On the Effect of Hyperparameters in Language Modeling for Computational Linguistics

Ruoxi Ning[1,2], Yongpeng Zhu[1], Qingcheng Zeng[3], Tatsuki Kuribayashi[4], Freda Shi[1,2]
[1]University of Waterloo, [2]Vector Institute, [3]Northwestern University, [4]MBZUAI
{ruoxi.ning, fhs}@uwaterloo.ca

Abstract

Training language models and examining their linguistic behaviors have been a common protocol in computational linguistics for studying linguistic phenomena and modeling human language processing. However, work in this area is often limited to proof-of-concept demonstrations with arbitrary model configurations, without considering hyperparameter sensitivity, an important source of variation in model performance. In this work, we replicate three prior studies (Chang and Bergen, 2022; Hu et al., 2020b; Kuribayashi et al., 2024) with hyperparameters varied within a practical range, and show that modest hyperparameter changes can alter some qualitative conclusions about models' linguistic abilities and even reverse the ranking of model performance. Our results highlight the risk that prior work may have reflected optimization artifacts rather than the genuine inductive biases of model classes, and that hyperparameter sensitivity should receive more attention as a factor that can meaningfully influence model behavior. We suggest future work to report the variation of performance across the configuration space to enhance the reliability and generalizability of conclusions.

Code: compling-wat/tune-linguistic-lms.

1 Introduction

Language models (LMs) have increasingly become a subject of linguistically motivated experiments on a wide range of topics from low-level fine-grained learning curves of grammar rules to high-level simulation of typological distribution of language structures (Regier, 2005; Wilcox et al., 2020; Misra and Mahowald, 2024; Yang et al., 2025, *inter alia*). One appeal of LM-based linguistic experiments stems from their enabling of controlled ablation of learning scenarios (Warstadt and Bowman, 2022) and precise tracking of learning dynamics. Thus, a considerable portion of work involves training models from scratch under controlled conditions. These studies provided new perspectives on how models acquire and understand word and grammar rules, on cross-architecture comparisons of linguistic ability, and on comparisons between human and model cognition.

Such experiments can often be limited to demonstrating the proof of concept with an arbitrary model instance, or training without model tuning. However, it is commonly accepted in the machine learning domain that hyperparameters including learning rate, initialization, optimization settings, and type of computation, affect the training dynamics and final behavior (Larochelle et al., 2007; Choi et al., 2019; Dodge et al., 2019; Shi et al., 2020, *inter alia*). Meanwhile, the concept of reproducibility has been a guiding principle in machine learning (Goodman et al., 2016; Belz et al., 2021). Much recent machine learning research cannot achieve solid reproducibility when performed with arbitrarily chosen hyperparameters (Bouthillier et al., 2019). Under-reporting or arbitrarily choosing hyperparameters will hinder the genuine observation of the model's behavior and the generalization the research claims, since the quantitative evaluation partly reflects the effect of hyperparameters.

In this paper, we systematically examine how hyperparameter variation influences linguistically oriented findings in neural networks. We revisit three representative linguistically motivated experiments on LMs, ranging from low-level word-learning curves to syntactic knowledge, and then to the typological distribution of linguistic features: (A) word acquisition patterns, (B) syntactic generalization ability, and (C) word order preferences. In each experiment, we conduct a systematic grid search to monitor the extent to which their reported effects are sensitive to hyperparameter changes. We identify which findings are most fragile under hyperparameter variation—whether in the ranking of model abilities or in the sign and significance of statistical tests—and offer recommendations on

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 41863–41880

designing experiments and reporting results to support more stable conclusions. We summarize our contributions as follows:

1. We replicate three established studies in computational linguistics and find that part of their conclusions exhibit low stability under modest variations, suggesting that limited hyperparameter variation can already shift some linguistic outcomes.
2. Through discussions on the effect of hyperparameter tuning, types of fragile conclusions, and suggestions for stable conclusions, we offer recommendations for future work to improve the reliability and generalizability of findings in linguistically motivated experiments with LMs.

2 Related Work

Linguistically-motivated studies with language models. The intersection of linguistics and language modeling has produced a broad body of research evaluating whether neural networks capture linguistic structure from natural language data. Studies have examined phonological patterns, morphological productivity, syntactic competence, and semantic composition (Alper and Averbuch-Elor, 2023; Ismayilzada et al., 2025; Mitchell and Lapata, 2010, *inter alia*). Common approaches include probing tasks, targeted syntactic evaluations, psycho-linguistically inspired behavioral tests, or verification related to information theory rules (Marvin and Linzen, 2018; Futrell et al., 2019; Meister et al., 2021; Xu et al., 2025, *inter alia*). Extending these efforts, computational studies have further investigated whether neural models exhibit linguistic competence by linking next-word prediction probabilities to human behavioral and cognitive measures such as surprisals, frequency effects, and neural responses (Portelance and Jasbi, 2024a; Beinborn and Hollenstein, 2023). These findings suggest a degree of alignment between model predictions and human processing, and typically hypothesize that training yields stable and well-formed linguistic representations. Most of them rarely consider how optimization settings may influence the observed linguistic patterns.

Hyperparameter tuning. Hyperparameter tuning strongly affects neural model behavior and observations from model training (Larochelle et al., 2007; Choi et al., 2019; Dodge et al., 2019; Shi et al., 2020, *inter alia*). Small changes in learning rate, initialization, optimizer, etc., are crucial factors in determining whether the model will fall into a local minimum and whether high accuracy can ultimately be achieved (Bishop, 2006; Arnold et al., 2024; Franceschi et al., 2024, *inter alia*). Recent empirical work has criticized drawing firm conclusions from models trained with arbitrary hyperparameters, emphasizing instead the need to examine how tuning choices affect performances (Larochelle et al., 2007; Choi et al., 2019; Dodge et al., 2019). Cooper et al. (2021) further argues that the process of hyperparameter optimization itself is worth systematic study.

Our work lies at the intersection of the two lines of research above. Rather than proposing new methods or tuning algorithms, we revisit three established linguistically motivated experiments under controlled hyperparameter variation, and assess which conclusions remain stable and which shift once tuning is taken into account.

3 Experiment

This section documents the replication procedure on our machine, the observed results when training across several groups of hyperparameters, and discussions of their stability under hyperparameter changes across the three chosen experiments.

3.1 Experiment A: Word Acquisition Patterns

The language-acquisition ability of neural networks, especially their word-acquisition patterns, is an indicator of their linguistic ability (Portelance and Jasbi, 2024b). Learning theory has classified the acquisition of word-level patterns as distributional learning since it relies on the distributional statistics of input texts (Lenci, 2018; Boleda, 2020). Discussions have been made on the divergence of models and humans in how they learn languages (Huebner et al., 2021; Chang and Bergen, 2022; Evanson et al., 2023), and exclusive patterns that machines learn expressions (Misra and Mahowald, 2024; Constantinescu et al., 2024).

We first replicate the word acquisition experiment of Chang and Bergen (2022), which compares neural models' word acquisition patterns with humans'. Four model architectures (LSTM, bidirectional LSTM, BERT, GPT-2) trained on a mixture of BookCorpus and WikiText-103 are evaluated on their word learning curves. During training, the models' checkpoints are recorded at a specified frequency for further analysis, including fitting the learning curve, recording output token probabili-

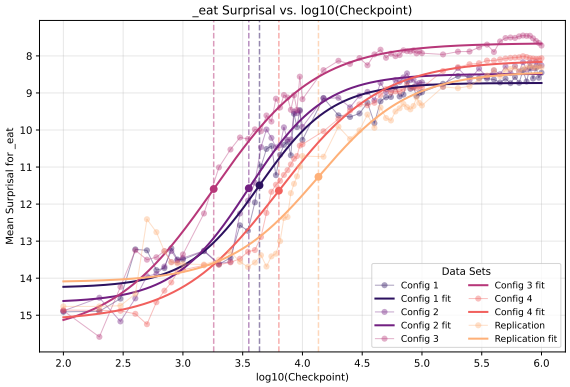
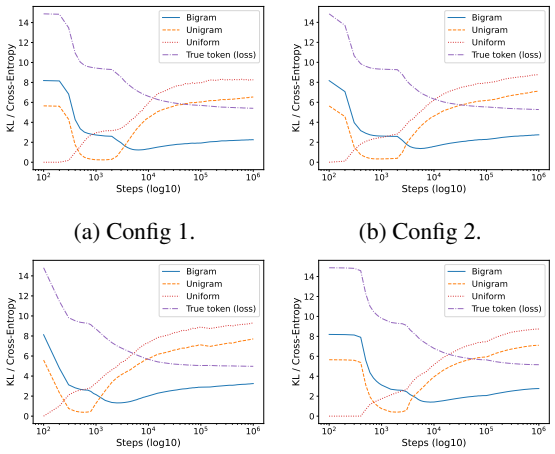


Figure 1: Learning curves of the word 'eat' for the LSTM model trained on each configuration.



(a) Config 1. (b) Config 2.

(c) Config 3. (d) Config 4.

Figure 2: Curves of KL-divergence between the real distributions and learned predictions during training across different configurations.

ties, and calculating age-of-acquisition indicators. The details of data, model, and training procedure are in Appendix A.1.

The conclusions of this paper can be summarized into the following ones: (A1) identified factors can predict words' age of acquisition (AoA) in language models, by the effects of five predictors (lexical class, word length, and concreteness), and language models exhibit a different learning pattern from children; (A2) language models acquire unigram and bigram statistics early in training.

Hypothesis and metrics. For each word to be evaluated, its AoA is defined to be the midpoint of a sigmoid curve fitted over the word surprisals obtained on all saved checkpoints. A likelihood ratio test is conducted on five quantitative indicators, log-frequency, mean length of utterance (MLU), n-chars, concreteness, and lexical classes, to assess the significance of each predictor in predicting the AoA. A set of Kullback–Leibler divergence curves between the model's word predictions and the unigram and bigram probabilities of the words is plotted to determine the acquisition trend of unigrams and bigrams.

Replication. We replicated the main results of this paper, including the surprisal curve and the five predictors in predicting the AoA for the specific word 'eat'. Results are in Figure 1.

Results across configurations. We sampled four sets of configurations on which we trained the models. Since the training data is extensive, the training time varies from 2 to 14 days per model on four A40 GPUs, depending on the number of layers and hidden dimension size.

For the case study word "eat," these configurations produce sigmoid functions that are mostly similar, consistent with Chang and Bergen (2022)'s characterization (A1) that neural models exhibit staged acquisition of lexicons. Hyperparameters primarily shift the position and the slope of the curves' transitions. Examination of finer-grained conclusions about how predictors contribute to the AoA shows relatively robust findings as well, with effects varying only in significance, not in sign, as shown in Table 1.

Regarding the claim about the unigram and bigram learning curves (A2), our hyperparameter variation also shows that the initially observed pattern holds across all our experiments, as illustrated in Figure 2. The KL-divergence curves between the model predictions and the unigram and bigram probabilities both increase after decreasing, matching the claim that the models always first overfit on unigrams and bigrams. As references, the curve for the uniform distribution keeps increasing, while that for the one-hot distribution, which equals the cross-entropy loss, remains decreasing.

Observations from experiment A provide sufficient evidence on the stability of its conclusions, though a limited hyperparameter space is explored. These observations align with Michaelov et al. (2025)'s finding that language models' behavior during word acquisition, which is a procedure highly dependent on the distributional statistics of training data, is consistent across model architectures.

Config	PPL	Log-freq	MLU	n-chars	Concreteness	Lexical class	R^2	HP
Base	54.80	*** (-)		*** (-)		***	0.93	
1	42.43	*** (-)		** (-)		***	0.88	Dropout -
2	38.75	*** (-)	* (+)	** (-)		***	0.88	#layers +
3	31.43	*** (-)	* (+)	*** (-)		*	0.88	#layers +, Batch size +
4	35.73	*** (-)	* (+)	** (-)		***	0.91	Layer size +, LR -
Children		*** (-)	*** (+)	** (+)	*** (-)	***	0.43	

Table 1: Coefficients in experiment A. Perplexity, significant predictors, and R^2 are reported in the original experiment and in the other four sets of configurations. Significant predictors are marked by asterisks ($p < 0.05$ *; $p < 0.01$ **; $p < 0.001$ ***). Signs of coefficients are notated in parentheses. The R^2 denotes the adjusted R^2 in a regression using all five predictors. The HP records the differences in hyperparameters between the corresponding setting and the base setting, with colors separating the model parameter and training parameter.

3.2 Experiment B: Syntactic Generalization Ability

We then move to the syntactic generalization experiment, where neural models are evaluated on their syntactic abilities to judge unseen ungrammatical syntactic tests. The general evaluation procedure is to train LMs on a natural language corpus, evaluate them on proposed syntactic test suites, and compare their performances. Recent research assessing the linguistic ability of neural language models has also increasingly focused on their syntactic skills, with the aim of understanding whether and how such models acquire human-like grammatical competence (Gulordava et al., 2018; Hu et al., 2020a; Gauthier et al., 2020; Wilcox et al., 2021; Finlayson et al., 2021).

With this motivation, Hu et al. (2020b) evaluate syntactic generalization – applying learned grammar rules to new contexts across a diverse range of neural architectures, including LSTMs and Transformers on 34 syntactic tests based on English. By plotting the relationship between test perplexities and models, they reveal differences in syntactic generalization ability across architectures. Details of data, model, and training procedure are in Appendix A.2.

Conclusions from this paper can be summarized into that, (B1) the results dissociate model perplexity and performance in the syntactic generalization task, suggesting that the two metrics represent complementary features of a language model's syntactic knowledge, and (B2) model architecture plays a more critical role than training data scale in yielding correct syntactic generalization.

Hypothesis and metrics. Besides obtaining the test set perplexity after training on next-token-prediction tasks, each model checkpoint is evaluated on a syntactic test suite involving 34 tests proposed by Hu et al. (2020b). Each test judges whether a model can distinguish ungrammatical sentences by producing higher surprisals from grammatical ones. The accuracy on all tests defines the syntactic generalization (SG) score for that model checkpoint for further analysis. The SG scores and perplexities are averaged across three training runs with three random seeds for each training configuration.

Results across configurations. Although the paper does not release the complete code or the training configuration, we seek to match the numbers and procedures described in the paper. We ensured the number of tokens and UNK tokens in data preprocessing matched those reported in the paper, detailed in Appendix A.2. We randomly selected five configurations from a commonly used parameter space. For each configuration, the model is trained three times with different random seeds until convergence on the validation set. All training instances yield validation-set perplexities similar to or smaller than those reported in the paper. Besides, a GPT-2 pretrained model (Radford et al., 2019) is loaded from Hugging Face's open-sourced weights[1] and tested on one L40S GPU, yielding similar SG scores (originally 0.7511, ours 0.7423). The evidence above confirms our replication is reliable.

Figure 3 shows the relationship between test perplexities and syntactic generalization (SG) scores across all hyperparameter configurations for LSTM and GPT-2. Same as Hu et al. (2020b), we observe that perplexities and SG scores are not monotonically related, since checkpoints with similar perplexities can exhibit markedly different SG scores (e.g., Configurations 1, 4, and 5 for LSTM LG, and Configurations 3, 4, and 5 for GPT-2 LG). This

[1] https://huggingface.co/openai-community/gpt2

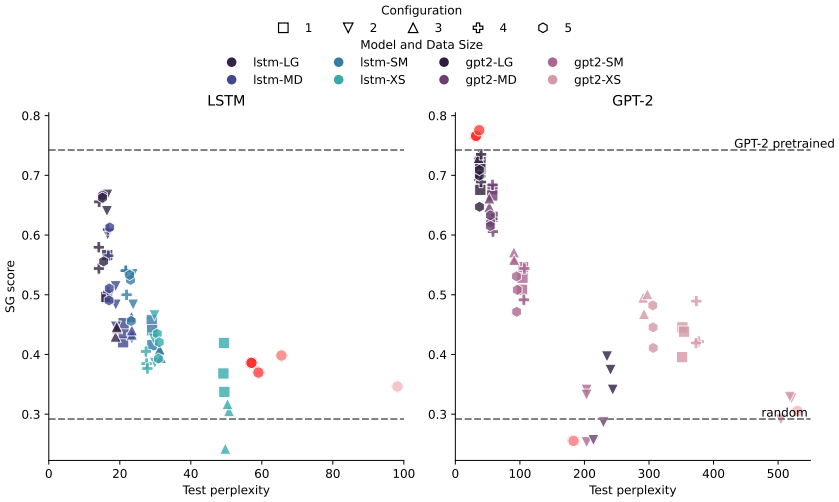


Figure 3: Test perplexities and SG scores for LSTM and GPT-2 on all configurations, each dot denoting one distinct training instance. Red dots indicate the results reported in the original paper. Same as Hu et al. (2020b), since GPT-2's losses have been converted from token-count-based into word-count-based, the perplexities of GPT-2 will increase in numbers, and thus perplexities are not directly comparable among model architectures.

confirms the original paper's claim that perplexity and SG behave as complementary indicators of model quality rather than a single goodness metric, marking the stability of the coarse claim (B1).

In contrast, conclusion B2 is not fully preserved in our results. We fitted a linear regression model with model architecture class, data size, and model-specific configuration names (i.e, LSTM-1 and GPT-2-1 treated as distinct levels) as independent variables to predict SG scores, using all training instances as independent data points. Across configurations, both data size and model architecture show strong associations with SG scores. With GPT-2 as the base architecture level and LG as the base data size level, the model predicts a 0.6764 (***) SG score. Switching the architecture to LSTM will result in 0.0569 (**) less SG score unit; while changing the data size to MD, SM, and XS will respectively lead to 0.0679 (*), 0.1238 (***), and 0.1991 (***) less SG score unit (adjusted R^2 = 0.85). These coefficients suggest that across these five hyperparameter variations, SG varies significantly with both data size and architecture, unlike what was observed in the original paper.

Results above show that experiment B is partially sensitive to hyperparameter variations. Across its conclusions, the coarse one claiming the specific linguistic ability is a distinct metric from the perplexity in next-token-prediction task still holds, echoing existing debate on the divergence of general model performance evaluation and specific linguistic abilities. (Beinborn and Hollenstein, 2023; Millière, 2024) In contrast, model ranking by linguistic ability can be easily affected by hyperparameter tuning, since well-chosen hyperparameters lead to better-trained models. Examining the model performances, it can be inferred that the conclusion difference is likely due to the under-trained models in the original paper. Our LSTMs' perplexities are notably lower and SG scores are higher than those reported in the original paper, possibly due to differences in the training framework, the GPU, or the hyperparameter choice, leading to under-tuned models.

3.3 Experiment C: Word Order Preference

Human languages exhibit a bias in part-of-speech orderings; e.g., SOV is more frequent than SVO. Recent research claims that neural networks exhibit human-like word-order preferences. White and Cotterell (2021) and Kuribayashi et al. (2024) demonstrated that cognitive-driven neural networks, such as those with memory limits, can learn languages whose word order occurs more frequently in human languages.

We replicate the setup of (Kuribayashi et al., 2024) to test whether different model architectures exhibit preferences over word orders that align with humans, denoted by their empirical distribution in natural languages. We interpret such preferences as a consequence of inductive bias, where architectural constraints restrict the hypothesis class (i.e., the set of functions a model can represent) prior to learning to prevent overfitting on training data

(Shalev-Shwartz and Ben-David, 2014).

Due to the lack of natural language data for most word orders, all neural models were trained on 64 artificial languages, each with a distinct word order. After training, the final test perplexity for each model is documented. A correlation between the perplexity and the real-world frequency of each word order taken from WALS (Dryer and Haspelmath, 2013) is then calculated as the indicator of the similarity to human preferences. The details of data, model, and training procedure are in Appendix A.3.

Conclusions of this experiment include (C1) typologically frequent word orders tend to have lower perplexity, and (C2) this relationship is better estimated by LMs with cognitively plausible biases, specific parsing strategies, and memory limitations. Due to computational constraints, we focus on the experiment related to memory limitations in C2, which assumes that the simple recurrent neural network (SRN), the most memory-limited model, has the most human-like word-order preference, LSTM is in the middle, and Transformer is the least.

Hypothesis and metrics. The original experiment adopts Pearson's r correlation of the grammar's negative test perplexity obtained from training and the real-world data distribution in WALS, as the indicator of the model's human-like preference. A higher correlation is assumed to indicate that a model prefers frequent word-order patterns in natural languages. However, during replication, we found that most of the correlations reported in the paper or in our replication have high p-values, indicating insignificance (see Appendix D). Continuing to use Pearson's r thus does not yield meaningful comparisons. We hence adopted a linear regression, defined as follows, to determine whether a model assigns lower PPLs to typologically frequent word orders.

$$\begin{aligned}\text{Test Perplexity} = \beta_0 \\ + \boldsymbol{\beta}_M^\top \mathbf{M} + \beta_F F \\ + \boldsymbol{\beta}_{FM}^\top (F \circ \mathbf{M}) \\ + \boldsymbol{\beta}_d^\top \mathbf{d} + \boldsymbol{\beta}_{dM}^\top (\mathbf{d} \circ \mathbf{M}) \\ + \epsilon,\end{aligned}$$

where $\mathbf{d}$ denotes the 6-digit feature vector (excluding the third digit), each as a 2-level categorical variable; F the frequency as a continuous value, $\mathbf{M}$ the model-class categorical indicator, and $\boldsymbol{\beta}$ the associated coefficients. The term $\mathbf{d} \circ \mathbf{M}$ denotes the element-wise interactions between the grammar-digit features and the model class. β_F and $\boldsymbol{\beta}_{FM}$ capture the frequencies' effect on the training perplexity, and how this effect varies across each model architecture. $\boldsymbol{\beta}_d$ and $\boldsymbol{\beta}_{dM}$ encode the effect of each specific grammar switch digit's effect on the test-set perplexity, and how well each model can make use of this digit. Thus, the paper's research questions can be translated to (C1) if more frequent word orders are preferred by language modes (β_F, $\boldsymbol{\beta}_{FM} < 0$), and (C2) if memory-more-limited models capture the natural word order distributions better than memory-less-limited models ($|\beta_{F,SRN}| > |\beta_F| > |\beta_{F,Trans}|$, where the vertical bars stand for absolute value). These paper-related coefficients $\boldsymbol{\beta}_M$, β_F, and $\boldsymbol{\beta}_{FM}$ will be focused on in the replication. Finer-grained analysis on the switch digit's effect is detailed in Appendix E.4.

Replication. We first replicate the experiments by running the code released by Kuribayashi et al. (2024) on our machine. The resulting perplexities reached the same level as those reported in the original paper, and the coefficients exhibited similar magnitudes, signs, and statistical significance. Details of replication can be referred to in Appendix A.3.

Results across configurations. Across all configurations in Table 2, β_0 denotes the perplexity levels when LSTM is set as the base architecture level. β_{SRN} and β_{Trans} correspondingly show that switching the model architecture to SRN will generally increase the test-set perplexities, while for Transformer, the perplexities decrease. Meanwhile, out of four configurations where β_F and β_{FM} are significant, most of them are negative, indicating that increasing the frequency variable decreases test-set perplexity. This aligns with the original paper's claim (C1) that languages with more frequent word orders in WALS are easier for language models. However, given the lack of significance across the remaining configurations, the results from either the original paper or the replication run are considered to support this claim. Besides, since the significant negative terms appear only in configurations 1 and 4 for SRN and configuration 6 for LSTM (at the base architecture level), it is hard to argue that C1 is robust across hyperparameters.

The claim (C2) concerns architecture-specific word-order preferences, where memory-limited models exhibit more human-like preferences and

Config	β_0	β_{SRN}	β_{Trans}	β_F	$\beta_{F,\text{SRN}}$	$\beta_{F,\text{Trans}}$	HP
Base	9.03 (***)	20.75 (***)	1.39 (*)				
Repl	6.35 (***)	13.93 (***)	0.29 (*)		-7.59 (*)		
1	6.47 (***)	16.78 (***)	0.23 (***)		-7.35 (*)		Batch size +
2	6.44 (***)	14.86 (***)					Dropout -
3	6.24 (***)	2.12 (***)	0.36 (***)				Early-stop +
4	6.47 (***)	16.78 (***)			-7.35 (*)		Layer size +
5	7.12 (***)	20.27 (***)	-0.58 (*)				#layer +
6	42.86 (***)	108.43 (***)	-32.42 (***)	-25.93 (***)	29.57 (**)		LR -
Best	6.24 (***)	0.29 (***)					

Table 2: Estimated coefficients related to the original paper's conclusion for all configurations in Experiment C. Coefficients with no significance are denoted as '-'. Significance codes: '***' for $p < 0.001$, '**' for $p < 0.01$, and '*' for $p < 0.05$. 'Repl' denotes the replication experiment. All adjusted R^2's are above 0.9. The HP records the differences in hyperparameters between the corresponding setting and the base setting, with colors separating the model parameter and training parameter.

correspond, in our linear regression, to the frequency-model interaction terms (β_{FM}). In our results, SRNs show stronger negative associations with perplexity across several configurations (1 and 4), whereas LSTMs and Transformers do not show significant effects in these settings. The expected order made by the assumption that LSTMs' negative effect should be greater than Transformers' is seldom presented, except in configuration 6. Configuration 6 also reverses the expected order between SRN and LSTM. Moreover, this expected order is not presented among the best configurations selected based on average validation-set perplexities over all languages (configuration 3 of LSTM, 3 of SRN, and 5 of Transformer), further undermining the original paper's conclusion C2.

The instability of experiment C mainly results from that its metrics involve model ranking and depend heavily on the Pearson correlation, a statistical test that requires checking the significance. However, the original paper's omission of p-values has led to the misreporting of insignificant results, which plausibly explains the high instability observed across hyperparameter settings.

4 Discussion

4.1 Effects of Hyperparameter Tuning on Conclusions

Since experiments A and C are inappropriate for further analysis due to either stability or instability across hyperparameters, we chose a hyperparameter-sensitive experiment, experiment B, to further examine the variance in evaluation metrics related to linguistic findings as the hyperparameters change. Besides, analysis on experiment B would benefit from generalization since it trains the model on real-world datasets.

A Bayesian optimization (Snoek et al., 2012; Murphy, 2012) is performed based on a Gaussian process[2], fit to the five existing training runs. The metric for determining the best model is average validation perplexity on the XS data size. In each optimization iteration, the configuration that maximizes the acquisition function, defined by the expected improvement (EI), is chosen as the next configuration to train on. The Bayes tuning continues for five iterations per model (denoted configurations 6 to 10). Model perplexities and SG scores on each configuration are detailed in Appendix E.2.

A likelihood-ratio test is performed to assess the effect of hyperparameters on the models' syntactic ability by iteratively comparing a full model and a reduced model that preserves all other hyperparameters except the one being tested. Table 3 presents the signs and significance of each hyperparameter as a predictor of the SG scores of each model architecture. For both models, the number of layers (LSTM layers or Transformer blocks), dropout rate, and the initial learning rates are significant predictors of the model's SG score.

4.2 Effects of Early Stopping vs. Convergence

Research performing linguistic-driven experiments trains models with different numbers of steps, either with a fixed number of epochs (Aurnhammer and Frank, 2018; Chang and Bergen, 2022; Kurib-

[2]The Gaussian process is built with Matérn kernel following the default of the *wandb* package (Biewald, 2020).

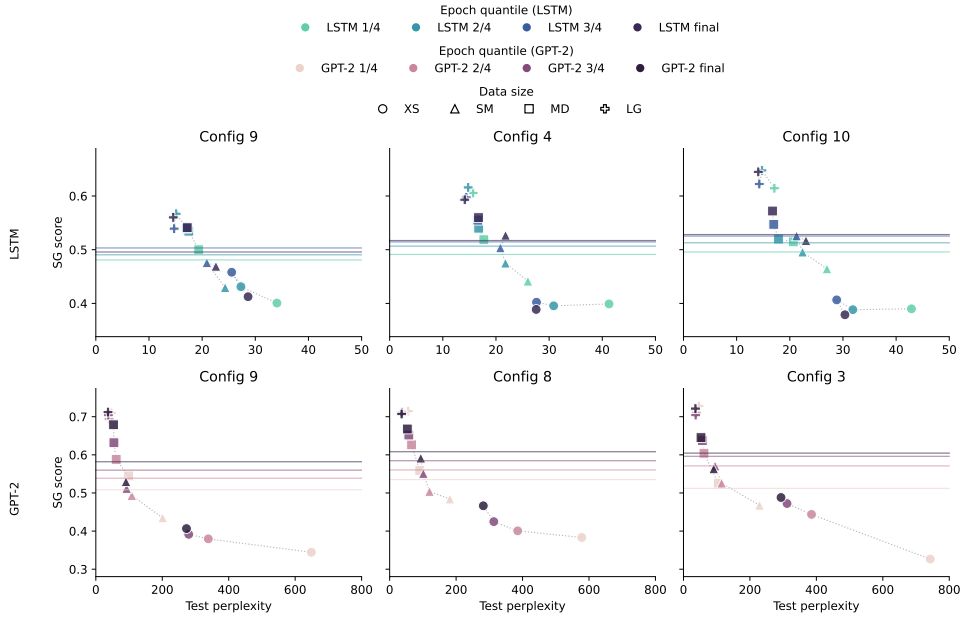


Figure 4: The dynamics of test-set perplexities SG scores with respect to the training procedure. Early stopping is decided by convergence on validation-set, with the best checkpoint loaded. LSTM Config 9 misses a 1/4 XS point since the corresponding training instances converge within three epochs.

Hyperparameter	**Sign and Significance**
Hidden Size	
Embedding Dimension	
Number of Layers	*** (+)
Dropout Rate	*** (-)
Initial Learning Rate	*** (+)

Hyperparameter	**Sign and Significance**
Hidden Size	
Number of Layers	*** (+)
Number of Heads	
Dropout Rate	*** (-)
Attention Dropout Rate	
Initial Learning Rate	*** (+)

Table 3: Hyperparameters as significant predictors of SG scores, in LSTM model (up) and GPT-2 model (bottom). Signs determined by the sign of coefficients in the overall regression, and significance by the likelihood ratio test.

ayashi et al., 2024, *inter alia*), or until the training converges on the validation set (White and Cotterell, 2021; Hu et al., 2020b; Wilcox et al., 2020, *inter alia*). However, whether a researcher can freely choose to train for a certain number of epochs or to train until convergence is under discussion.

This section further examines experiment B on the training dynamics, focusing on changes in their evaluation metrics. Based on the results from the Bayes tuning above, we compare their SG scores at the 1/4, 2/4, and 3/4 positions, as well as at the final checkpoints. The visualization is shown in Figure 4. The trajectories of the points demonstrate that as the training procedure continues, the SG scores for each data size generally increase, though in occasional cases, the scores at the final checkpoints are lower than those at the 3/4 checkpoints. The divergence across data sizes differs the most among these trajectories, as SG scores and perplexities diverge significantly at the early checkpoints but converge by the final checkpoints. Analysis on the divergence across runs can be affected by the choice of training epochs. Additionally, the SG scores for each epoch quantile are indicated by horizontal lines. Choosing to train for a certain number of epochs (e.g., 1/4) before convergence can mislead to a fake observation that LSTMs and GPT-2 have similar SG scores.

4.3 Suggestions on Experiments for Stability across Hyperparameters

On tuning. Hyperparameters can act as a substantial source of variability besides commonly accepted sources like random seeds or training data order. Empirical conclusions should be tuned over several rounds to produce better-trained models that better reflect the learning potential of specific model architectures. In addition, it is worth noting that the number of training epochs affects the model's behavior.

Fragile conclusion. Our results reveal a gradient of sensitivity to hyperparameters across linguistic phenomena. Fine-grained lexical findings, such as staged acquisition and frequency effects, remain stable across a wide range of training configurations. Models' syntactic ability ranking shows moderate sensitivity. Typological word-order claims, which rely on subtle interactions among architectures, grammatical rules, and statistical tests, are the most sensitive to hyperparameter sweeps. This pattern suggests that claims relying on higher-order interactions require stronger robustness checks than claims based on lower-level distributional signals.

Reporting standard. Experimental reports should document the hyperparameter range faithfully, tuning procedures, and other relevant sources of randomness (e.g., seeds, initialization, hardware), e.g., as the best practices Ulmer et al. (2022) suggest for research in the NLP field. Architectural differences should be interpreted cautiously since they are not stable across hyperparameters. In addition to reporting best-performing checkpoints, researchers should report how conclusions vary under tuning. Additionally, statistical analyses should report their significance faithfully.

5 Conclusion and Discussions

This work advocates for the consideration of hyperparameter stability in linguistically motivated experiments with language models. Our claim has been supported by the replication of three representative research studies and rigorous examination of the variance of their conclusions across different hyperparameters: we find that the results from the experiments above are all affected, to some extent, by hyperparameter choices, and that some observations alter or reverse the original conclusion. From the perspective of experimental design, linguistically-driven experiments that observe models' distributional learning signals, such as word learning curves, tend to be stable. Rankings of model capabilities and significance of statistical tests are more sensitive to hyperparameter choices, which are common metrics for higher-level signals such as syntactic ability or language-typological preferences. Conclusions related to training trajectories are particularly sensitive to variation induced by hyperparameter choices. To improve the reproducibility of future experiments, we suggest future work to view hyperparameter choice as a source of variation equally important as other sources, such as random seed and GPU type, and conclude from adequately trained models and robustness checks across settings.

Limitations

Experiments. Our analysis is based on three representative experiments, each tied to a single dataset or grammar per phenomenon. While this covers a broad range of linguistic claims, it does not capture the full diversity of tasks or datasets used in LM-based linguistics.

Number of seeds. Experiment A runs on only one seed, which may restrict our ability to estimate variance in that setting.

Configuration space. Our sweeps explore hyperparameter sensitivity within a restricted configuration space, since training on each hyperparameter set is itself time-consuming. Some hyperparameters that are already commonly considered to be optimal for each model family (e.g., optimizer type, LR schedule, and weight decay) are not covered. This means our sweep does not cover the full range of available training dynamics.

Acknowledgements

We thank the anonymous reviewers and the area chair for their valuable comments. This work is supported in part by an NSERC Discovery Grant (RGPIN-2024-04395) and a Canada CIFAR AI Chair Award to FS, and a Cohere Scholarship to RN.

References

Morris Alper and Hadar Averbuch-Elor. 2023. Kiki or bouba? sound symbolism in vision-and-language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Christian Arnold, Luka Biedebach, Andreas Küpfer, and Marcel Neunhoeffer. 2024. The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*, 12(4):841–848.

Christoph Aurnhammer and Stefan Frank. 2018. Comparing gated and simple recurrent neural network architectures as models of human sentence processing.

Lisa Beinborn and Nora Hollenstein. 2023. *Cognitive plausibility in natural language processing*. Springer.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. A systematic review of reproducibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393. Association for Computational Linguistics.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.

Xavier Bouthillier, César Laurent, and Pascal Vincent. 2019. Unreproducible research is reproducible. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 725–734. PMLR.

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

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. Bllip 1987-89 wsj corpus release 1. *(No Title)*.

Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. 2019. On empirical comparisons of optimizers for deep learning. *ArXiv preprint*, abs/1910.05446.

Ionut Constantinescu, Tiago Pimentel, Ryan Cotterell, and Alex Warstadt. 2024. Investigating Critical Period Effects in Language Acquisition through Neural Language Models.

A. Feder Cooper, Yucheng Lu, Jessica Forde, and Christopher De Sa. 2021. Hyperparameter optimization is deceiving us, and how to stop it. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 3081–3095.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (v2020.4)*. Zenodo.

Stephen Merity et al. 2016. Wikitext-103. *ArXiv preprint*, abs/1609.07843.

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

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843. Association for Computational Linguistics.

Luca Franceschi, Michele Donini, Valerio Perrone, Aaron Klein, Cédric Archambeau, Matthias Seeger, Massimiliano Pontil, and Paolo Frasconi. 2024. Hyperparameter Optimization in Machine Learning.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76. Association for Computational Linguistics.

Steven N. Goodman, Daniele Fanelli, and John PA Ioannidis. 2016. What does research reproducibility mean? *Science translational medicine*, 8(341):341ps12–341ps12.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

Jennifer Hu, Sherry Yong Chen, and Roger Levy. 2020a. A closer look at the performance of neural language models on reflexive anaphor licensing. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 323–333. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020b. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744. Association for Computational Linguistics.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646. Association for Computational Linguistics.

Mete Ismayilzada, Defne Circi, Jonne Sälevä, Hale Sirin, Abdullatif Köksal, Bhuwan Dhingra, Antoine Bosselut, Duygu Ataman, and Lonneke Van Der Plas. 2025. Evaluating morphological compositional generalization in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1270–1305. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.

Tatsuki Kuribayashi, Ryo Ueda, Ryo Yoshida, Yohei Oseki, Ted Briscoe, and Timothy Baldwin. 2024. Emergent Word Order Universals from Cognitively-Motivated Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14522–14543. Association for Computational Linguistics.

Hugo Larochelle, Dumitru Erhan, Aaron C. Courville, James Bergstra, and Yoshua Bengio. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 473–480. ACM.

Alessandro Lenci. 2018. Distributional models of word meaning. *Annual review of Linguistics*, 4(1):151–171.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202. Association for Computational Linguistics.

Clara Meister, Tiago Pimentel, Patrick Haller, Lena Jäger, Ryan Cotterell, and Roger Levy. 2021. Revisiting the Uniform Information Density hypothesis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 963–980. Association for Computational Linguistics.

James A. Michaelov, Roger P. Levy, and Benjamin K. Bergen. 2025. Language model behavioral phases are consistent across architecture, training data, and scale.

Raphaël Millière. 2024. Language models as models of language.

Kanishka Misra and Kyle Mahowald. 2024. Language Models Learn Rare Phenomena from Less Rare Phenomena: The Case of the Missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53. Association for Computational Linguistics.

Eva Portelance and Masoud Jasbi. 2024a. The roles of neural networks in language acquisition. *Language and Linguistics Compass*, 18(6):e70001.

Eva Portelance and Masoud Jasbi. 2024b. The Roles of Neural Networks in Language Acquisition. *Language and Linguistics Compass*, 18(6):e70001.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Terry Regier. 2005. The emergence of words: Attentional learning in form and meaning. *Cognitive Science*, 29(6):819–865.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Haoyue Shi, Karen Livescu, and Kevin Gimpel. 2020. On the role of supervision in unsupervised constituency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7611–7621. Association for Computational Linguistics.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2960–2968.

Dennis Ulmer, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. 2022. Experimental standards for deep learning in natural language processing research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2673–2692. Association for Computational Linguistics.

Alex Warstadt and Samuel R Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In Shalom Lappin and Jean-Philippe Bernardy, editors, *Algebraic Structures in Natural Language*, pages 17–60. CRC Press.

Jennifer C. White and Ryan Cotterell. 2021. Examining the inductive bias of neural language models with artificial languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 454–463. Association for Computational Linguistics.

Ethan Wilcox, Pranali Vani, and Roger Levy. 2021. A targeted assessment of incremental processing in neural language models and humans. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 939–952. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger Levy. 2020. On the Predictive Power of Neural Language Models for Human Real-Time Comprehension Behavior.

Tianyang Xu, Tatsuki Kuribayashi, Yohei Oseki, Ryan Cotterell, and Alex Warstadt. 2025. Can Language Models Learn Typologically Implausible Languages?

Changbing Yang, Franklin Ma, Freda Shi, and Jian Zhu. 2025. LingGym: How Far Are LLMs from Thinking Like Field Linguists?

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.

Appendix

A Background of Each Experiment

A.1 Exp A: Word Acquisition

Data. The training data is a mixture of BookCorpus (Zhu et al., 2015) and WikiText-103 (et al., 2016).[3] The input sequences are paired and tokenized by a SentencePiece tokenizer (Kudo and Richardson, 2018).

Training. Due to the size of the training data, the training of this experiment takes days to finish. Besides, since (Michaelov et al., 2025) confirms the consistency of distributional signals over model architectures, we only use one architecture, the LSTM model, to replicate the experiment. For each set of hyperparameters, an LSTM model is trained on the dataset for 1M steps. All training instances take place on 4 A40 GPUs, with AdamW as the optimizer and Lambda LR scheduler. The checkpoint recording frequency is as described in Appendix C. The optimization space involved in this experiment includes the hyperparameters in Appendix B. We trained the LSTM on four configurations.

Detail of analysis. The sigmoid function is fitted through the SSlogis method in the R language. During the determination of the signs and significance of predictors of AoAs, a likelihood ratio test is performed between the full regression model and a reduced model with all predictors except the target single one. The signs of predictors are obtained from the general linear regression. In the KL-divergence curves, the unigram probability is calculated by raw word frequency in the corpus, while the bigram probability is calculated by the conditional probability of a word depending on the previous word.

A.2 Exp B: Syntactic Generalization

Data. The corpus used for training is the BLLIP dataset (Charniak et al., 2000), divided into four sizes: XS, SM, MD, and LG. For LSTM, the

[3]The exact versions of the two datasets we adopted are recorded in the link https://huggingface.co/datasets/compling/word-acquisition

dataset is tokenized through a word tokenizer. The same UNK removal procedure is performed on the tokenized data as well. The vocab size after UNK removal matches the vocab size reported in the paper. For Transformer, the dataset is tokenized with the pre-trained GPT-2 tokenizer with a fixed vocab size. We verified that our word counts after data segmentation and unkification, and our token counts after tokenization, match the original paper.

Training. Two models, LSTM and GPT-2, are trained with the Hugging Face Trainer,[4] each on one H100 GPU. Both models use the AdamW optimizer and the linear scheduler. Each model has been trained three times on different random seeds for each configuration. Parameter space is documented in Appendix B.

A.3 Exp C: Word Order Preference

Data. The training data are artificial languages generated from one probabilistic context-free grammar (PCFG). This PCFG is designed to vary across six types of generation rules, each denoted as a switch (e.g., S $\rightarrow$ NP VP $[d_1 == 0]$ | VP NP $[d_1 == 1]$ is a rule yielding the values 'on' and 'off' for switch S). Thus, the grammar generates $2^6 = 64$ languages, each characterized by a distinct word-order. The identifier of the language is thus denoted as a switch vector $\mathbf{b} \in \{0, 1\}^6$, corresponding to the 7-digit grammar name with the 3rd digit marginalized. For each grammar, we sample 20,000 sentences to train each model. Each of such training instances is repeated 3 times to reduce the effect of randomness.

Training. Three models, the simple recurrent neural network (SRN), long short-term memory (LSTM), and decoder-only Transformer (Transformer), are compared to verify this conclusion. The SRNs are assumed to be the most memory-limited, LSTMs in the middle, and Transformers are assumed to be the least memory-limited. On each grammar and each configuration, each model is trained on three separate but equivalent datasets. All models are trained with the AdamW optimizer and the Inverse-Sqrt scheduler. Each training instance is run on a single L40 GPU. The hyperparameters involved in the tuning space from the *fairseq* training framework (Ott et al., 2019) are detailed in Appendix B. By varying a single hyperparameter each time, we selected seven configs and trained on them.[5]

[4]The original paper trains GPT-2 on the TensorFlow framework.

[5]Embedding size and hidden size are bound to change together.

Detail of replication. A comparison of the test-set perplexities between the original paper and the replicated runs is presented in Table 4. The sizes of the perplexities roughly match those in the original paper.

Experiment	SRN	LSTM	Transformer
Base	28.9020	8.8328	10.5339
Repl.	23.0733	7.7554	7.2007

Table 4: Test-set perplexities from replication of experiment C, comparing with results open-sourced by the original paper.

B Tuning Space

Random sampling space. The hyperparameter configurations randomly sampled for three experiments are recorded in Table 5.

Bayes tuning space. The Bayes tuning space of experiment B is documented in Table 6.

C Implementation Difference in Each Experiment

Data deduplication in Kuribayashi et al. (2024). In Kuribayashi et al. (2024)'s data generation procedure, which is a direct reusing the open-source PCFG and code from White and Cotterell (2021)'s paper, around six hundred sentences out of 100,000 sentences were overlapping between the training set and the test set during data generation. We regenerated the same number of sentences after adding a duplication removal procedure.

Checkpointing frequency in Chang and Bergen (2022). Due to the storage limit, we slightly reduced the checkpointing frequencies to the following ones.

- Checkpointing at every 100 steps during the first 1,000 steps.
- Checkpointing at every 500 steps during the first 10,000 steps.
- Checkpointing at every 5,000 steps during the first 100,000 steps.
- Checkpointing at every 50,000 steps during the first 1,000,000 steps.

D P-value in Experiment C and change of the metric.

Kuribayashi et al. (2024) originally adopts Pear-

Config	batch size	learning rate	number layers	embedding size	dropout
Replication	64	5e-4	3	768	0.1
1	64	5e-4	3	768	0.0
2	64	5e-4	3	1,536	0.3
3	256	5e-4	4	768	0.1
4	256	5e-5	4	1,536	0.3

(a) Configurations for experiment A, the word acquisition experiment.

Config	embedding size	hidden size	num layers	num heads	dropout	attention dropout	learning rate
LSTM 1	128	512	4	-	0.1	-	5e-5
LSTM 2	64	256	2	-	0.0	-	1e-3
LSTM 3	256	256	2	-	0.1	-	5e-5
LSTM 4	128	512	2	-	0.1	-	1e-3
LSTM 5	256	256	4	-	0.1	-	1e-3
LSTM 6	128	512	2	-	0.1	-	9.05e-4
LSTM 7	64	512	2	-	0.1	-	9.83e-4
LSTM 8	64	512	2	-	0.0	-	9.79e-4
LSTM 9	256	512	2	-	0.0	-	9.85e-4
LSTM 10	256	512	3	-	0.0	-	9.9e-3
GPT-2 1	768	3,072	12	12	0.0	0.0	5e-5
GPT-2 2	384	1,536	4	4	0.3	0.0	1e-3
GPT-2 3	768	3,072	12	12	0.1	0.1	5e-5
GPT-2 4	384	1,536	12	12	0.1	0.1	5e-5
GPT-2 5	768	3,072	6	6	0.1	0.1	5e-5
GPT-2 6	1024	4,096	8	8	0.1	0.1	1e-6
GPT-2 7	786	3,072	12	12	0.1	0.1	1e-5
GPT-2 8	768	3,072	12	12	0.1	0.1	9.9e-5
GPT-2 9	768	3,072	8	6	0.1	0.1	6.2e-5
GPT-2 10	768	3,072	8	6	0.1	0.1	1e-6

(b) Configuration for experiment B, the syntax test experiment. The symbol'-' denotes not applicable hyperparameters to a model. Configurations 6 to 10 are selected through Bayes tuning.

Config	batch size	dropout	epoch	hidden size	number layers	learning rate
Replication	256	0.3	10	512/64	2	5e-4
1	512	0.3	10	512/64	2	5e-4
2	256	0.1	10	512/64	2	5e-4
3	256	0.3	5000	512/64	2	5e-4
4	256	0.3	10	1024/128	2	5e-4
5	256	0.3	10	512/64	4	5e-4
6	256	0.3	10	512/64	2	5e-5

(c) Configurations for experiment C, the word order preference experiment. Since this experiment focuses on cross-model-architecture comparison, each configuration is shared across models.

Table 5: Configurations of all experiments performed.

Model	Hyperparameter	Space	Type
LSTM	Embedding size	{64, 128, 256}	Discrete
	Hidden size	{256, 512}	Discrete
	Numebr of layers	{2, 3, 4}	Discrete
	Dropout rate	{0.0, 0.1, 0.2, 0.3}	Discrete
	Initial learning rate	(1e-5, 1e-3)	Continuous variable, log-uniformly sampled
GPT-2	Embedding size	{384, 512, 768, 1024}	Discrete
	Number of layers	{6, 8, 12}	Discrete
	Number of heads	{6, 8, 12}	Discrete
	Dropout rate	{0.0, 0.1, 0.2, 0.3}	Discrete
	Attention Dropout	{0.0, 0.1, 0.2}	Discrete
	Initial learning rate	(1e-6, 1e-4)	Continuous variable, log-uniformly sampled

Table 6: Hyperparameter optimization space of experiment B during Bayes tuning.

son's r between the grammars' frequencies and test perplexities on models as the indicator of the models' human-like word order preferences. Upon checking the original data, we found that many correlation tests yielded p-values > 0.05, indicating statistical insignificance. Our replication with the same methods also yields high p-values. Scatter plots in Figure 5 provide further evidence of the weak correlation. Thus, we would regard Pearson's r an insufficiently reliable metric in modeling this research question.

However, an insignificant correlation between two variables does not rule out their joint effect on a third variable. Since the goal is to assess how model architecture modulates the relationship between typological frequency and perplexity, we instead use the linear regression model described in the main text, treating architecture and frequency as predictors of test-set perplexity. This allows us to directly estimate how architectural differences affect the extent to which natural word-order frequency predicts perplexity.

E Fine-grained Inspection on Hyperparameters' Effect on Model Behaviors

E.1 Experiment A: Fitted Parameters of Sigmoid Function

The fitted midpoints (Xmid) in Table 7 range by almost one log-step unit across configurations, while the scale parameter varies subtly as well, with some settings producing an earlier and steeper transition (e.g., configurations 2 and 3) and others a later transition. Thus, while the existence of a stable acquisition stage for a given word is robust, the precise inferred AoA is hyperparameter-dependent.

Config	Upper	Lower	Xmid	Scale
Replication	14.1073	8.4197	4.1340	0.3809
1	14.2582	8.7297	3.6408	0.3039
2	14.6603	8.4857	3.5512	0.3152
3	15.5302	7.6546	3.2575	0.4304
4	15.1542	8.1232	3.8027	0.4235

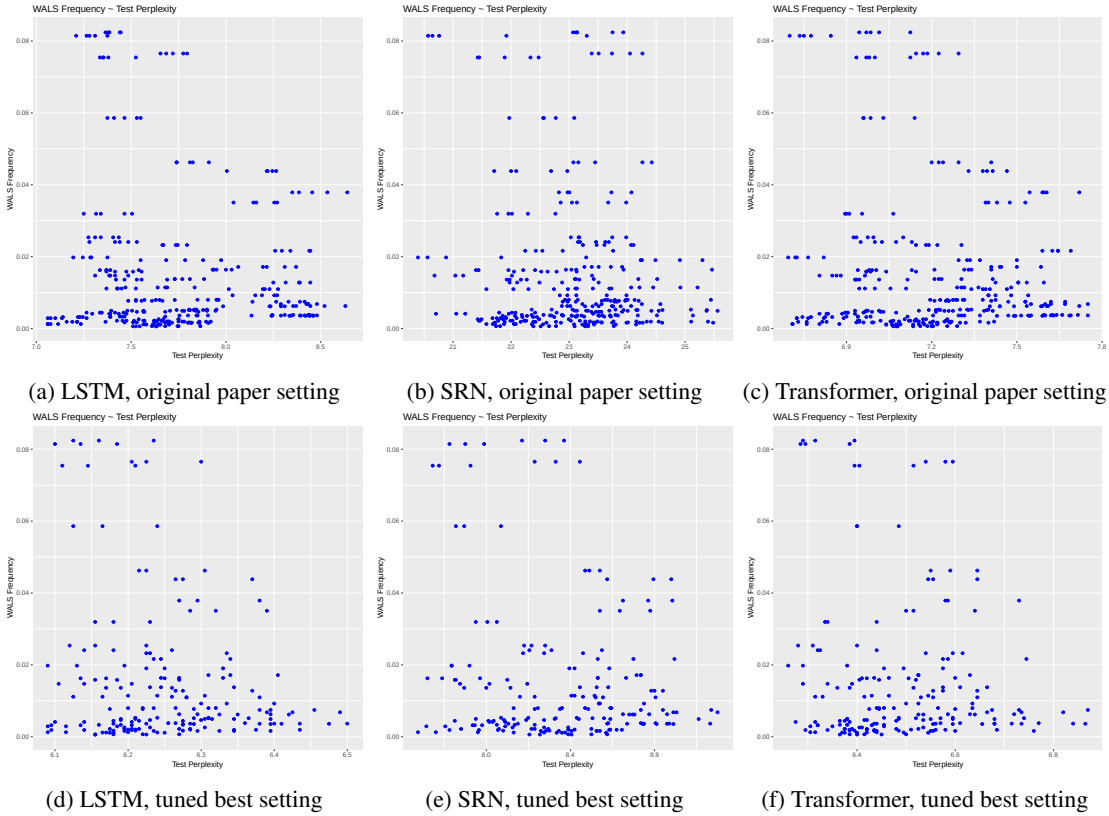
Table 7: In experiment A, parameters taken from fitted sigmoid curves on tested configurations. 'Upper' and 'lower' indicate the max and min y-values on the curve. 'Xmid' denotes the x-value of the midpoint of the curve. 'Scale' is a factor describing the range of the x-axis. The greater the 'scale' is, the flatter the curve will be.

E.2 Experiment B: Effect of Tuning on SG Scores

Figure 6 illustrates the SG score changes for the five randomly chosen configurations and the Bayes tuning. Following the plots in Hu et al. (2020b), each point denotes the delta of the score on a specific test suite with the average score across all models, configurations, and test suites. Although the SG scores of models fluctuate on configurations yielding high perplexities, they stabilize as configurations with lower perplexities are found. Thus, the ranking of architectures in terms of syntactic ability becomes easier to decide.

E.3 Experiment B: Syntax Ability by Test Suite

The syntactic bars per ability for each model is illustrated in Figure 7.



(a) LSTM, original paper setting (b) SRN, original paper setting (c) Transformer, original paper setting

(d) LSTM, tuned best setting (e) SRN, tuned best setting (f) Transformer, tuned best setting

Figure 5: An example of poor correlation with high p-values in Pearson correlation testing: WALS frequency and model perplexity of grammars in Experiment C. The low correlation and high probability to reject the correlation demonstrated by these scatter plots hinder both drawing a conclusion and observing a variance resulting from hyperparameters.

E.4 Experiment C: Word Order Preference by Switch

Table 8 presents the effect of a specific grammar switch digit on the perplexity β_d and its interaction with the model architecture class β_{dM}. A reference to the meaning of each digit is presented in Table 9. In most cases with LSTM as the base architecture level, the effect of a single digit is not significant, except in configuration 7, where tuning the switch digit 1 on (VP NP) and digit 5 off (Adj N) decreases test-set perplexity. The architecture SRN has significant effects on test-set perplexities with the digits 2 (predicate-object order), 5 (adjective-noun order), 6 (attribute clause), and 7 (case agreement) in most cases. Switching the model architecture to Transformer does not present a significant effect within our hyperparameter space.

F Generative AI Usage

We used GPT-5.2 to assist with writing. Generative AI is only used to check the code and grammar. Any AI-generated content is strictly reviewed, verified, and modified by a human, and never directly appears in code or paper.

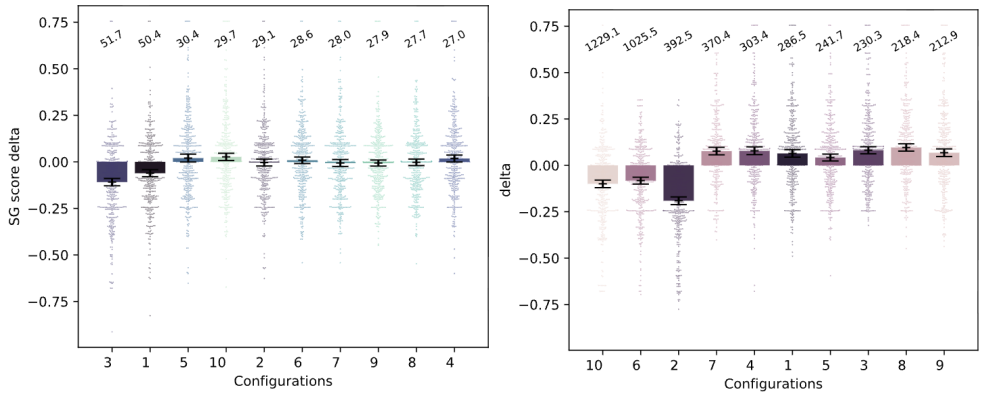


Figure 6: SG scores of each configuration, ranked by validation set perplexity on XS set. Each point represents the score delta relative to the average SG score across all models. Average validation-set perplexities are annotated over the bars. Error bars represent 95% confidence interval through bootstrapping.

Config	β_{d}	β_{dM}	HP
Base		$d_6 \times SRN^{*}$(-), $d_7 \times SRN^{*}$(+)	
Repl		$d_2 \times SRN^{***}$(+), $d_5 \times SRN^{**}$(+), $d_6 \times SRN^{**}$(-), $d_7 \times SRN^{***}$(-)	
1		$d_2 \times SRN^{***}$(+), $d_6 \times SRN^{***}$(-), $d_7 \times SRN^{**}$(-)	Batch size +
2		$d_2 \times SRN^{***}$(+), $d_6 \times SRN^{***}$(-), $d_7 \times SRN^{*}$(-)	Dropout -
3			Early-stop +
4		$d_2 \times SRN^{***}$(+), $d_6 \times SRN^{***}$(-), $d_7 \times SRN^{**}$(-)	Layer size +
5		$d_2 \times SRN^{***}$(+), $d_7 \times SRN^{**}$(+)	#layer +
6	d_1^{*}(-), d_5^{**}(+)		
Best			LR -

Table 8: Estimated coefficients for all configurations on models' grammar-digit preferences in Experiment C. $\beta_{\mathbf{5}\bullet}$ summarizes the domain-model interactions ($d_1, \ldots, d_7$ * model). Coefficients with no significance are denoted as '-'. Significance codes: '***' for p < 0.001, '**' for p < 0.01, and '*' for p < 0.05. 'Repl' denotes the replication experiment. All adjusted R^2's are above 0.9. The HP records the differences in hyperparameters between the corresponding setting and the base setting, with colors separating the model parameter and training parameter.

Switch Digit	b = 0	b = 1	Additional Description
1	S → NP VP	S → VP NP	-
2	VP → NP V	VP → V NP	-
3	S_Comp → S Comp	S_Comp → Comp S	Object clause
4	NP* → NP Prep NP*	NP* → NP* Prep NP	-
5	NP → Adj N	NP → N Adj	-
6	NP → VP Rel N	NP → N Rel VP	Attribute clause
7	N_Subj/Obj → N Subj/Obj	N_Subj/Obj → Subj/Obj N	Case agreement

Table 9: Grammar definition of the artificial language adopted in the word order preference experiment.

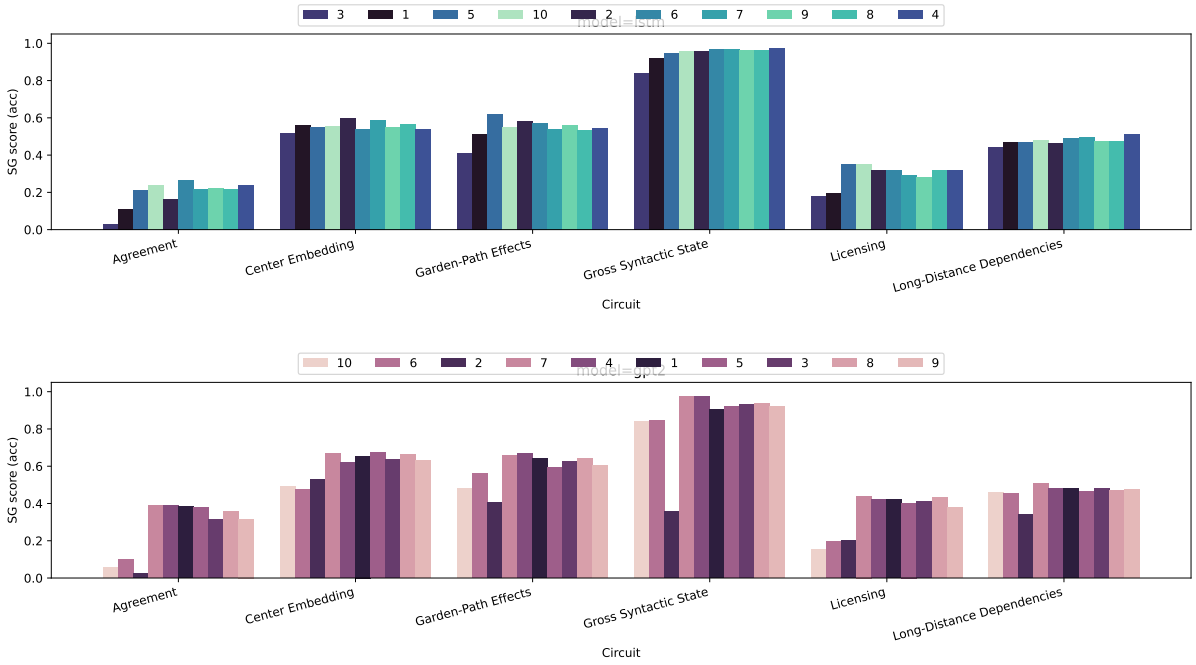


Figure 7: SG scores per syntactic ability of each configuration, ranked by validation set perplexity on XS set.