A Graph-Theoretical Framework for Analyzing the Behavior of Causal Language Models

Rashin Rahnamoun and Mehrnoush Shamsfard
Shahid Beheshti University, Tehran, Iran
rahnamounrashin@gmail.com and m-shams@sbu.ac.ir

Abstract

Recent progress in natural language processing has popularized causal language models, but their internal behavior remains poorly understood due to the high cost and reliance on large-scale benchmarks in existing analysis methods. To address these challenges, we introduce a graph-theoretical framework for analyzing causal language models. Our method constructs graphs from model outputs by linking high-probability token transitions and applies classical metrics to capture linguistic features of model behavior. Based on previous works, none have examined or applied graph analysis from this perspective. For the first time, a macroscopic view of the overall behavior of a language model is provided by analyzing the mathematical characteristics of small sample graphs derived from the generated outputs. We first discuss the metrics theoretically, then demonstrate how they work through experiments, followed by some applications of this graph-theoretical framework in natural language processing tasks. Through experiments across training steps and model sizes, we demonstrate that these metrics can reflect model evolution and predict performance with minimal data. We further validate our findings by comparing them with benchmark accuracy scores, highlighting the reliability of our metrics. In contrast to existing evaluation methods, our approach is lightweight, efficient, and especially well-suited for low-resource settings. Our implementation codes are available at this GitHub repository. [1]

1 Introduction

Recent progress in natural language processing (NLP) has expanded the scope of applications for language models. To further improve their performance and uncover additional use cases, it is important to understand how these models function and what enables their capabilities. Among these, causal language models are particularly prevalent, yet there is still much to learn about their internal mechanisms. Causal language models are a type of language model that predict the next token in a sequence based on the previous tokens, without considering future tokens. A deeper understanding of these models could reveal novel approaches and enhance their effectiveness.

Current approaches to analyzing language models often rely on large-scale experiments or trial-and-error techniques, which are both time-consuming and costly. To overcome these limitations, we propose a mathematical framework grounded in graph theory that enables low-cost, interpretable analysis of causal language models. To the best of our knowledge, this is the first framework that systematically analyzes causal language models through a graph-theoretical lens. This approach provides an alternative to expensive evaluation benchmarks and reduces dependency on extensive hardware resources.

The use of graph-based techniques in language modeling is well established. Graphs has long been intertwined with NLP, from syntactic parsing to modern tasks such as prompting (Jin et al., 2024; Jiang et al., 2023), reasoning (Chen et al., 2024; Tang et al., 2025), language modeling, and retrieval-augmented generation (Wang et al., 2024b; Ye et al., 2024; Wang et al., 2024a; Sun et al.; LUO et al.; Wang et al., 2024c). These applications demonstrate that graphs are powerful tools for uncovering underlying patterns and structures in language data, motivating the development of a graph-theoretical framework for model analysis.

Our method constructs graphs from the outputs of causal language models, linking selected and non-selected tokens based on their transition probabilities. We analyze these graphs using classical metrics such as density (Coleman and Moré, 1983), spectral entropy (De Domenico and Bia-

[1] https://github.com/rarahnamoun/A-Graph-Theoretical-Framework

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 20042–20073
November 4-9, 2025 ©2025 Association for Computational Linguistics

monte, 2016), and graph energy (Li et al., 2012). These measurements capture distinct aspects of model behavior and can be aggregated across prompts to reveal stable, interpretable patterns, even with minimal input data.

We evaluate our approach through two experiments. The first examines models at various training steps and parameter sizes, showing how different metrics reflect model evolution. The second focuses on spectral entropy and uses a small sample of prompts to predict model performance, achieving results consistent with human-annotated benchmarks. Unlike recent evaluation methods that require high-quality datasets or costly infrastructure (Maia Polo et al., 2024; Feng et al., 2024; Saranathan et al., 2024; Wu et al., 2025; Liusie et al., 2024), these recent works address only one of the limitations in low-resource setups, not all of them. Moreover, most of them rely on sampling strategies to select prompts from large-scale benchmarks, which are often unavailable in many low-resource datasets. Our graph-theoretical framework is lightweight and applicable in low-resource scenarios. This graph-theoretical framework opens new directions for analyzing causal language models through interpretable, low-cost metrics.

Using only a few samples, this method generates an expanded graph with thousands of nodes and edges, and analyzing the structure of this graph can help predict the models behavior on related samples.

It is not only a method for evaluating low-resource benchmark models with minimal data but also a framework for comparing and analyzing models with different features from a graph-theoretical perspective. Moreover, these graph-theoretical metrics can be particularly useful in the early stages of training or for models with a smaller parameter count, where classical metrics such as accuracy or datasets with well-formed textual structures fail to provide reliable evaluation because weaker models often produce incomplete and irregular outputs that differ significantly from well-formed texts.

2 Related Work

2.1 LLMs and Graphs

Using graphs with LLMs has many applications and approaches. Historically, knowledge graphs have been used in various NLP tasks. Recent notable works (Madaan et al., 2022; Sun et al.; LUO et al.; Wang et al., 2024c; Zhang et al., 2024) and many more have focused on this integration more deeply. Some recent publications (Hu et al., 2024; Chen et al.; Yu et al., 2025) have focused on node classification using graphs. Another recent study focused on graph reasoning (Chen et al., 2024; Tang et al., 2025) and proposed models based on graph structures (Wang et al., 2024b; Ye et al., 2024; Wang et al., 2024a), There are also works on graph-based retrieval-augmented generation (He et al., 2024; Gutiérrez et al.) and prompts related to graphs (Jin et al., 2024; Jiang et al., 2023)

2.2 NLP from a Graph Theory Perspective

A recent study explored the intersection of graphs and natural language as complex systems, (Stanisz et al., 2024) analyzing various aspects of network topology and linguistic structure, such as word co-occurrence in texts, using both mathematical and statistical methods. Another work by Wachs-Lopes and Rodrigues (2016) presents a graph-based model for human natural language and analyzes it using various graph-theoretical measurements. In this recent work, a word embedding model was constructed from low-resource texts, and a complex network graph was built using cosine similarity as the edges; however, no further prediction or evaluation was conducted (Rahnamoun and Rahnamoun, 2025).

In fact, there remains a significant gap between recent advances in NLP and findings from the graph-theoretical perspective. Many publications in this area are not recent and do not demonstrate new effects or results.

2.3 Efficient Evaluation for LLMs

Recently, many studies have been conducted to efficiently evaluate LLMs. We investigate some notable ones here. First, a work by Maia Polo et al. (2024) introduced a method for sampling from large datasets while avoiding prompts that yield typical or uninformative results. Another important recent work by Feng et al. (2024) also uses informative samples to extract more meaningful insights about LLMs from large datasets. Other recent studies, such as the one by Saranathan et al. (2024), follow a similar approach.

A novel method introduced by Wu et al. (2025) uses self-explanation for natural language generation (NLG) tasks, achieving performance up to 20 times faster at runtime compared to others. This is accomplished through prompt engineering but is effective only for large-scale language models capable of generating appropriate responses for prompt-based evaluations.

Another work by Liusie et al. (2024) presents a more innovative approach aimed at reducing the cost of using LLMs as evaluation judges by decreasing the number of comparisons in NLG tasks.

However, none of these works address low-resource languages or settings. High-quality, large-scale benchmarks are often unavailable in such contexts. Moreover, models with many parameters remain expensive to evaluate, even when the number of comparisons is reduced. They require powerful hardware and often ignore LLMs that do not follow instruction templates properly. As a result, only a few competitive models are included, which limits the fairness and generalizability of evaluations in low-resource settings.

Algorithm 1 Causal Language Model Analysis Flow

```
Require: Prompts $\mathcal{P} = \{p^{(1)}, \dots, p^{(n)}\}$; model $M$; tokenizer $\tau : \Sigma^* \to$ Vocab; token limit $m \in \mathbb{N}^+$; top-$k \in \mathbb{N}^+$; threshold $\theta \in (0, 1)$
Initialize score list $\mathcal{F} \leftarrow [\,]$
for each $p^{(i)} \in \mathcal{P}$ do
    $\mathbf{x}^{(i)} \leftarrow \tau(p^{(i)})$
    tokens $(t_{|\mathbf{x}^{(i)}|+1}, \dots, t_{|\mathbf{x}^{(i)}|+m})$ from $M$
    $G^{(i)} = (V, E, w) \leftarrow (\emptyset, \emptyset, \{\})$
    for $j = 0$ to $m - 1$ do
        $P_j \leftarrow \mathrm{softmax}(\mathrm{logits}_j)$
        $\{(p_1, v_1), \dots, (p_k, v_k)\} \leftarrow$ Top-$k(P_j)$
        $u \leftarrow \tau^{-1}(t_{|\mathbf{x}^{(i)}|+j})$
        Add $u$ to $V$
        for $\ell = 1$ to $k$ do
            if $p_\ell > \theta$ then
                $v \leftarrow \tau^{-1}(v_\ell)$
                Add $v$ to $V$
                Add edge $(u, v)$ to $E$
                $w(u, v) \leftarrow p_\ell$
            end if
        end for
    end for
    Compute graph score: $f_i \leftarrow f(G^{(i)})$
    Append $f_i$ to $\mathcal{F}$
end for
return $S \leftarrow \frac{1}{n} \sum_{i=1}^{n} f_i$
```

3 Methodology

Our methodology follows three simple steps: first, we construct graphs from prompts; then, we analyze each graph separately using graph-theoretical metrics; and finally, we aggregate the results to draw conclusions about the language model's behavior. The following sections will explain these steps in detail.

3.1 Word Transition Sampling Graphs

To construct the sampling graph from a causal language model, we include both the actual generated words (the output text) and the top candidate words with high probabilities that were not selected during output generation. These words are treated as nodes in the graph. The graph also contains transitions between words, represented as edges, which reflect the probabilistic relationships between the nodes.

Definition 1. *Given a prompt p, its tokenized form* $\mathbf{t} = (t_1, \dots, t_n)$*, and a sampling run generating output tokens* $(t_{n+1}, \dots, t_{n+m})$*, the* probabilistic word transition graph *is a, weighted graph:*

$$G = (V,\, E,\, w) \tag{1}$$

where:

- $V = \{\mathit{decode}(t) \mid t \in \{t_{n+1}, \dots, t_{n+m}\}\}$, *where* $\mathit{decode}\colon \mathcal{V} \to \Sigma^*$ *maps tokens to their string representations in the output alphabet* Σ.
- $E \subseteq V \times V$ *is the set of edges, where an edge* $(u, v) \in E$ *exists if* $v = \mathit{decode}(t')$ *for some token* t' *such that the transition probability* $w(u, v)$ *exceeds a specified threshold* $\theta \in (0, 1)$.
- $w\colon E \to [0, 1]$ *is a weight function assigning transition probabilities, i.e.,* $w(u, v) = \mathbb{P}(t_{i+1} = t' \mid t_1, \dots, t_i)$*, where* $t' = \mathit{encode}(v)$ *and* $\mathit{encode}\colon \Sigma^* \to \mathcal{V}$ *is the inverse of decode.*

The graph connects previous words to subsequent words, with each edge representing a transition from one word to the next, reflecting the probabilistic relationships between them. Transition probabilities are filtered to include only those that exceed the threshold, ensuring that only significant transitions are represented in the final graph.

3.2 Analyzing the Structure of the Graphs

To interpret the behavior of causal language models from a structural perspective, we construct and analyze word transition sampling graphs 3.1. By applying principles from spectral graph theory and information theory, we examine how the structure reflects predictability, diversity, and influence in the main model's output.

In the following, we present key theoretical formulations that bridge graph characteristics to linguistic patterns.

3.2.1 Graph Density

Word density is the first metric used in our theoretical framework, defined as follows:

Definition 2 (Graph Density (Coleman and Moré, 1983))**.** *The* graph density $\mathcal{D}(G)$ *is defined as the ratio of the number of edges to the maximum possible number of edges in a simple graph with* $|V|$ *vertices:*

$$\mathcal{D}(G) = \frac{2|E|}{|V|(|V|-1)}. \tag{2}$$

$\mathcal{D}(G)$ *lies in the interval* $[0, 1]$*, where* $\mathcal{D}(G) = 1$ *corresponds to a complete graph.*

By using graph density and applying it to the word transition sampling graph mathematical model, which is described in Section 3.1, we proved Theorem 1 (see the proof in Appendix A.1). For validation, we used a well-known mathematical model, $|E^{(t)}| = \alpha|V^{(t)}|^{\beta}$ (Barabási, 2002), to analyze graph growth over time. See Appendix A.1 for details on how the parameters were set for the linguistic task, the specifics of this mathematical modeling, and why it is appropriate for analyzing word transition sampling graphs. This theorem shows that the number of word diversities increases regardless of whether word transitions increase or decrease, resulting in a decrease in density. Therefore, for greater textual diversity, we prefer a decrease in density.

Theorem 1. *In a word transition graph with vertices as unique words and edges as transitions, if* $|V|$ *increases, the density* $D = \frac{2|E|}{|V|(|V|-1)}$ *decreases for large* $|V|$*, under:*

- $|E| = \alpha|V|^{\beta}$, $\beta < 2$
- $|E| = \alpha|V|^{\gamma}$, $\gamma < 1$ *as* $|E|$ *decreases*

3.2.2 Spectral Entropy

There are many definitions and different aspects of calculating the spectral entropy of a graph (e.g., Von Neumann entropy) (De Domenico and Biamonte, 2016; Liu et al., 2022), but in our case, the spectral entropy is defined as follows:

Definition 3 (Spectral Entropy)**.** *Let* G *be a graph with adjacency matrix* A *and degree matrix* D*. The combinatorial Laplacian is* $L = D - A$*. Let* $\lambda_1, \ldots, \lambda_n$ *be the eigenvalues of* L*, excluding those smaller than a threshold* ε*, and define the normalized eigenvalue probabilities* $\frac{\lambda_i}{\sum_{\lambda_j \in \Lambda} \lambda_j}$*. The spectral entropy* $H(G)$ *is then*

$$H(G) = -\sum_{\lambda_i \in \Lambda} \frac{\lambda_i}{\sum_{\lambda_j \in \Lambda} \lambda_j} \ln\left(\frac{\lambda_i}{\sum_{\lambda_j \in \Lambda} \lambda_j}\right). \tag{3}$$

Adding edges to a connected graph can cause its eigenvalues to increase, decrease, or remain unchanged, depending on the structure and placement of the added edges (Guo et al., 2018); therefore, analyzing the effect of edge addition or removal on spectral-based metrics is generally intractable for arbitrary graphs and must be approximated or studied under specific conditions.Analyzing spectral entropy in general graphs can be challenging; however, previous findings indicates that higher entropy corresponds to more complex graph structures and a more uniform distribution of eigenvalues (Chung, 1997). For our purposes, this suggests a greater diversity and complexity in textual relationships, which is a crucial factor in understanding the overall behavior and performance of causal language models. As previously noted, analyzing eigenvalue dependencies in general graph structures presents significant theoretical challenges. However, for a key structural property of textual graphs, Theorem 2 (see proof in Appendix A.2) establishes an important relationship: as the graph scales, the ratio of possible transitions to vocabulary elements provably decreases.

Theorem 2. *Under the word transition sampling graph growth model, let* $|E| = \alpha|V|^{\beta}$ *with* $\beta < 2$*. Then, the maximum degree* Δ *satisfies:*

$$\lim_{|V| \to \infty} \frac{\Delta}{|V|} = 0. \tag{4}$$

3.2.3 Graph Energy

The final component of our framework introduces the concept of graph energy, which quantifies network connectivity.

Definition 4 (Graph Energy (Balakrishnan, 2004; Li et al., 2012)). *Let G be a graph with adjacency matrix A. Let $\lambda_1, \dots, \lambda_{|V|}$ be the eigenvalues of A, where each λ_ℓ satisfies the eigenvalue equation*

$$A\vec{v}_\ell = \lambda_\ell \vec{v}_\ell.$$

Then, the graph energy $\mathcal{E}(G)$ *is defined as*

$$\mathcal{E}(G) = \sum_{\ell=1}^{|V|} |\lambda_\ell|, \quad \text{where } A\vec{v}_\ell = \lambda_\ell \vec{v}_\ell. \tag{5}$$

Lower energy values correspond to more fragmented graph structures, while higher energy indicates greater connectivity. From a linguistic perspective, we naturally prefer more strongly connected word graphs as they yield richer semantic information. Theorem 3 establishes an upper bound for graph energy in our word transition sampling model, with this bound increasing as the number of edges and nodes grows (see Appendix A.3 for proof).

Theorem 3. *Let $G^{(t)} = (V^{(t)}, E^{(t)})$ be a graph evolving over time t, where $|V^{(t)}|$ is the number of vertices and $|E^{(t)}| = \alpha|V^{(t)}|^\beta$ with constants $\alpha > 0$ and $1 < \beta < 2$. For word transition sampling graphs constructed from natural language sequences, the graph energy $\mathcal{E}(G^{(t)})$ is upper-bounded by*

$$\mathcal{E}(G^{(t)}) \leq \sqrt{2\alpha}\,|V^{(t)}|^{\frac{\beta+1}{2}}. \tag{6}$$

As $|V^{(t)}|$ increases over time, this bound increases, implying that $\mathcal{E}(G^{(t)})$ grows with t.

3.3 Prompt-to-Graph Pipeline for Model Behavior Prediction

The procedure for analyzing the behavior of a causal language model M involves three main steps. Given a set of prompts $\mathcal{P} = \{p^{(1)}, \dots, p^{(n)}\}$, we first generate model outputs by sampling from M, obtaining for each prompt $p^{(i)}$ a sequence of output tokens $\mathbf{y}^{(i)} = (t_{|\mathbf{x}^{(i)}|+1}, \dots, t_{|\mathbf{x}^{(i)}|+m})$, where $\mathbf{x}^{(i)} = \tau(p^{(i)})$ is the tokenized input.

In the second step, for each output $\mathbf{y}^{(i)}$, we construct a corresponding probabilistic word transition graph $G^{(i)} = (V^{(i)}, E^{(i)}, w^{(i)})$ as defined in Definition 3.1. This graph encodes both the actual sampled words and their top-k alternatives with significant probabilities above a threshold θ, capturing local word-level dynamics.

Finally, in the third step, we analyze each graph $G^{(i)}$ using structural metrics from spectral graph theory and information theory, such as spectral entropy, graph energy, and density. Let $f : \mathcal{G} \to \mathbb{R}$ be a measurement function applied to a graph $G \in \mathcal{G}$. We aggregate these scores over the dataset:

$$S = \frac{1}{n} \sum_{i=1}^{n} f(G^{(i)}), \tag{7}$$

producing an overall quantitative profile of the models structural behavior.

This process allows us to characterize how a model like M responds to different prompt types in terms of its generative structure. Even from a limited number of samples, this approach offers a macroscopic view into the latent organizational patterns and transition dynamics learned by the causal language model (see Algorithm 1).

4 Experiments

To bridge the theoretical framework introduced in Section 3 with real-world applications in natural language processing, we conducted experiments using two approaches. First, we showed how the theoretical concepts reveal the diverse behaviors of the model across various training steps and parameter settings. These observations were then linked to theoretical aspects using established natural language processing benchmarks to assess the applicability of our framework in practice. Second, based on the evaluated results from paper (Biderman et al., 2023) across different models and benchmark datasets, we compared our method, which requires only a small number of sampled prompts to assess model performance, with these existing results. This comparison highlights the practicality and efficiency of our framework, particularly in low-resource settings, where extensive data or computational resources are limited. By requiring minimal input. All experiments were conducted using an NVIDIA Tesla K80 GPU with 12GB of VRAM and 12 GPU hours.

4.1 Datasets

Following the well-known benchmarks employed in paper (Biderman et al., 2023) for various tasks and evaluated on different large language models, we utilized the datasets below to cover different tasks and aspects of our models. The reason for selecting some of the same models and benchmarks used in paper (Biderman et al., 2023) is that they

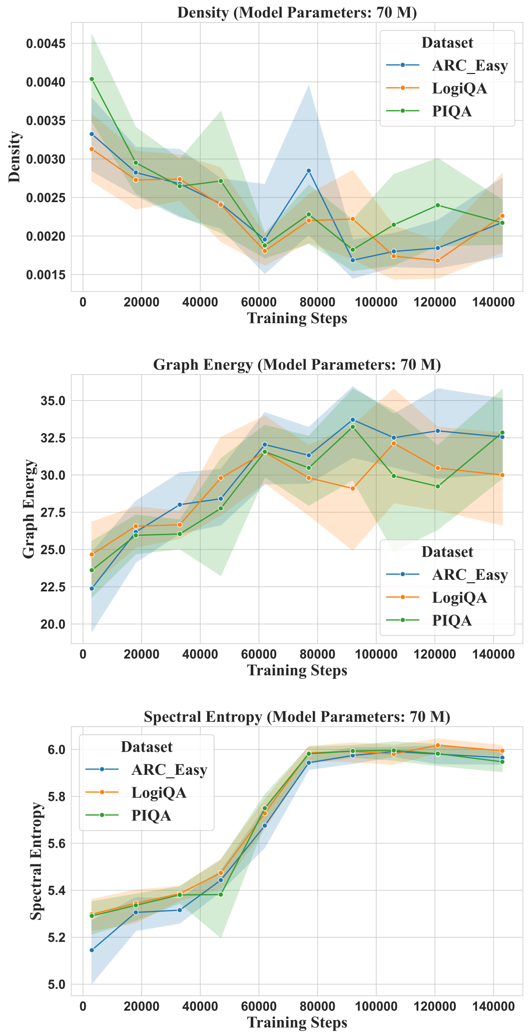


Figure 1: Pythia model with 70M parameters on ARC-Easy, LogiQA, and PIQA datasets across different training steps. The visualizations represent trends in spectral entropy, graph energy, and density.

evaluated their model, Pythia (which we used to experiment with our theoretical concepts), at different training steps and compared its performance with various other models of similar parameter sizes.

ARC-Easy. We used the ARC-Easy subset of the AI2 Reasoning Challenge (ARC) (Clark et al., 2018). It contains 5.2K rows of grade-school science question data that can be answered by simpler models, making it more suitable for models with fewer than 1 billion parameters that cannot perform reliably on more challenging reasoning tasks.

PIQA. The PIQA dataset (Bisk et al., 2020) contains 21K examples focused on physical commonsense reasoning, a task that remains challenging for current natural language processing models despite humans achieving approximately 95% accuracy.

LogiQA. We have utilized LogiQA (Liu et al., 2021), a comprehensive dataset comprising 8,678 question-answer pairs designed to evaluate logical reasoning abilities in natural language understanding.

4.2 Models

For our experiments, we used three models: Pythia (Biderman et al., 2023), BLOOM (Le Scao et al., 2023), and OPT (Zhang et al., 2022), selecting versions with fewer than 2 billion parameters due to our limited access to computational resources.

Pythia. Designed to enable detailed research on training dynamics and model behavior, Pythia (Biderman et al., 2023) is a suite of 16 large language models ranging from 70M to 12B parameters, all trained on public data in identical order with 154 checkpoints each.

BLOOM. BLOOM (Le Scao et al., 2023) is an open-access decoder-only transformer model trained on multilingual data, developed collaboratively to promote accessible and transparent large language model research. It is available in a range of sizes, from 560 million to 176 billion parameters.

OPT. The model OPT (Zhang et al., 2022) is a publicly available suite of decoder-only transformer models, ranging from 125M to 175B parameters, designed to support open research with full access to model weights.

4.3 Results

To empirically validate the theoretical relationships discussed in Section 3.2, we conducted a series of experiments to examine their correspondence with real-world model behavior. For the initial set of experiments, we employed the Pythia model at three different parameter scales: 70M, 160M, and 410M. Pythia provides 154 checkpoints across the training process, allowing for a detailed analysis of how large language models evolve over time. We selected 10 equally spaced checkpoints from this set to investigate how our proposed metrics behave at different stages of training.

From each of the three datasets, we randomly selected 15 prompts. For the 70M model, we extended each prompt using a top-k sampling strategy with $k = 20$, constructing corresponding

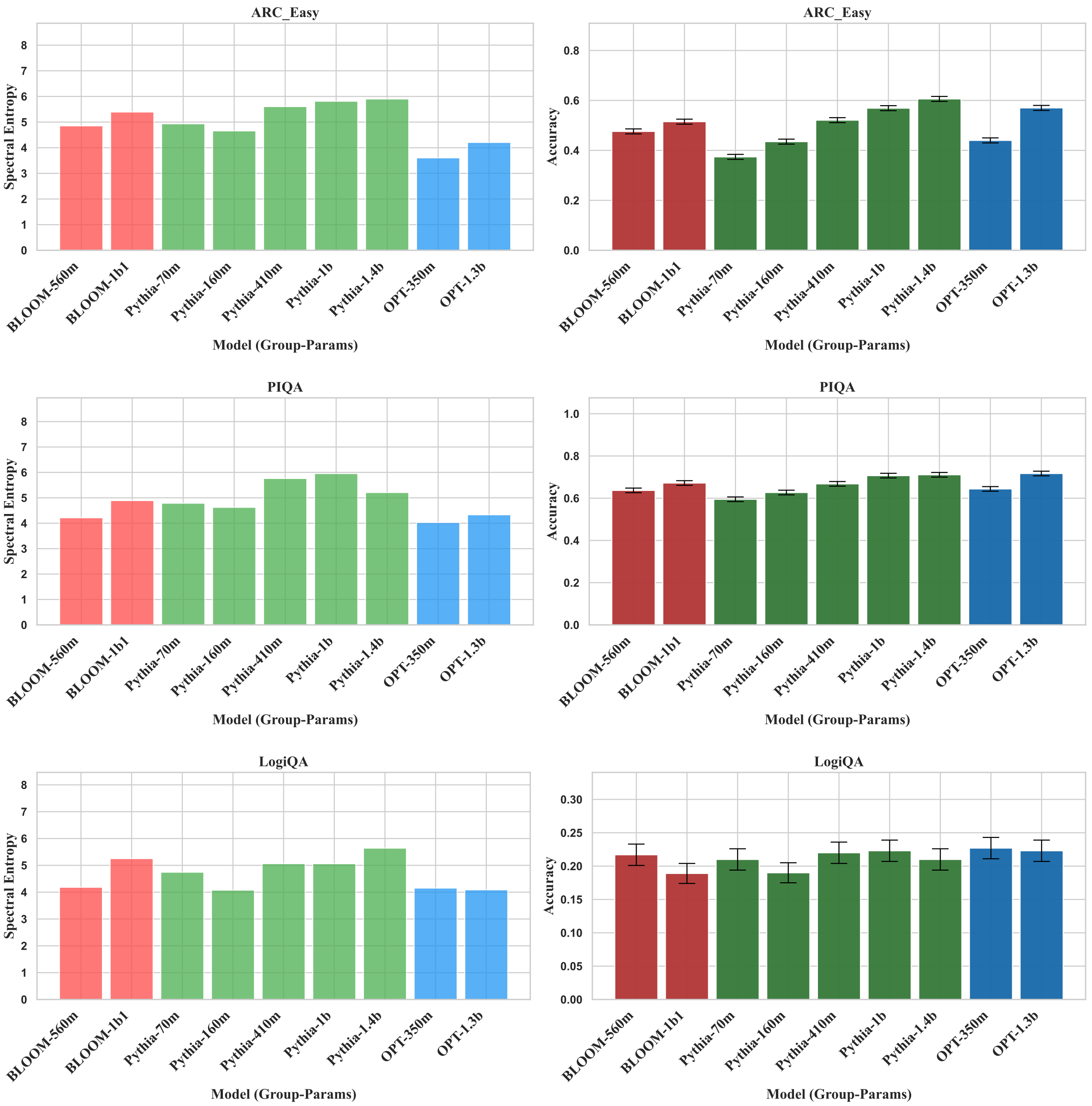


Figure 2: The left bar chart shows spectral entropy values for models Pythia, BLOOM, and OPT across different parameter counts on the ARC-Easy, PIQA, and LogiQA datasets. The right bar chart shows benchmark accuracies for each dataset, with standard error bars included

graphs to analyze model behavior. The choice of $k = 20$ was determined empirically and reflects a trade-off between computational feasibility and hardware constraints. For larger models, higher values of k may be required to obtain more representative or coherent outputs. The reason for using only a few examples is that, by a factor of k, this method constructs a graph with thousands of nodes and edges, which differs from using basic prompts as samples for evaluating models. It effectively expands the samples by constructing large graphs, and analyzing the features of these graphs enriches the prompts with additional textual information, even when only a few examples are available.

As outlined in Sections 3.2.2, 3.2.3, and 3.2.1, our theoretical framework suggests that linguistic quality can be associated with higher values of spectral entropy and graph energy, and lower values of graph density. As shown in Figure 1. Additional experimental results for models with 160M and 410M parameters are provided in Appendix C.

As explained in Section 3.2.2, spectral entropy captures more complex and linguistically significant patterns compared to the other metrics, making it the best candidate for evaluating model performance. For our experiments, we used only 15 random prompts from each dataset and set the graph construction parameter to top-k = 90. In general, increasing the value of k improves the quality of the graph construction and the resulting analysis. This choice was made experimentally to balance the large model sizes, which are in the millions of parameters, with computational feasibility. Additionally, we compared our results against benchmark accuracies and their standard errors reported for different models in the official Pythia paper (Biderman et al., 2023). The results are presented in Figure 2.

4.4 Discussion and Analysis

As shown in Figure 1, variations in graph structure cause fluctuations that can obscure clear trends, since different topologies affect spectral entropy, graph energy, and density differently. However, as explained in Section 3.2, the ideal scenario involves increasing graph energy and spectral entropyindicating richer linguistic representationsalongside decreasing density. Our experiments with the 70M parameter model confirm this: spectral entropy and graph energy increase during training, while density decreases. This aligns with findings in the Pythia study (Biderman et al., 2023), where extended training improved performance across benchmarks. These results support the hypothesis that the proposed metrics capture meaningful linguistic characteristics of causal language models.

Furthermore, as discussed in Appendix B, most of these metrics do not exhibit strong correlations across different datasets and settings, indicating that each captures a unique aspect of model behavior. Therefore, all three should be considered jointly to evaluate and guide the development of high-performing language models.

Another important factor in analyzing larger models, such as the 410M variant, is the choice of the top-k parameter. In our experiments, we selected top-$k = 90$ to balance accuracy and computational cost for 410M parameters model. For models with a greater number of parameters, using a larger k is generally necessary to fully capture the graph structure and ensure meaningful analysis. Notably, we do not use a simple prompt for analysis; instead, we expand a single prompt into a large graph with many nodes and edges. With a larger k, even a small change can produce a richer and larger graph containing more features and information, although this comes with an increased computational cost for analysis. However, increasing k also significantly raises computational demands. This constraint makes the analysis of larger models more time-consuming and may limit the clarity of increasing or decreasing trends in the metrics compared to models with fewer parameters.

Additional results, presented in Figure 2, compare spectral entropy values with the official benchmark accuracies of Pythia, (Biderman et al., 2023) reported across different model sizes. These benchmark results are based on comparisons made by related entities between the models generated outputs and the corresponding reference answers. The results, as shown in Figure 2, indicate that in most cases and across different models, spectral entropy measurements derived from only a small number of prompt samples can closely approximate those obtained from full benchmark evaluations. While this method offers an approximation rather than an exact match, it remains informative. For example, in the PIQA dataset (Figure 2), the spectral entropy value for Pythia-1.4b shows some discrepancy when compared to its corresponding accuracy on the same dataset.

As explained in Section 3.2, spectral entropy was chosen among several metrics for its ability to capture meaningful linguistic patterns with low computational cost. Using top-k sampling expands the models response space beyond a single deterministic output, enabling analysis of both generated and unselected tokens via word transition sampling graphs. This approach extends a single prompt to offer deeper insights into the models behavior. By selecting an appropriate top-k and using only a few samples and datasets, model performance can be approximated without large benchmarks or costly human evaluation. Unlike the problems faced by recent works in efficient LLMs evaluation discussed in Section 2.3, which mainly do not address challenges related to low-resource settings and limited access. While human or large-scale evaluations better reflect real-world performance, this method provides a fast, lightweight alternative for evaluating language models, especially in low-resource settings with limited data, computation, or time, offering meaningful behavioral insights from minimal input.

This method can be particularly useful in the early stages of training or for models with a small number of parameters, because it does not rely on the textual structure of model outputs. Instead, it analyzes the structure of the graphs constructed from model output prompts. For irregular or weak model outputs, this approach provides a more suitable evaluation than classical metrics such as accuracy, which require high-quality model outputs. This method is also helpful when human annotation is expensive or inaccessible during the evaluation stage.

5 Conclusions

Today, many approaches in Natural Language Processing (NLP) heavily rely on trial-and-error methods and extensive benchmarking to gain deeper insights into the behavior of large language models and to enhance their performance. However, foundational mathematical methods and behavior modeling have often been overlooked. This oversight can be particularly limiting in contexts where computational, informational, financial, or human resources are constrained.

In this paper, we introduce, for the first time, a novel approach: by sampling the behavior of language models and constructing graphs based on these samples, we analyze the overall behavior of the model using techniques from graph theory. We present several theoretical and practical aspects of this method and support them with experimental evidence. This includes experiments such as analyzing the effect of training steps on graph-theoretical metrics, and evaluating the accuracy of different models using only a few prompt samples.

This work opens a new direction, suggesting that prompt-based graph structures derived from model outputs can be analyzed to reveal new features and behavioral patterns from different perspectives. Although we focus only on three such properties in this study, the method can be extended to many other applications. Each use of this approach represents a significant step toward understanding causal language behavior and evaluating language models in resource-constrained environments.

Limitations

Due to limited access to computational resources, we were unable to perform inference or run experiments on large-scale language models with high parameter counts (e.g., models with tens of billions of parameters). As a result, it remains an open question whether the proposed graph-theoretical metrics retain their utility and interpretability when applied to such large models, and whether their behavior diverges significantly from that observed in smaller-scale models.

Another critical limitation of our method lies in its reliance on the generation of sufficiently long outputs from language models. The construction of meaningful graphs depends on having rich textual sequences; therefore, in scenarios where the language model produces only short or option-based responses, the approach becomes ineffective. To address this, prompts must be designed to elicit explanatory or elaborative responses from the model.

Furthermore, spectral entropy, one of the key metrics used in our framework, shows promise as a comparative tool for evaluating different models or various configurations of the same model under a fixed benchmark. However, its applicability across different benchmarks is limited. It is not appropriate for comparing the performance of different models across different benchmarks in order to determine which models perform better on which benchmarks.

References

Paolo Allegrini, Paolo Grigolini, and Luigi Palatella. 2004. Intermittency and scale-free networks: a dynamical model for human language complexity. *Chaos, Solitons & Fractals*, 20(1):95–105.

R Balakrishnan. 2004. The energy of a graph. *Linear Algebra and its Applications*, 387:287–295.

Albert-László Barabási. 2002. The new science of networks. *Cambridge MA. Perseus*.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science*, 286(5439):509–512.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. 2024. Graphwiz: An instruction-following language model for graph problems. *arXiv preprint arXiv:2402.16029*.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). In *The Twelfth International Conference on Learning Representations*.

Fan R. K. Chung. 1997. *Spectral Graph Theory*. American Mathematical Society, Providence, RI.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Thomas F. Coleman and Jorge J. Moré. 1983. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM Journal on Numerical Analysis*, 20(1):187–209.

Manlio De Domenico and Jacob Biamonte. 2016. Spectral entropies as information-theoretic tools for complex network comparison. *Physical Review X*, 6(4):041062.

Paul Erdős and Alfréd Rényi. 1959. On random graphs i. *Publ. math. debrecen*, 6(290-297):18.

Kehua Feng, Keyan Ding, Kede Ma, Zhihua Wang, Qiang Zhang, and Huajun Chen. 2024. Sample-efficient human evaluation of large language models via maximum discrepancy competition. *arXiv preprint arXiv:2404.08008*.

Ji-Ming Guo, Pan-Pan Tong, Jianxi Li, Wai Chee Shiu, and Zhi-Wen Wang. 2018. The effect on eigenvalues of connected graphs by adding edges. *Linear Algebra and its Applications*, 548:57–65.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. 2024. Lets ask gnn: Empowering large language model for graph in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1396–1409.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and 1 others. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 163–184.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and 1 others. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Taoying Li, Jie Bai, Xue Yang, Qianyu Liu, and Yan Chen. 2018. Co-occurrence network of high-frequency words in the bioinformatics literature: Structural characteristics and evolution. *Applied Sciences*, 8(10):1994.

Xueliang Li, Yongtang Shi, and Ivan Gutman. 2012. *Graph energy*. Springer Science & Business Media.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3622–3628.

Xuecheng Liu, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2022. On the similarity between von neumann graph entropy and structural information: Interpretation, computation, and applications. *IEEE Transactions on Information Theory*, 68(4):2182–2202.

Adian Liusie, Vatsal Raina, Yassir Fathullah, and Mark Gales. 2024. Efficient llm comparative assessment: A product of experts framework for pairwise comparisons. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6835–6855.

LINHAO LUO, Yuan-Fang Li, Reza Haf, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1384–1403.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson de Oliveira, Yuekai Sun, and Mikhail Yurochkin. 2024. Efficient multi-prompt evaluation of llms. *Advances in Neural Information Processing Systems*, 37:22483–22512.

Bernard J McClelland. 1971. Properties of the latent roots of a matrix: the estimation of π-electron energies. *The Journal of Chemical Physics*, 54(2):640–643.

Rashin Rahnamoun and Ramin Rahnamoun. 2025. Semantic analysis of jurisprudential zoroastrian texts in pahlavi: A word embedding approach for an extremely under-resourced, extinct language. In *Proceedings of the New Horizons in Computational Linguistics for Religious Texts*, pages 23–41, Abu Dhabi, UAE. Association for Computational Linguistics.

Oliver Riordan and Alex Selby. 2000. The maximum degree of a random graph. *Combinatorics, Probability and Computing*, 9(6):549–572.

Gayathri Saranathan, Mahammad Parwez Alam, James Lim, Suparna Bhattacharya, Soon Yee Wong, Martin Foltin, and Cong Xu. 2024. DELE: Data efficient LLM evaluation. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Tomasz Stanisz, Stanisław Drożdż, and Jarosław Kwapień. 2024. Complex systems approach to natural language. *Physics Reports*, 1053:1–84.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Jianheng Tang, Qifan Zhang, Yuhan Li, Nuo Chen, and Jia Li. 2025. Grapharena: Evaluating and exploring large language models on graph computation. In *The Thirteenth International Conference on Learning Representations*.

Guilherme Alberto Wachs-Lopes and Paulo Sergio Rodrigues. 2016. Analyzing natural human language from the point of view of dynamic of a complex network. *Expert Systems with Applications*, 45:8–22.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024a. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Advances in Neural Information Processing Systems*, 37:5950–5973.

Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian J McAuley. 2024b. Instructgraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In *ACL (Findings)*.

Jiapu Wang, Sun Kai, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2024c. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Advances in Neural Information Processing Systems*, 37:8384–8410.

Meng-Chen Wu, Md Mosharaf Hossain, Tess Wood, Shayan Ali Akbar, Si-Chi Chin, and Erwin Cornejo. 2025. Seeval: Advancing llm text evaluation efficiency and accuracy through self-explanation prompting. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7357–7368.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. 2024. Language is all a graph needs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1955–1973.

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. 2025. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13087–13095.

Qinggang Zhang, Junnan Dong, Hao Chen, Daochen Zha, Zailiang Yu, and Xiao Huang. 2024. Knowgpt: Knowledge graph based prompting for large language models. *Advances in Neural Information Processing Systems*, 37:6052–6080.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Theoretical Properties of Word Transition Sampling Graphs

A.1 Proof of the Effects of Textual Diversity

Let $G^{(t)} = (V^{(t)}, E^{(t)})$ evolve over time t, where time refers to the generative steps of a causal language model. Let $|V^{(t)}|$ be the number of nodes and $|E^{(t)}|$ the number of edges at time t, with edge growth governed by

$$|E^{(t)}| = \alpha |V^{(t)}|^{\beta}, \tag{8}$$

where $\alpha > 0$, $1 < \beta < 2$. Word transition graphs (Allegrini et al., 2004), (Li et al., 2018) suggest that they may follow a power-law degree distribution, in which common stop words (e.g., “by”, “and”, “the”) typically emerge as hubs due to their high frequency and widespread co-occurrence. This is the reason β has been chosen between 1 and 2 in our assumptions.

Although in our modeling we increase the number of edges per V as vocabulary selections and edges and nodes with top k are increasing, it means that for a fixed α, we can approximate the addition of edges as if $k+1$ candidates were being considered per node. That is, during each generative step, the number of new nodes (vocabularies) increases roughly by k, and the number of potential edges increases proportionally. This gives us the local approximation: Finally, this is also reflected in the graph density, which evolves as:

$$\Delta|E^{(t)}| \approx k \cdot \Delta|V^{(t)}|. \tag{9}$$

However, in reality we never reach the full top-k count due to either a minimum selection threshold or the probability distribution over transitions. That is, while k transitions may be available, the model often selects fewer because many candidate transitions do not exceed the threshold or have low transition probabilities. Hence, the actual increase in edges satisfies:

$$\Delta|E^{(t)}| \leq k \cdot \Delta|V^{(t)}|. \tag{10}$$

Moreover, many of the added new vocabularies are repeated across time steps. As a result, many transitions are also repeated and do not contribute to novel edge formation. This phenomenon further slows the effective graph expansion, and the number of unique nodes and edges grows more slowly than a naive k-based estimation would suggest.

Only when k is chosen to be very small (i.e., $k \ll |V^{(t)}|$) does the actual edge growth approach the ideal case of adding k transitions per vocabulary per step. On the other hand, if we consider the case where $0 < \beta < 1$, then the number of edges grows sub linearly with respect to the number of nodes. This leads to a graph with very few edges compared to the number of nodes. As the number of vertices increases, the graph becomes increasingly sparse and exhibits disconnected components and structural fragmentation. Such behavior is incompatible with the empirical structure of vocabulary transition graphs produced by causal language models, which tend to be well-connected due to the frequent co-occurrence and recurrence of vocabulary terms. Therefore, $\beta \in (0, 1)$ is not a plausible regime for modeling realistic generative language graphs.

Values $\beta > 2$ imply an unrealistically dense vocabulary transition graph, where the number of word-to-word transitions grows faster than the square of the vocabulary size, suggesting that almost every word co-occurs with nearly every other wordan implausible scenario in natural language where word usage is selective and context-dependent. Therefore, the power-law relation with $1 < \beta < 2$ remains a realistic bound in the context of vocabulary transition networks in generative models. Finally, this is also reflected in the graph density, which evolves as:

$$D(t) = \frac{2|E(t)|}{|V(t)|(|V(t)|-1)} \approx 2\alpha|V(t)|^{\beta-2}. \tag{11}$$

To derive this, substitute $|E(t)| = \alpha|V(t)|^{\beta}$ into the density formula:

$$D(t) = \frac{2|E(t)|}{|V(t)|(|V(t)|-1)} = \frac{2 \cdot \alpha|V(t)|^{\beta}}{|V(t)|(|V(t)|-1)}. \tag{12}$$

Simplify the expression by factoring out terms:

$$D(t) = \frac{2\alpha|V(t)|^{\beta}}{|V(t)| \cdot |V(t)|(1-\frac{1}{|V(t)|})} = \frac{2\alpha|V(t)|^{\beta-1}}{|V(t)|(1-\frac{1}{|V(t)|})}. \tag{13}$$

For large $|V(t)|$, the term $1 - \frac{1}{|V(t)|} \approx 1$, so:

$$D(t) \approx \frac{2\alpha|V(t)|^{\beta-1}}{|V(t)|} = 2\alpha|V(t)|^{\beta-1-1} = 2\alpha|V(t)|^{\beta-2}. \tag{14}$$

This approximation holds because as $|V(t)| \to \infty$, the denominator $|V(t)|(|V(t)|-1) \approx |V(t)|^2$.

Lemma 1. *Let $|V| \uparrow, |E| \uparrow$. Assume standard growth: $|E| = \alpha|V|^{\beta}$, $\beta < 2$. Then:*

$$\lim_{|V|\to\infty} D = 0. \tag{15}$$

Proof. Consider $D(t) \approx 2\alpha|V(t)|^{\beta-2}$. Since $\beta < 2$, the exponent $\beta - 2 < 0$. As $|V(t)| \to \infty$, the term $|V(t)|^{\beta-2}$ decreases because the negative exponent causes the value to approach zero:

$$\lim_{|V|\to\infty} |V(t)|^{\beta-2} = \lim_{|V|\to\infty} \frac{1}{|V(t)|^{2-\beta}} = 0, \tag{16}$$

since $2 - \beta > 0$. Thus, $D(t) = 2\alpha|V(t)|^{\beta-2} \to 0$. □

Lemma 2. *Let $|V| \downarrow, |E| \downarrow$. Assume $|E| = \alpha|V|^{\beta}$, $\beta > 1$. Then:*

$$\lim_{|V|\to 0} D = \infty. \tag{17}$$

Proof. The density is:

$$D(t) = \frac{2\alpha|V(t)|^{\beta}}{|V(t)|(|V(t)|-1)}. \tag{18}$$

As $|V(t)| \to 0$, the denominator $|V(t)|(|V(t)| - 1) \to 0$, while the numerator $2\alpha|V(t)|^{\beta} \to 0$.

Since $\beta > 1$, the denominator decreases faster. Rewrite:

$$D(t) = \frac{2\alpha|V(t)|^{\beta}}{|V(t)|^2(1-\frac{1}{|V(t)|})} \approx \frac{2\alpha|V(t)|^{\beta-2}}{1-\frac{1}{|V(t)|}}. \quad (19)$$

As $|V(t)| \to 0$, $|V(t)|^{\beta-2} \to \infty$ because $\beta - 2 < 0$, and the denominator $1 - \frac{1}{|V(t)|}$ becomes negative and approaches zero, but the graph structure is undefined for $|V| \leq 1$. □

Lemma 3. *Let $|V| \uparrow, |E| \downarrow$. Assume $|E| = \alpha|V|^{\gamma}$, $\gamma < 1$. Then:*

$$\lim_{|V|\to\infty} D = 0. \quad (20)$$

Proof. Substitute $|E| = \alpha|V|^{\gamma}$:

$$D(t) = \frac{2\alpha|V(t)|^{\gamma}}{|V(t)|(|V(t)|-1)} = \frac{2\alpha|V(t)|^{\gamma}}{|V(t)|^2(1-\frac{1}{|V(t)|})}. \quad (21)$$

For large $|V(t)|$:

$$D(t) \approx \frac{2\alpha|V(t)|^{\gamma}}{|V(t)|^2} = 2\alpha|V(t)|^{\gamma-2}. \quad (22)$$

Since $\gamma < 1$, $\gamma - 2 < 0$, so as $|V(t)| \to \infty$:

$$\lim_{|V|\to\infty} |V(t)|^{\gamma-2} = 0, \quad (23)$$

implying $D(t) \to 0$. The decay is faster than in the first lemma because $\gamma - 2 < \beta - 2 < 0$. □

Lemma 4. *Let $|V| \downarrow, |E| \uparrow$. Let $|V| \to |V|'$, $|E| \to |E|'$ with $|E|' > |E|$, $|V|' < |V|$. Then density increases if edge growth outpaces node reduction.*

Proof. Compare the density before and after:

$$D = \frac{2|E|}{|V|(|V|-1)}, \quad D' = \frac{2|E|'}{|V|'(|V|'-1)}. \quad (24)$$

Since $|V|' < |V|$, the denominator $|V|'(|V|' - 1) < |V|(|V|-1)$. If $|E|' > |E|$, the numerator increases. Thus, if the increase in $|E|'$ is significant relative to the decrease in $|V|'$, then $D' > D$. □

Lemma 5. *Let $|E| = \alpha|V|^{\beta}$, with $\beta < 2$. Then:*

$$\lim_{|V|\to\infty} \frac{2|E|}{|V|(|V|-1)} = 0. \quad (25)$$

Proof. Given $|E| = \alpha|V|^{\beta}$, the density is:

$$D = \frac{2\alpha|V|^{\beta}}{|V|(|V|-1)} \approx \frac{2\alpha|V|^{\beta}}{|V|^2} = 2\alpha|V|^{\beta-2}. \quad (26)$$

Since $\beta < 2$, $\beta - 2 < 0$. As $|V| \to \infty$:

$$\lim_{|V|\to\infty} |V|^{\beta-2} = 0, \quad (27)$$

implying $D \to 0$. □

Lemma 6. *Let $|E| = \alpha|V|^{\beta}$, with $\beta < 2$. Then:*

$$\lim_{|V|\to\infty} \frac{2|E|}{|V|(|V|-1)} = 0. \quad (28)$$

Proof. As shown:

$$D(|V|) = \frac{2\alpha|V|^{\beta}}{|V|(|V|-1)} = \frac{2\alpha|V|^{\beta}}{|V|^2(1-\frac{1}{|V|})}. \quad (29)$$

For large $|V|$, $1 - \frac{1}{|V|} \approx 1$, so:

$$D(|V|) \approx \frac{2\alpha|V|^{\beta}}{|V|^2} = 2\alpha|V|^{\beta-2}. \quad (30)$$

Since $\beta < 2$, $\beta - 2 < 0$, thus:

$$\lim_{|V|\to\infty} |V|^{\beta-2} = \lim_{|V|\to\infty} \frac{1}{|V|^{2-\beta}} = 0, \quad (31)$$

because $2 - \beta > 0$. Therefore:

$$\lim_{|V|\to\infty} D(|V|) = 2\alpha \cdot 0 = 0. \quad (32)$$

□

Theorem 1. *In a word transition graph with vertices as unique words and edges as transitions, if $|V|$ increases, the density $D = \frac{2|E|}{|V|(|V|-1)}$ decreases for large $|V|$, under:*

- *$|E| = \alpha|V|^{\beta}$, $\beta < 2$ (Eq. 15).*
- *$|E| = \alpha|V|^{\gamma}$, $\gamma < 1$ as $|E| \downarrow$ (Eq. 20).*

Proof. Define density $D(t) = \frac{2|E(t)|}{|V(t)|(|V(t)|-1)}$ Eq. 11 and using Lemmas 1 to 6.
Case 1: $|V| \uparrow$, $|E| \uparrow$, $|E| = \alpha|V|^{\beta}$, $\beta < 2$. By Eq. 11, $D(t) \approx 2\alpha|V(t)|^{\beta-2}$. Eq. 15 shows $\beta - 2 < 0$, so as $|V| \to \infty$, $|V(t)|^{\beta-2} \to 0$ Eq. 16, hence $D(t) \to 0$.
Case 2: $|V| \uparrow$, $|E| \downarrow$, $|E| = \alpha|V|^{\gamma}$, $\gamma < 1$. By Eq. 22, $D(t) \approx 2\alpha|V(t)|^{\gamma-2}$. Eq. 20 shows $\gamma - 2 < 0$, so $|V(t)|^{\gamma-2} \to 0$ Eq. 23, hence $D(t) \to 0$.

Conclusion: In both cases, $D(t) \to 0$ as $|V| \to \infty$ (Eq. 15, Eq. 20). □

Definition 5 (Random Graph (Erdős and Rényi, 1959))**.** *A random graph $G_{ER}(|V|, p)$ is a graph where each possible edge between any two distinct nodes in a vertex set V is included independently with probability $p \in [0, 1]$. The expected number of edges is:*

$$\mathbb{E}[|E|] = p \cdot \binom{|V|}{2}. \quad (33)$$

Definition 6 (Scale-Free Graph (Barabási and Albert, 1999))**.** *A* Scale-Free *graph is a graph whose degree distribution follows a power law:*

$$P(k) \sim k^{-\gamma}, \tag{34}$$

where k is the node degree and $\gamma \in (2, 3)$ for most real-world networks.

Corollary 1. *Let $G_{\text{ER}} \sim \mathcal{G}(|V|, p)$ be an Erdős Rényi random graph and let G_{SF} be a scale-free graph with degree distribution*

$$P(k) \sim k^{-\gamma}, \quad 2 < \gamma < 3.$$

Assume the number of edges grows as $|E| = \alpha|V|^{\beta}$ for some constants $\alpha > 0$ and $1 < \beta < 2$. Then the densities satisfy

$$\begin{aligned} D_{\text{ER}}(|V|) &\approx p = \frac{2|E|}{|V|(|V|-1)} = \Theta(|V|^{\beta-2}), \\ D_{\text{SF}}(|V|) &= \Theta\left(\frac{1}{|V|}\right). \end{aligned} \tag{35}$$

Hence,

$$\lim_{|V|\to\infty} D_{\text{ER}}(|V|) = 0, \quad \lim_{|V|\to\infty} D_{\text{SF}}(|V|) = 0, \tag{36}$$

but the density $D_{\text{ER}}(|V|)$ decays slower than $D_{\text{SF}}(|V|)$.

Proof. Consider the Erdős Rényi random graph G_{ER}. The expected number of edges is

$$\mathbb{E}[|E|] = p\binom{|V|}{2} = p\frac{|V|(|V|-1)}{2}. \tag{37}$$

The density is defined as

$$D_{\text{ER}}(|V|) = \frac{2\mathbb{E}[|E|]}{|V|(|V|-1)} = p. \tag{38}$$

If the edge count grows as $|E| = \alpha|V|^{\beta}$, then

$$D_{\text{ER}}(|V|) = \frac{2\alpha|V|^{\beta}}{|V|(|V|-1)} \approx \frac{2\alpha|V|^{\beta}}{|V|^2} = \Theta(|V|^{\beta-2}). \tag{39}$$

For the scale-free graph G_{SF}, the average degree is constant due to the degree distribution, implying

$$|E| = \Theta(|V|)$$

. Hence,

$$D_{\text{SF}}(|V|) = \frac{2|E|}{|V|(|V|-1)} \approx \frac{2|V|}{|V|^2} = \Theta\left(\frac{1}{|V|}\right). \tag{40}$$

Since $1 < \beta < 2$ implies $\beta - 2 < -1$, we have

$$|V|^{\beta-2} > \frac{1}{|V|} \quad \text{for large } |V|.$$

Thus, $D_{\text{ER}} > D_{\text{SF}}$ for large $|V|$. □

A.2 Maximum Degree in Terms of Vertex Count

We consider a growing graph model in which both the number of nodes and the number of edges increase. The theorem below analyzes the effect of node insertion on maximum degree.

Theorem 2. *Under the word transition sampling graph growth model, let $|E| = \alpha|V|^{\beta}$ with $\beta < 2$. Then, the maximum degree Δ satisfies:*

$$\lim_{|V|\to\infty} \frac{\Delta}{|V|} = 0. \tag{41}$$

Proof. The sum of degrees equals:

$$2|E| = 2\alpha|V|^{\beta}. \tag{42}$$

The average degree is:

$$d_{\text{avg}} = \frac{2|E|}{|V|} = 2\alpha|V|^{\beta-1}. \tag{43}$$

Let $0 < p < 1$, $q = 1 - p$, and b fixed. For $G \in \mathcal{G}(|V|, p)$, the probability that every vertex has degree at most (Riordan and Selby, 2000)

$$p|V| + b\sqrt{|V|pq} \tag{44}$$

is

$$\Pr\left(\Delta(G) \le p|V| + b\sqrt{|V|pq}\right) = (c(b) + o(1))^{|V|}, \tag{45}$$

where $c(b)$ is the root of a certain equation independent of p, with

$$c(0) \approx 0.6102 > \frac{1}{2}. \tag{46}$$

For an Erdős-Rényi graph $G(|V|, p)$, the expected number of edges is:

$$\mathbb{E}[|E|] = \frac{|V|(|V|-1)}{2}p. \tag{47}$$

Given $|E| = \alpha|V|^{\beta}$, we set:

$$\frac{|V|(|V|-1)}{2}p \approx \alpha|V|^{\beta}. \tag{48}$$

Solving for p:

$$p \approx \frac{2\alpha|V|^{\beta}}{|V|(|V|-1)} \approx \frac{2\alpha|V|^{\beta}}{|V|^2} = 2\alpha|V|^{\beta-2}, \tag{49}$$

since $|V|(|V|-1) \approx |V|^2$ for large $|V|$. However, aligning with the theorems definition, we use:

$$p = \frac{2\alpha|V|^{\beta-1}}{|V|-1}. \tag{50}$$

Since $\beta < 2$, $\beta - 1 < 1$, so $|V|^{\beta-1} \to 0$ as $|V| \to \infty$, implying $p \to 0$.

The maximum degree Δ is bounded with high probability by:

$$\Delta \leq p|V| + b\sqrt{|V|pq}. \tag{51}$$

Substitute $p = 2\alpha|V|^{\beta-2}$:

$$p|V| = (2\alpha|V|^{\beta-2})|V| = 2\alpha|V|^{\beta-1}. \tag{52}$$

Since $q \approx 1$, we have:

$$|V|pq \approx |V|p = 2\alpha|V|^{\beta-1}, \tag{53}$$
$$\sqrt{|V|pq} \approx \sqrt{2\alpha} \cdot |V|^{(\beta-1)/2}. \tag{54}$$

Thus:

$$\Delta \leq 2\alpha|V|^{\beta-1} + b\sqrt{2\alpha} \cdot |V|^{(\beta-1)/2}. \tag{55}$$

Dividing by $|V|$:

$$\frac{\Delta}{|V|} \leq 2\alpha|V|^{\beta-2} + b\sqrt{2\alpha} \cdot |V|^{(\beta-1)/2-1}. \tag{56}$$

The first term has exponent $\beta - 2 < 0$, so $2\alpha|V|^{\beta-2} \to 0$. The second terms exponent is:

$$\frac{\beta-1}{2} - 1 = \frac{\beta-3}{2} < 0, \tag{57}$$

since $\beta - 3 < -1$. Thus, $|V|^{(\beta-3)/2} \to 0$. Both terms vanish as $|V| \to \infty$, so:

$$\lim_{|V|\to\infty} \frac{\Delta}{|V|} = 0. \tag{58}$$

For scale-free graphs with $P(d) \sim d^{-\gamma}$, the maximum degree scales as (Barabási, 2002):

$$\Delta \sim |V|^{\frac{1}{\gamma-1}}. \tag{59}$$

Since $\gamma > 2$, we have:

$$\frac{\Delta}{|V|} \sim |V|^{\frac{1}{\gamma-1}-1} = |V|^{\frac{2-\gamma}{\gamma-1}} \to 0. \tag{60}$$

□

A.3 Proof of Graph Energy Upper-bound

Theorem 3. *Let $G^{(t)} = (V^{(t)}, E^{(t)})$ be a graph evolving over time t, where $|V^{(t)}|$ is the number of vertices and $|E^{(t)}| = \alpha|V^{(t)}|^\beta$ with constants $\alpha > 0$ and $1 < \beta < 2$. For both Erdős–Rényi and scale-free models, the graph energy $\mathcal{E}(G^{(t)})$ is upper-bounded by*

$$\mathcal{E}(G^{(t)}) \leq \sqrt{2\alpha}\,|V^{(t)}|^{\frac{\beta+1}{2}}. \tag{61}$$

As $|V^{(t)}|$ increases over time, this bound increases, implying that $\mathcal{E}(G^{(t)})$ grows with t.

Proof. Consider a graph $G^{(t)} = (V^{(t)}, E^{(t)})$ evolving over time t, with $|V^{(t)}|$ vertices and edges governed by:

$$|E^{(t)}| = \alpha|V^{(t)}|^\beta, \quad \alpha > 0, \quad 1 < \beta < 2. \tag{62}$$

Let $n_t = |V^{(t)}|$, so $|E^{(t)}| = \alpha n_t^\beta$. As t increases, n_t increases, and since $\beta > 1$, $|E^{(t)}|$ grows superlinearly with n_t.

For $G^{(t)}$, the adjacency matrix $A^{(t)}$ has eigenvalues $\lambda_1, \ldots, \lambda_{n_t}$, each satisfying $A^{(t)}\vec{v}_\ell = \lambda_\ell \vec{v}_\ell$. The graph energy is:

$$\mathcal{E}(G^{(t)}) = \sum_{\ell=1}^{n_t} |\lambda_\ell|. \tag{63}$$

McClelland (McClelland, 1971) provides an upper bound for the graph energy:

$$\mathcal{E}(G^{(t)}) \leq \sqrt{2|E^{(t)}||V^{(t)}|}. \tag{64}$$

Substituting the edge growth law:

$$\sqrt{2|E^{(t)}||V^{(t)}|} = \sqrt{2(\alpha n_t^\beta)n_t} = \sqrt{2\alpha n_t^{\beta+1}}. \tag{65}$$

Define the upper bound as:

$$U(n_t) = \sqrt{2\alpha n_t^{\beta+1}}. \tag{66}$$

Since $\beta + 1 > 2$, $U(n_t)$ is monotonically increasing in n_t. As t increases, n_t grows, so the upper bound increases over time.

For an Erdős-Rényi graph $G^{(t)} \sim G(n, p)$, the edge probability at time t is:

$$p_t = \frac{2|E^{(t)}|}{n_t(n_t-1)} \approx \frac{2\alpha n_t^\beta}{n_t^2} = 2\alpha n_t^{\beta-2}. \tag{67}$$

The expected number of edges matches $|E^{(t)}|$, and the upper bound $\sqrt{2\alpha n_t^{\beta+1}}$ applies. Since $p_t \to 0$

but $n_t p_t = 2\alpha n_t^{\beta-1} \to \infty$, and the bound increases as n_t and $|E^{(t)}|$ grow.

For a scale-free graph, assume a generative model (e.g., preferential attachment (Barabási, 2002)) where the total number of edges at time t is $|E^{(t)}| = \alpha n_t^{\beta}$. The degree distribution follows $P(k) \sim k^{-\gamma}$, and the upper bound remains:

$$\sqrt{2|E^{(t)}||V^{(t)}|} = \sqrt{2\alpha n_t^{\beta+1}}, \tag{68}$$

which increases with n_t, consistent with the growth in $|V^{(t)}|$ and $|E^{(t)}|$.

For both Erdős-Rényi and scale-free graphs, the upper bound on $\mathcal{E}(G^{(t)})$ increases as $|V^{(t)}|$ and $|E^{(t)}|$ grow over time. □

B Independence of Graph-Based Metrics in Language Model Evaluation

We present a detailed empirical investigation of the relationships between the proposed metrics spectral entropy, graph energy, and graph density using correlation analysis across different datasets and model scales. To visualize these relationships, we computed pairwise correlations on subsets of metric data drawn from various checkpoints and datasets. For each pair of metrics, we generated joint distribution plots with marginal histograms to examine their bivariate behavior, applying linear scales to accurately capture metric variability. Additionally, a comprehensive correlation matrix heatmap was created. The analysis consistently revealed low correlations among the metrics, indicating that each captures distinct and complementary dimensions of model behavior. Figures 3 to 9 show various correlation analyses on the PIQA dataset across models 70, 160, and 410 m parameters. Specifically, Figures 3 and 4 illustrate correlations between density and graph energy or spectral entropy for models 70 and 160 m parameters. Figures 5 and 6 extend these comparisons to model 410 m parameters and other metric pairs. Finally, Figures 7 through 9 display correlation matrices for all three models.

Figures 10 to 16 present correlation results on the ARC-Easy dataset for models 70, 160, and 410 m parameters. The early figures compare density with graph energy and spectral entropy (Figures 10 and 12), while later ones (Figures 13) show correlations involving graph energy versus spectral entropy. The last set (Figures 14 to 16) provide correlation matrices per model.

Figures 17 to 23 summarize correlation analyses on the LogiQA dataset for models 70, 160, and 410 m parameters. Initial figures (Figures 17 and 18) depict correlations between density and graph energy or spectral entropy for models 70 and 160 m parameters. Figures 19 and 20 expand these to model 410 m parameters and other pairs, while Figures 21 through 23 show correlation matrices.

C Experimental Results for Pythia Models with 160 M and 410 M Parameters

Figures 24 and 25 show the behavior of the density metric across training steps for the Pythia models with 160 million and 410 million parameters, respectively. In both models, density decreases as the number of training steps increases, consistently across the PIQA, ARC-Easy, and LogiQA datasets.

Conversely, Figures 26 and 27 illustrate that graph energy increases over the training steps for both models and all datasets.

Similarly, spectral entropy shows an increasing trend with training progression as depicted in Figures 28 and 29.

Together, these results demonstrate consistent trends in graph-based metrics as training advances, revealing important insights into the structural evolution of models with varying parameter sizes on multiple datasets.

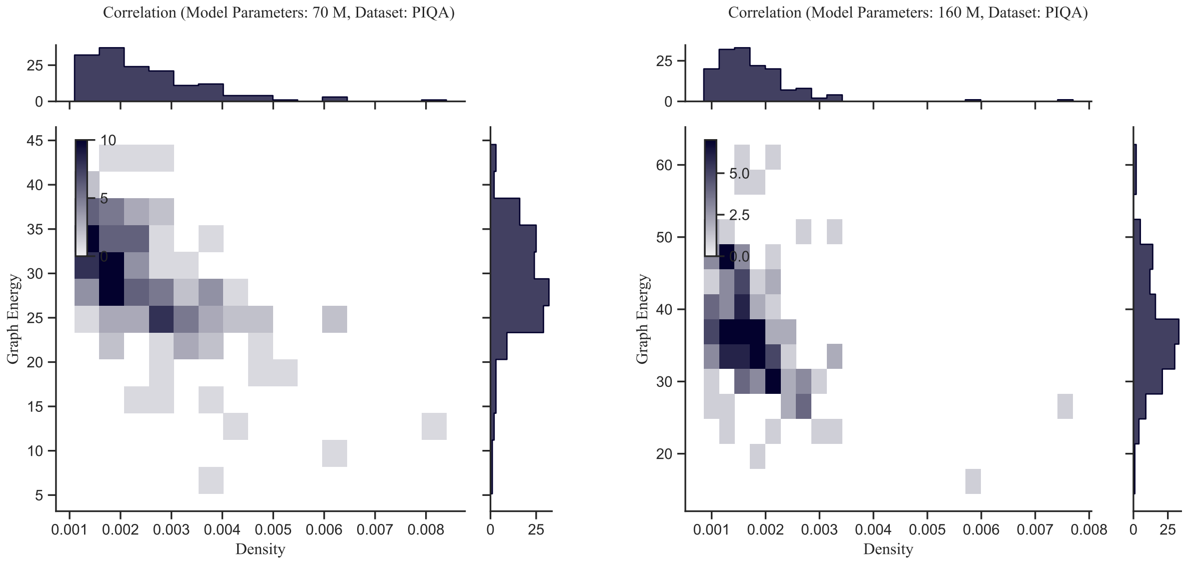


Figure 3: PIQA dataset: Correlation Density vs Graph Energy for Model 70M (left) and Model 160M (right).

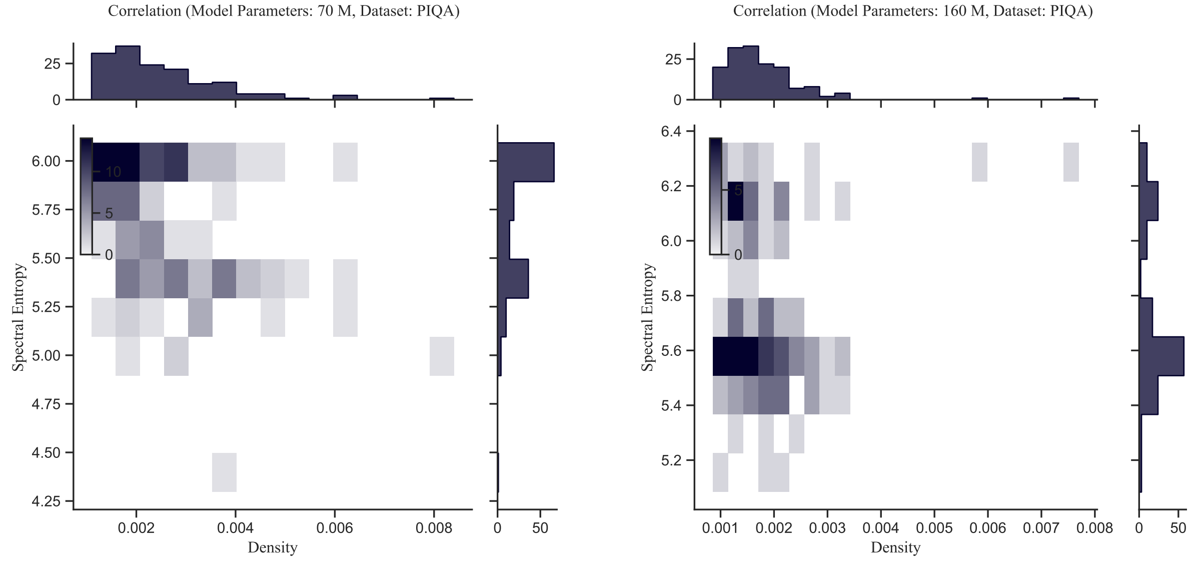


Figure 4: PIQA dataset: Correlation Density vs Spectral Entropy for Model 70M (left) and Model 160M (right).

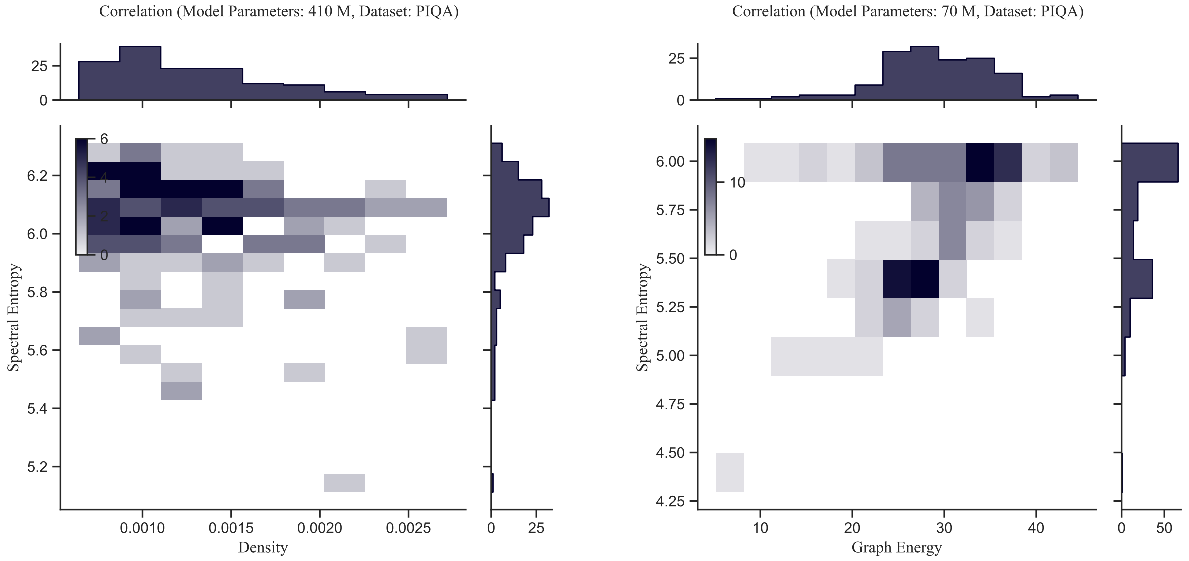


Figure 5: PIQA dataset: Correlation Density vs Spectral Entropy for Model 410M (left) and Correlation Graph Energy vs Spectral Entropy for Model 70M (right).

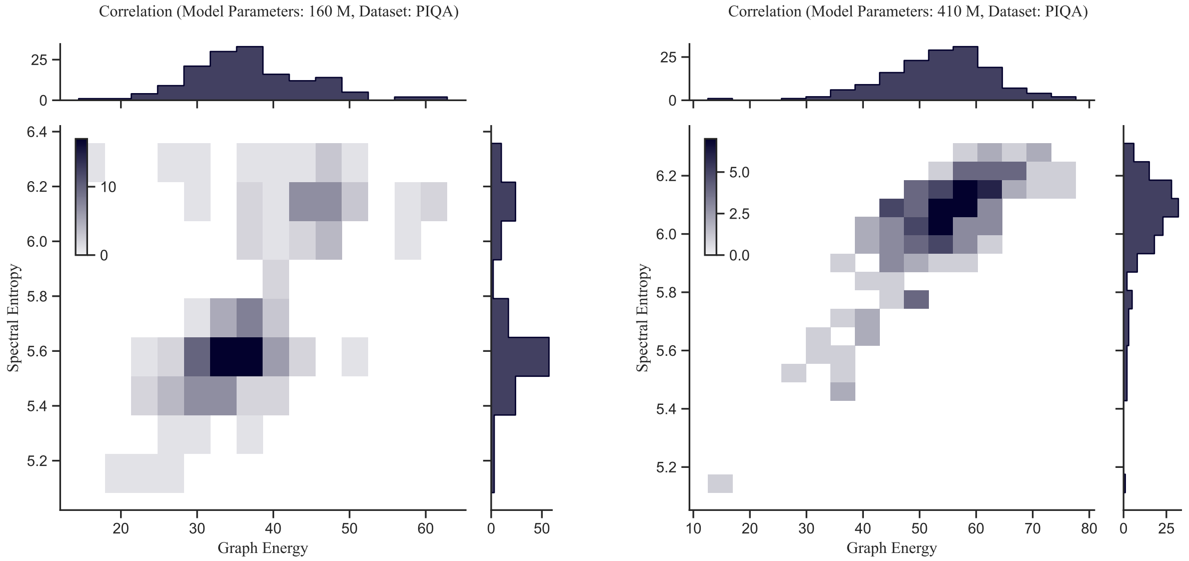


Figure 6: PIQA dataset: Correlation Graph Energy vs Spectral Entropy for Model 160M (left) and Model 410M (right).

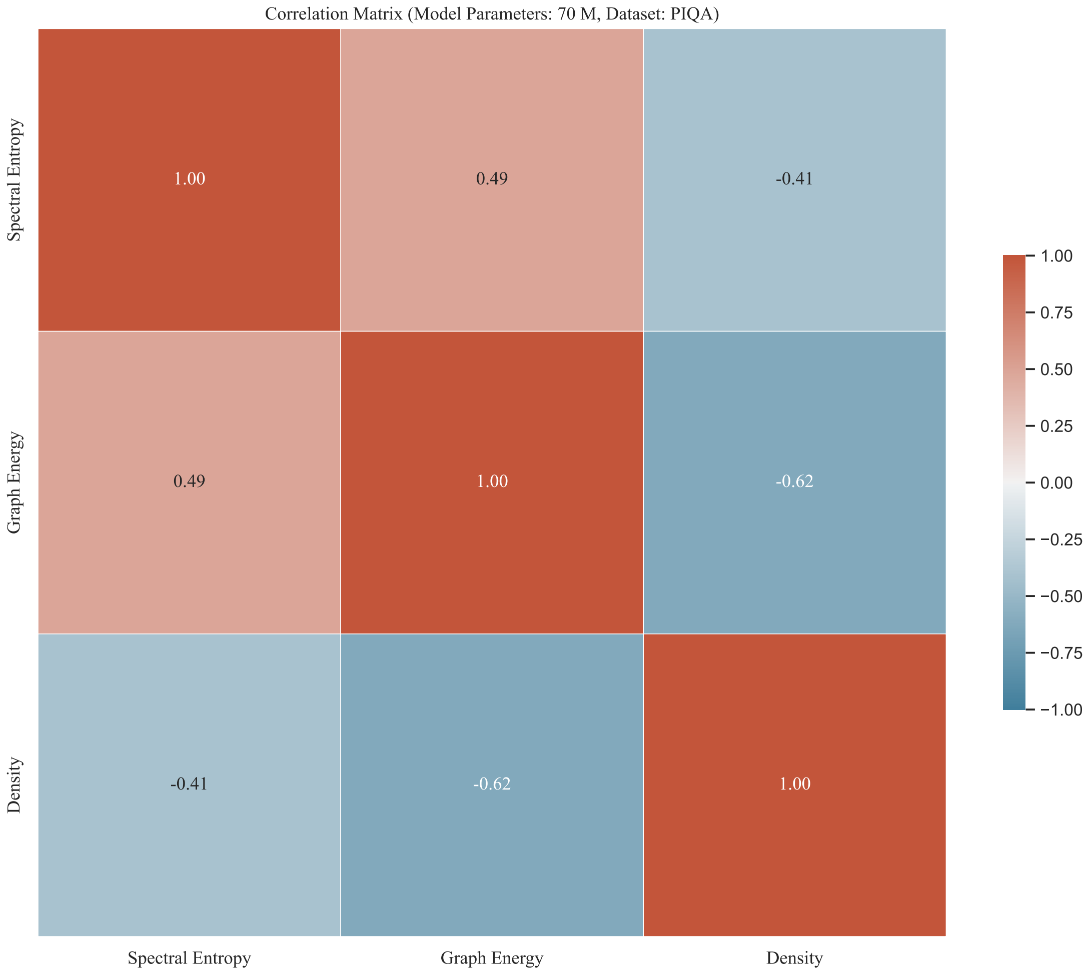


Figure 7: PIQA dataset: Correlation Matrix for Model 70M.

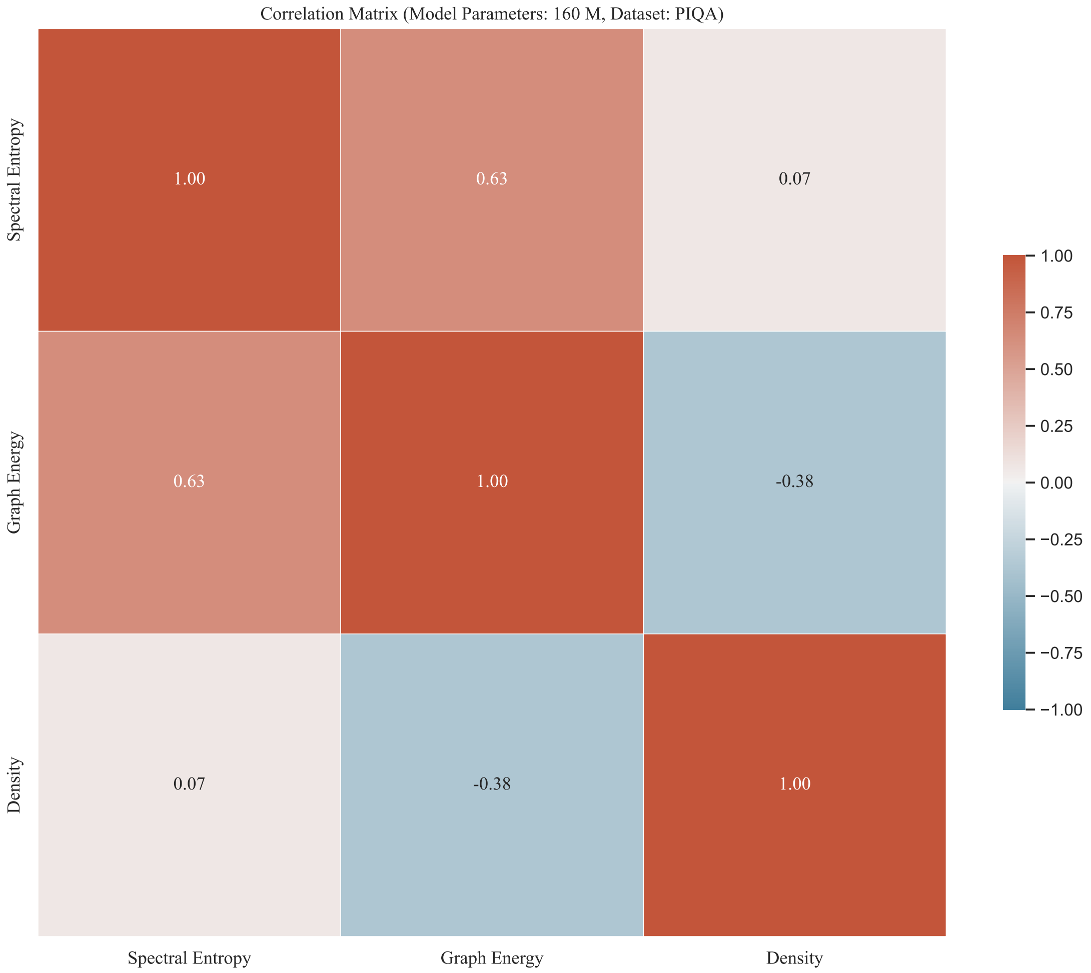


Figure 8: PIQA dataset: Correlation Matrix for Model 160M.

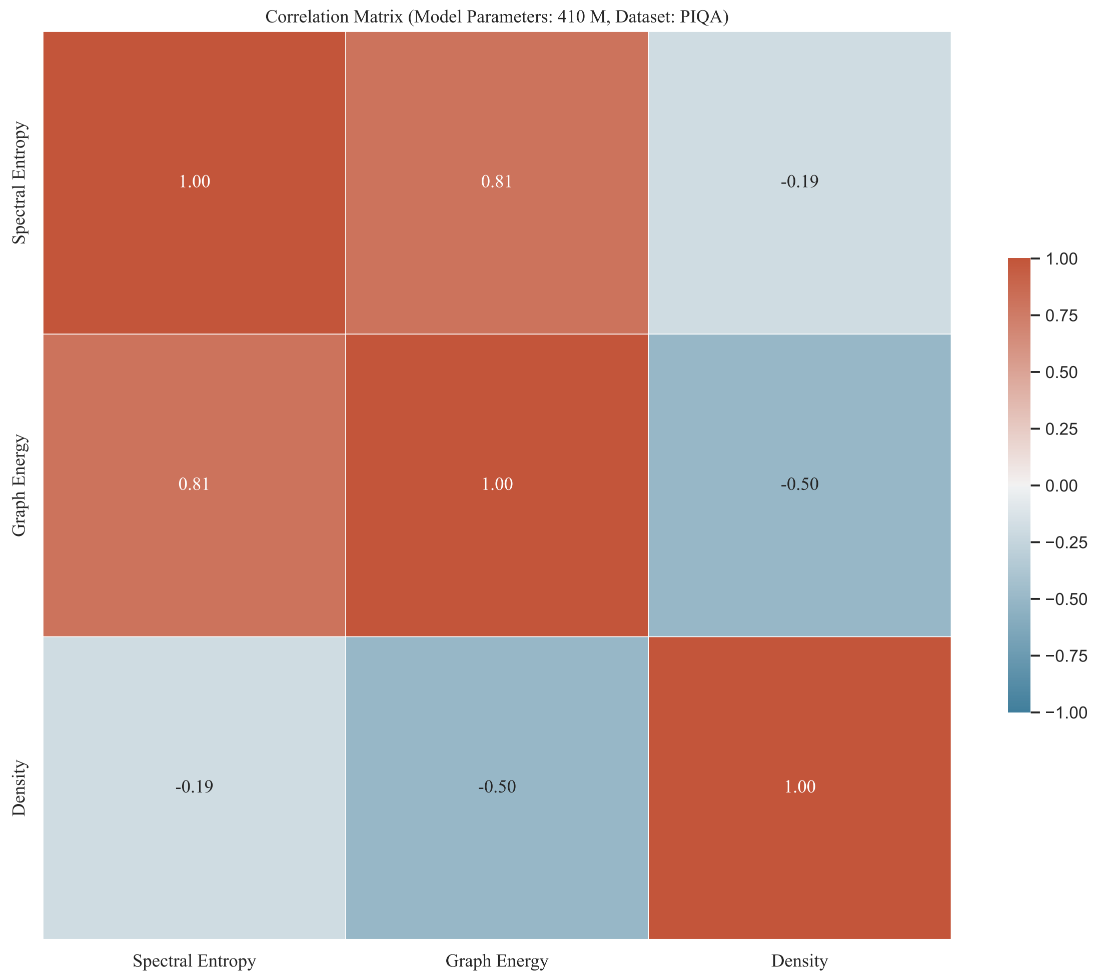


Figure 9: PIQA dataset: Correlation Matrix for Model 410M.

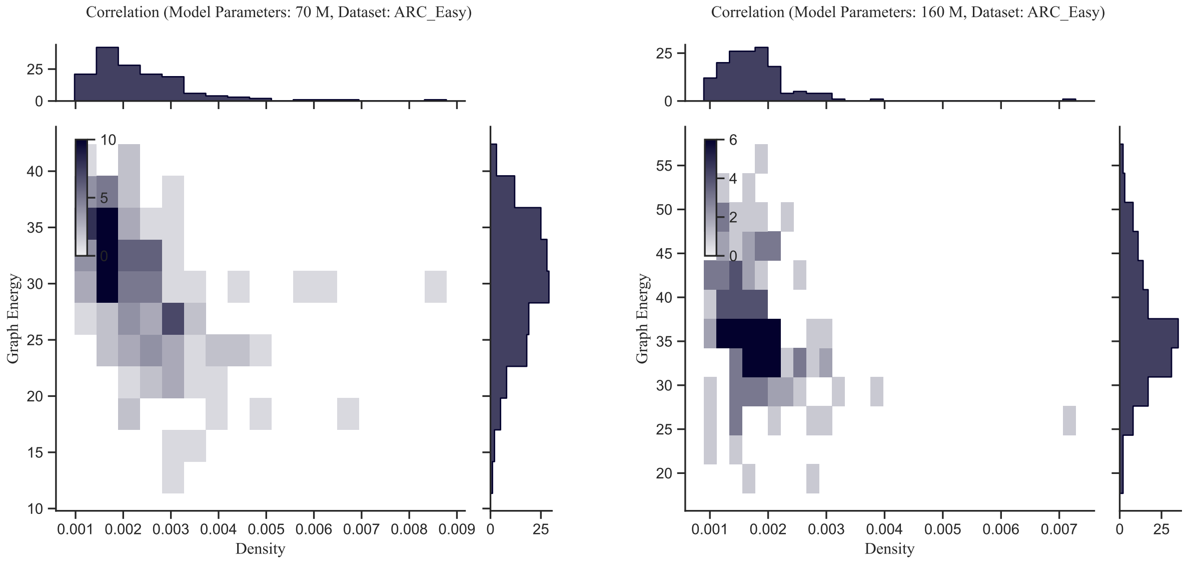


Figure 10: ARC-Easy dataset: Correlation Density vs Graph Energy for Model 70M (left) and Model 160M (right).

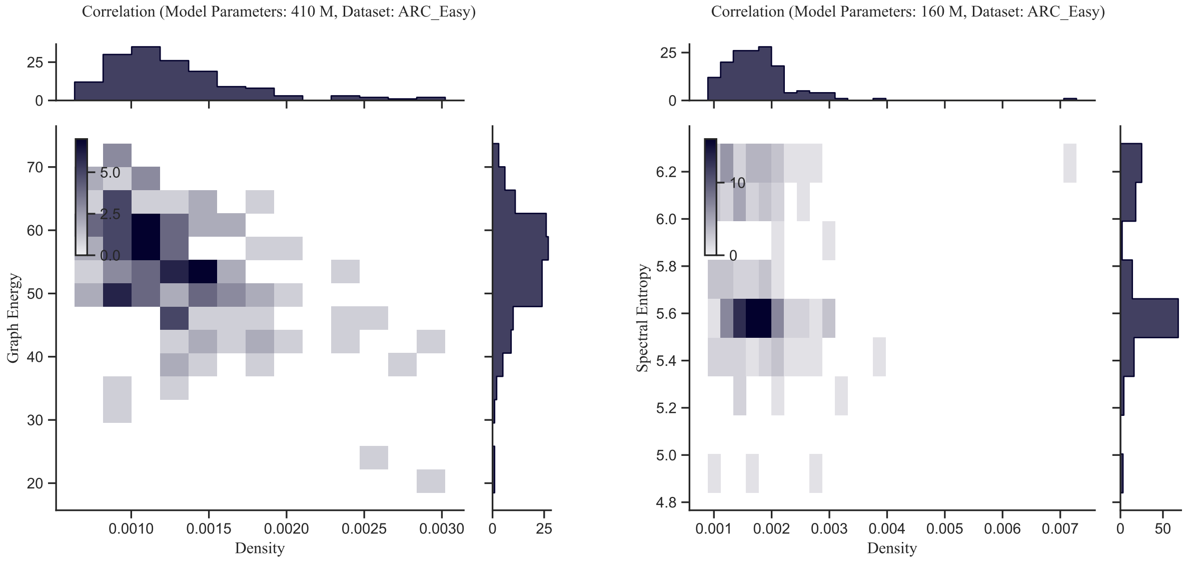


Figure 11: ARC-Easy dataset: Correlation Density vs Graph Energy for Model 410M (left) and Correlation Density vs Spectral Entropy for Model 160M (right).

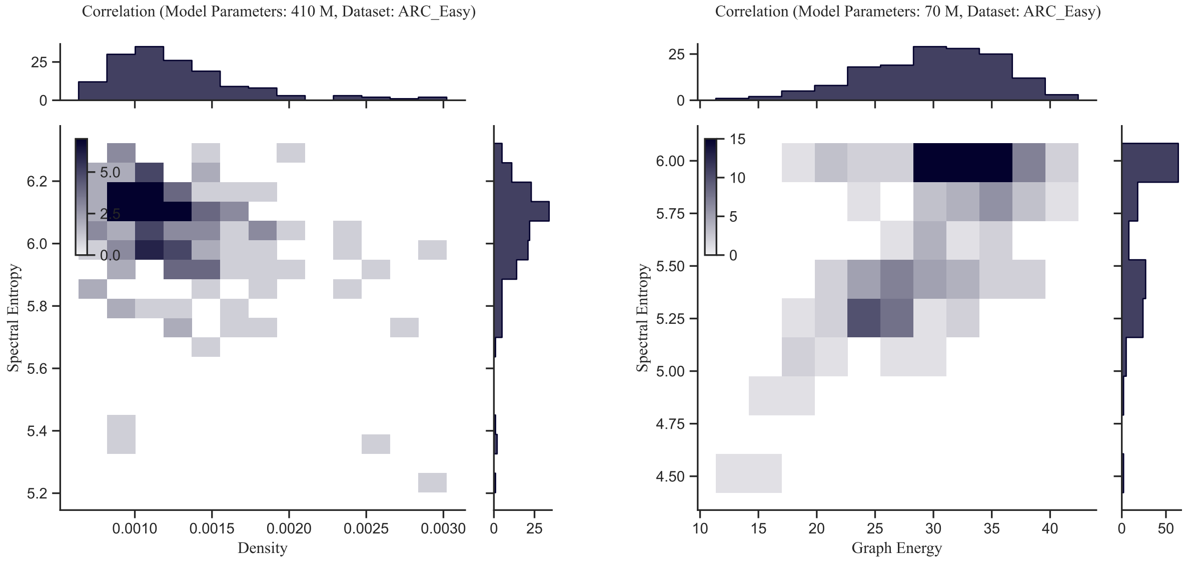
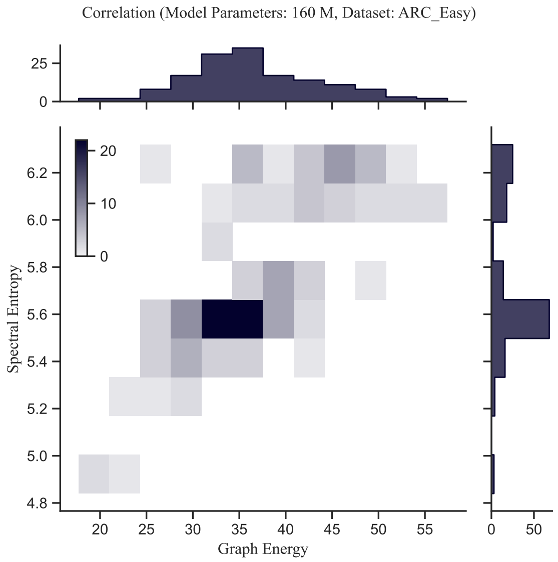


Figure 12: ARC-Easy dataset: Correlation Density vs Spectral Entropy for Model 410M (left) and Correlation Graph Energy vs Spectral Entropy for Model 70M (right).



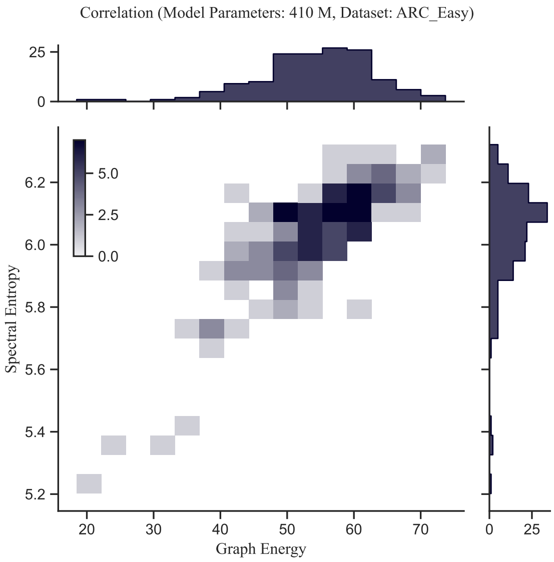


Figure 13: ARC-Easy dataset: Correlation Graph Energy vs Spectral Entropy for Model 160M (left) and Model 410M (right).

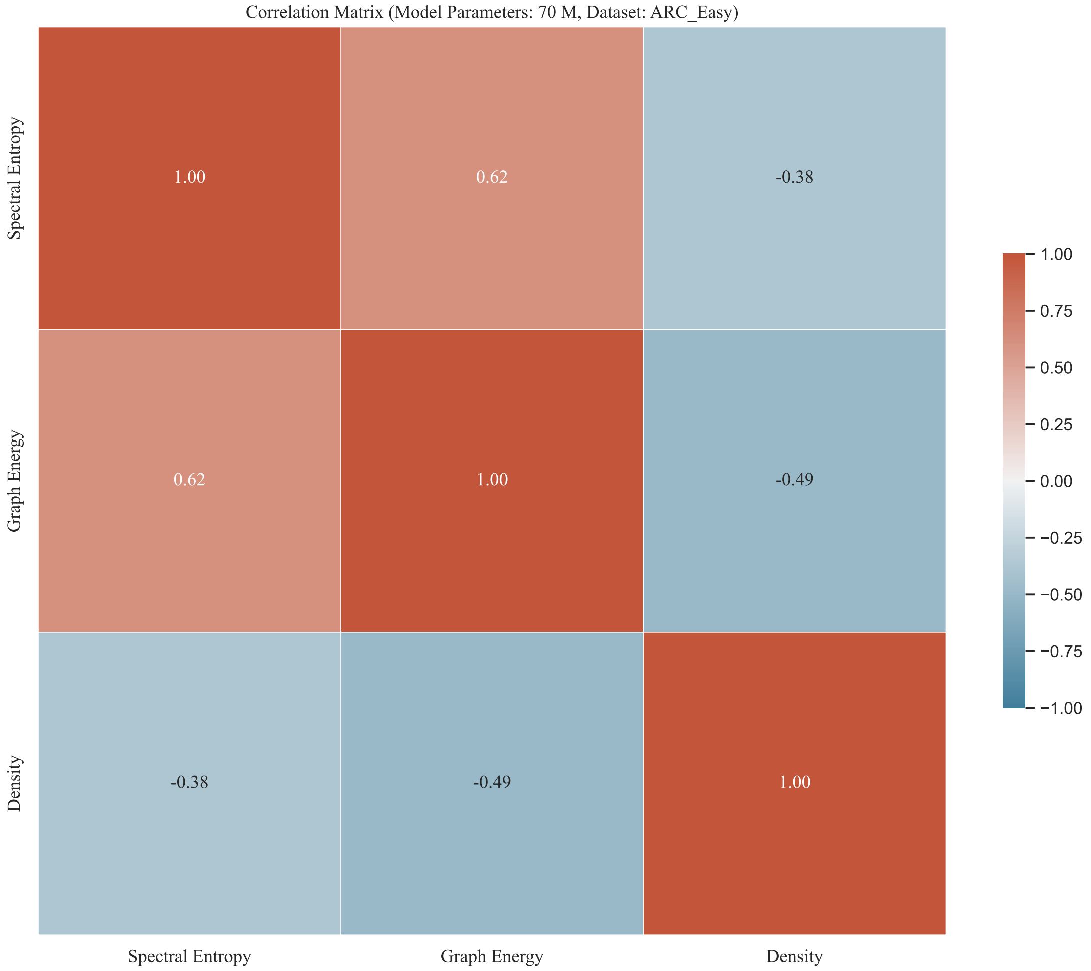


Figure 14: ARC-Easy dataset: Correlation Matrix for Model 70M.

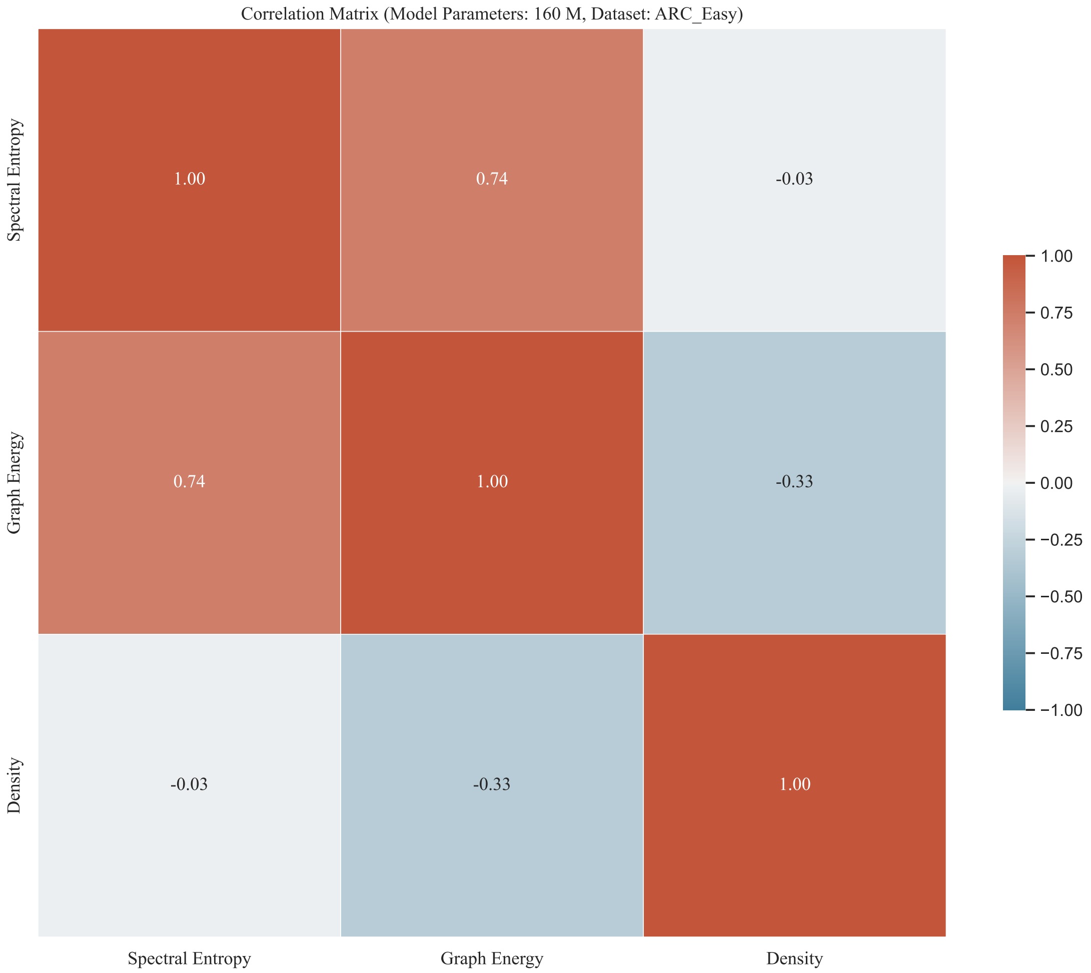


Figure 15: ARC-Easy dataset: Correlation Matrix for Model 160M.

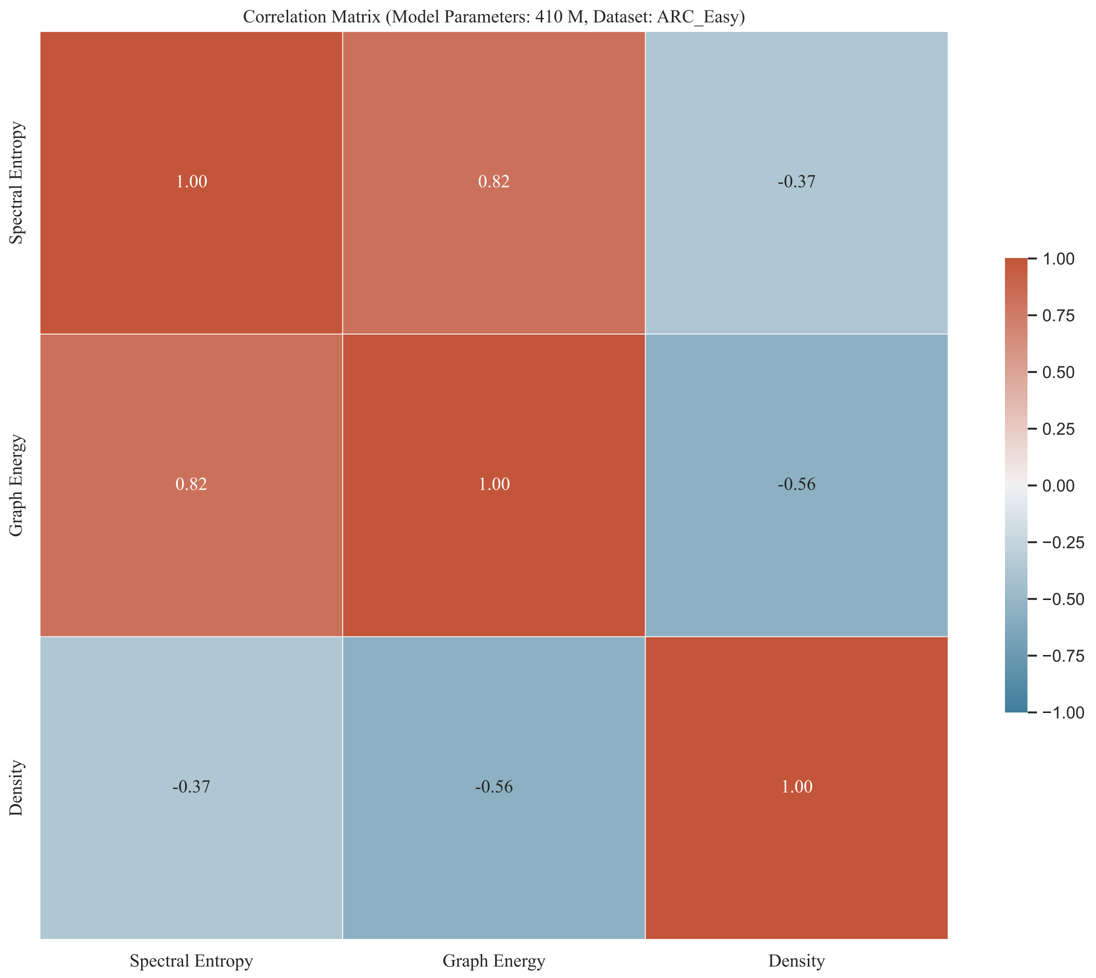


Figure 16: ARC-Easy dataset: Correlation Matrix for Model 410M.

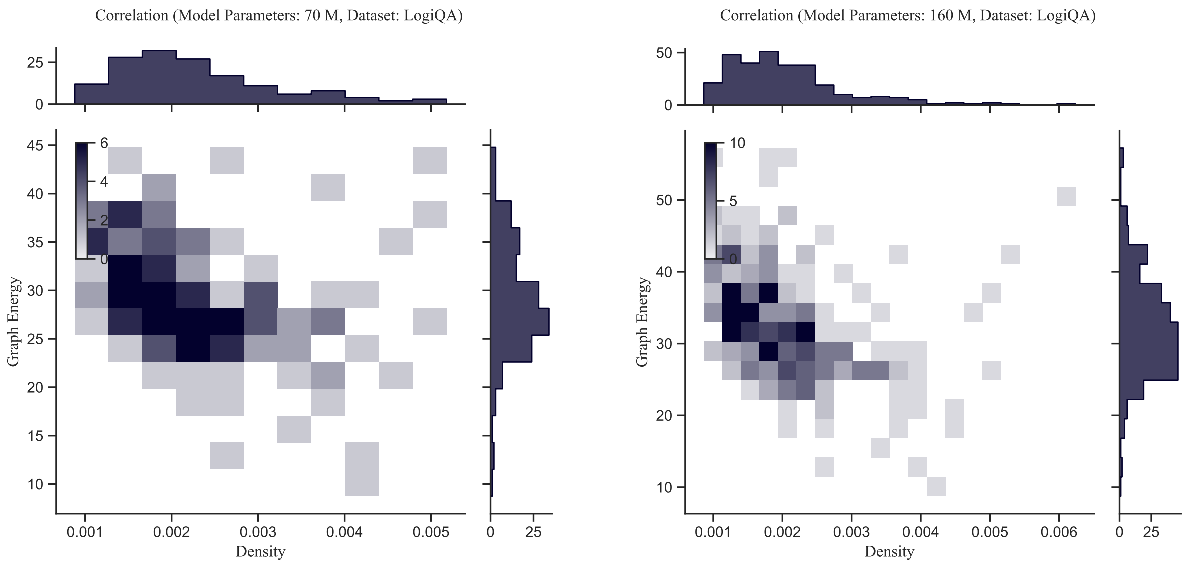


Figure 17: LogiQA dataset: Correlation Density vs Graph Energy for Model 70M (left) and Model 160M (right).

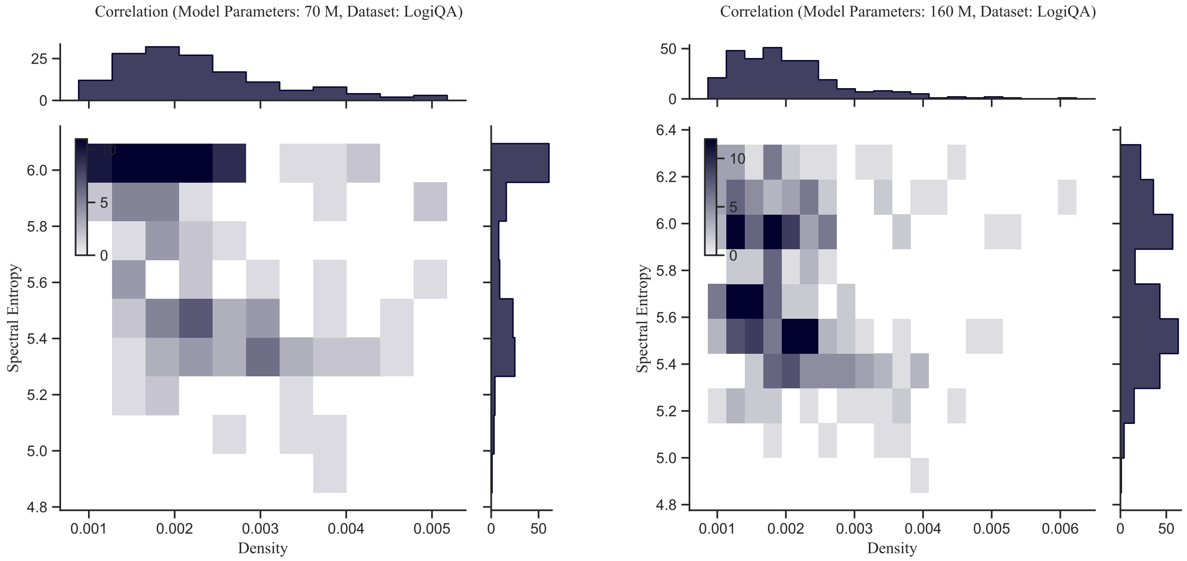


Figure 18: LogiQA dataset: Correlation Density vs Spectral Entropy for Model 70M (left) and Model 160M (right).

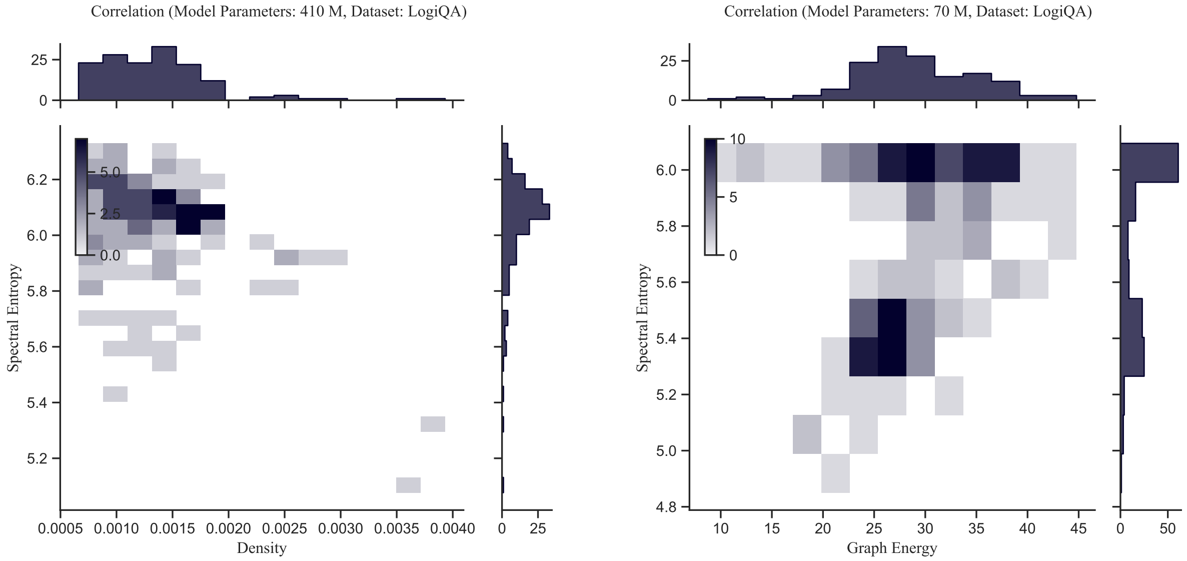


Figure 19: LogiQA dataset: Correlation Density vs Spectral Entropy for Model 410M (left) and Correlation Graph Energy vs Spectral Entropy for Model 70M (right).

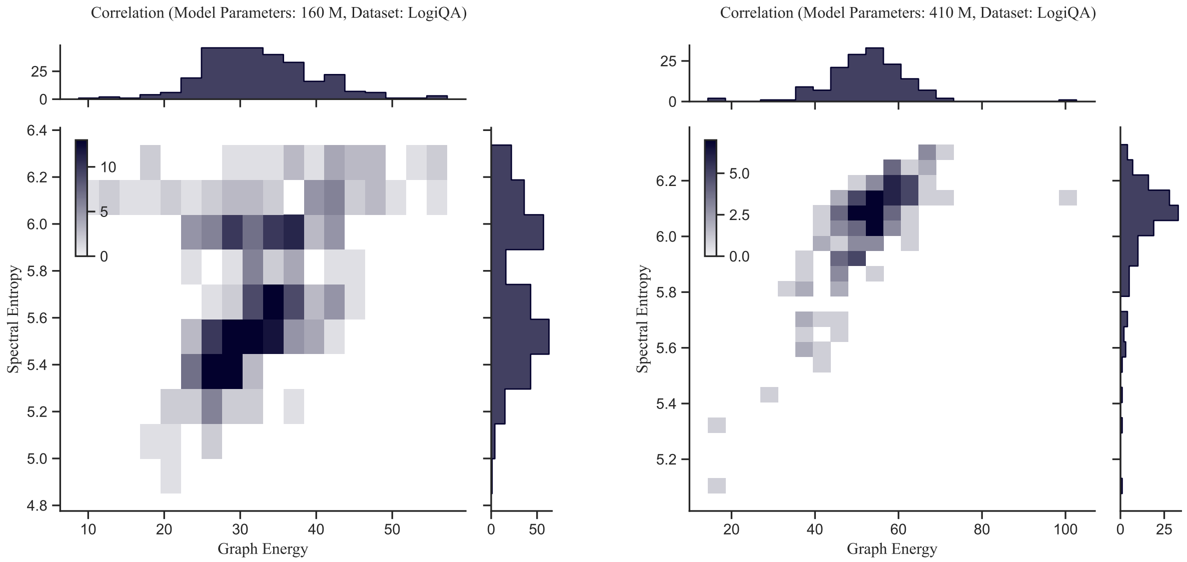


Figure 20: LogiQA dataset: Correlation Graph Energy vs Spectral Entropy for Model 160M (left) and Model 410M (right).

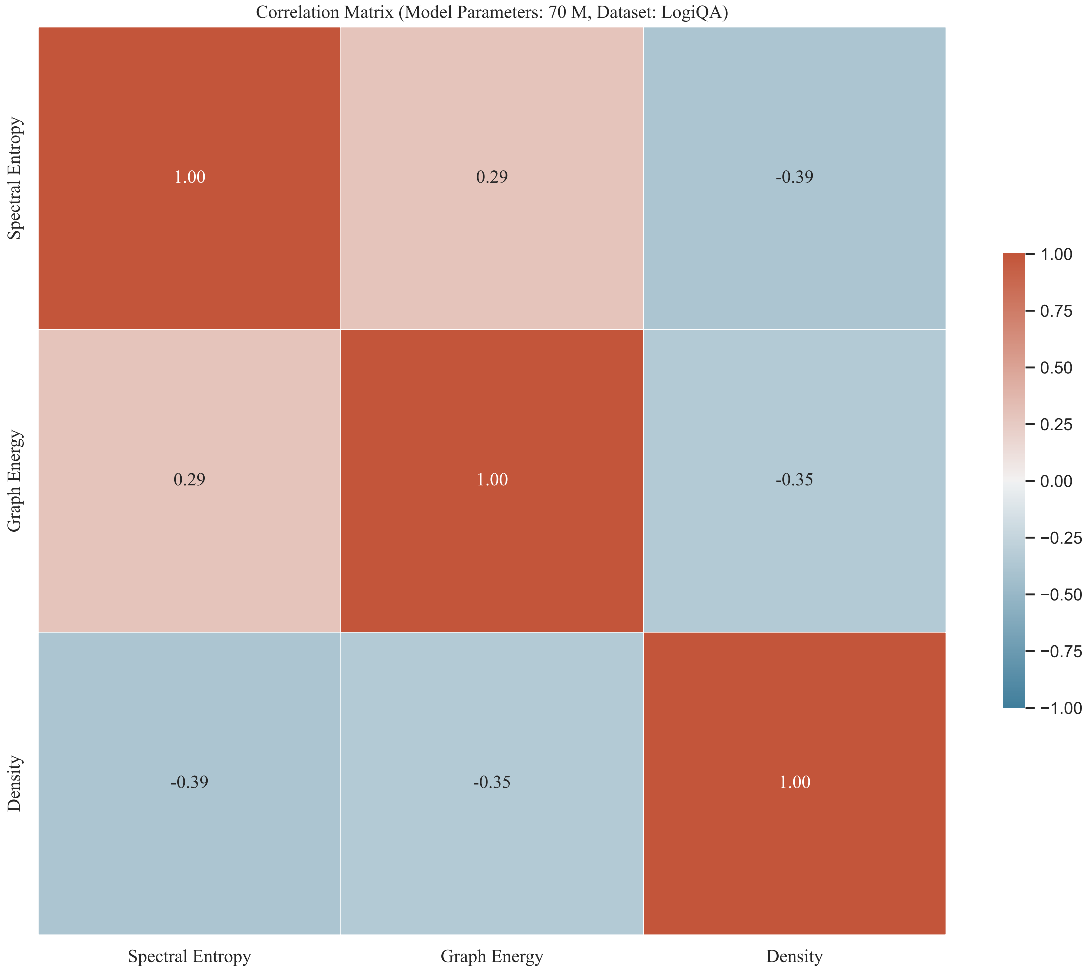


Figure 21: LogiQA dataset: Correlation Matrix for Model 70M.

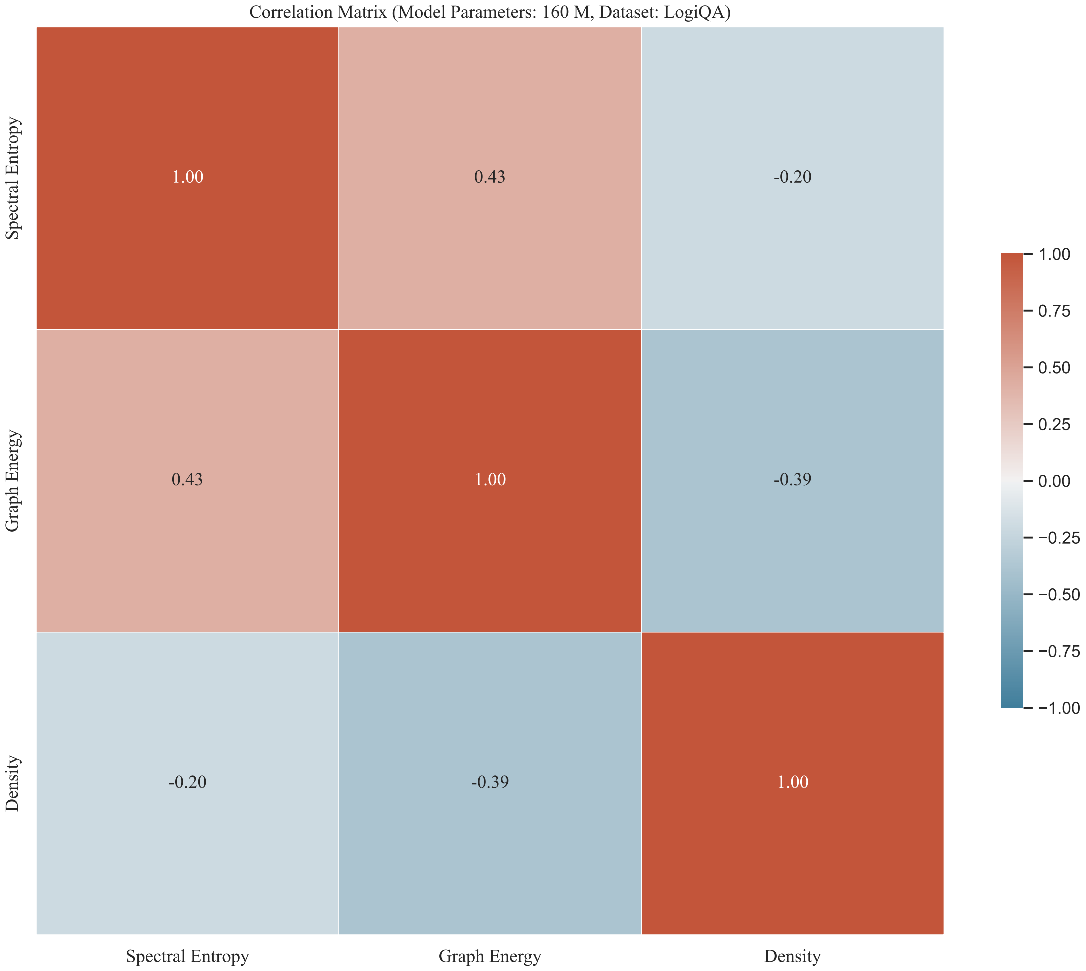


Figure 22: LogiQA dataset: Correlation Matrix for Model 160M.

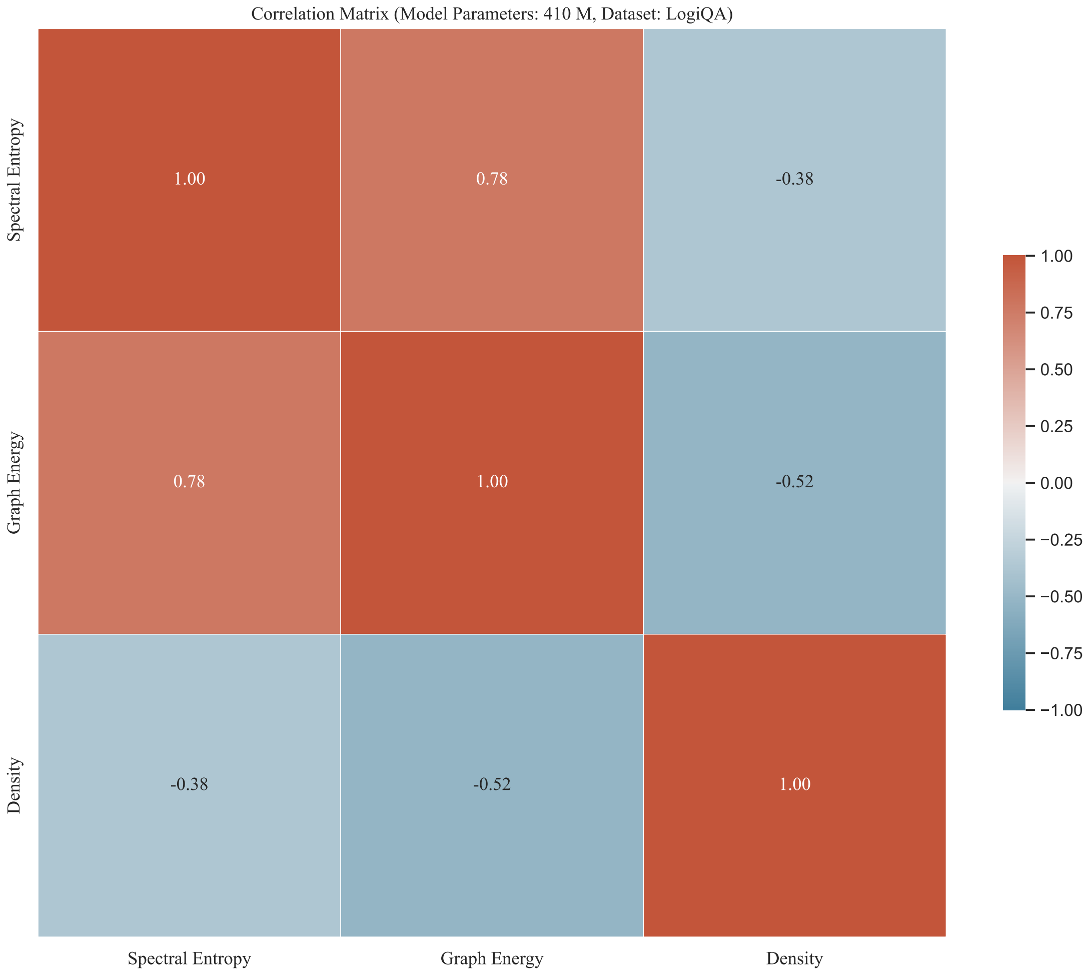


Figure 23: LogiQA dataset: Correlation Matrix for Model 410M.

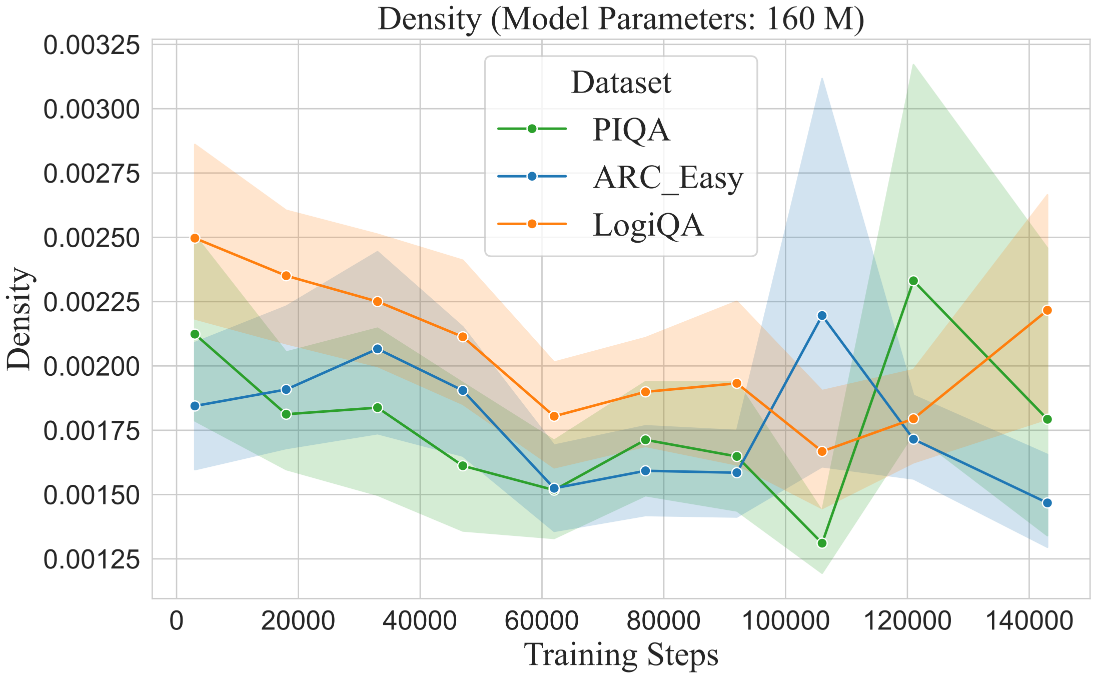


Figure 24: Density metric (y-axis) vs Training Steps (x-axis) for Model 160 across PIQA, ARC-Easy, and LogiQA datasets. Density decreases as training steps increase.

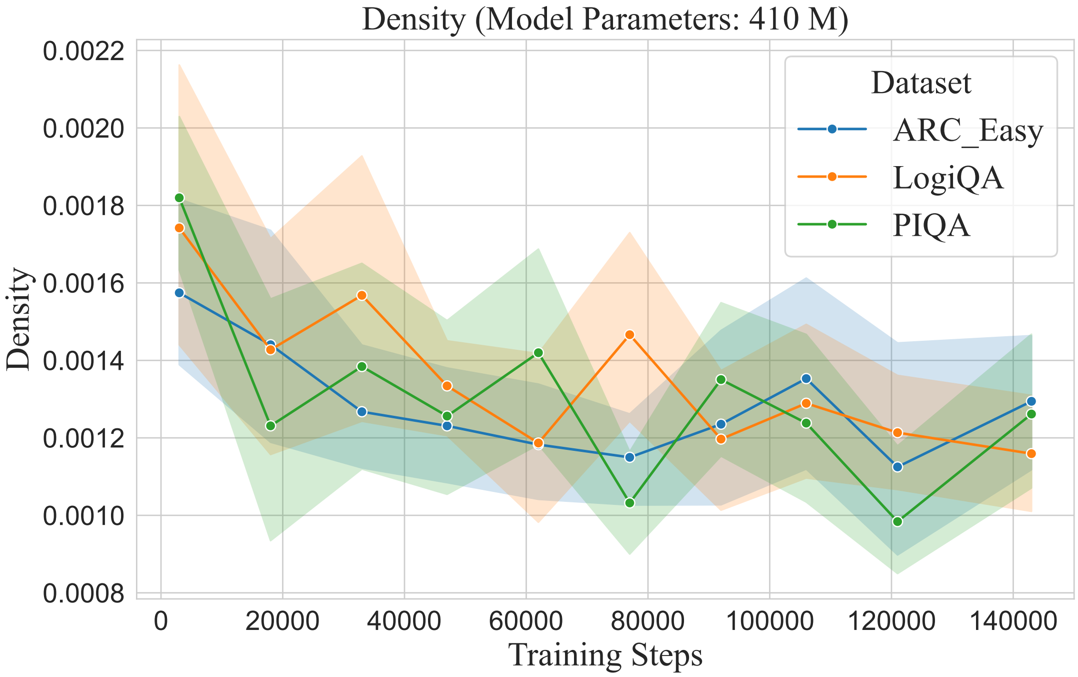


Figure 25: Density metric (y-axis) vs Training Steps (x-axis) for Model 410 across PIQA, ARC-Easy, and LogiQA datasets. Density decreases as training steps increase.

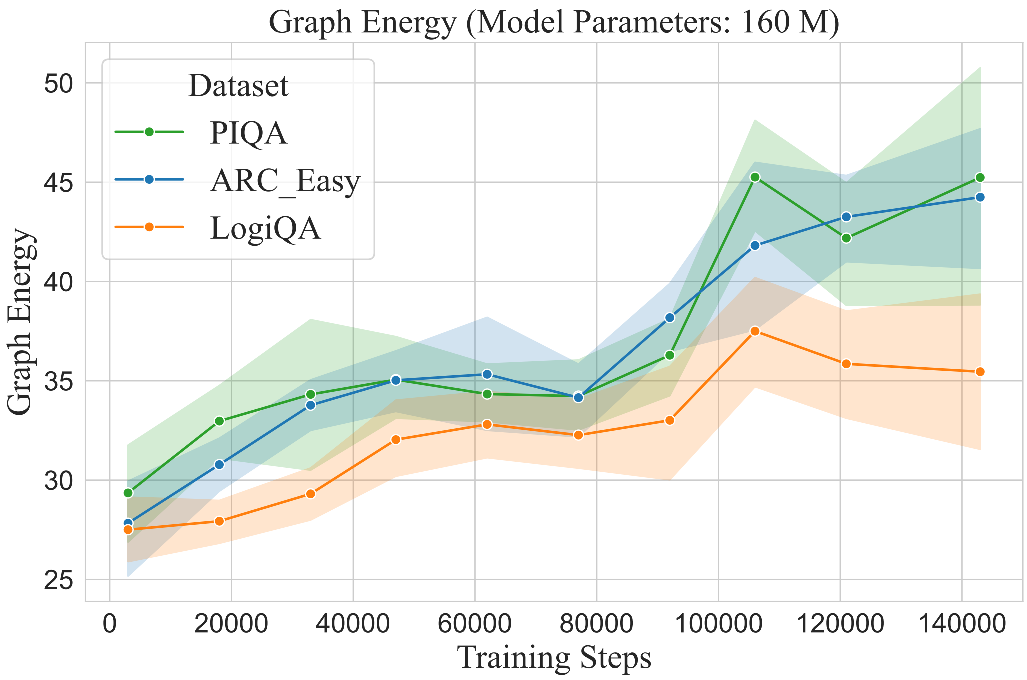


Figure 26: Graph Energy metric (y-axis) vs Training Steps (x-axis) for Model 160 across PIQA, ARC-Easy, and LogiQA datasets. Graph Energy increases as training steps increase.

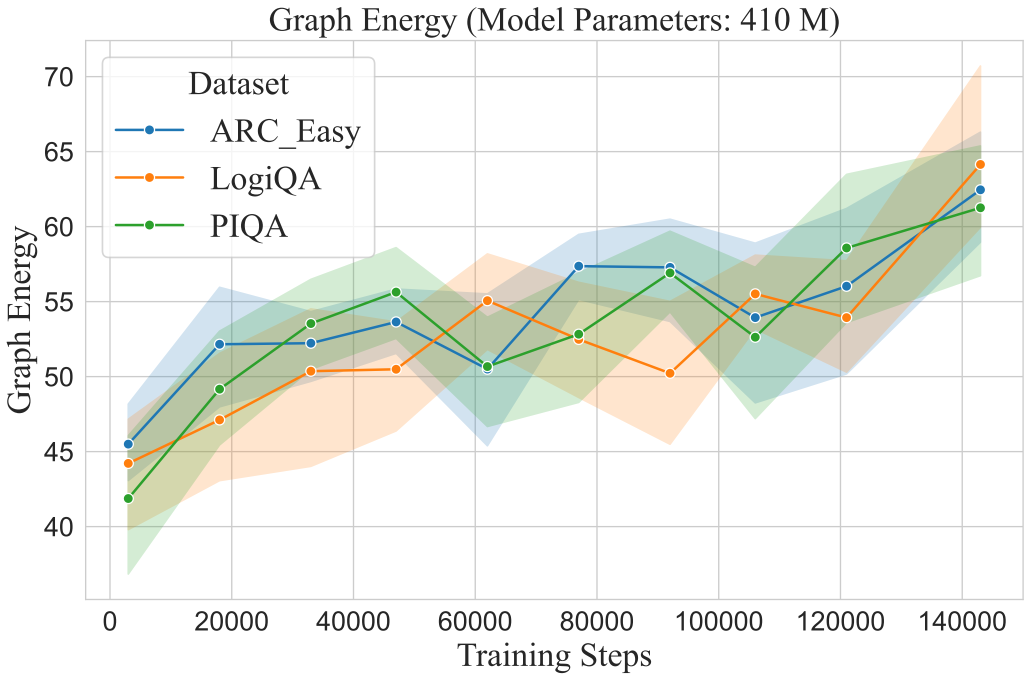


Figure 27: Graph Energy metric (y-axis) vs Training Steps (x-axis) for Model 410 across PIQA, ARC-Easy, and LogiQA datasets. Graph Energy increases as training steps increase.

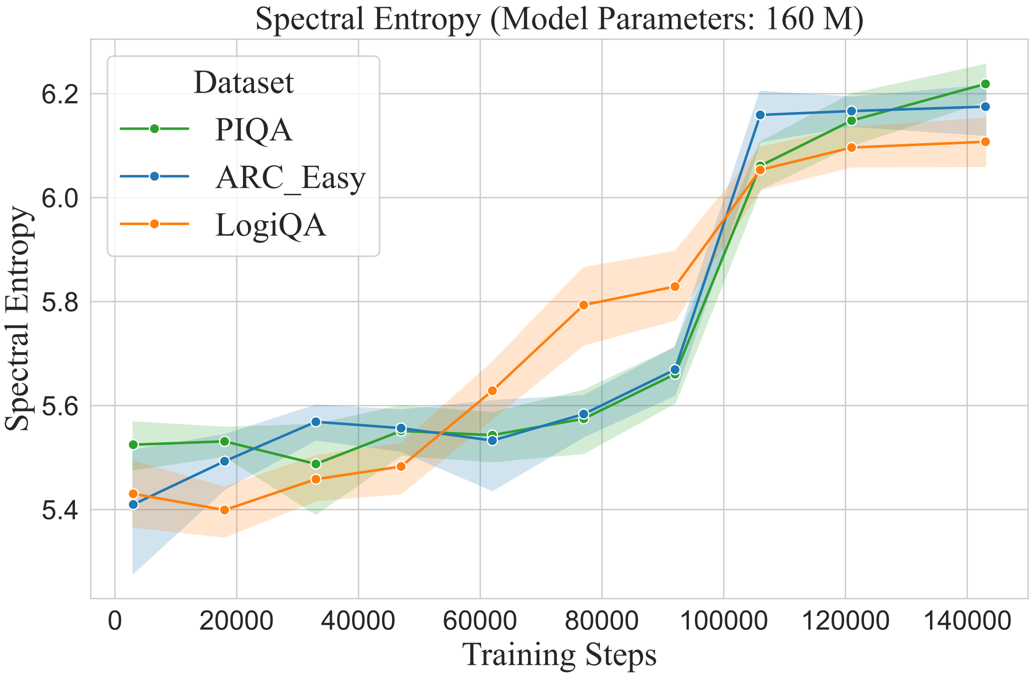


Figure 28: Spectral Entropy metric (y-axis) vs Training Steps (x-axis) for Model 160 across PIQA, ARC-Easy, and LogiQA datasets. Spectral Entropy increases as training steps increase.

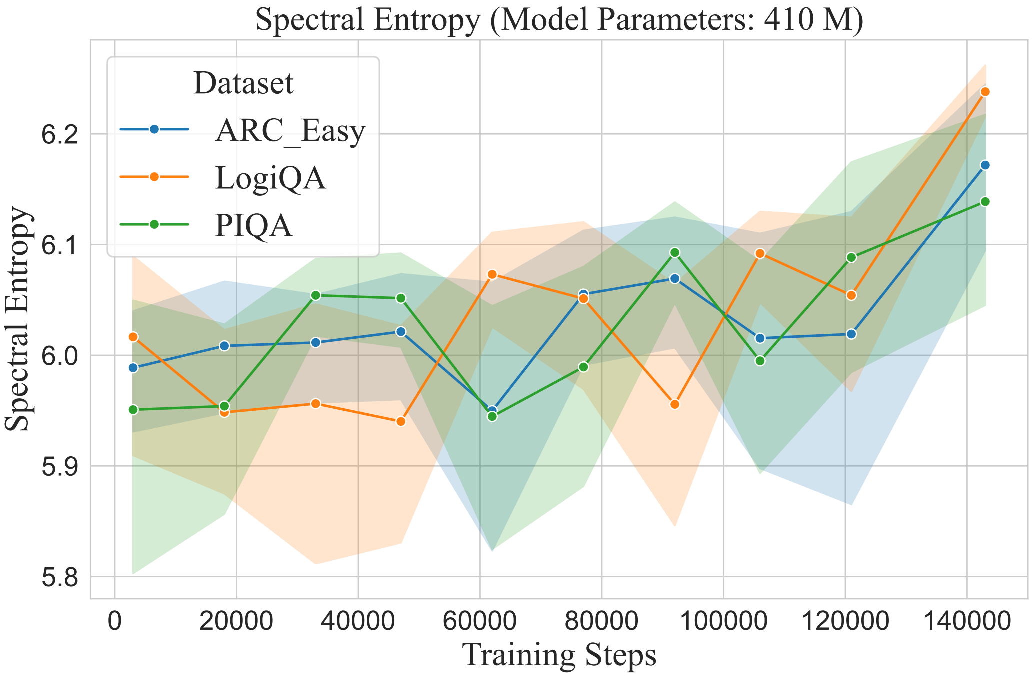


Figure 29: Spectral Entropy metric (y-axis) vs Training Steps (x-axis) for Model 410 across PIQA, ARC-Easy, and LogiQA datasets. Spectral Entropy increases as training steps increase.