
Understanding Transformers via N -gram Statistics

Timothy Nguyen
Google DeepMind
timothycnguyen@google.com

Abstract

Transformer based large-language models (LLMs) display extreme proficiency with language yet a precise understanding of how they work remains elusive. One way of demystifying transformer predictions would be to describe how they depend on their context in terms of simple template functions. This paper takes a first step in this direction by considering families of functions (i.e. rules) formed out of simple N -gram based statistics of the training data. By studying how well these rulesets approximate transformer predictions, we obtain a variety of novel discoveries: a simple method to detect overfitting during training without using a holdout set, a quantitative measure of how transformers progress from learning simple to more complex statistical rules over the course of training, a model-variance criterion governing when transformer predictions tend to be described by N -gram rules, and insights into how well transformers can be approximated by N -gram rulesets in the limit where these rulesets become increasingly complex. In this latter direction, we find that for 79% and 68% of LLM next-token distributions on TinyStories and Wikipedia, respectively, their top-1 predictions agree with those provided by our N -gram rulesets.

1 Introduction

This paper is an attempt to answer the following

Question: *How does a transformer-based large language model (LLM) make use of its context when predicting the next token?*

Our approach proceeds via studying the statistical properties of training data. This is perhaps the most natural place to start even though it is not exhaustive (e.g. it does not include in-context learning [5]). The reasons to understand LLM behavior in terms of the statistics of their training data are plenty. First, the functional form of how LLMs use their training data is not well-understood (though there has been progress on understanding memorization [22, 6]). Second, the over-reliance of LLMs on training data statistics leads to brittleness (e.g. the “reversal curse” [3]) and the perpetuation of dataset biases [12]. Understanding the nature of this statistical dependence can lead to improved and more informed dataset curation and training methods. Finally, in various scenarios, the performance of LLMs on downstream tasks are found to be correlated with frequency of relevant training data [26, 10, 16, 17]. A better understanding of this phenomenon would allow better steering of models towards desired performance levels.

We can think of the complexity of an LLM next token prediction (regarded as a probability distribution over tokens) along two axes: form and selection. Form refers to the functional form of the prediction as a function of the context, e.g. whether the prediction is some explicit function of associated training data statistics (see Figure 1). Selection refers to which functional form, chosen from a set of functional templates, suitably describes the transformer prediction (supposing the choice set is sufficiently rich). As a first nontrivial step, one might hope that an approximate model for an LLM is that each of its next token predictions can be roughly described by simple statistical rules from

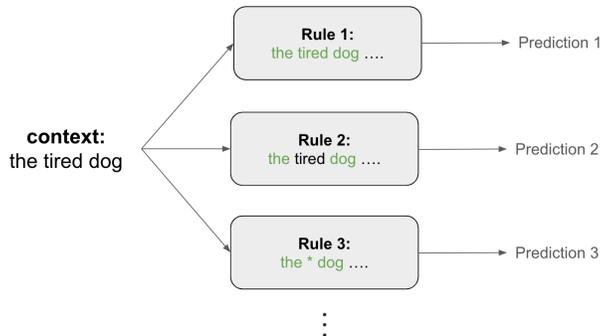


Figure 1: **Illustration of rule approximation.** Given a context, different N -gram based rules formed out of the context will yield different next-token predictive distributions. In the above example, the context consists of three tokens. The first rule uses all three tokens of the context and makes a prediction based on the corresponding 4-gram rule derived from the training data; the second rule uses only the first and last tokens to form a corresponding 3-gram rule (and so the next token “slept” will be assigned less weight than the first rule since the “tired” token is ignored); and the third rule makes a prediction using the N -gram statistics obtained from aggregating over three token contexts from the training data where the second token is arbitrary (i.e. the second token is marginalized). Given a list of such rules, one can ask which rule’s predictive distribution best matches that of the transformer.

the context (simple form) even if the mechanism for its rule selection remains hidden (complex selection)¹. This paper is an attempt to see how far this perspective can be pushed, and fortuitously we obtain additional insights for understanding LLM behavior along the way. The statistical rules we consider, which are based on N -grams, are defined in Section 4, with Figure 1 showing some examples.

We perform our main investigations on the TinyStories [11] dataset, with supporting experiments on Wikipedia to confirm our results remain robust at larger scales. The use of TinyStories is for practical reasons: its small size makes training models and aggregating N -gram statistics computationally efficient, yet it is complex enough to capture basic natural language statistics (those occurring in simple children’s stories).

Below is a summary of our observations and contributions:

1. (Approximation Criterion) We observe that next token LLM predictions tend to be well-approximated by N -gram rules when the predictions have low variance across different training runs². In particular, this includes predictions conditioned on contexts with sufficiently high count in the training data. (Section 5)
2. (Curriculum Learning Dynamics) By grouping our N -gram rulesets in terms of complexity (as measured by the amount of context they use), we discover the various ways in which the learning dynamics of LLMs implement a statistical type of curriculum learning, in which easier rules are eventually supplanted by more complex ones. (Section 6.1)
3. (Overfitting Criterion) Based on our analysis of approximating LLM predictions by N -gram rules, we propose a simple and novel procedure for detecting overfitting of LLMs during training. The procedure makes no use of holdout data and it makes quantitatively precise the intuition that overfitting corresponds to a model memorizing long context at the expense being able to generalize through making use of subcontext. (Section 6.2)
4. (Approximation Strength) We study how well LLM predictions can be approximated by our N -gram rulesets, noting that significant gains in top1-accuracy occur as we increase ruleset

¹It is important to emphasize that we seek a *descriptive* approximation of a transformer, rather than an *explanatory* one. A description merely requires that we can provide a post-hoc, per-instance approximation of transformer predictions in terms of an available rule; an explanation means we provide reasons for and thus can predict in advance why and when a particular rule approximates transformer predictions. Hence, we make the distinction between form (description) and selection (explanation).

²Different runs have different dataset shuffles.

complexity and diversity, whereby we achieve up to 79% top1-accuracy on TinyStories (Tables 2 and 5). We also visually ground these approximations with concrete examples (Figure 5), which may form the basis for dataset attribution methods in future work. Corresponding experiments on Wikipedia are shown in the Appendix. (Section 7)

We also open source our training datasets and related N -gram statistics so that others can verify and build upon our work.³

2 Related Work

Rule extraction methods for neural networks have been studied in quite different settings, e.g. [15, 21]. Some recent works have performed N -gram analyses for large-language models in the setting of in-context learning [1, 23] and associative recall [2]. The “infini-gram” model [19] compares LLM predictions with the single N -gram rule given by retrieving the largest possible matching context from the training data. Our work uses shorter but more sophisticated N -gram rules. In [28], an approach to understanding how LLMs process N -grams is carried out at the level of individual neurons. This complements our dataset-based work, which treat models as a black box. See also [27] which studies how transformers can represent N -gram models. In [9], the evolution of the type of N -gram statistics that transformers learn during training is analyzed in the setting of synthetic Markov chain data, in contrast to our natural language setting. Other works studying the learning trajectory of language models include [7, 8]. There is a large literature on building more sophisticated N -gram models, e.g. [18, 13]. Such models could have been incorporated into our set of rules, but for simplicity we choose not to include them.

3 Experimental Setup

We train standard decoder-only transformer models on the TinyStories [11] dataset (480M tokens) consisting of children’s stories synthetically generated from GPT-4 using templated prompts. The value of this dataset lies in its linguistic simplicity, whereby it is possible to model language well on the dataset using very small models. Unless stated otherwise, our experiments use a 160M parameter model trained for 4 epochs, which achieves a loss of around 1.11 nats on the validation set. We train for 4 epochs since we use learning rate warmup and cosine learning rate decay and we want to ensure all datapoints receive updates with a high learning rate (this way all N -gram statistics have a fair chance of being learned during training). For overfitting experiments in Section 6.2, we train a 1.4B model for 10 epochs. In the Appendix, we include additional corresponding experiments on Wikipedia (from MassiveText [25]) with a single epoch of training in order to validate that our results are of a general nature and extend to more complex datasets. For a fixed dataset, the only source of randomness among different runs are different dataset shuffles. Full experimental details are described in the Appendix.

4 N -Gram Rules

The attention layer within a transformer is in essence a soft context-selection mechanism. The N -gram rules we consider will be loosely modeled on this mechanism. Namely, given a context we will form a derived context in which each token will either be kept, discarded, or marginalized, which is meant to mimic positive attention, no attention, and semantic invariance⁴, respectively. More formally, we proceed as follows:

Given a regular expression⁵ α , all contexts from the training data can be retrieved which match the regular expression. This allows us to define a corresponding rule that defines for us a distribution

³https://github.com/google-deepmind/transformer_ngrams

⁴For instance, the next token distribution for the context “... the tired dog” may be insensitive to replacing “tired” with “brown” or “furry”. Statistics which thus marginalize over all extant substitutions for “tired” yield a crude but generally applicable way of capturing semantic invariance. One can imagine an attention mechanism for which there is a many-to-one mapping of keys to a particular value that might implement semantic invariance.

⁵Our regular expressions operate on tokens not string characters, since our contexts are formed out of sequences of tokens.

over tokens t :

$$R_\alpha(t) = \frac{\#\{\alpha t\}}{\#\{\alpha *\}} \quad (1)$$

where the numerator and denominator are the counts for the N -grams from the training data matching the concatenated regular expressions αt and $\alpha *$, respectively, where $*$ is wildcard (single) character match⁶. (Thus the N -grams in the numerator end with t while those in the denominator can end with any token.) Observe that the next-token predictions of a vanilla N -gram model are obtained by letting $R_\alpha(t)$ vary over all ordinary token sequences α of length $N - 1$.

Given σ , a symbol from the the alphabet $\{*, -, +\}$, consider the following operation which maps a token t to a regular expression:

$$S_\sigma(t) = \begin{cases} t & \sigma = + \\ * & \sigma = * \\ \epsilon & \sigma = - \end{cases} \quad (2)$$

where ϵ is the empty regular expression. Given now a sequence $\sigma = \sigma_{-N} \cdots \sigma_{-2} \sigma_{-1}$, define S_σ on a sequence of tokens $C = C_{-N} \cdots C_{-2} C_{-1}$ by tokenwise application of (2) and concatenation⁷:

$$S_\sigma(C) = S_{\sigma_{-N}}(C_{-N}) \cdots S_{\sigma_{-2}}(C_{-2}) S_{\sigma_{-1}}(C_{-1}). \quad (3)$$

Thus (3) defines a regular expression which we can think of as fuzzy matching for a subset of a context C (the fuzziness arising from the presence of wildcard matches). For notational convenience, we assume σ is left padded with $-$ symbols, so that we can define $S_\sigma(C)$ for $\text{len}(\sigma) < \text{len}(C)$. Finally, define

$$R_\sigma(t|C) = R_{S_\sigma(C)}(t) \quad (4)$$

for C with $\text{len}(C) \geq \text{len}(\sigma)$. The collection of (4) for various σ defines our N -gram rules under consideration⁸. Each such rule is a function which maps a context C to a next token distribution. We refer to $S_\sigma(C)$ as the rule context for $R_\sigma(t|C)$.

As concrete examples, let $\sigma = + - * +$. If $C = C_{-5} C_{-4} C_{-3} C_{-2} C_{-1}$, then $S_\sigma(C) = C_{-4} * C_{-1}$ and

$$R_{+-*+}(t|C) = \frac{\#\{C_{-4} * C_{-1} t\}}{\#\{C_{-4} * C_{-1} *\}} \quad (5)$$

is a rule which yields a next token distribution based on a particular combination of 4-gram model statistics: it retrieves all three token contexts in the training data whose first token is C_{-4} and last token is C_{-1} and marginalizes over the second token. Likewise, the rules

$$R_{+ + - -}(t|C) = \frac{\#\{C_{-4} C_{-3} t\}}{\#\{C_{-4} C_{-3} *\}} \quad R_{+ + * * *} = \frac{\#\{C_{-4} C_{-3} * * t\}}{\#\{C_{-4} C_{-3} * * *\}} \quad (6)$$

are respectively a trigram model with context $C_{-4} C_{-3}$ (all other tokens receiving a $-$ are dropped) and a model which uses four tokens of context but marginalizes over the two most recent ones.

When σ consists of all $+$ symbols, we get vanilla N -gram rules derived from the suffix of C . When σ consists of \pm symbols, we get vanilla N -gram rules derived from subsets of C . Varying the length and the entries of σ yields the following rulesets⁹:

$$\mathcal{R}_M^{\text{suffix}} = \{R_\sigma(t|\cdot) : |\sigma| \leq M, \sigma_i = + \text{ for all } i\} \quad (7)$$

$$\mathcal{R}_M^{\text{subgram}} = \{R_\sigma(t|\cdot) : |\sigma| \leq M, \sigma_i = \pm \text{ for all } i\} \quad (8)$$

$$\mathcal{R}_M^{\text{all}} = \{R_\sigma(t|\cdot) : |\sigma| \leq M\}. \quad (9)$$

The parameter M controls how much of the context is being used for the rules.

⁶We use $*$ (i.e. glob notation) instead of the standard $.$ symbol for readability purposes.

⁷The empty regular expression does nothing under concatenation and does not contribute to the length of the resulting sequence.

⁸For $\sigma = \emptyset$, we define R_σ to be the unigram distribution.

⁹There is some redundancy among the σ 's in terms of the rules they determine: for instance, in between any two $+$ consecutive symbols, permuting the order of $-$ and $*$ will determine the same rule. Also in practice, we can assume the first entry of σ is a $+$ since marginalizing the first token is equivalent to reducing the context length. From this, it follows that the number of distinct rules in $\mathcal{R}_M^{\text{all}}$ is 2, 5, 13, 34, 89, 233, 378, for $M = 1, \dots, 7$, respectively.

Table 1: **Terminology associated to a context C** . Here \mathcal{R} is some reference ruleset under consideration. The superscript on $p^{(i)}(t|C)$ is meant to denote the predictions of the i th model. In Section 5, we consider rules that are fixed across model runs (where we have five models) whereas elsewhere we will only have a single model (and thus optimal rules will be model specific).

<i>optimal rule distance</i> : the minimum (possibly averaged over runs) distance between LLM predictions and rule predictions	$\min_{r \in \mathcal{R}} \text{avg}_i d(p^{(i)}(t C), p_r(t C))$
<i>optimal rule</i> : a rule achieving the optimal distance	$\text{argmin}_{r \in \mathcal{R}} \text{avg}_i d(p^{(i)}(t C), p_r(t C))$
<i>model variance</i> : the average of the pairwise distance between LLM predictive distributions over different runs	$\text{avg}_{\substack{i,j \\ \text{distinct runs}}} d(p^{(i)}(t C), p^{(j)}(t C))$

5 Approximating Transformer Predictions with Rules

Let $p(t|C)$ denote the next-token distribution of an LLM conditioned on the context C and for notational similarity, write $p_r(t|C)$ for $r(t|C)$, where r is one of the rules defined in Section 4. We wish to measure how similar these distributions are (higher similarity corresponds to a better rule description). To that end, we use the variational distance to measure the difference of two distributions (we discuss our choice and others in the Appendix):

$$d(p, q) = \frac{1}{2} \sum_{\alpha} |p_{\alpha} - q_{\alpha}|, \quad (10)$$

where the summation is over the vocabulary index (i.e. the components of the probability vectors). Since variational distance may be lacking in concrete interpretability, we will sometimes use top1-accuracy to measure similarity, defined by

$$\text{top-1-acc}(p, q) = \frac{|\text{argmax}(p) \cap \text{argmax}(q)|}{|\text{argmax}(p) \cup \text{argmax}(q)|} \quad (11)$$

(in general, the argmax of a probability distribution is a set due to potential ties among maximal probabilities). When the argmaxes in (11) are singletons, top1-accuracy just measures agreement between greedy predictions.

Given a context C , we want to understand how $d(p(t|C), p_r(t|C))$ varies with different rules r and in particular if it can be made small. To that end, we introduce some terminology:

We are interested in determining the optimal rule $p_r(t|C)$ (as defined in Table 1) and if it has small optimal rule distance then we regard the rule as being a good description of the corresponding transformer predictions.¹⁰ As a first step, note there is a distinguished rule

$$p_{\text{full}}(t|C) = \frac{\#\{Ct\}}{\#\{C*\}} \quad (12)$$

whose rule context is the full unmodified context C .¹¹ This is because (roughly) the language-modeling objective aims to make $p(t|C)$ similar to $p_{\text{full}}(t|C)$.¹² All other rules in our rulesets are “subleading” in that they drop or marginalize over tokens in the context C . Our goal is to quantify which rules, either (12) or subleading ones, are optimal rules and what their optimal rule distances are.

One of our main findings is an *approximation criterion*: contexts that have low model-variance tend to have low optimal rule distance. In particular, this includes contexts with sufficiently high frequency in the training data. The latter situation is to be expected: the more often a context C occurs in the

¹⁰In practice, we will only have one model available and our optimal rules are computed per-context and per-model. In this section, we have available five models for use in computing optimal quantities.

¹¹That is, $p_{\text{full}}(t|C)$ is the invocation of the rule corresponding to $\sigma = + \dots + \in \mathcal{R}_{|C|}^{\text{suffix}}$ applied to C .

¹²See Section C for additional discussion.

training data, the more the minimization of the cross entropy loss objective encourages the network to make predictions close to $p_{\text{full}}(t|C)$.

The novel aspect of our approximation criterion is the *sufficiency* of low model-variance situation even in cases when the context is rare.¹³ We present the case of 7-gram contexts in Figure 2 to corroborate the approximation criterion, with additional examples relegated to the Appendix. We sample around six-thousand 7-grams from the training data, sampling from logarithmically spaced buckets based on counts, and plot various relations between counts, model variances, and rule distances. For simplicity, we consider the ruleset $\mathcal{R} = \mathcal{R}_7^{\text{suffix}}$ to limit the number of rules under consideration. Our analysis of Figure 2 can be summarized as follows:

Plot (a) shows how with increasing count of the number occurrences of the context C in the training data, LLM predictions become nearer to $p_{\text{full}}(t|C)$, which in this case, is the vanilla 8-gram rule. Nevertheless, for all but the highest of counts, we have a large spread of distances: even for unique 7-gram contexts, some predictions are well-approximated by $p_{\text{full}}(t|C)$ while others are close to having disjoint-support (distance equal to 1). Plot (c) also shows that while model variance decreases with count of the context (as expected) we have a large spread of model variances for contexts with intermediate or low counts. Since the contexts whose predictions have high model variance can be regarded as “noise”, one can ask whether those contexts with low model variance have some structure. Plots (b) and (c) address this question. While for (b), we see that the 8gram-rule has a large spread when plotted against model variance, there is a significant reduction in outliers in (d) when the y-axis is the optimal distance to rules in $\mathcal{R}_7^{\text{suffix}}$. Concretely, the transition from (b) to (d) is a way of visualizing LLMs performing back-off, whereby LLMs rely on statistics from subsets of the context. Moreover, the lower left portion of (d) is a manifestation of our approximation criterion: contexts that yield consistent predictions across model runs (i.e. sufficiently low model variance) are indicative of rule-like behavior (in this case, good approximation with a suffix N -gram rule formed out of the context).

We believe our approximation criterion and its corresponding analyses have significance beyond the experiments carried out here since they (i) highlight that naive count-based statistics do not provide the strongest signal in terms of how LLMs leverage dataset statistics (since as Figure 2(a) shows, high count can still have high model variance) (ii) suggest that LLM predictions that have low-variance are likely the ones that are amenable to description (or even explanation) by some underlying dataset statistic (with high-variance predictions being regarded as noise). We leave a more systematic exploration of (ii) to future work.

6 Learning Dynamics

6.1 Curriculum Learning

We can track how well LLM predictions are described by our N -gram rules over the course of training by tracking optimal rule distance as a function of train step. Here optimal rule distance is defined as in Table 1 with \mathcal{R} any of the rulesets (7)-(9), and we will measure how optimal rule distances vary with maximum context length M (the resulting analyses are similar for “all”, “subgram”, and “suffix” rules so we show our analysis for “all”).

Figure 3 summarizes our results. Early in training, LLM predictions acquire structure and thus become approximable by rule predictors. However, with further training, LLM predictions eventually diverge from simpler rules (small context length) while continuing to increase in similarity with more complex rules (larger context length). Moreover, the rightmost plot of Figure 3 shows that $\text{top1-acc}(p_{\text{gt}}(t|C), p_r(t|C))$ increases over the course of training for optimal $r \in \mathcal{R}_M^{\text{all}}$ (for $M > 1$), where p_{gt} is the ground-truth distribution regarded as a one-hot distribution, showing that the rule selection improves with time. Altogether, this shows that LLMs undergo a curriculum style learning, in which their predictions gradually move away from simpler rules to more complex and effective rules.

¹³Necessity is a given. Predictions which have high variance cannot be well approximated by a single model-independent rule. We use five runs in our analysis here since approximation by a rule that remains fixed across models yields a fortiori approximation by a per-model rule.

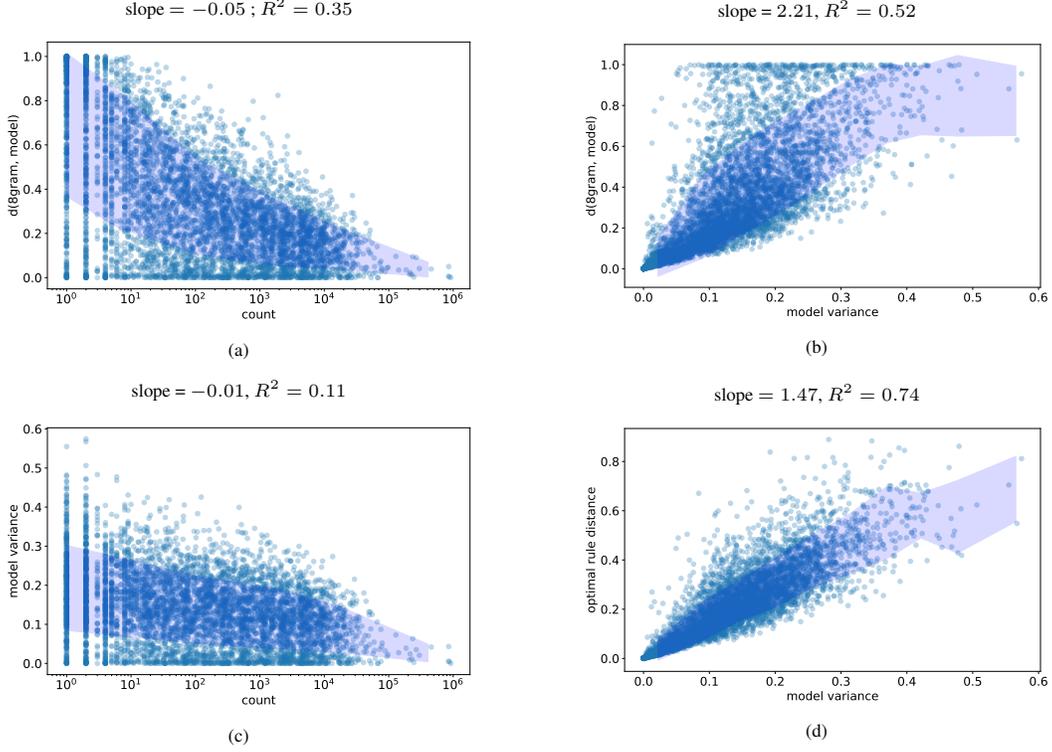


Figure 2: **TinyStories 7-grams**. Every point in the above plots represents a 7-gram context. Shaded regions are obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. Slope and R^2 values of plots are with respect to the linear fit of the data. Optimal rule distances and model variances are computed with respect to five model runs. (a): $d(p(t|C), p_{\text{full}}(t|C))$ vs count of C . (b): $d(p(t|C), p_{\text{full}}(t|C))$ vs model variance. (c): model variance vs count of C . (d): similar to (b) but now the y-axis is optimal rule distance of the optimal rule from $\mathcal{R}_7^{\text{suffix}}$. Model size: 160M.

6.2 Early Stopping Criterion

Our investigations of approximating LLMs with rules given by limited contexts naturally lead us to consider LLMs with limited context. The latter have predictive distributions given by

$$p_n(t|C) = p(t|C_{-n} \cdots C_{-1}) \quad (13)$$

where n is the maximum context length. In Figure 4, we plot the loss of an LLM trained to overfit (train loss decreases while validation loss increases) along with its limited context versions for $1 \leq n \leq 7$. For the limited context models with $n > 1$, we see that on *both* the train and validation set, the two respective loss curves track each other closely and both eventually go up. This suggests the following picture: an overfitting LLM is spending capacity to minimize train loss by memorizing the full context at the expense of using capacity to learn statistics of subcontext, i.e. the reduced context in (13). This manifests itself both during training (where subcontext arises from a subset of a larger memorized context) and during validation (where subcontext arises from the partial overlap between novel context and the train set).

Our discovery suggests a simple and computationally inexpensive early stopping criterion: during training, evaluate the transformer on train data consisting of short contexts and when this quantity begins increasing, stop training. Significantly, this method involves no holdout set and is a training dataset intrinsic criterion.

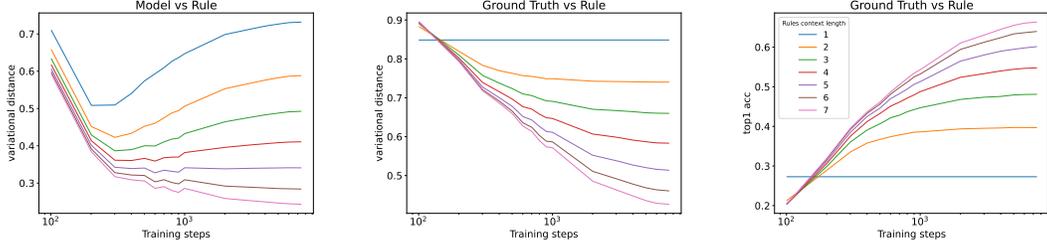


Figure 3: **Training Dynamics.** *Left:* Models reach their lowest distance to more complex rules later in training. For rules with four tokens of context or less, the variational distance eventually starts increasing later in training. For six and seven tokens of context, the variational distance continues to decrease. *Center & Right:* The optimal rule selected always has nonincreasing distance and nondecreasing top1-accuracy relative to the ground truth (interpreted as a one-hot distribution p_{gt}), despite distances to model predictions eventually increasing or plateauing for rules with less than six tokens of context. This shows that the optimal rule selection is improving with additional training even if the optimal rule distance with respect to model predictions is not improving. (One can imagine the rule predictions as a mesh in probability space, with LLM predictions navigating this space through training. The distance to the mesh may plateau but which rule is closest can continue to change.) Model size: 160M.

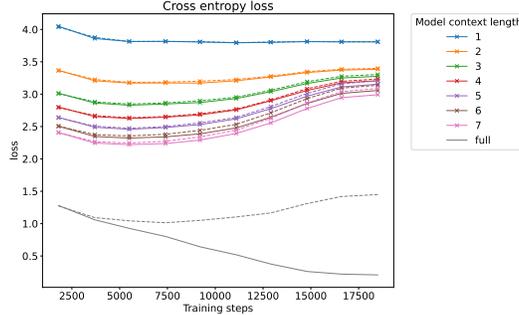


Figure 4: **Overfitting Detection.** We plot both train loss (solid lines) and validation loss (dashed lines) for the full transformer and limited context length transformers (the latter are marked with an "x" for emphasis) on TinyStories. Unlike the full transformer which overfits, those with limited context length have train and validation loss curves closely following each other. Model size: 1.4B.

7 Rule Performance

Finally, addressing our main question from the introduction, we track how well our rulesets describe LLM predictions (in the sense of Section 5) as a whole at inference time. Here, the utility of our N -gram rules defined in Section 4 becomes apparent, since on a holdout set, there will be novel contexts and being able to drop or marginalize context tokens aid in being able to retrieve or aggregate corresponding training dataset statistics. In Table 2, we show the average top1-accuracy between the optimal rule from our various rulesets and LLM predictions on 100 random stories from the validation set. Here, we include as baseline backoff_M , the single rule given by the predictive model which performs "stupid backoff" [4] using M tokens of context.¹⁴

We see significant gains in accuracy at large M when adding additional types of rules. In the end, we are able to obtain 78% top1-accuracy between the per-prediction optimal rule and the LLM predictions, averaged over all tokens. This is perhaps a remarkably high figure, considering that the top1 accuracy of the model with respect to the ground truth on the validation set is 69.6%. At minimum, we have provided a precise quantification of structure in LLM next-token predictions: they are often matched (as measured by top token prediction) by some simple N -gram rule derived from

¹⁴That is, $p_{\text{backoff}_M}(t|C) = p_{\text{full}}(t|C_{-k} \cdots C_{-1})$ where $k \leq M$ is the largest value for which $C_{-k} \cdots C_{-1}$ occurs in the training data.

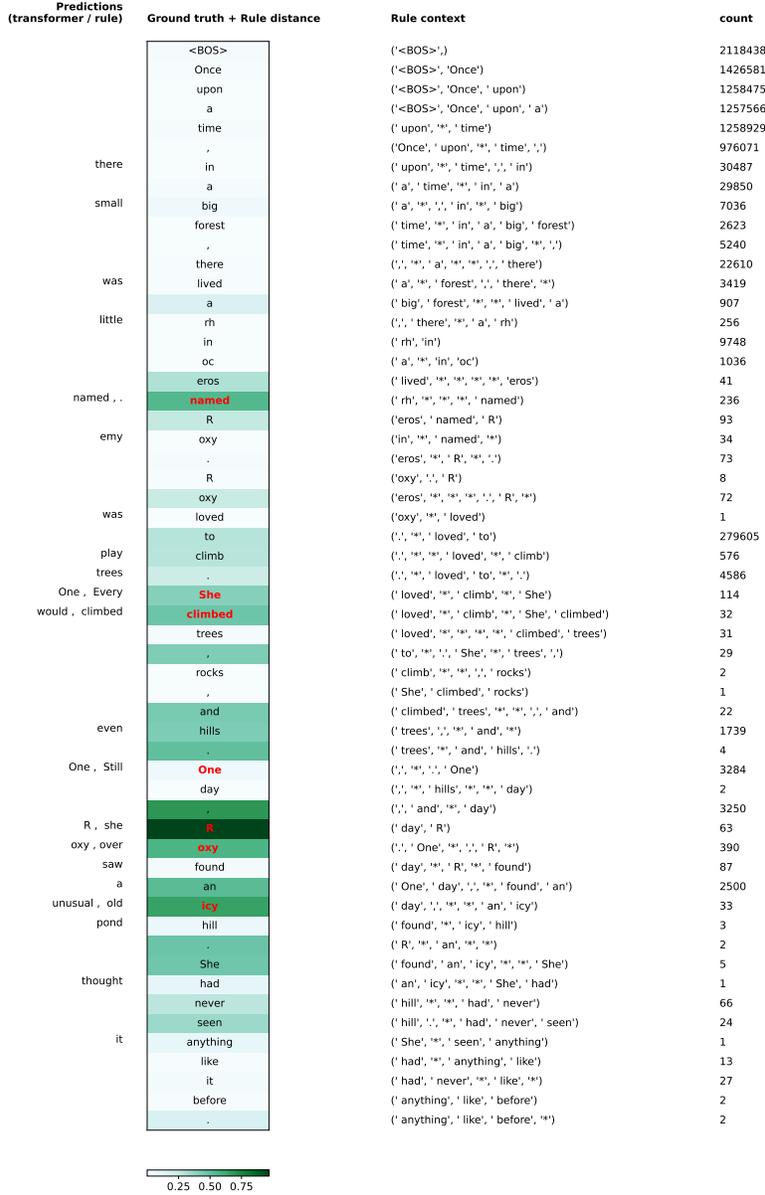


Figure 5: **Rule selection for a TinyStories validation sequence.** The above is a sequence from a heldout story. In the second and third columns are the ground truth, token by token, along with the rule context (as defined in Section 4) associated to the optimal rule from $\mathcal{R}_7^{\text{all}}$. The heatmap indicates the variational distance between optimal rule and LLM next token distributions at the given token. The first column shows at most two tokens, which are chosen as follows: If the LLM top-1 prediction disagrees with the ground truth, the LLM prediction is shown. If in addition, the rule selected makes a different top-1 prediction from the transformer, that token is shown as the second token and the corresponding ground truth token is colored red. Thus red tokens are precisely the locations of disagreement between LLM and optimal rule greedy predictions. The last column shows the number of contexts supporting the optimal rule. Model size: 160M.

the training data. See Section D.1 for some supplementary analysis. Using the L^∞ distance instead of the variational distance gives us a slightly higher result of 79%, see Table 5.

To ground our rule optimization procedure, we provide Figure 5 which shows side-by-side how LLM predictions compare with ground truth and optimal rule predictions in an example heldout story.

Table 2: **Top-1 accuracy of optimal rule.** We look at the average top1-accuracy of the optimal rule versus LLM predictions for rules of varying strength and maximum rule context length M . We compute this average over each token prediction from 100 random validation stories (around 22K tokens total). Model size: 160M.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	30.0	44.8	54.3	62.4	68.8	74.0	78.0
$\mathcal{R}_M^{\text{subgram}}$	30.0	44.5	53.1	60.0	64.8	68.5	71.1
$\mathcal{R}_M^{\text{suffix}}$	30.0	44.4	52.2	57.9	61.5	63.8	65.6
backoff_M	30.0	42.5	48.7	52.7	54.6	55.9	56.7

For instance, for the target token “climb” in “... Roxy loved to climb”, both the LLM and optimal rule R_α predict “play”, where $\alpha = “. * loved to”$. For target token “climb” in “... She climbed”, the LLM predicts “would” whereas the ground truth and R_α predict “climb”, where $\alpha = “loved to climb * She”$. In general, optimal rules provide the closest statistical match from the training data to the given LLM predictive distributions (from amongst our rulesets), and their top1-predictions can agree or disagree (as indicated by target token color). Additional examples, including those from Wikipedia, are shown in Section D. For interpretability purposes, we re-emphasize that our optimal rules currently only provide descriptions, not explanations. We leave the possibility of the latter for future work.

8 Conclusions and Limitations

Our work provides quantitative measures of how well the predictions of transformer-based LLMs are described (i.e. approximated) by simple N -gram rules. Such rules were motivated by the simplest token-level operations applied to the context (keep, ignore, or marginalize). The results we obtained in Section 7 imply that, at least on simple datasets like TinyStories and Wikipedia, LLM predictions contain much quantifiable structure insofar that they often can be described in terms of our simple statistical rules. Along the way, we also obtained novel discoveries into the statistical nature of overfitting, the occurrence of curriculum learning, and the relation between model-variance and approximability by N -gram rules. Altogether then, our work provides various avenues of progress in understanding how simple dataset statistics are reflected in LLM behavior.

On the other hand, it is intuitively clear that current state-of-the-art LLMs go well beyond invoking N -gram rules. A typical request to perform a nontrivial task (e.g. “Write a thirty line poem about mathematics that rhymes”) requires a high-level conceptual understanding of language that goes beyond simple literal token-level associations between the context and the training data that we consider here. Nevertheless, one can speculate that an analogue of our work could still apply: in general, an LLM might be performing some high-level rule application, whereby statistics formed out of distributional categories [24] instead of individual tokens are leveraged from the context. Formulating a correct and parsimonious set of rules, if it is at all possible, would be a nontrivial challenge to overcome and one which we leave to future work. Addressing such a challenge and being able to promote the descriptive approximations provided here to explanatory ones would provide a next step towards understanding how LLMs work.

Acknowledgements

The author would like to especially thank Senthoran Rajamanoharan for numerous conversations and an exceptionally discerning eye, which greatly improved the paper from earlier drafts. The author also thanks Jonathan Hale, Marcus Hutter, Matthew McGill, Nick Roy, Avraham Ruderman, and Daniel Tanis for helpful feedback and discussions. Finally, the author thanks Frank Perbet and Daniel Tanis for engineering support.

References

- [1] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms, 2024.

- [2] Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models, 2023.
- [3] Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "a is b" fail to learn "b is a", 2024.
- [4] Thorsten Brants, Ashok C. Papat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, 2007.
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [6] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- [7] 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.
- [8] Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. The grammar-learning trajectories of neural language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8281–8297, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [9] Benjamin L. Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning markov chains, 2024.
- [10] Yanai Elazar, Nora Kassner, Shauli Ravfogel, Amir Feder, Abhilasha Ravichander, Marius Mosbach, Yonatan Belinkov, Hinrich Schütze, and Yoav Goldberg. Measuring causal effects of data statistics on language model’s ‘factual’ predictions, 2023.
- [11] Ronen Eldan and Yuanzhi Li. Tinstories: How small can language models be and still speak coherent english?, 2023.
- [12] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey, 2024.
- [13] Joshua Goodman. A bit of progress in language modeling. *CoRR*, cs.CL/0108005, 2001.
- [14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA, 2024. Curran Associates Inc.
- [15] Henrik Jacobsson. Rule extraction from recurrent neural networks: Ataxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

- [16] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org, 2023.
- [17] Cheongwoong Kang and Jaesik Choi. Impact of co-occurrence on factual knowledge of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
- [18] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1, 1995.
- [19] Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens, 2024.
- [20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.
- [21] Clayton Mcmillan, Michael Mozer, and Paul Smolensky. The connectionist scientist game: Rule extraction and refinement in a neural network. 09 1991.
- [22] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models, 2023.
- [23] 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. <https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html>.
- [24] Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of English words. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, Ohio, USA, June 1993. Association for Computational Linguistics.
- [25] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d’Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorraine Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher, 2022.
- [26] Yasaman Razeghi, IV RobertL.Logan, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Conference on Empirical Methods in Natural Language Processing*, 2022.
- [27] Anej Svete and Ryan Cotterell. Transformers can represent n -gram language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6845–6881, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

- [28] Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional, 2023.

A Choice of Distance Measure

We choose variational distance since it is a symmetric and bounded distance function (unlike the KL divergence). Symmetry means we do not have to make a choice between computing the distance between model predictions and rule predictions or vice versa. Boundedness ensures that when we measure average distance across tokens, large outliers do not dominate the average. In fact, for the KL divergence, since $KL(p||q)$ is infinite when $p > 0$ whenever $q = 0$, were we to use KL divergence, we would have to set p equal to rule predictions and q equal to model predictions (since rule predictions are typically sparse). To avoid such constraints and potential pathologies, we choose the variational distance. Another possible choice is the Jensen-Shannon distance, but we found it gives similar results to variational distance.

It is worth noting that while the L^∞ metric

$$d_{L^\infty}(p, q) = \max_{\alpha} |p_{\alpha} - q_{\alpha}| \tag{14}$$

often gives slightly better results for the rule-approximation analysis of Section 7, it has a failure mode when comparing two very high entropy distributions. If p and q are two distributions such that p_{α} and q_{α} are all small, then their L^∞ distance will be small even though their variational distance can be large. Thus, the L^∞ distance is not suitable for the curriculum learning analysis in Section 6.1, since at initialization when the model makes uniform predictions on tokens, it will have low L^∞ distance with many bad N -gram rules that are high entropy. Hence, for the sake of using a consistent metric in the main body of the paper, we use the variational distance when comparing probability distributions, although as the numbers of Section D show, for well-trained models the L^∞ distance often yields better rule approximation.

B Additional Experimental Details

Our transformer architecture and training procedure is based on that of Chinchilla [14]. The architecture hyperparameters are as follows:

Table 3: Model specifications.

Model	Layers	Number Heads	$d_{\text{key}}/d_{\text{value}}$	d_{model}
160M	12	16	64	896
420M	12	16	128	1536
1.4B	24	16	128	2048

We use a linear learning rate warmup of 1000 steps up to a maximum value of 2×10^{-4} and then use a cosine learning rate decay. We use weighted Adam optimizer [20] with weight decay 10^{-4} . Our models are trained using TPU accelerators. The 160M and 420M models use 16 TPU accelerators while the 1.4B models use 64 TPU accelerators per run. We use a batch size of 128 sequences with each sequence consisting of 2048 tokens.

Our training datasets (TinyStories and MassiveText Wikipedia) are prepared as follows. After tokenizing the individual documents (stories for TinyStories and articles for Wikipedia), we concatenate them all into one long sequence, with each document separated by the $\langle \text{BOS} \rangle$ token¹⁵. The full sequence is then subdivided into contiguous sequences of length 2048 (with padding as needed) and then shuffled to form a static dataset of shuffled sequences. We refer to the previous procedure as "chunking". Crucially, observe that chunking results in most sequences not starting with the $\langle \text{BOS} \rangle$ token (hence a model will be trained to predict the next token conditioned on incomplete contexts, as desired).

For TinyStories experiments, we train 160M models for 4 epochs except for the overfitting experiments in Section 6.2 where we train 1.4B models for 10 epochs. We use the train and validation splits provided by HuggingFace¹⁶. For Wikipedia experiments, we train a 1.4B model for a single epoch.

¹⁵Attention masks are used so that tokens only attend to those from the same document

¹⁶Available at <https://huggingface.co/datasets/roneneldan/TinyStories>

We have train and validation splits based on using choosing random sets of disjoint documents. Our Wikipedia train set has 4.4B tokens. In places where we perform several training runs (Section 5), the only source of variance (randomness) among the runs are different dataset shuffles. The only exception to the above is in Section D.3, where all model sizes are trained for 1 epoch (for TinyStories, we observed overfitting of the 1.4B model around 4 epochs so we switch to 1 epoch for all model sizes for a fair comparison).

Our tokenizer¹⁷ uses byte-pair encoding trained on MassiveText with a vocabulary size of 32,678.

B.1 N -gram statistics

The computation of N -gram statistics of the training data is formed after chunking (as described above), so that they correspond to the N -gram statistics seen by models during training. In particular, tokens which are contiguous in a story but separated by the chunking will not contribute to the N -gram statistics. We used a distributed map-reduce system to tabulate N -gram counts in the most naive manner. Using sliding windows of size N and aggregating across train documents, we are able to compute N -gram counts for all occurring N -grams and store them in SQL databases. (We ignore those invalid N -grams where $\langle \text{BOS} \rangle$ occurs not as the first token). Note that the number of rows of such N -gram databases is bounded by at most the size of the training corpus times N .

As an aside, we note that for the analysis in Section 6.1, we used our static N -gram rules computed from the entire training data. We do not compute statistics based on the training dataset seen up to the point in training. However, for the purposes of our analysis, this distinction is immaterial (and in practice, the distinction between two sets of statistics will, for the dominant N -grams, be small with sufficiently large batch size).

C Additional Approximation Criterion Analyses

We provide additional commentary and experimental settings for our analysis in Section 5.

C.1 Full-context vs Subcontext

As noted in footnote 12, there is usually a mismatch between the contexts that N -gram rules and LLMs receive during training: the latter can receive very long contexts (up to one less than the number of tokens in a document) while the former typically receives very short contexts (in our case, up to 7 tokens). Concretely, while a bigram model is trained on consecutive pairs of tokens (c, t) , an LLM is rarely trained so as to optimize $p(t|c)$. Indeed, given a training sequence x , only the target for the first token of x has context consisting of a single token; the other targets will have more tokens of context accordingly. Thus, it is unclear how well LLM predictions $p(t|c)$ should match bigram rule predictions as c varies over the vocabulary set, since LLMs almost always receive c within a much larger context. More generally, it is unclear how well $p(t|C)$ matches $p_{\text{full}}(t|C)$. Nevertheless, because in practice LLMs learn how to use context effectively, LLMs manage to learn $p(t|C)$ despite being optimized for $p(t|\tilde{C})$ with \tilde{C} a context containing C as a suffix.

As a measure of how much training context “dilutes” the LLM ability to learn the bigram distribution, in Figure 6 we plot the distance between LLM predictions and the bigram rule for two LLMs: one trained in the usual fashion with full context and one trained with only one token of context (concretely, a token can only attend to itself in attention layers). In both cases, we have the same pattern of increased count leads to lower rule distance. However, the transformer trained with context length equal to one has much lower distances since it cannot learn anything else other than the bigram rule. The difference between the variational differences of the two models is thus a measure of the dilution an LLM has in learning a bigram rule owing to receiving surrounding context.

As an aside, we note how for both models, a context with low count has difficulty being learned. In this way, one can regard the inability to learn rules for low count contexts as being due to a failure of optimization, something that could be addressed in the future by improved optimization methods.

¹⁷Trained using <https://github.com/google/sentencepiece>

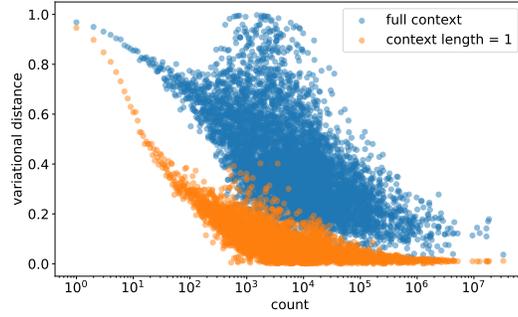


Figure 6: **Comparison with TinyStories bigram model.** We evaluate transformer models (trained with either full context or context length equal to one) on all 22.8K distinct unigrams of TinyStories and record the corresponding variational distance with the bigram rule. Grouping unigrams based on count and averaging the variational distances result in the above scatterplots. Model size: 160M.

C.2 TinyStories Unigram Context

We repeat Figure 2 for the simplest case of unigram context. In this case, there is only one rule (the bigram rule) and so there are only three plots to consider.

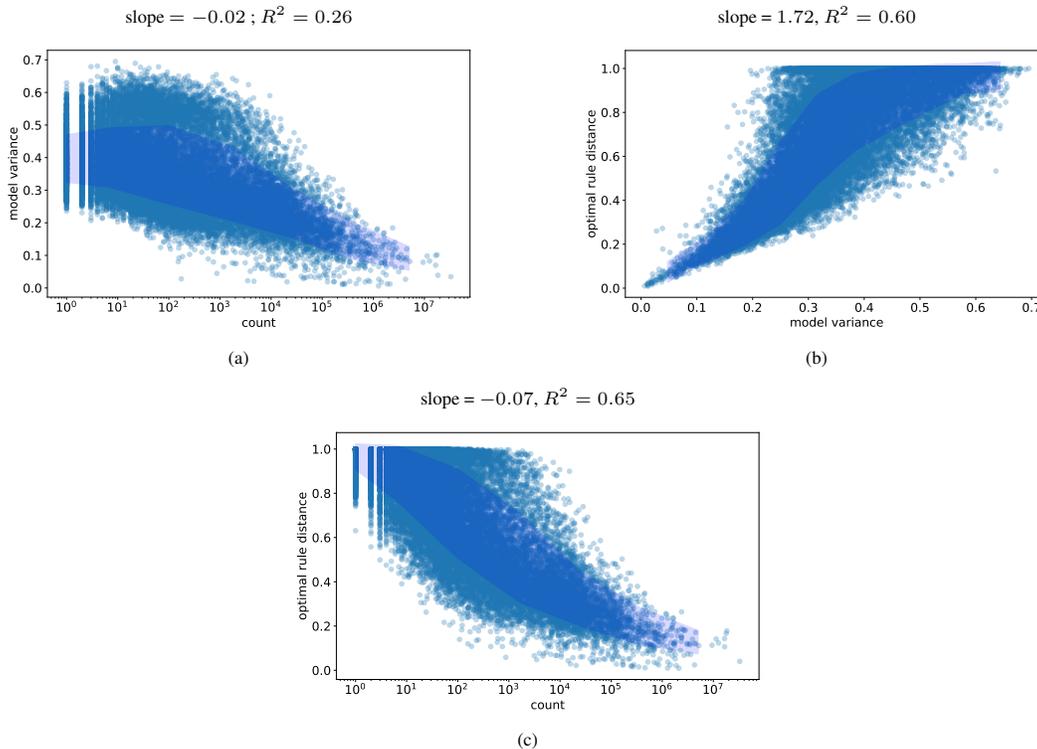


Figure 7: **TinyStories 1-grams.** Every point in the above plots represents a 1-gram context (all 22.8K from TinyStories). Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. Slope and R^2 values of plots are with respect to the linear fit of the data given by their axes. Optimal rule distances and model variances are computed with respect to five model runs. (a): model variance vs count of C (b): $d(p(t|C), p_{\text{full}}(t|C))$ vs model variance (c): $d(p(t|C), p_{\text{full}}(t|C))$ vs count. Model size: 160M.

C.3 Tinystories Bigram Context

Next, consider the case when there are two tokens of context. To get a more fine-grained analysis, we consider the case of full-context bigrams, i.e. those starting with the $\langle \text{BOS} \rangle$ token. This is because such bigrams do not appear within a larger context and so a transformer’s corresponding predictions are more fair to compare with those of N -gram models (both are trained using equal contexts). Conveniently, there are only 691 full-context bigrams in this case and so we do not have to randomly sample a subset.

We will consider the ruleset $\mathcal{R}_2^{\text{all}}$ for which there are three relevant N -gram rules of interest: one which uses the entire bigram of context (a trigram model), one which uses only the last token (a bigram model), and one which uses only the first token (the next token distribution of $\langle \text{BOS} \rangle$).¹⁸ We will refer to these as the trigram, bigram, and $\langle \text{BOS} \rangle$ rule respectively.

We plot an analog of Figure 2 for full-context bigrams in Figure 8. As with the analysis for Figure 2(a), a large count leads to good approximation with $p_{\text{full}}(t|C)$, which is the vanilla trigram rule at present. However lower counts lead to a spread in approximability (some low counts have high error while others have low ones). In (c), we plot a variation in which the x -axis is the maximum of the count of C and the unigram C_{-1} . What the poor fit in (c) indicates is that whether a prediction is well-described by a rule is not a simply determined by whether a subcontext of C occurs often. Given the small number of rules, we now color code the optimal rule of each full-context bigram (as indicated by the legend in (b)). In passing from (b) to (d), we see how the outliers in the upper left of (b) move towards the bottom once the large distance from the trigram model is replaced with the optimal rule distance. These are bigrams whose rules are well approximated by bigram or $\langle \text{BOS} \rangle$ rules and are misspecified when trying to be approximated by the trigram rule. In accord with our Approximation Criterion, contexts with low model variance are well approximated by N -gram rules (the lower left of (d)).

C.4 Wikipedia 6-gram contexts

We plot the analog of Figure 2 in Figure 9 but for contexts consisting of 6-grams from Wikipedia. We also subsample as before, from logarithmically spaced buckets, for a total of around 6.8K total contexts. We get nearly identical behavior as with TinyStories. Our Approximation Criterion is thus not specific to small datasets like TinyStories.

¹⁸It turns out that the $\langle \text{BOS} \rangle^*$ rule (given by R_{+^*}) in $\mathcal{R}_2^{\text{all}}$ never occurs as an optimal rule for full-context bigrams and so can be ignored in this example.

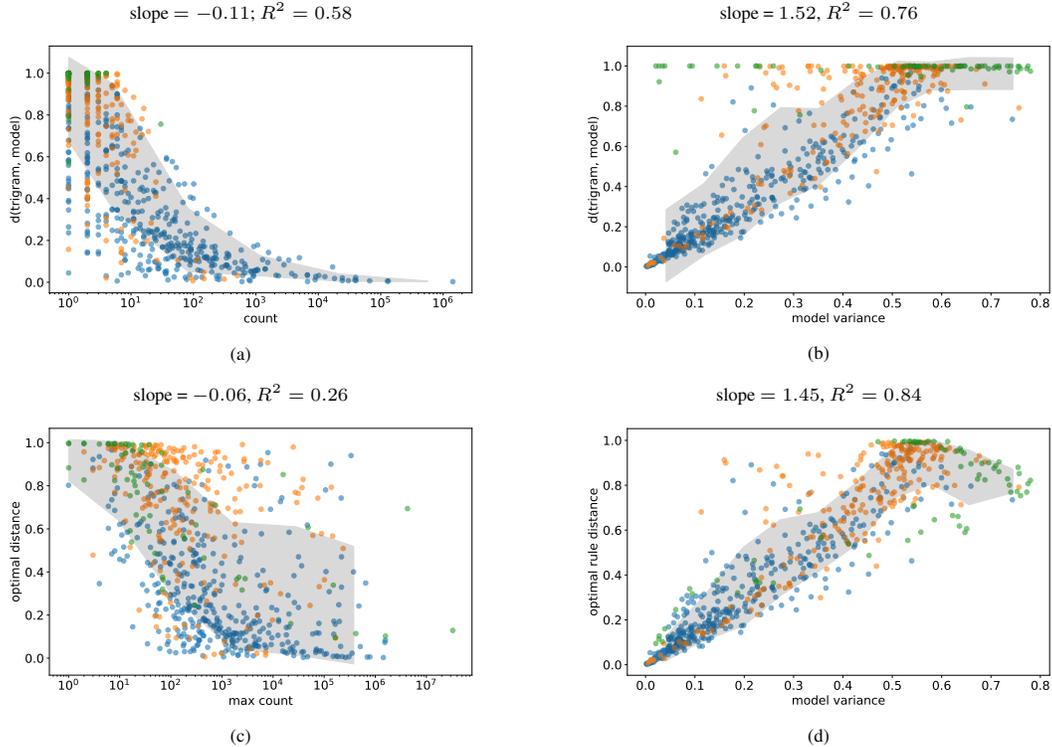


Figure 8: **TinyStories full-context bigrams**. Every point in the above plots represents a full-context bigram C from among the 691 distinct ones in TinyStories. Points are colored by which N -gram rule is the optimal rule, among those in $\mathcal{R}_2^{\text{all}}$, for transformer prediction given C . Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. (a): $d(p(\cdot|C), p_{\text{trigram}}(\cdot|C))$ vs count of C . (b): $d(p(\cdot|C), p_{\text{trigram}}(\cdot|C))$ vs model variance. (c): Optimal rule distance vs the greater of the bigram count of C and the unigram count of C_{-1} . (d): Similar to upper right but now the y-axis is optimal rule distance. Five model runs are used to compute optimal rule distance and model variance. Model size: 160M.

D Rule Performance: Additional Analysis and Examples

D.1 TinyStories

We supplement Table 2 with Table 4 to show how optimal rule distances decrease with increasing rule strength. This is to preclude a trivial situation in which by having sufficiently many rules (say a one-hot distribution for every vocabulary token), one can have a ruleset that for any model prediction always returns an optimal rule with 100% top-1 accuracy! Such coarse rules will not in general yield small optimal distances however¹⁹ and our variational distances decreasing in Table 4 shows that our rulesets are truly better approximating the predictions with increasing strength.

We also include the analog of Tables 2 and (4 but with the variational distance replaced with the L^∞ distance in Tables 5 and 6. We see that in fact the top-1 accuracy numbers are slightly better in this case.

D.2 Wikipedia

We present analogous results for Section 7 using a 1.4B model trained on Wikipedia. In Table 7 we present the analogue of Table 2 on ten holdout Wikipedia chunks (a total of 10×2048 tokens). The

¹⁹Both the variational and L^∞ distances between a one-hot distribution and a distribution which is uniform on n tokens are at least $\frac{n-1}{n}$. Thus, whenever an LLM has at least two roughly valid options, we expect a one-hot distribution to be at least of distance 0.5 from the LLM prediction.

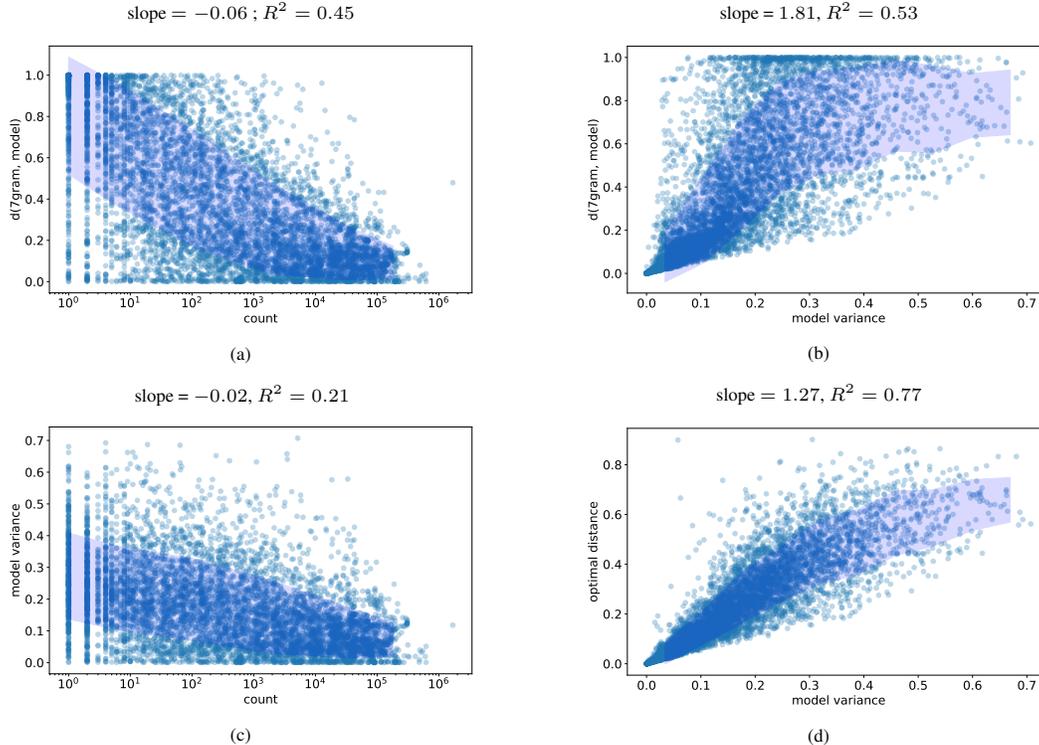


Figure 9: **Wikipedia 6-grams**. Every point in the above plots represents a 6-gram context. Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. Slope and R^2 values of plots are with respect to the linear fit of the data given by their axes. Optimal rule distances and model variances are computed with respect to five model runs. (a): $d(p(t|C), p_{\text{full}}(t|C))$ vs count of C . (b): $d(p(t|C), p_{\text{full}}(t|C))$ vs model variance. (c): model variance vs count of C . (d): similar to (b) but now the y-axis is optimal rule distance of the optimal rule from $\mathcal{R}_6^{\text{suffix}}$. Model size: 160M.

Table 4: **Optimal rule distance (TinyStories, variational distance)**. We look at the average optimal rule distance with LLM predictions for rules of varying strength and maximum context length M . We compute this average over each token prediction from 100 random TinyStories validation stories (around 22K tokens total). Model size: 160M.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	0.738	0.597	0.507	0.434	0.37	0.316	0.274
$\mathcal{R}_M^{\text{subgram}}$	0.738	0.598	0.513	0.449	0.399	0.362	0.335
$\mathcal{R}_M^{\text{suffix}}$	0.738	0.598	0.519	0.465	0.425	0.399	0.381
backoff $_M$	0.738	0.606	0.539	0.503	0.482	0.472	0.466

top-1 accuracy when using optimal rules from $\mathcal{R}_7^{\text{all}}$ and the L^∞ distance for rule selection is 67.7%. As with TinyStories, we see significant gains in accuracy when we increase rule strength. Achieving the number 67.7% (versus the corresponding 78.9% number for TinyStories from Table 5), perhaps a surprisingly a high score, is the result of two competing factors: on the one-hand, Wikipedia is a more complex dataset (which makes prediction harder), while on the other hand, the training data has more N -grams and thus more rules. Our model achieves 54.9% top-1 accuracy on the 10 holdout Wikipedia chunks (Table 12) which is substantially lower than the top-1 accuracy of the optimal rule.

Table 5: **Top-1 accuracy of optimal rule (TinyStories, L^∞ distance)**. The analogue of Table 2 but with L^∞ distance instead of variational distance for selecting the optimal rule. Model size: 160M.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	30.0	45.1	55.0	63.2	69.6	75.0	78.9
$\mathcal{R}_M^{\text{subgram}}$	30.0	44.7	53.5	60.4	65.1	68.5	71.2
$\mathcal{R}_M^{\text{suffix}}$	30.0	44.5	52.6	58.0	61.4	63.6	65.2
backoff $_M$	30.0	42.5	48.7	52.7	54.6	55.9	56.7

Table 6: **Optimal rule distance (TinyStories, L^∞ distance)**. The analogue of Table 4 but with L^∞ distance instead of variational distance. Model size: 160M.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	0.557	0.443	0.371	0.309	0.253	0.207	0.170
$\mathcal{R}_M^{\text{subgram}}$	0.557	0.444	0.376	0.322	0.28	0.248	0.225
$\mathcal{R}_M^{\text{suffix}}$	0.557	0.446	0.383	0.338	0.305	0.282	0.267
backoff $_M$	0.557	0.456	0.407	0.385	0.378	0.379	0.382

D.3 Model Scaling

In this section, we present results about how our rule approximation changes with model size. For both TinyStories and Wikipedia, we train models of size 160M, 420M, and 1.4B for one epoch²⁰.

In Tables 11 and 12, there is a clear trend towards improved model performance with scale: we obtain lower cross entropy loss, higher top-1 accuracy, and lower model distance to the ground truth distribution regarded as a one-hot distribution²¹. Note that for the latter, distance from model predictions to the ground truth distribution is the same with respect to the variational or L^∞ distance and is given by $1 - p$ where p is the probability assigned by the model to the ground truth.

On the other hand, entries in Tables 13-20 measuring the changes in rule approximation with scale are much more modest (i.e. are more stable) by comparison. They also show mixed results, with L^∞ results worsening but some variational distance results slightly improving with scale (for large context length rules). We leave a more thorough investigation of how model scale affects approximability by N -gram rules to future work.

²⁰Since the 1.4B starts showing signs of overfitting around 4 epochs on TinyStories, we train models for 1 epoch in this section unlike 4 epochs elsewhere.

²¹It would be more proper to aggregate statistics over the holdout set to compute a ground truth distribution that takes into the account the relative frequencies of the next token given the context. However, since most contexts will be unique, this more refined computation will not affect the corresponding result much.

Table 7: **Top-1 accuracy of optimal rule (Wikipedia, L^∞ distance)**. We look at the average top-1 accuracy between optimal rule and LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 10 holdout Wikipedia sequences each consisting of 2048 tokens. Model size: 1.4B.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	24.3	39.7	49.8	55.5	60.3	64.3	67.7
$\mathcal{R}_M^{\text{subgram}}$	24.3	39.2	48.3	52.7	55.5	57.6	59.0
$\mathcal{R}_M^{\text{suffix}}$	24.3	38.8	47.1	50.3	51.5	52.3	52.7
backoff $_M$	24.3	36.8	42.9	43.9	43.7	43.8	43.9

Table 8: **Optimal rule distance (Wikipedia, L^∞ distance)**. We look at the average L^∞ distance between optimal rule and LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 10 holdout Wikipedia sequences each consisting of 2048 tokens. Model size: 1.4B.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	0.48	0.369	0.29	0.237	0.202	0.173	0.150
$\mathcal{R}_M^{\text{subgram}}$	0.48	0.371	0.295	0.249	0.226	0.209	0.198
$\mathcal{R}_M^{\text{suffix}}$	0.48	0.375	0.303	0.265	0.250	0.242	0.238
backoff $_M$	0.48	0.394	0.359	0.368	0.387	0.398	0.405

Table 9: **Top-1 accuracy of optimal rule (Wikipedia, variational distance)**. We look at the average top-1 accuracy between optimal rule and LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 10 holdout Wikipedia sequences each consisting of 2048 tokens. Model size: 1.4B.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	24.3	39.2	48.8	54.4	58.5	61.9	65.0
$\mathcal{R}_M^{\text{subgram}}$	24.3	39.0	47.9	52.3	54.9	56.8	58.3
$\mathcal{R}_M^{\text{suffix}}$	24.3	38.7	46.9	50.3	51.5	52.4	52.8
backoff $_M$	24.3	36.8	42.9	43.9	43.7	43.8	43.9

Table 10: **Optimal rule distance (Wikipedia, variational distance)**. We look at the average variational distance between optimal rule and LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 10 holdout Wikipedia sequences each consisting of 2048 tokens. Model size: 1.4B.

ruleset / context length	1	2	3	4	5	6	7
$\mathcal{R}_M^{\text{all}}$	0.768	0.635	0.543	0.484	0.446	0.413	0.388
$\mathcal{R}_M^{\text{subgram}}$	0.768	0.636	0.549	0.498	0.472	0.453	0.440
$\mathcal{R}_M^{\text{suffix}}$	0.768	0.638	0.556	0.513	0.497	0.488	0.483
backoff $_M$	0.768	0.656	0.609	0.597	0.598	0.599	0.600

Table 11: **TinyStories metrics**. How average cross entropy loss, top-1 accuracy, and model distance to the ground truth scale with model size on a holdout set of 100 stories.

model size	eval loss	eval acc	eval distance
160M	1.43	63.2	0.485
420M	1.28	65.9	0.452
1.4B	1.22	66.9	0.439

Table 12: **Wikipedia metrics.** How average cross entropy loss, top-1 accuracy, and model distance to the ground truth scale with model size on a holdout set of 10 Wikipedia chunks.

model size	eval loss	eval acc	eval distance
160M	2.63	50.0	0.617
420M	2.43	52.3	0.590
1.4B	2.26	54.9	0.562

Table 13: **Top-1 accuracy of optimal rule with model scale (TinyStories, variational distance).** How top-1 accuracy of the optimal rule varies with model size and rule context length. Optimal rule from is selected from $\mathcal{R}_7^{\text{all}}$ using variational distance.

model size / context length	1	2	3	4	5	6	7
160M	31.4	47.1	56.7	64.3	69.8	74.5	77.9
420M	30.7	45.9	55.5	63.3	69.2	74.1	77.9
1.4B	30.5	45.9	55.3	63.4	69.6	74.3	78.2

Table 14: **Optimal rule distance with model scale (TinyStories, variational distance).** How optimal rule distance varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using variational distance.

model size / context length	1	2	3	4	5	6	7
160M	0.692	0.545	0.46	0.398	0.347	0.306	0.275
420M	0.711	0.566	0.479	0.411	0.355	0.310	0.274
1.4B	0.718	0.574	0.486	0.417	0.359	0.311	0.274

Table 15: **Top-1 accuracy of optimal rule with model scale (TinyStories, L^∞ distance).** How top-1 accuracy of the optimal rule varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using L^∞ distance.

model size / context length	1	2	3	4	5	6	7
160M	31.4	47.5	57.5	65.2	71.4	76.1	79.6
420M	30.7	46.3	56.3	64.2	70.5	75.3	79.2
1.4B	30.5	46.2	56.0	64.1	70.5	75.3	79.3

Table 16: **Optimal rule distance with model scale (TinyStories, L^∞ distance).** How optimal rule distance varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using L^∞ distance.

model size / context length	1	2	3	4	5	6	7
160M	0.482	0.368	0.301	0.248	0.204	0.169	0.141
420M	0.511	0.398	0.329	0.272	0.223	0.184	0.153
1.4B	0.522	0.408	0.338	0.280	0.230	0.189	0.157

Table 17: **Top-1 accuracy of optimal rule with model scale (Wikipedia, variational distance).** How top-1 accuracy of the optimal rule varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using variational distance.

model size / context length	1	2	3	4	5	6	7
160M	25.9	40.8	49.5	54.2	57.7	61.0	63.8
420M	24.7	39.8	49.0	54.2	58.1	61.6	64.3
1.4B	24.3	39.2	48.8	54.4	58.5	61.9	65.0

Table 18: **Optimal rule distance with model scale (Wikipedia, variational distance)**. How optimal rule distance varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using variational distance.

model size / context length	1	2	3	4	5	6	7
160M	0.737	0.606	0.527	0.477	0.445	0.418	0.397
420M	0.753	0.621	0.534	0.480	0.445	0.414	0.391
1.4B	0.768	0.635	0.543	0.484	0.446	0.413	0.388

Table 19: **Top-1 accuracy of optimal rule with model scale (Wikipedia, L^∞ distance)**. How top-1 accuracy of the optimal rule varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using L^∞ distance.

model size / context length	1	2	3	4	5	6	7
160M	25.9	41.2	50.4	55.9	60.4	64.4	67.8
420M	24.7	40.3	50.0	55.7	60.4	64.3	67.7
1.4B	24.3	39.7	49.8	55.5	60.3	64.3	67.7

Table 20: **Optimal rule distance with model scale (Wikipedia, L^∞ distance)**. How optimal rule distance varies with model size and rule context length. Optimal rule is selected from $\mathcal{R}_7^{\text{all}}$ using L^∞ distance.

model size context length	1	2	3	4	5	6	7
160M	0.429	0.321	0.253	0.208	0.179	0.154	0.135
420M	0.455	0.346	0.271	0.222	0.190	0.163	0.143
1.4B	0.480	0.369	0.290	0.237	0.202	0.173	0.150

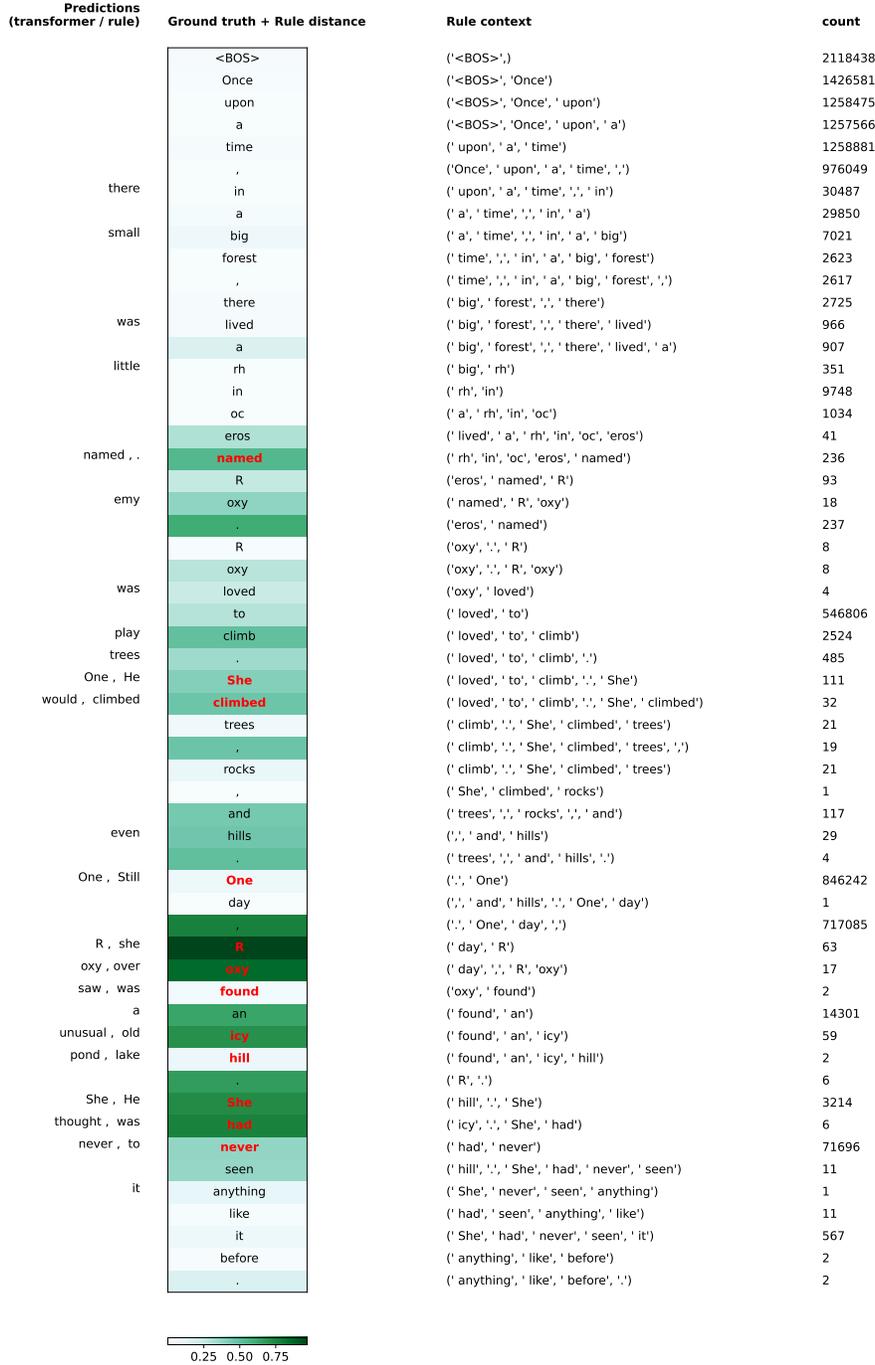


Figure 10: Rule selection for a TinyStories heldout sequence using $\mathcal{R}_7^{\text{subgram}}$. Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_7^{\text{subgram}}$ instead of $\mathcal{R}_7^{\text{all}}$. Model size: 160M.

D.4 Rule Selection

Here we supplement our example in Figure 5 by showing how the smaller rulesets $\mathcal{R}_7^{\text{subgram}}$ and $\mathcal{R}_7^{\text{suffix}}$ compare in Figures 10 and 11. As expected, the top1 accuracy between transformer predictions and optimal rule predictions decrease with smaller rulesets.

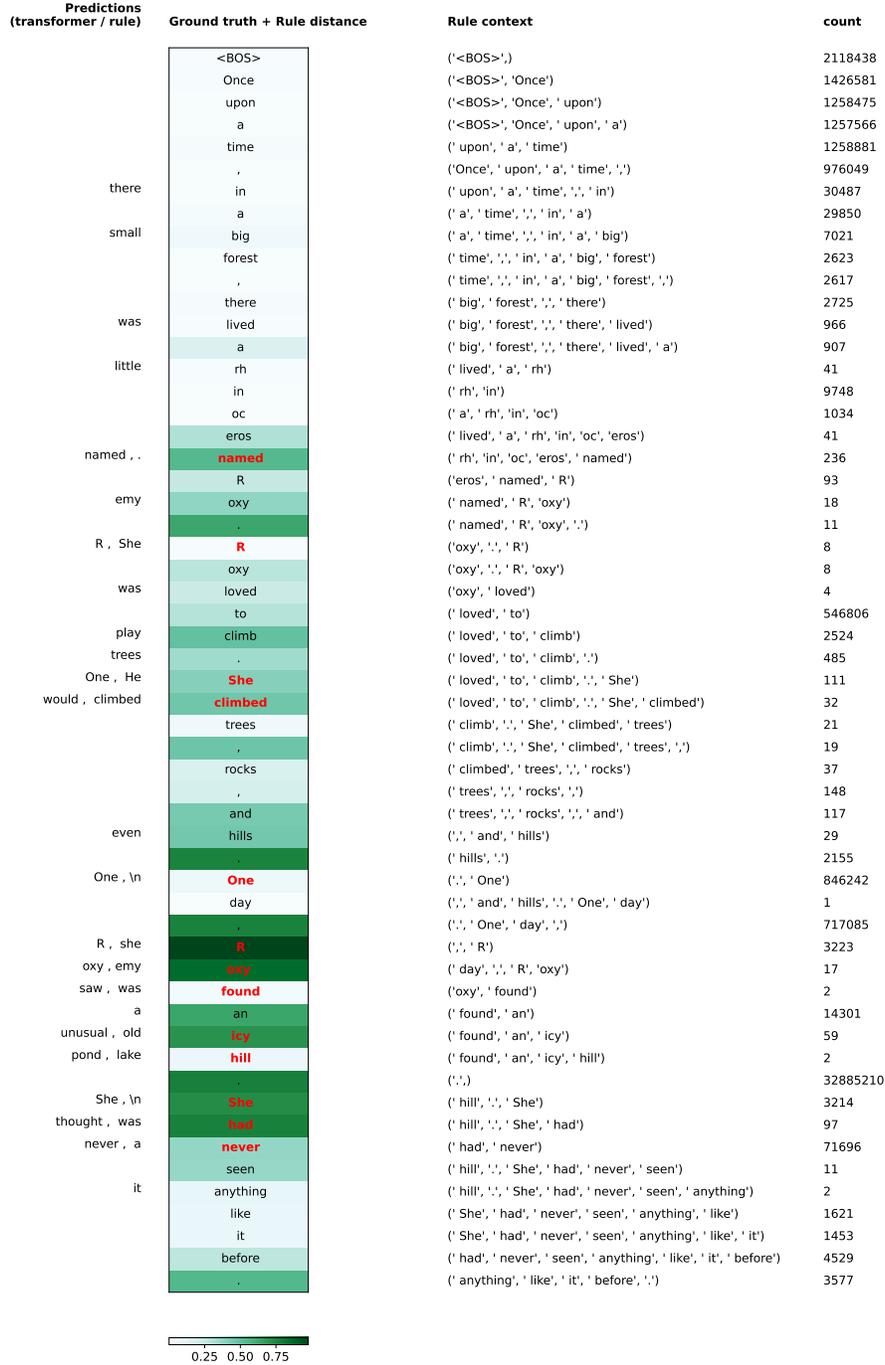


Figure 11: Rule selection for a TinyStories heldout sequence using $\mathcal{R}_7^{\text{suffix}}$. Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_7^{\text{suffix}}$ instead of $\mathcal{R}_7^{\text{all}}$. Model size: 160M.

We also ground our rule approximation on Wikipedia by providing two concrete examples in Figures 12 and 13.

Predictions (transformer / rule)	Ground truth + Rule distance	Rule context	count
	video	(' video',)	525296
game	was	(' video', ' was')	35709
released	directed	(' video', ' directed')	1837
	by	(' was', ' directed', ' by')	37813
the	Michael	(' was', '*', '*', ' Michael')	5779
H ,	Sal	(' Michael', ' Sal')	240
	omon	(' was', '*', ' by', '*', ' Sal', ' omon')	66
, ,	and	(' was', ' directed', '*', '*', '*', ' and')	4339
produced ,	prem	(' omon', ' prem')	2
	iered	(' Michael', '*', '*', ' prem', '*')	18
	on	(' and', ' prem', '*', '*')	7480
	C	(' and', '*', ' on', ' C')	334
BC	MT	(' on', ' C', ' MT')	720
	on	(' iered', '*', ' C', '*', ' on')	150
October ,	February	(' prem', '*', '*', '*', ' on', ' February')	226
		(' iered', ' on', ' February', '*')	1429
	1	(' on', ' February', ' ', '1')	51089
	5	(' MT', '*', ' February', '*', '5')	1
	,	(' February', '1', ',')	3
		(' on', ' February', '*', '1', ' ')	8
	2	(' February', '5', '*', '*', '2')	2
	0	('1', '5', '*', '*', '2', '0')	193674
1	0	('1', '5', '*', '*', '2', '*', '0')	55855
9	6	(' ', ' ', '0')	143834
	.	(' ', '2', '*', '0', '*', ' ', ' ')	4435
\n	G	(' ', '2', '0', '*', '*', ' ', ' ', 'G')	2584
aga , r	AC	('0', '*', ' ', '*', '*', 'AC')	86
V .	cut	('AC', ' cut')	6
	the	(' G', 'AC', '*', ' the')	61
video ,	ending	('6', '*', '*', 'AC', '*', ' the', '*')	19
	of	(' ', 'AC', '*', '*', '*', ' of')	2
	the	(' G', 'AC', '*', ' the')	61
video ,	video	(' the', ' ending', '*', ' the', '*')	2003
,	out	(' cut', ' the', '*', '*', ' out')	25
of	because	(' video', '*', ' because')	93
it	of	(' out', ' because', ' of')	689
the	its	(' of', '*', ' video', '*', ' its')	19
sexual	suggestive	(' because', ' of', '*', ' suggestive')	18
nature ,	language	(' because', '*', ' its', ' language')	9
.	Keith	(' of', ' Keith')	1262
'	tells	(' because', '*')	842884
his ,	the	(' of', ' suggestive', '*', '*', '*', '*')	61
audience , ,	audience	(' Keith', ' tells', '*', '*')	24
to ,	,	(' tells', ' the', ' audience', ' ', ' ')	16
"	referring	(' the', '*', ' ', ' referring')	661
	to	(' tells', '*', '*', ' ', ' ', ' to')	29
the	him	(' referring', '*', ' him')	924
as	shooting	(' referring', ' him', '*')	23
a	the	(' shooting', ' the')	2570
video ,	video	(' to', ' him')	114486
.	as	(' to', ' him', ' as')	6703



Figure 13: **Rule selection for a Wikipedia heldout sequence.** Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_7^{\text{all}}$ and with variational distance replaced with the L^∞ metric for measuring distances between probability distributions. Model size: 1.4B.

E Broader Impacts

Large language-models are having significant impacts on society, due to their use as question-answer tools and natural language generators. A better understanding of such language models will only serve to improve their capabilities. Our work here presents steps towards a fundamental understanding of language models, albeit in a small-scale regime far removed from those relevant for production systems. Given how far removed our work is from realistic datasets and use cases, we do not anticipate any direct negative broader impacts of our work.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Claims explicitly mentioned in abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss limitations in the introduction (description vs explanation) and conclusion (N -gram rules are certainly not the whole story).

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA] .

Justification: We have no theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, the experimental setup (architecture, training setup, dataset preparation) are described in detail in the main body and appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not release the code used to run our experiments (though there is nothing proprietary or novel about the model or the training procedure, and both are described in detail). The datasets we use are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, experimental settings are explained by dedicated sections in the main paper and appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes] (Partially)

Justification: Our Section 5 provides error bars and R^2 values. In other sections, only individual model runs are given.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Explained in Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our experiments use small, public datasets and involve no sensitive material or topics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We include a brief discussion in the Appendix of how our work contains no direct societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, datasets used are referenced.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing or human subjects involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No crowdsourcing or human subjects involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.