq \alpha$          ▷ $\alpha$ represents $P(\mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$
    make_abab ← random.random() $\leq \beta$          ▷ $\beta$ represents $P(\mathsf{B} \mid \mathsf{A}, \mathsf{B}, \ldots, \mathsf{A})$
    if is_aba(w) then
        if make_abab then
            $w_t \sim \mathcal{D}_{w_{t-1}}(\cdot \mid \mathcal{B}_t(w_{t-1}))$
        else
            if make_aba then
                $w_t \sim \mathcal{D}_{w_{t-1}}(\cdot \mid \mathcal{U}_{<t}\backslash\mathcal{B}_t(w_{t-1}))$
            else
                $w_t \sim \mathcal{D}(\cdot \mid \mathcal{V}\backslash\{\mathcal{U} \cup \mathcal{B}_t(w_{t-1})\})$
    else
        if make_aba then
            $w_t \sim \mathcal{D}_{w_{t-1}}(\cdot \mid \mathcal{U}_{<t})$
        else
            $w_t \sim \mathcal{D}_{w_{t-1}}(\cdot \mid \mathcal{V}\backslash\mathcal{U}_{<t})$
    add $w_t$ to s
    update $\mathcal{U}_t$ and $\mathcal{B}_t$
```

where the estimation of $H(T)$ is detailed in Appendix I.

Peakedness loss ($\mathcal{L}_P$). We find that the matrix optimization often suffers from a degenerate matrix, where the desired properties are satisfied by allocating a very large probability mass to a single token in each row. To mitigate this problem, we include the mean of row-wise largest probabilities:

$$\mathcal{L}_P = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \max T_{i,:}, \tag{12}$$

where $T_{i,:}$ is the i-th row of the matrix T.

Within-category loss ($\mathcal{L}_{WC}$) and across-category loss ($\mathcal{L}_{AC}$). As defined in Section 5, the within-category similarity is the mean similarity of all pairs of words within a category, whereas the across-category similarity is the mean similarity of all pairs of words from different categories. For the $+$C configuration, we set the target within- and across-category similarities to be 0.4 and 0.1, respectively, and both were set to be 0.1 for the $-$C configuration. The squared differences between the actual and desired within/across category similarities were included as loss terms. The within-category loss is defined as:

$$\mathcal{L}_{WC} = (\text{WC}_{\text{target}} - \text{WC}(T))^2, \tag{13}$$

where WC is within-category similarity:

$$\frac{1}{N} \sum_{c \in C} \sum_{w, w' \in c, w \neq w'} sim(w, w') \tag{14}$$

Similarly, across-category loss is defined as:

$$\mathcal{L}_{WC} = (\text{WC}_{\text{target}} - \text{WC}(T))^2, \tag{15}$$

where AC is across-category similarity:

$$\frac{1}{N}\sum_{c,c'\in C, c\neq c'}\sum_{w\in c}\sum_{w'\in c'} sim(w,w') \tag{16}$$

We optimize the matrix with these loss terms for 5,000 steps with $\lambda_1 = 100$, $\lambda_2 = 0.01$, $\lambda_3 = 0.1$, $\lambda_4 = \lambda_5 = 5$.

I Conditional Entropy

Conditional entropy is defined as:

$$H(X \mid Y) = \sum_{y\in Y} P(y)\left[\sum_{x\in X} P(x\mid y)\log\frac{1}{P(x\mid y)}\right] \tag{17}$$

For a transition matrix $T \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $\mathcal{V}$ is the vocabulary, it can be expressed as:

$$H(T) = \sum_{i=1}^{|\mathcal{V}|} P(w_i)\left[\sum_{j=1}^{|\mathcal{V}|} P(w_j\mid w_i)\log\frac{1}{P(w_j\mid w_i)}\right] \tag{18}$$

Since the transition matrix T is a row-stochastic matrix, and each row sums to 1, the conditional probability $P(w_j|w_i)$ is an entry $T[i,j]$. The marginal probability $P(w_i)$ can be estimated by obtaining the stationary distribution of the transition distribution T, which is a left eigenvector with the eigenvalue of 1. Assume a ground-truth stationary distribution π; this stationary distribution, which is the unigram distribution the transition matrix converges to, should remain unchanged after transitions:

$$\pi T = \pi \tag{19}$$

Eigenvector is a vector that only gets scaled by a factor λ after a linear transformation L. Hence, we can find the stationary unigram distribution π by finding the left eigenvector with the eigenvalue of 1 of a linear transformation T.

Now that we have the stationary distribution π, Equation (18) can be expressed as a matrix multiplication:

$$H(T) = \sum_{i=1}^{|\mathcal{V}|} \pi[i]\cdot H(T[i,]) \tag{20}$$

J Inter- and Intra-Group Similarity Scores

We first assign $\frac{|\mathcal{V}|}{|\mathcal{C}|}$ tokens into each category $c \in \mathcal{C}$, thereby creating creating $|\mathcal{C}|$ groups with disjoint members. We then define inter-group similarity as the average cosine similarity between words (i.e., each of the $|\mathcal{V}|$ rows of the transition matrix $\mathcal{T} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$) from a given category c and words from a different category c':

$$\frac{1}{N}\sum_{c,c'\in\mathcal{C}, c\neq c'}\sum_{w\in c}\sum_{w'\in c'} sim(w,w') \tag{21}$$

where N is the number of such word pairs. Within-category similarity is likewise defined as the average similarity between all pairs of words that belong to the same category. If a distribution has a high within-category similarity but a lower across-category similarity, it means that categories exist in this distribution.

K Summary of Data Configurations

Table 3 summarizes the properties as well as the actual statistics of each of these distributions. $H(\cdot)$ measures the conditional entropy of the distribution, estimated by taking the sum of the entropies of

	Properties			**Statistics**			
	Marginal	LD	CAT	H$(\cdot)$	Intra-group	Inter-group	$\mathrm{D_{KL}}(\cdot \mid\mid \text{target})$
Zipf$_{[+\mathrm{D}+\mathrm{C}]}$	Zipfian	✓	✓	6.2142	0.3987	0.1007	0.0001
Zipf$_{[+\mathrm{D}-\mathrm{C}]}$		✓	✗	6.1988	0.0999	0.1010	0.0001
Zipf$_{[-\mathrm{D}-\mathrm{C}]}$		✗	✗	9.5239	-	-	-
Unif$_{[+\mathrm{D}+\mathrm{C}]}$	Uniform	✓	✓	6.5388	0.3719	0.0010	4e-6
Unif$_{[+\mathrm{D}-\mathrm{C}]}$		✓	✗	6.4759	0.0873	0.0104	0.0001
Unif$_{[-\mathrm{D}-\mathrm{C}]}$		✗	✗	13.2734	-	-	-
Gaus$_{[+\mathrm{D}+\mathrm{C}]}$	Gaussian	✓	✓	6.2435	0.3995	0.1003	0.0001
Gaus$_{[+\mathrm{D}-\mathrm{C}]}$		✓	✗	6.2407	0.0999	0.1002	0.0001
Gaus$_{[-\mathrm{D}-\mathrm{C}]}$		✗	✗	12.3490	-	-	-

Table 3: Markov Processes used for pretraining data generation in Experiment 3. Each row represents a matrix that defines the Markov Process. The **Properties** column lists desired properties the matrix was optimized for, and the **Statistics** column summarizes the actual statistical properties each matrix had at the end of the optimization process. LD and CAT represents the binary variables local dependency and categoriality, respectively. H measures the entropy of the data, and KL divergence measures the fit between the desired distribution (as shown in the **Properties** column) and the actual marginal distribution of the generated matrix.

each row $\mathbf{r} \in \mathbb{R}^{|\mathcal{V}|}$, weighted by the stationary distribution (see Appendix I for details). Note that the last row of each block, marked by the $-$D configuration, has a higher entropy. This is because, for the $-$D configuration, each row of the matrix is identical to each other, and each word in this distribution is i.i.d. Hence, once the shape of the marginal distribution is specified, the entropy of this matrix is automatically determined. For example, Unif$_{[-\mathrm{D}-\mathrm{C}]}$ is by definition the maximally entropic distribution over $|\mathcal{V}|$ items. $\mathrm{D_{KL}}(\cdot \mid\mid \text{target})$ is the KL divergence between the desired marginal distribution and the actual marginal distribution of the generated matrix. We can see that the divergence is very small for all distributions. Intra-group similarities, inter-group similarities, and $\mathrm{D_{KL}}(\cdot \mid\mid \text{target})$ for the Dist$_{[-\mathrm{D}-\mathrm{C}]}$ configs are 1, 1, 0, respectively, by definition.