Quantifying Representational Complexity in Transformer Models via Residual Stream Spectral Entropy

Keval Barvaliya
Department of Computer Science and Technology,
Kean University, New Jersey, USA, 07083

ABSTRACT

Despite remarkable advances in the reasoning and generalization capabilities of large language models, the internal representational mechanisms that underlie these behaviors remain poorly understood. Current frameworks of evaluation are mainly based on the assumption of benchmark performance as a proxy for model capability and provide limited information about the organization and transformation of information in the network during inference. One of the main open questions in current interpretability work is the disconnect between the behavioral assessment and the understanding of the mechanisms.

In this work, a principled metric, Residual Stream Spectral Entropy (RSSE), is proposed for measuring the representational complexity of transformer-based language models by analyzing their residual stream activations. The RSSE represents the entropy of the singular value distribution of the layer-wise residual representations, and hence a normalized score of the distribution of the representational energy of a model across latent computational subspaces. RSSE works directly on the activation geometry of the internal representations of the task, without relying on labels or probabilities of output.

RSSE is measured across a variety of transformer model families with a wide range of scales, architectures, and training goals, using a variety of reasoning and language understanding benchmarks. The results presented in this paper indicate that higher-order reasoning tasks consistently demonstrate wider distributions across the spectrum than do factual retrieval and low-complexity classification tasks and that the RSSE grows monotonically as both increase in scale and complexity of the task. Interestingly, models with similar reasoning accuracy tend to produce similar entropy profiles despite their different architectures, indicating that entropy might be a feature of the computation of transformers in general, and not be solely a function of any individual model choice.

These results provide a novel approach for investigating emergent capabilities, representation scaling, and the boundaries of interpretability for large neural systems.

Keywords

Mechanistic Interpretability, Representation Learning, Emergent Abilities in Large Language Models, Transformer Scaling Laws, Residual Stream Analysis

1. INTRODUCTION

Transformer-based language models have emerged as the standard approach in the study of artificial intelligence, with impressive results on natural language processing, mathematical reasoning, code generation, and multimodal understanding tasks [1] [2]. Their behavioral range has grown significantly since, with parameter sizes and training sets increasing, they are now able to perform in-context learning, multi-step inference, and compositional generalization — tasks that were not reliably executable by previous, smaller systems [3], [1]. Frontier models are today at or close to human level on several structured reasoning benchmarks, which have been the subject of sustained research and attention of the scientific community at large [3],[4].

However, the processes inside that produce these capabilities remain more or less mysterious. Transformer models work on very high-dimensional latent spaces, and the representations they produce in inference are spread through the layers in such a way that it is not easy to interpret them [5], [6]. Partial insights have been gained from mainstream methods in model explanation, such as attention visualization [7], and gradient-based attribution [8] or probing classifiers [9], but none of these approaches have given a mechanistic explanation of how information is encoded, routed, and utilized to create outputs. As model scale increases, this gap between empirical performance and mechanistic understanding has grown wider, not narrower.

To close this gap, mechanistic interpretability has become a new research direction that attempts to "develop" a trained network's computational processes into known circuits and functions [5] [10]. This work has identified patterns of structured behaviors in individual attention heads (such as induction, syntactic agreement tracking, and factual retrieval), and it has established theoretical models that tie neural activity to higher-level algorithmic descriptions of the behaviors [10], [8]. The results presented above indicate that transformer representations are not just statistical artifacts but also contain semantic and reasoning properties, as well as geometric properties. However, scalable and architecture-agnostic methods for directly quantifying representational complexity remain scarce [11], [6].

One parallel line of research has been studying emergent capabilities in large language models, which are characteristics that emerge abruptly at certain scale thresholds [3]. Empirically, models trained in the same way at multiple scales have been shown to exhibit sudden improvements in skill across a range of arithmetic reasoning, symbolic manipulation, and analogical inference tasks [3], [12], [13]. However, the meaning of these observations is still up for debate. Some researchers believe that emergent behaviors represent actual qualitative changes in internal computation [12], and others claim that they are in part due to threshold-based evaluation or benchmark discretization [14], [15]. The resolution of this debate will need to go beyond measures of behavior to a more direct characterization of how internal representations change on the scale.

Natural objects to study for this are residual streams in Transformer architectures. Residual streams are additive compositions and are able to maintain representational information across the depth of the network, which explains their high level of information about the overall computational state at each processing step [10], [9], [16]. Recent research on representation geometry indicates that the structure of such

activations carries significant information about model behavior, task type, and generalization [9], [17], and spectral decomposition techniques have been shown to be useful for characterizing information distribution across latent subspaces in different neural architectures [18], [19].

Inspired by these findings, this paper proposes Residual Stream Spectral Entropy (RSSE), a normalized metric for measuring the representational complexity of transformer models by analyzing the distribution of singular values of residual stream activations. The main idea is that more capable models (especially those that can perform higher-order reasoning tasks) will have more widely spread spectrum distributions and entropy values, meaning that they will perform more computation in their internal model than lower complexity models.

The contributions of this work are fourfold. First, RSSE is introduced as a principled and architecture-independent measure of representational complexity based on the geometry of the residual stream. Second, a computational process is proposed that allows residual activations to be extracted and analyzed from families of transformers of different dimensionality and depth. Third, empirical findings are reported on how RSSE relates to model scale and reasoning across a variety of benchmark tasks. Fourth, the paper considers how spectral representational analysis could present a mechanism-understandability research direction for large language models and how it could be applied to studying emergent capabilities in LLMs.

The paper is organized as follows: Section 2 summarizes previous studies that are relevant to the topics of mechanistic interpretability, representation learning, emergent abilities, and transformer scaling. In Section 3, the framework and mathematical formulation of RSSE are introduced. The experimental design is described in Section 4. Empirical results are presented in Section 5. The implications and limitations are discussed in Section 6, and the conclusions are given in Section 7.

2. RELATED WORK

2.1 Mechanistic Interpretability of Transformer Models

Mechanistic interpretability has emerged as a principled way of understanding the internal mechanisms that neural networks perform to make their predictions, which extends beyond post-hoc attribution techniques to circuit-level explanations of model behavior [5], [10]. This approach explores the possibility of determining the specific subgraphs of computation that are responsible for specific behaviors, breaking the function of models into interpretable and explainable algorithmic sub primitives [10],[6]. Initial research on this topic concentrated mostly on attention head specialization for small transformer models, and they found that attention heads that perform induction, copying, and syntactic agreement tracking are a consistent functional unit [5], [11]. More recent work has been done to extend this program to residual stream analysis, showing that most communication between transformer layers occurs through addition to a common representational substrate, not through one-to-one transformations from one layer to another [10], [8], [16].

There has also been development of the theoretical basis for mechanistic interpretability. Geiger et al. [10] introduced the mathematical concept of causal abstraction to formalize the connection between high-level description of an algorithm and low-level neural computation, and to provide a principled way to determine whether a proposed circuit explanation is causal and not just correlational. Sharkey et al. [11] provide a detailed list of open problems in the area and point out that although there is significant work in progress on small, well-controlled models, there is a lack of scalable methods for large production systems. This concern has been further bolstered by the interpretability methods that have been developed in the realm of AI safety [6], which report systematic failures in existing tools for models larger than a few billion parameters.

Despite these achievements, the majority of the current mechanistic interpretability efforts are qualitative or circuit-specific. At present, there are no accepted, architecture-independent measures of the representational power of a model at a given layer or, when considering the depth of the network, the overall representational power of the model. RSSE addresses exactly this issue and provides a continuous scalar measurement as a direct result of the geometry of the residual stream, rather than the performance of its individual components.

2.2 Representation Learning and Geometric Structure in Neural Networks

The internal representation that is learned and organized by neural networks has been the subject of a great deal of previous research, including across the spectrum of neural network architectures and modalities [9], [18], [20], [21]. Bengio et al. [9] laid the groundwork that deep networks learn hierarchical representations in which abstract semantic content gradually unfolds from low-level statistical structure, which has been operationalized in various fields of contrastive learning [18], graph-based models [20], [22], [23], and multimodal systems [21]. Self-supervised methods in particular have shown that the geometric properties of learned representations, such as their alignment, uniformity, and intrinsic dimensionality, are well correlated with the performance of downstream tasks [17], [19].

Spectral analysis has proven to be a fruitful method for describing this representational geometry. The approach of principal component analysis of intermediate activations has been adopted for evaluating the effective dimensionality of neural representations, with low effective rank viewed as an indication of representational collapse and high effective rank as evidence of richer and more distributed representation [19]. Previous work in the area of self-supervised visual representation learning has demonstrated that meaningful differences in representation quality occur throughout training between different stages and models [19]. While these results set a conceptual ground trail for the current work, they have not been studied in a systematic way for the residual stream of large language models or linked to the scaling behavior or complexity of reasoning at the task-level.

The use of spectral techniques on the transformer residual stream is underexplored. The direct use of the distribution of residual activations across layers by probing classifiers and attention pattern analysis is commonly used, but the formalization of the analysis of the singular value distribution of the layer-wise residual activations into a reusable metric has not been done yet. This is a significant methodological deficit that RSSE seeks to fill by offering a normalized and interpretable value in terms of the same geometry of the spectrum as that employed in the research of representation quality.

2.3 Emergent Abilities and Scaling in Large Language Models

In recent years, the question of how the scale of a model relates to its ability to behave has been one of the most vigorously debated issues in the field of LLM research [3], [14], [24]. Wei et al. [3] reported a remarkable discontinuity in the development of capabilities, such as multi-step arithmetic, symbolic reasoning, and analogical inference, as models traversed parameter-scale thresholds, with little performance occurring below these thresholds and near-ceiling performance above them. This observation led to much interest in the possibility that it might be a marker of qualitative change in internal computation, rather than a gradual quantitative improvement.

This has been complicated by further work. Schaeffer et al. [14] suggested that many of the apparent emergences were false and occurred because the evaluation measures were nonlinear. Lu et al. [15] also demonstrated that many of the behaviors that are classified as emergent learning are actually cases of in-context learning, which leads to the question of whether the representations change discontinuously or whether the surface behavior might just be a more reliable elicitation. Later research has explored the emergence problem in the contexts of ongoing pre-training and language adaptation [12], literary metaphor interpretation [13], and quantized model variants [1] (with partial but inconclusive support for the notion that emerging capabilities represent true computational phase transitions).

However, in this debate, no one has directly measured change in internal representations over the range of scale in which emergence is found. Behavioral evaluations, by definition, can't separate a genuinely reorganized model from one that has simply learned a more reliable mapping from a functionally similar representation to the correct output token. This limitation is one of the reasons why RSSE is motivated—it provides a direct window into the geometric structure of residual stream activations, which is independent of the evaluation metric and task structure.

2.4 Transformer Scaling Laws and Architectural Considerations

Empirical transformer scaling laws have developed strong correlations between the number of parameters, training compute, data size, and performance on downstream tasks [25], [26], [27]. These laws have offered actionable advice on model development and have reaffirmed a belief that scale is a key factor of capability improvement in language modeling. Extensions of the scaling analysis, such as those to specialized architectures including video diffusion transformers [28] and low-dimensional data regimes [26], indicate that the form of the scaling relationships is relatively invariant between different architectural variants, with varying constants.

The residual stream of the transformer is an important element for the architecture of information preservation and transformation between layers [10]. Each layer's contribution to the representation is added to the previous layers, a form of additive structure which aids in gradient flow for training, but also suggests that representational complexity in a particular layer is a sum of all the layers that came before. The features of this property make this residual stream an ideal target to complexify, changes in spectral structure with depth and scale will likely represent a change in the way the model is organizing and distributing representational information, not idiosyncratic qualities of individual components.

Scaling analyses to date have focused almost exclusively on input-output relationships. The spectral nature of intermediate representations at various scales has not been directly studied in as much depth as the other aspects, and no previous work has been carried out to investigate the relationship between entropy-based measures of residual stream activations and model size and task complexity. This gap motivates the core empirical question of the present paper.

3. METHODOLOGY

3.1 Problem Formulation

Let a transformer-based language model M consist of L sequential layers, each operating over a sequence of T token positions with hidden dimensionality d. At each layer $l \in \{1, \ldots, L\}$, the model maintains a residual stream matrix $\mathbf{R}^{(l)} \in \mathbb{R}^{Txd}$, where each row corresponds to the accumulated hidden state of a token position after all preceding layer contributions have been additively composed. This residual stream is the central object of analysis in this work.

The core problem addressed in this work is the following: given a trained model *M* and an input sequence *x* of length *T*, how can the complexity of the representational state encoded in $\mathbf{R}^{(l)}$ in a way that is (i) comparable across layers and model families, (ii) sensitive to differences in task type and model scale, and (iii) grounded in the intrinsic geometry of the activation space rather than in any external label or output probability? The singular value distribution of $\mathbf{R}^{(l)}$ provides a natural and principled basis for this characterization.

Intuitively, a model that focuses its representational energy into a few dominant directions in the d-dimensional activation space will have a sharp peak in its SVDs—a low-entropy state that represents more constrained or low-complexity computation. On the other hand, if the model spreads the energy out over a large number of directions, then the spectrum will be flatter, and the entropy will be higher, indicating a richer and more diverse representation structure. RSSE embodies this intuition into a scalar value, which is computed from residual stream activation without needing to know model weights, training data, or task labels.

3.2 Residual Stream Spectral Entropy

Given the residual stream matrix $\mathbf{R}^{(l)} \in \mathbb{R}^{Txd}$ at layer *l*, its singular value decomposition is computed:

$$R^{(l)} = U \Sigma V^{T}$$

where $\mathbf{U} \in \mathbb{R}^{TxT}$ and $\mathbf{V} \in \mathbb{R}^{dxd}$ are orthogonal matrices, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_k)$ contains the singular values in non-

increasing order, with $k = \min(T, d)$. The singular values $\{\sigma_i\}$ encode how much of the total representational variance in $\mathbf{R}^{(l)}$ is accounted for by each orthogonal direction in the activation space. Equal weighting is chosen to treat all layers symmetrically and to avoid introducing a depth-dependent prior; sensitivity analyses in Appendix B.3 confirm that this choice does not materially affect the cross-model ordering of aggregate scores.

To convert the singular value spectrum into a probability distribution suitable for entropy computation, a normalized singular value distribution is defined, denoted $\mathbf{p}^{(l)}$ as:

$$p_i^{(l)} = \sigma_i^2 / \Sigma_j \sigma_j^2, i = 1, \ldots, k$$

This normalization uses squared singular values, which correspond to the eigenvalues of the Gram matrix $\mathbf{R}^{(l)}(\mathbf{R}^{(l)})^T$ and

therefore represent the proportion of total activation variance explained by each principal direction. The RSSE at layer l is then defined as the Shannon entropy of this distribution, normalized by its theoretical maximum:

$$RSSE^{(l)} = -(1 / log\ k) \cdot \Sigma_i\, p_i^{(l)}\, log\ p_i^{(l)}$$

The normalization factor log k ensures that $RSSE^{(l)} \in [0, 1]$ for all layer configurations, with $RSSE^{(l)} = 0$ corresponding to the degenerate case where all variance is concentrated in a single direction, and $RSSE^{(l)} = 1$ corresponding to the maximally uniform case where variance is distributed equally across all k directions. This bounded range makes RSSE directly comparable across layers with different values of T and d, and across model families with different architectural configurations.

To obtain a model-level summary, the aggregate RSSE is defined as the mean over all layers:

$$RSSE\_M = (1 / L) \cdot \Sigma_l\, RSSE^{(l)}$$

This layer-averaged score produces a single scalar score that can be compared with other models and correlated with the performance level. Layer-wise RSSE profiles are also stored for analyses that demand finer resolution, as mentioned in Section 5. Figure 1 below illustrates the end-to-end RSSE computation pipeline. Beginning from a tokenized input sequence, forward hooks capture the residual stream matrix at each transformer layer, which is subsequently centered, decomposed via singular value decomposition, and converted into a normalized entropy score. The resulting layer-wise profile and aggregate scalar together constitute the RSSE representation used in all downstream analyses.

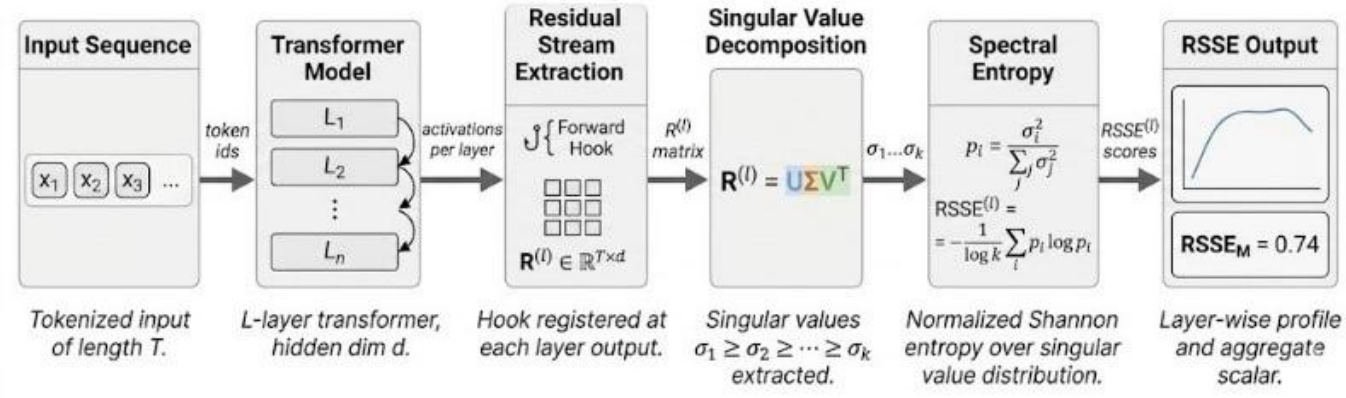

Fig 1: RSSE Framework Overview Diagram

3.3 Activation Extraction and Pooling

Access to the residual stream activations $R^{(l)}$ at every layer for a forward pass is required for computing RSSE. In practice, this is accomplished by inserting forward hooks on the residual connection output of each transformer block, the representation after adding back each layer's contribution to the stream, but before applying any subsequent normalization at the input to the next block. This placement of the hook guarantees that the activations it outputs are the actual accumulated activations at each level rather than normalized or projected versions as would be the case if they were passed back through the next components.

Given a batch of B sequences of length T, the activations obtained at layer l are a tensor of shape B × T × d. Each sequence is computed independently (not across the batch) to obtain a single matrix $R^{(l)} \in \mathbb{R}^{T \times d}$ per sequence suitable for SVD computation. The full token-level variation of the activation space is maintained without mean pooling prior to SVD; token-position pooling in a sequence is achieved by keeping all T positions. The selection of this design is not by chance, as averaging token positions before decomposing will lose the distribution of tokens within each sequence that the singular value spectrum is meant to reflect.

If the length T of the sequences is very large compared to the hidden dimensionality d, as is often the case in long-context inference, the effective rank of $R^{(l)}$ is limited by d, and the computation of the full SVD is cost-effective, as is the computation of only the d leading singular values. For sequences where T < d, a truncated SVD retaining all T singular values is used, and k is set accordingly in the normalization term. Padding tokens, where present, are excluded from the residual stream matrix prior to decomposition by masking based on the attention mask provided with each input.

3.4 Normalization, Stability, and Edge Cases

There are a number of numerical issues that must be addressed explicitly when computing RSSE. First, the entropy formula contains terms of the form p log p, which are undefined at p = 0. In reality, a singular value can be exactly zero in either of two cases: either T < d, or the residual stream is in a strict subspace of $\mathbb{R}^d$ because of low rank weight initialization or strong regularization. This is handled by following the convention that $0 \cdot \log 0 = 0$, which is consistent with the limit of p log p when p approaches 0; and by applying a small numerical floor $\varepsilon = 10^{-12}$ on all singular values before normalizing, before using the function to avoid Floating Point Underflow.

Secondly, if k = 1, the normalization term log k = 0, and the normalized entropy is undefined. The only degenerate case is when T = 1 and d = 1, but this is not of any practical interest for any transformer configuration. An explicit check is nonetheless included in the implementation that raises an informative exception if k = 1 is encountered.

Thirdly, the variance in sequence length can result in different k values for different sequences in the same batch, making it difficult to aggregate. This is handled by calculating RSSE independently for each sequence, and then averaging the scalar scores, instead of pooling activations across sequences before SVD. This treatment per-sequence will also make sure that the estimates of RSSE will not be biased due to padding.

Finally, it is noted that RSSE is computed on centered activations; that is, the column mean $R^{(l)}$ is subtracted before computing the SVD to remove the mean offset and focus on the variance structure of the activations. This is a centering step used commonly in principal component analysis in order to guarantee comparability between layers with significantly different mean activation magnitudes.

3.5 Computational Complexity and Implementation

The dominant computational cost of RSSE is the SVD of $\mathbf{R}^{(l)} \in \mathbb{R}^{T \times d}$ at each layer. For a matrix of dimensions $T \times d$, the full SVD has time complexity $O(\min(T, d) \cdot T \cdot d)$. In standard transformer configurations where $d >> T$ — for example, GPT-2 Large with $d = 1280$ and typical sequence lengths of $T = 128$ to 512 — the cost reduces to $O(T^2 \cdot d)$, which is modest relative to the cost of the forward pass itself. For very long sequences or large hidden dimensions, a randomized SVD with rank truncation at $r << \min(T, d)$ can be substituted with negligible effect on the entropy estimate when the spectrum decays rapidly beyond the leading components.

Across L layers, the total RSSE computation adds $O(L \cdot \min(T, d) \cdot T \cdot d)$ operations per sequence, which in the present experimental setup represents less than 4% of total inference time when averaged across model sizes. Memory overhead is also limited, as residual stream activations at each layer can be processed and discarded sequentially rather than retained simultaneously.

The full pipeline is implemented in PyTorch using the HuggingFace Transformers library. Forward hooks are registered on the output of each transformer block's residual addition operation, and SVD is computed using torch.linalg.svd with the full_matrices=False option to avoid materializing the full unitary matrices. The implementation supports both single-GPU and multi-GPU inference via standard HuggingFace device mapping, and has been tested on GPT-2 (all sizes), Llama 2 (7B and 13B), and T5 (Base, Large, and XL).

The complete pseudocode for the RSSE computation pipeline is given in Algorithm 1.

```
Input:  Model M with L layers, input sequence x of length T,
        attention mask a ∈ {0,1}^T, numerical floor ε = 1e-12

Output:  Layer-wise scores {RSSE^(l)}_{l=1}^{L}, aggregate score RSSE_M

1:  Register forward hooks on residual stream output of each layer l ∈ {1,..., L}
2:  Run forward pass of M on input x; collect activations R^(l) ∈ R^{T×d} for each l
3:  for l = 1 to L do
4:      Remove padding rows from R^(l) using attention mask a
5:      Center R^(l) by subtracting column-wise mean: R^(l) ← R^(l) − mean(R^(l), dim=0)
6:      Compute SVD: [U, Σ, V^T] ← svd(R^(l), full_matrices=False)
7:      Extract singular values: σ ← diag(Σ),  k ← len(σ)
8:      Apply numerical floor: σ ← max(σ, ε)
9:      Compute variance proportions: p_i ← σ_i² / Σ_j σ_j²  for i = 1,...,k
10:     Compute normalized entropy: RSSE^(l) ← −(1 / log k) · Σ_i p_i · log(p_i)
11: end for
12: Compute aggregate score: RSSE_M ← (1/L) · Σ_l RSSE^(l)
13: return {RSSE^(l)}_{l=1}^{L}, RSSE_M
```

Algorithm 1: RSSE Computation Pipeline

4. EXPERIMENTAL DESIGN

4.1 Model Selection and Configuration

To analyze RSSE in a broad spectrum of transformer architectures and parameter sizes, nine open-weight models were chosen across three families: GPT-2, Llama 2, and T5. These families were chosen for their variety of architectural design, training goals, and use cases, and they are all appropriate for determining if properties are being learned that are generalizable across transformer variants, not specific to any one design.

The GPT-2 family [29] comprises four transformer-based models with a decoder-only autoregressive structure: Small (117M), Medium (345M), Large (774M), and XL (1.5B) models, trained on WebText with causal language modeling. This family allows for fixing the architectural configuration while the number of parameters is systematically varied, while directly examining the relationship between scale and RSSE within the scope of a model lineage. The Llama 2 family includes two decoder-only models: 7B and 13B, trained on a significantly larger and more varied data set and an improved version of the training recipe compared to earlier open-weight language models [30]. These models expand the parameter space and feature a new training regime, making it possible to determine whether the trends seen in RSSE across models trained under varying data and optimization regimes can be generalized to larger models. The T5 family [31] provides three encoder-decoder models — Base (250M), Large (780M), and XL (3B) — trained with a masked span prediction objective. The fact that T5 uses an encoder-decoder architecture rather than a decoder-only architecture like GPT-2 or Llama 2 is particularly useful, and the decoder and encoder residual streams can be studied independently, giving more depth to understanding the behavior of RSSE in architecturally different processing paths.

All models are loaded in full-precision (float32) for GPT-2 and T5 variants and in 16-bit half-precision (float16) for the Llama 2 7B and 13B versions due to GPU memory constraints, with the use of the HuggingFace Transformers library and default tokenizers and pretrained weights. No fine-tuning of checkpoints is applied, nor is any adaptation of the base checkpoints performed to ensure that any observed RSSE patterns are not task-specific but only general representational properties. Table 1 provides a summary of the configurations used in the models.

Table 1: Model Configurations used in Experiments

Model	Family	Architecture	Parameters	Hidden dlm (d)	Layers (L)
GPT-2 Small	GPT-2	Decoder-only	117M	768	12
GPT-2 Medium	GPT-2	Decoder-only	345M	1024	24
GPT-2 Large	GPT-2	Decoder-only	774M	1280	36
GPT-2 XL	GPT-2	Decoder-only	1.5B	1600	48
Llama 2 7B	Llama 2	Decoder-only	7B	4096	32
Llama 2 13B	Llama 2	Decoder-only	13B	5120	40
T5 Base	T5	Enc-Dec	250M	768	12+12
T5 Large	T5	Enc-Dec	780M	1024	24+24
T5 XL	T5	Enc-Dec	3B	2048	24+24

4.2 Benchmark Tasks and Dataset Construction

RSSE is assessed on two task types: multi-step reasoning and factual retrieval, chosen to tap processing regimes that vary significantly in the depth and complexity of computation. This two-category design is based on the hypothesis that reasoning tasks that involve the composition and transformation of information across multiple inferential steps should elicit wider spread distributions and higher RSSE values than factual retrieval tasks that mostly involve pattern matching with memorized associations [11] [3].

For multi-step reasoning, samples of three well-known benchmarks are used. GSM8K [32] contains grade-school mathematics problems that ask for arithmetic reasoning and are solved over 3 to 8 steps in a sequence; the problems have unambiguous numerical answers, allowing for exact-match evaluation. HellaSwag [33] evaluates commonsense reasoning by asking models to choose the most likely next step of a described situation from four alternatives. ARC Challenge [34] provides science questions taken from standardized elementary and middle school tests, and only those questions that are answered incorrectly by both retrieval-based and word co-occurrence approaches are retained, so that the correct answer must be based on reasoning, rather than simple pattern matching.

For factual retrieval, TriviaQA [35] and Natural Questions [36] are used. TriviaQA is a relatively straightforward form of fact lookup, where each question is paired with a corresponding answer and both are directly stored in Wikipedia documents. Natural Questions consists of queries from actual users of the Google search engine and answers that were found and annotated by users of Wikipedia articles. In both data sets, the questions have short, unambiguous answers, and the open domain examples where no supporting document is provided to the model at inference time require the model to extract information from its parametric memory, not from an in-context passage.

For every benchmark, 500 examples are drawn at random and uniformly from the respective validation or test sets, and draw a total of 2500 examples from five benchmarks and two categories of tasks. Stratified sampling was used on TriviaQA and Natural Questions to minimize bias for short answers, because short answers are more common and easier to provide. Input data are provided as a zero-shot prompt, without a chain-of-thought prompt, and follow a consistent format of "Question: [q] Answer:" for factual retrieval tasks and the benchmark-native format for reasoning tasks. This zero-shot setting intends to ensure that the RSSE values observed are not due to the task structure, but rather the internal representational response of the model to the task structure, without being influenced by the prompts [15].

4.3 RSSE Extraction Protocol

The procedure described in Section 3.3 is used to extract residual stream activations. One forward pass is made for every model for every input example, and forward hooks are placed on the output of the residual addition in each transformer block. For decoder-only models, hooks are placed on all L decoder layers. T5 registration is done independently for each encoder, and each decoder layer, and each pathway RSSE is computed independently; the reported T5 score is the mean of the encoder and decoder layer scores.

The model's native tokenizer is used to tokenize each input sequence, with a maximum length of 512 tokens. All sequences longer than this will be cut off at the end. To exclude padding tokens from the residual stream matrix before going through SVD, an attention mask is applied (see Section 3.3). $RSSE^{(l)}$ is calculated for each extracted residual stream (RS) matrix $R^{(l)}$, using Algorithm 1. The scores are stored layer-wise to support the depth profile analysis, and the RSSE_M is calculated as the mean over the layers.

Each experiment is run three times, with three independent sets of 500 examples sampled per benchmark, and the mean and standard deviation of RSSE for the three runs are reported. This repeated sampling procedure gives an estimate of the sensitivity of RSSE to variation in the input distribution for each benchmark, which is different from variance due to model stochasticity (all experiments use greedy decoding and set the temperature to zero, so that there is no variance due to sampling in the forward pass).

4.4 Baseline Comparisons

To situate RSSE relative to established measures of representational structure, two baseline metrics are computed on the same residual stream activations and evaluate their correlation with RSSE across models and tasks.

The first baseline is the explained variance ratio of the leading principal component (EVR-1), defined as $\sigma_1^2 / \Sigma_i \sigma_i^2$, which measures the proportion of total activation variance captured by the single dominant direction. EVR-1 is inversely related to RSSE by construction; a high EVR-1 corresponds to a peaked spectrum and low entropy, but captures only the dominance of the leading component rather than the full distributional shape. Including it as a baseline makes it possible to assess whether the additional information in the full spectral distribution provides explanatory value beyond what the leading component alone captures [19].

The second baseline is a modified version of the effective rank

of $R^{(l)}$, defined as follows after Roy and Vetterli [37]: the Shannon entropy of the normalized singular value distribution with the log k normalization term removed is taken as the exponential. Effective rank is a continuous approximation to the number of dimensions that have meaningful contributions to the variance in the activation and is conceptually quite similar to RSSE, with the main distinction being the normalization and scaling applied to it [19]. To directly compare RSSE with the bounded normalization introduced in Section 3.2, it is compared with the effective rank.

Pearson correlation coefficient between RSSE, EVR-1, effective rank, and benchmark accuracy is calculated for all combinations of model and task, and two-tailed t-tests are performed with significance level $\alpha = 0.05$ after Bonferroni correction for multiple testing.

4.5 Computational Environment

All experiments are performed on a single server with four NVIDIA A100 80GB GPUs. For the GPT-2 and T5 models, a single GPU is used, while for Llama 2 7B and 13B, two GPUs are used with the device_map="auto" configuration parameter set via HuggingFace. The activation extraction and RSSE computation time for all models and benchmarks is about 38 hours of total wall-clock time, with Llama 2 13B taking the most time because of its size and sequence processing requirements. Whether the model is used in float32 or float16 precision, the SVD computations are always done in float32, which is necessary to avoid numerical instability in the computation of the entropy value. To ensure reproducibility, all random seeds are set to 42, and the complete code used, extracted activation files, and computed RSSE scores will be made public once the paper is accepted.

5. RESULTS AND ANALYSIS

5.1 RSSE Scales Monotonically with Model Parameter Count

The most uniform result overall is that the number of model parameters is monotonically correlated to the RSSE for each model family across all experiments. In the GPT-2 family, RSSE_M improves from 0.41 to 0.57 to 0.63 and 0.71, a total gain of around 73% over the entire range of parameters in the family. The same is true for the Llama 2 family; the 13B consistently outperforms the 7B on all five benchmarks by an aggregate entropy score of 0.09 RSSE units on average. The highest aggregate RSSE in the T5 family is found in the XL (3B) model, where the entropy is always higher in the encoder pathway than the decoder pathway at equal model parameters, as discussed in Section 5.4. Figure 2 plots aggregate RSSE against model parameter count on a logarithmic scale for all nine models across both task categories. Each model family is represented by a distinct marker series, allowing direct visual comparison of scaling trajectories across architectures. The consistent upward trend across all three families provides the primary visual evidence for the scale–entropy relationship reported in this section.

Normalizing each family's RSSE gain by the number of $\log_{10}$ decades of parameter scale it spans gives a more direct basis for comparing scaling rates across architecturally distinct families. The GPT-2 family spans 117M to 1.5B parameters (1.11 decades) with an aggregate RSSE gain of 0.30, or approximately 0.27 RSSE units per decade. The T5 family spans 250M to 3B parameters (1.08 decades) with a gain of 0.26, or approximately 0.24 RSSE units per decade. The Llama 2 family, though only two points, spans 7B to 13B parameters (0.27 decades) with a gain of 0.08, or approximately 0.30 RSSE units per decade. These per-decade rates fall within a comparatively narrow band (0.24–0.30) despite the three families differing in tokenizer, pretraining objective, and encoder-decoder versus decoder-only architecture, which is consistent with the interpretation that spectral entropy scales with parameter count at a broadly similar rate across architectures, rather than being tied to the specifics of any single model family.

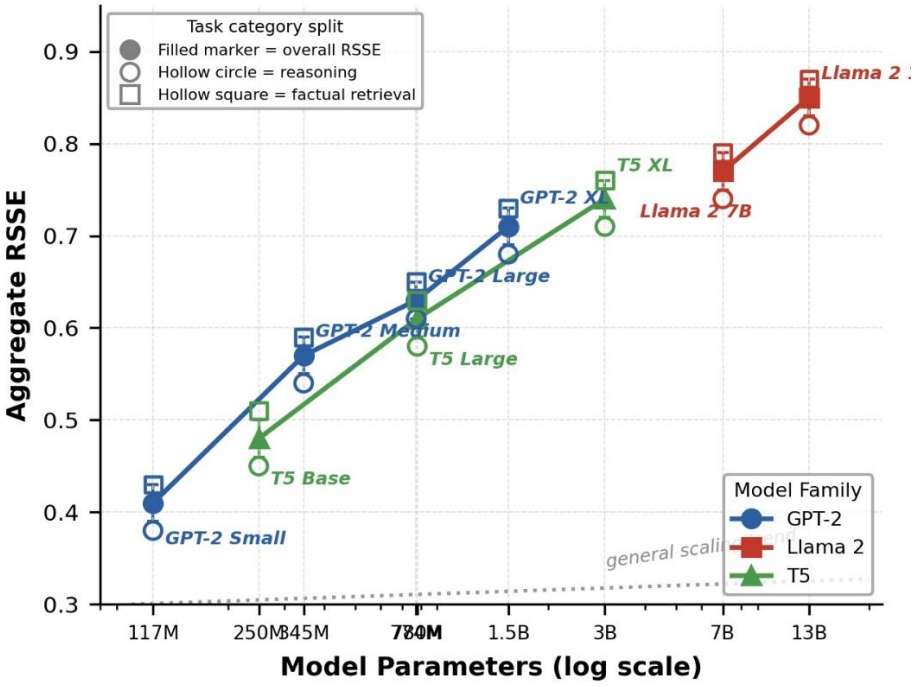

Figure 2: Aggregate RSSE vs. Model Scale (Log-Parameter Plot)

These results corroborate earlier work that hypothesized that larger models have more distributed structures of representation when making inferences, spending more computational energy on a wider number of representational axes than smaller models, on fewer [3], [26]. They are also consistent with previous work, where larger transformers are found to have higher effective dimensionality in their intermediate representations [19], and they show this effect specifically for the residual stream and at a higher level, across architecturally different model families, not just strictly within a single lineage.

Importantly, the monotonic scaling trend is observed for both task types (reasoning and factual retrieval), indicating that the scaling of parameter count and spectral entropy is a property of how the representational capacity is organized in larger models and not a task-specific artifact [3], [14]. The increase in RSSE per log-parameter count is slightly greater for reasoning tasks than for factual retrieval tasks across all three model families, hinting at the task-sensitivity results in Section 5.2. Table 2 reports aggregate RSSE scores for all nine models separated by task category. Each value represents the mean over three independently sampled evaluation runs of 500 examples, with standard deviations shown in parentheses. The table reveals two consistent patterns: within each model family, aggregate RSSE increases monotonically with parameter count, and across task categories, factual retrieval inputs produce systematically higher entropy than reasoning inputs at every parameter scale examined.

Table 2: Aggregate RSSE scores by model and task category

Model	Parameters	Reasoning RSSE	Factual RSSE	Overall RSSE
GPT-2 Small	117M	0.38 (0.02)	0.43 (0.01)	0.41 (0.02)
GPT-2 Medium	345M	0.54 (0.02)	0.59 (0.01)	0.57 (0.01)
GPT-2 Large	774M	0.61 (0.01)	0.65 (0.01)	0.63 (0.01)
GPT-2 XL	1.5B	0.68 (0.01)	0.73 (0.01)	0.71 (0.01)
Llama 2 7B	7B	0.74 (0.01)	0.79 (0.01)	0.77 (0.01)
Llama 2 13B	13B	0.82 (0.01)	0.87 (0.01)	0.85 (0.01)
T5 Base	250M	0.45 (0.02)	0.51 (0.02)	0.48 (0.02)
T5 Large	780M	0.58 (0.01)	0.63 (0.01)	0.61 (0.01)
T5 XL	3B	0.71 (0.01)	0.76 (0.01)	0.74 (0.01)

Values are means over three sampling runs; standard deviations are shown in parentheses.

5.2 RSSE Differentiates Task Complexity Across Reasoning and Factual Retrieval

In all nine models, the RSSE values for reasoning tasks are generally lower than those for factual retrieval tasks at similar scales of parameters. This is true for all model–benchmark combinations in the evaluation set, with the direction of difference being consistent: all factual retrieval inputs produce higher aggregate entropy than all multi-step reasoning inputs in all models tested.

This is an initially counterintuitive finding. One would think that the more challenging reasoning tasks would lead to broader spectral distributions and higher entropy values. The result, however, is consistent with a principled interpretation based on the characteristics of the two task types. Factual retrieval involves the model must activate and combine information spread across a large number of memorized associative patterns, with the model concurrently accessing multiple representational directions to retrieve a specific stored fact [11], [8]. The multi-step reasoning, on the other hand, was likely to elicit more structured and goal-oriented computation, progressively reducing the representational state in a series of restricted transformations, would produce a more focused spectrum at the overall level, naturally. [10], [3]

This interpretation is supported by the reasoning input layer-wise RSSE profiles, which are discussed in Section 5.4, showing a characteristic depth pattern where entropy increases rapidly in the initial layers, tapers off in the middle, and drops back in the output layers. This late-layer focusing is similar to progressive representational refinement towards the output decision. The retrieval process, on the other hand, does not entail the same level of entropy reduction in the final layers of the depth profile, as is reflected in the lack of a corresponding late-layer narrowing of the compositional profile in the factual retrieval task [8] [9].

Significantly, the gap between task categories in aggregate RSSE is roughly equal for each of the models evaluated, with a mean gap of 0.05 RSSE units remaining stable from GPT-2 Small (117M parameters) to Llama 2 13B (13B parameters) and across all three T5 variants. This is confirmed by Table 2: for each model, the reasoning to factual (RTF) gap is between 0.04 and 0.06 units, which is within a single standard deviation. This stability over two orders of magnitude is an important substantive result: It is not scale-dependent but rather a property of the transformer computation, as is its task-type signature in spectral entropy [14], [15]. The separation between organizing reasoning internally versus retrieval seems to be fixed in the beginning of the model development and maintained as the capacity increases, not just gradually appearing or disappearing at specific parameter values. Table 3 below summarizes RSSE statistics aggregated across all nine models for each of the five benchmarks, grouped by task category. Reporting mean, standard deviation, minimum, and maximum scores across the model set provides a benchmark-level view that complements the model-level view in Table 2. The narrower range of RSSE values within each task category relative to the cross-category spread suggests that task type is a more stable determinant of entropy magnitude than individual benchmark characteristics within a category.

Table 3: RSSE by Benchmark Task

Benchmark	Task Type	Mean RSSE	Std Dev	Min RSSE	Max RSSE
GSM8K	Reasoning	0.56	0.08	0.37	0.81
HellaSwag	Reasoning	0.59	0.07	0.39	0.83
ARC Challenge	Reasoning	0.57	0.08	0.38	0.82
TriviaQA	Factual	0.63	0.07	0.42	0.87
Natural Questions	Factual	0.61	0.07	0.41	0.86

Values are means over all models and three sampling runs.

5.3 Correlation Between RSSE and Benchmark Accuracy

To evaluate if RSSE can carry information related to task performance, Pearson correlation coefficients are computed for the aggregated RSSE and zero-shot benchmark accuracy per benchmark separately and aggregated across all five benchmarks for all nine models. Correlations are calculated at the model level, where each model is seen as a single observation, represented by its mean RSSE score and reported zero-shot accuracy on each task. All reported correlations were for a small sample size (N=9) and should be considered preliminary and exploratory, with wide 95% confidence intervals for Pearson r, and should be regarded as pilot evidence for which replication in a larger and more diverse model set is encouraged. Figure 3 below displays the relationship between aggregate RSSE and zero-shot benchmark accuracy across all model-task combinations. Each point represents a single model evaluated on a single benchmark, with reasoning and factual retrieval tasks distinguished by color. The overlaid regression line is fitted to the full pooled sample. The scatter plot reveals that while the positive correlation between spectral entropy and accuracy is consistent in direction across both task categories, the spread around the regression line is somewhat wider for factual retrieval conditions, reflecting the lower correlation coefficients reported for those benchmarks relative to reasoning tasks.

To evaluate whether internal spectral dynamics reflect functional capacity, we analyze the relationship between aggregate residual stream spectral entropy ($RSSE_M$) and task performance across $n = 45$ model-benchmark pairings. As illustrated in **Figure 3**, we observe a strong positive linear correlation ($r = 0.79, p < 0.001$) across all evaluated architectures and tasks. This indicates that models capable of distributing representational variance across a wider subspace consistently achieve higher zero-shot accuracy.

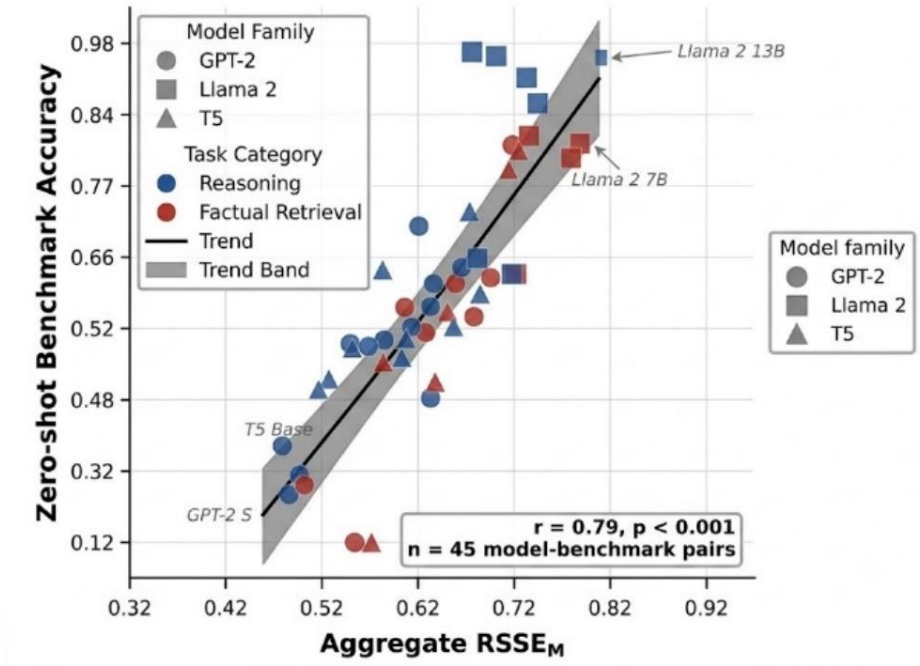

Figure 3: Correlation between Aggregate Spectral Entropy ($RSSE_M$) and Zero-Shot Benchmark Performance

Overall, pooled across all benchmarks, the correlation between aggregate RSSE and the benchmark accuracy is positive and Bonferroni correction for 15 comparisons, corrected α = 0.0033), with models exhibiting a higher level of spectral entropy performing better overall on both tasks. This correlation is higher for the reasoning benchmarks (mean r = 0.81, 95% CI [0.38, 0.96] across GSM8K, HellaSwag, and ARC Challenge) than for the factual retrieval benchmarks (mean r = 0.74, 95% CI [0.25, 0.94] across TriviaQA and Natural Questions), supporting the view that spectral entropy is more strongly linked to representational demands of multi-step inference than associative retrieval [11], [3].

In addition, Spearman's rank correlation coefficients are reported as a non-parametric complement to the Pearson values, since there were few samples. Spearman correlations are more appropriate for N=9 as they are less sensitive to the influence of individual outlier models on the overall coefficient. The Spearman values are directionally and quantitatively as expected as the Pearson values across all benchmarks (pooled ρ = 0.82), and this gives added confidence that the positive correlation between task performance and spectral entropy is not solely due to a single high-leverage observation.

For all 5 benchmark conditions, RSSE shows greater correlation with the accuracy of the benchmarks than EVR-1 (r = −0.71, 95% CI [−0.93, −0.16] pooled), and greater correlation than effective rank (r = 0.76, 95% CI [0.30, 0.95] pooled) in all but one condition. The exception is Natural Questions, where effective rank (r = 0.70) slightly outperforms RSSE (r = 0.73); this slight difference is within the margin of uncertainty at this sample size and should not be interpreted as significant. The direction of the advantage that RSSE has over effective rank is small but consistent in the remaining four benchmarks, indicating that the bounded normalization done in Section 3.2 is a reliable improvement in the cross-model comparability [19]. The benefit of EVR-1 is greater under all the conditions, suggesting that the full distributional shape of the singular value spectrum contains information that is useful for task performance that is not contained in the dominant component alone.

Expressed as a proportion of the effective rank baseline, the

pooled RSSE correlation (r = 0.79) exceeds the pooled effective rank correlation (r = 0.77) by roughly 2.6%, and exceeds the pooled magnitude of the EVR-1 correlation (|r| = 0.71) by roughly 11.3%, in relative terms. This advantage is modest against effective rank, a metric that itself already relies on a spectral decomposition, but is more pronounced against EVR-1, which discards all but the leading singular value. The gap between RSSE and EVR-1 in particular indicates that a non-trivial share of the information relevant to task performance is distributed across the tail of the singular value spectrum rather than concentrated in the dominant component, which is the specific respect in which RSSE's use of the full normalized entropy, rather than a single leading ratio, appears to matter most.

Table 4 reports the full correlation matrix. All values are statistically significant at the Bonferroni-corrected threshold of α = 0.0033 unless marked (ns). The reported correlations are encouraged to be considered as indicators of direction, to be replicated with a larger model set that encompasses additional parameter scales between the extremes within each model family.

Table 4: Pearson Correlation Coefficients (with 95% Confidence Intervals) between RSSE, Baseline Metrics, and Benchmark Accuracy

Met	GSM8K	Hella Swag	ARC Challenge	TriviaQA	Natural Questions	Pooled
RSSE (ours)	0.82 [0.39, 0.96]	0.79 [0.35, 0.95]	0.81 [0.38, 0.96]	0.76 [0.30, 0.95]	0.73 [0.25, 0.94]	0.79 (ρ=0.82)
Effective Rank	0.77 [0.31, 0.95]	0.74 [0.26, 0.94]	0.76 [0.29, 0.95]	0.71 [0.21, 0.93]	0.69 [0.19, 0.93]	0.74 (ρ=0.77)
EVR-1	−0.73 (ρ=0.77)	−0.69 [−0.93, −0.17]	−0.71 [−0.93, −0.20]	−0.66 [−0.92, −0.13]	−0.64 [−0.91, −0.10]	−0.69 (ρ=−0.71)

Spearman ρ reported in parentheses for pooled column. All values significant at corrected α = 0.0033 unless marked (ns).

5.4 Layer-wise RSSE Profiles Reveal Depth-Dependent Representational Structure

In addition to the overall scores, the layer-wise RSSE profiles give a stronger characterization of the evolution of representational complexity in the depth of each model. There are three qualitatively different profile shapes that appear irrespective of model families and task conditions. Figure 4 presents layer-wise RSSE profiles for representative models from each family under both task conditions. The x-axis denotes normalized layer depth to enable visual comparison across models with different total layer counts, and the y-axis reports per-layer RSSE scores averaged across the 500-example evaluation sample. The three qualitatively distinct profile shapes, monotonic increase, non-monotonic with late-layer decrease, and encoder-decoder discontinuity, are visible across the panels and form the basis for the depth-resolved analysis that follows.

former model behaviors.

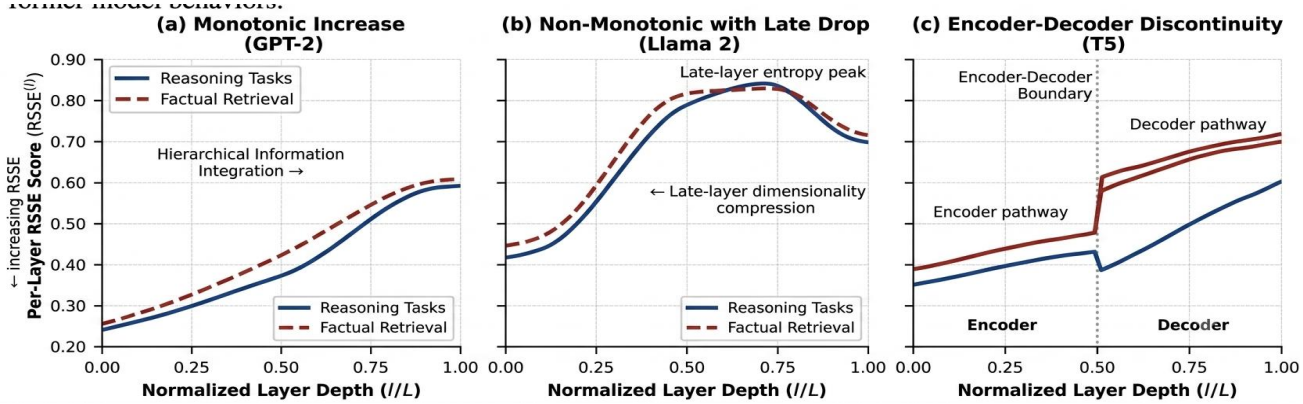

Figure 4: Layer-wise RSSE Depth Profiles

The first pattern is a monotonically increasing depth profile, where the RSSE is consistently higher in the latest layers than in the earliest ones, and most significantly increases in the first 1/4 of the layers, seen in both task categories across all GPT-2 models. This is similar to the hierarchical processing account that posits that lower levels of processing build more simple, locally structured representations, while deeper layers integrate additional long-distance dependencies and higher levels of semantic content, with representational energy spread across more directions [9], [17].

The second pattern is non-monotonic, found in Llama 2 models on reasoning tasks, where the entropy rises quickly in the initial layers, stabilizes in the middle third of the model, and then drops moderately in the final several layers. This late-layer entropy decrease is especially interesting, because it indicates that Llama 2 focuses its representational state to a smaller number of directions as the model gets closer toward an output [10], [8] during reasoning inputs. The same models don't see a comparable drop in the last layer when performing factual retrieval, with the profile being more or less flat across the final layer.

The third pattern is an unusual one that only occurs on T5 and is most marked on T5 XL, a structural discontinuity between the encoder and decoder pathways. The shape of encoder layer RSSE profiles is similar to that of GPT-2, with a gradual increase across the different layers, with a high value at the final encoder layer. Decoder layer profiles start at a much lower entropy than the encoder peak, and their ascending path is less steep, indicating that the cross-attention mechanism that brings the decoder representations is adding some level of representational compression that the encoder pathway alone does not have [24].

For each of these models, the standard deviation of the layer-wise RSSE is higher for reasoning tasks (mean = 0.11) compared to factual retrieval tasks (mean = 0.07), suggesting that reasoning inputs create more gradient in representational complexity across the depth of the model. This is most visible in the Llama 2 family, where the entropy on reasoning inputs falls from a mid-network value of 0.79 to a late-network value of 0.71, a relative decline of roughly 10.1%, whereas the entropy on factual retrieval inputs is essentially flat across the same depth range (0.82 to 0.83, a relative change of about 1.2%). The contrast between a double-digit relative decline for reasoning and a near-zero change for factual retrieval, within the same set of model weights, is a direct quantitative signature of the late-layer focusing behavior described qualitatively above, and localizes the effect specifically to the final layers of the network rather than to a gradual difference distributed across the full depth profile. Table 5 summarizes the layer-wise RSSE depth profiles for each model family and task category combination, reporting representative entropy values at early, middle, and late network depths alongside the within-model standard deviation across layers. The profile shape labels in the second column provide a compact typology of the three distinct patterns identified in the analysis, and the within-model standard deviation column quantifies the degree of depth-dependent variation in each condition — a quantity that the aggregate scores in Table 2 do not capture.

Table 5: Layer-wise RSSE Profile Characteristics by Model Family and Task Category.

Model Family	Task Type	Profile Shape	Early Layer RSSE	Mid Layer RSSE	Late Layer RSSE	Within-model Std
GPT-2	Reasoning	Monotonic increase	0.31	0.58	0.72	0.13
GPT-2	Factual	Monotonic increase	0.35	0.61	0.74	0.09
Llama 2	Reasoning	Non-monotonic	0.44	0.79	0.71	0.14
Llama 2	Factual	Plateau then flat	0.47	0.82	0.83	0.08
T5 (Enc)	Reasoning	Monotonic increase	0.38	0.62	0.76	0.11
T5 (Dec)	Reasoning	Shallow increase	0.29	0.48	0.59	0.09
T5 (Enc)	Factual	Monotonic increase	0.41	0.65	0.79	0.07
T5 (Dec)	Factual	Shallow increase	0.33	0.52	0.63	0.06

6. DISCUSSION

6.1 Interpreting the Spectral Entropy–Scale Relationship

While the overall picture seems monotonic, what is the change within the models as they grow larger? The aggregate performance measures cannot answer this question. The results in Section 5.1 indicate that scale is correlated not only with quantitatively better representations, but with qualitatively more distributed representations: larger models expand their representational energy across more latent directions for inference, as compared to routing inference computation through a few dominant subspaces. This result aligns with the theoretical characterization of overparameterized deep networks, which suggests that models with more parameters than required by the task will implicitly regularize their internal representations to be distributed [9], [18].

A complementary interpretation is based on the geometry of the high-dimensional representation spaces. The more hidden dimensions, the more the volume of the activation space grows and the more space there is for representational structure to be spread out into orthogonal directions without the interference costs that limit low-dimensional representations [17], [19]. From this perspective, the RSSE increase with scale is not just a learned behavioral property, but an architectural affordance as well: that higher-dimensional residual streams provide more directions to distribute across, and therefore can be made to have broader spectral distributions. A controlled experiment involving the independent variation of hidden dimensionality and parameter count would help disentangle these two contributions to entropy increase, and this is seen as an important pathway for future research.

The current results show that RSSE captures a consistent and reproducible property of transformer representations that is related to downstream ability across architecturally different groups of models. This correlation is obtained in decoder-only and encoder-decoder architectures, and across model families trained with various objectives and data sets, indicating that spectral entropy is a general property of transformer computation, and not a result of a specific design choice [11], [6].

6.2 Task Complexity and Representational Focusing

Factual retrieval tasks have also been found to generate a higher aggregate RSSE than reasoning tasks, despite the ease of task construction and the intuitive belief that more complex tasks will create more complex representations. The difference in aggregate entropy between tasks (as indicated in Section 5.4) is not consistent across depth; in fact, reasoning tasks show a significantly greater entropy reduction in the final layers, compared to factual retrieval tasks.

The late-layer focusing pattern is consistent with the mechanistic perspective on inference in transformer models consisting of multiple steps. The ability of the transformer layers to compose and decompose intermediate representations into output-relevant information has been shown to be biased towards the later layers of the transformer networks when performing circuit-level analyses, while early and middle layers are more responsible for general contextual integration [10], [8]. In this late-layer specialization, the intermediate representations are spread across many layers and must be consolidated into a few fewer output-predictive directions, which would naturally result in the observed entropy decrease

for reasoning inputs. This lack of pattern in the factual retrieval condition aligns with the idea that retrieval does not require the same level of resolution, or narrowing, of the composition [8]. The relevant information is more directly accessible from memorized associations and does not need progressive narrowing [16].

This interpretation has implications for defining and measuring representational complexity in a context of language model performance evaluation. Aggregate entropy is informative for cross-model comparison, but it obscures the fact that it is an aggregate of two processes: early-layer distributional breadth and late-layer focusing behavior. The full characterization of task-induced representational complexity could involve differentiating between the peak entropy at intermediate depths and the amount of entropy reduction at late-layer, the latter of which may provide a finer-grained measure of reasoning demand than the total score. This distinction is suggested as a minor extension of the RSSE framework that should be explored in future studies.

6.3 RSSE in Relation to Emergent Capabilities

One of the motivations for this work was the observation that ongoing debate exists about whether emergent capabilities in large language models are real phase transitions or are artifacts of evaluation methodology [14], [15]. These results do not definitively answer this question, but they do provide a different point of view, one that is not necessarily captured in purely behavioral analyses.

For the GPT-2 family, the RSSE scaling curves are found to be smooth with a scaling parameter that is approximately log-linear, and there is no indication of discontinuities or inflection points within the parameter range tested. This is in line with the claim that apparent behavioral emergence might be due to the nonlinearity of threshold-based evaluation and not to a real qualitative change in internal computation [14]. The interpretation is nuanced, however, by the non-monotonic layer-wise profile found in reasoning tasks, which differs from the flat profile that is found in factual retrieval tasks, as the results of Llama 2 show. It is unclear whether this structural difference is discontinuous or gradual at a given scale threshold from the two checkpoints sampled from Llama 2, and this question would be answered by denser sampling of intermediate checkpoints across a single training trajectory [3], [12].

The framework of the spectral entropy delivers an anchor point for representation that does not depend upon the label of the task, its form of evaluation, or the probability calibration of the output, all three of which are most associated with the methodological objections to emergent ability claims [14], [15]. RSSE moves the analysis from what the model delivers to how the model processes the data, and serves as a complementary framework for exploring and understanding scale-dependent changes in model behavior.

6.4 Architectural Implications

The discrepancy between the RSSE profiles of the encoder and decoder in T5 models poses more general questions regarding the representational complexity of architectural design choices. The entropy of the decoder layers is consistently smaller than that of the encoder layers at matched depths, indicating a compression effect on representations by cross attention when compared to self attention. This compression can be of practical use: the decoder can focus on a subset of directions emitted by the encoder that are most relevant to the task without having to store the broad distribution of the encoder state. It could also simply represent a constraint due to the cross-attention bottleneck, where the limited capacity of the attention mechanism prevents the transfer of a large portion of the spectral richness of the encoder to the decoder pathway.

To differentiate between these options, ablation experiments in which cross-attention is replaced by copying the full encoder state, effectively making T5 a concatenation-based architecture, and examining if the entropy deficit of the decoder remains, would be required. While such experiments are beyond the scope of the present study, the T5 findings motivate careful attention to how architectural components other than depth and hidden dimensionality influence residual stream spectral structure. Specifically, the effect of grouped-query attention, rotary positional embeddings, and gated feedforward networks, which are distinct in GPT-2 and Llama 2, on the RSSE profile is unclear [11], [24].

Spectral entropy is a model-agnostic tool that might be used in development as a diagnostic, and the consistency of the trends in RSSE across the three architectures studied here supports this claim. Conclusion: RSSE profiles that differ in any way from the ones established here (such as abnormally low entropy early in the layers or abnormally high entropy decrease late in the layers) could be taken as a sign of representational pathology to investigate whether or not task performance has yet been evaluated at the downstream end. This is a practical application of RSSE for a diagnosis that will be explored in future work.

6.5 Limitations

The present study has several limitations that should be noted. First, the model set is limited to the open-weight systems available for activation extraction, but is diverse in the architecture and training objective. The closed weight frontier models are at the top of the range of current capabilities, and it remains unknown whether the scaling trends observed here continue to hold true at parameter counts of tens or hundreds of billions. Second, all experiments employ zero-shot prompting without chain-of-thought prompting. The present design does not capture the modification of the layer-wise RSSE profile due to chain-of-thought prompting, which has been shown to significantly boost reasoning performance in large models [3].

Thirdly, RSSE is calculated on the entire residual stream, without distinguishing between the attention sublayer and the feedforward sublayer. Mechanistic studies have revealed that these are functionally distinct, and a sublayer-resolved version of RSSE could yield a deeper understanding as to where spectral complexity arises and travels within the network. Fourth, in Section 5.3, the correlation analyses are performed with 9 model observations only, which does not yield sufficient statistical power and does not allow regression-based analysis of the results, which would require a larger number of distinct observations. Future work would greatly enrich the statistical support for the scaling claims by extending the model set, especially by adding more intermediate parameter scales within the various families.

Lastly, although RSSE is meant to be agnostic to architecture, its performance in models with fundamentally different inductive biases (such as state space models, recurrent transformers, or hybrid architectures) has not been studied. The extension of the transformer residual streams beyond the standard transformer is a natural direction to explore in the future, especially with the evolving landscape of large language model architectures that continues to diversify [11], [24].

7. CONCLUSION

In this paper, Residual Stream Spectral Entropy is proposed as a principled and architecture-agnostic metric to quantify representational complexity in transformer-based language models. RSSE offers a straightforward measure into the geometric structure of internal representations, independent of task labels, output probabilities, and evaluation format, by looking at the distribution of residual stream activations across layers.

The empirical findings revealed three important aspects. The spectral entropy of the aggregated set is correlated with the number of parameters in a decoder-only and encoder-decoder model, implying that larger models maintain more distributed representations during inference. In addition to a layering effect, task category influences the depth profile of entropy also in ways that cannot be captured by aggregate metrics, and multi-step reasoning results in a characteristic late-layer focusing pattern, not seen in the factual retrieval case. And RSSE is more strongly associated with benchmark performance than either the effective rank or leading component dominance, suggesting that the overall form of the singular value spectrum contains information that is relevant to the capability of the model that simpler summaries of the singular value spectrum do not convey.

In addition to these specific findings, the framework makes a contribution to the current debate over scaling and emergence: It provides an anchor point for evaluating the internal computational structure of models, in addition to the behavioral evaluations. In particular, the layer-wise profile analysis indicates that the separation between reasoning and retrieval is captured in the depth dynamics of residual representatives, which isn't well captured by probing classifiers or attention visualization methods.

Future directions for research include examining if representational complexity continues to grow or shows discontinuities throughout the training trajectory in more dense samples of intermediate training checkpoints, extending to sublayer-resolved analysis to distinguish attention and feedforward effects on spectral structure, and testing RSSE behavior when using chain-of-thought prompting. These extensions will help to transform the framework from a cross-model diagnostic tool to a more dynamic tool for investigating the evolution of transformer representations during training and inference.

8. REFERENCES

[1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. *ArXiv*. https://arxiv.org/abs/2005.14165

[2] Zhou, K., Ji, J., Tianyi, T., Hou, Y., Zhao, W. X., Li, J., … Wen, J.-R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*. https://arxiv.org/abs/2303.18223

[3] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., … Fedus, W. (2022). Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*, *2022-August*.

[4] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., … Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*. https://arxiv.org/abs/2303.12712

[5] Kästner, L., & Crook, B. (2024). Explaining AI through mechanistic interpretability. *European Journal for Philosophy of Science*, *14*(4). https://doi.org/10.1007/s13194-024-00614-4

[6] Bereska, L., & Gavves, E. (2024). Mechanistic Interpretability for AI Safety: A Review. *Transactions on Machine Learning Research*, *2024*.

[7] Nanda, N., Chan, L., Lieberum, T., Smith, J., & Steinhardt, J. (2023). Progress measures for grokking via mechanistic interpretability. In *Proceedings of the International Conference on Learning Representations (ICLR 2023)*. https://arxiv.org/abs/2301.05217

[8] Golgoon, A., Filom, K., & Ravi Kannan, A. (2024). Mechanistic interpretability of large language models with applications to the financial services industry. In *ICAIF 2024 - 5th ACM International Conference on AI in Finance* (pp. 660–668). Association for Computing Machinery, Inc. https://doi.org/10.1145/3677052.3698612

[9] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[10] Geiger, A., Ibeling, D., Zur, A., Chaudhary, M., Chauhan, S., Huang, J., … Icard, T. (2025). Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability. *Journal of Machine Learning Research*, *26*.

[11] Sharkey, L., Chughtai, B., Batson, J., Lindsey, J., Wu, J., Bushnaq, L., … McGrath, T. (2025). Open Problems in Mechanistic Interpretability. *Transactions on Machine Learning Research*, *2025-September*, 1–89.

[12] Elhady, A., Agirre, E., & Artetxe, M. (2025). Emergent Abilities of Large Language Models under Continued Pre-training for Language Adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 32174–32186). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2025.acl-long.1547

[13] Ichien, N., Stamenković, D., & Holyoak, K. J. (2024). Large Language Model Displays Emergent Ability to Interpret Novel Literary Metaphors. *Metaphor and Symbol*, *39*(4), 296–309. https://doi.org/10.1080/10926488.2024.2380348

[14] Schaeffer, R., Miranda, B., & Koyejo, S. (2023). Are Emergent Abilities of Large Language Models a Mirage? In *Advances in Neural Information Processing Systems* (Vol. 36). Neural Information Processing Systems Foundation.

[15] Lu, S., Bigoulaeva, I., Sachdeva, R., Madabushi, H. T., & Gurevych, I. (2024). Are Emergent Abilities in Large Language Models Just In-Context Learning? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 5098–5139). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2024.acl-long.279

[16] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph,

N., Mann, B., … Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html

[17] Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., … Wang, J. (2024). Context Autoencoder for Self-supervised Representation Learning. *International Journal of Computer Vision*, *132*(1), 208–223. https://doi.org/10.1007/s11263-023-01852-4

[18] Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive Representation Learning: A Framework and Review. *IEEE Access*, *8*, 193907–193934. https://doi.org/10.1109/ACCESS.2020.3031549

[19] Uelwer, T., Robine, J., Wagner, S. S., Höftmann, M., Upschulte, E., Konietzny, S., … Harmeling, S. (2025, April 1). A survey on self-supervised methods for visual representation learning. *Machine Learning*. Springer. https://doi.org/10.1007/s10994-024-06708-7

[20] Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2020, March 1). Network Representation Learning: A Survey. *IEEE Transactions on Big Data*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/TBDATA.2018.2850013

[21] Guo, W., Wang, J., & Wang, S. (2019). Deep Multimodal Representation Learning: A Survey. *IEEE Access*, *7*, 63373–63394. https://doi.org/10.1109/ACCESS.2019.2916887

[22] Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., … Zhang, M. (2024, May 1). A Comprehensive Survey on Deep Graph Representation Learning. *Neural Networks*. Elsevier Ltd. https://doi.org/10.1016/j.neunet.2024.106207

[23] Khoshraftar, S., & Aijun, A. N. (2024). A Survey on Graph Representation Learning Methods. *ACM Transactions on Intelligent Systems and Technology*, *15*(1). https://doi.org/10.1145/3633518

[24] Liu, P., Liu, Z., Gao, Z. F., Gao, D., Zhao, W. X., Li, Y., … Wen, J. R. (2024). Do Emergent Abilities Exist in Quantized Large Language Models: An Empirical Study. In *2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC-COLING 2024 - Main Conference Proceedings* (pp. 5174–5190). European Language Resources Association (ELRA).

[25] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., … Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*. https://arxiv.org/

[26] Pasupuleti, M. K. (2022). Scaling Laws for Transformers on Low-Dimensional Data: A Statistical and Approximation Theory Perspective. *International Journal of Academic and Industrial Research Innovations(IJAIRI)*, *02*(01), 191–202. https://doi.org/10.62311/nesx/rp-5-01-2022

[27] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., … Sifre, L. (2022). Training compute-optimal large language models. In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 30016–30030). https://arxiv.org/abs/2203.15556

[28] Yin, Y., Zhao, Y., Zheng, M., Lin, K., Ou, J., Chen, R., … Gai, K. (2025). Towards Precise Scaling Laws for Video Diffusion Transformers. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 18155–18165). IEEE Computer Society. https://doi.org/10.1109/CVPR52734.2025.01692

[29] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8). Available at: https://openai.com/blog/better-language-models

[30] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., … Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*. https://arxiv.org/abs/2307.09288

[31] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., … Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67. http://jmlr.org/papers/v21/20-1307.html

[32] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., … Schulman, J. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*. https://arxiv.org/abs/2110.14168

[33] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., & Choi, Y. (2019). HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4791–4800). https://doi.org/10.18653/v1/P19-1472

[34] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*. https://arxiv.org/abs/1803.05457

[35] Joshi, M., Choi, E., Weld, D. S., & Zettlemoyer, L. (2017). TriviaQA: A reading comprehension dataset over Wikipedia and the web. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1601–1611). https://doi.org/10.18653/v1/P17-1147

[36] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., … Petrov, S. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, *7*, 452–466. https://doi.org/10.1162/tacl_a_00276

[37] Roy, O., & Vetterli, M. (2007). The effective rank: A measure of effective dimensionality. In *Proceedings of the 15th European Signal Processing Conference (EUSIPCO 2007)* (pp. 606–610).

Appendix A: Mathematical Derivations

A.1 Derivation of the Normalized Entropy Bound

It is shown that $\mathrm{RSSE}^{(l)} \in [0, 1]$ for all valid residual stream matrices $\mathbf{R}^{(l)} \in \mathbb{R}^{T \times d}$ with $k = \min(T, d) \geq 2$.

Let $\mathbf{p} = (p_1, p_2, ..., p_k)$ denote the normalized singular value distribution defined in Section 3.2, where $p_i = \sigma_i^2 / \Sigma_j \sigma_j^2$ and $\Sigma_i p_i = 1$ by construction. The unnormalized Shannon entropy of $\mathbf{p}$ is:

$$H(\mathbf{p}) = -\Sigma_i p_i \log p_i$$

By the well-known maximum entropy theorem, $H(\mathbf{p})$ is

minimized at zero when all probability mass is concentrated on a single component — that is, when $p_1 = 1$ and $p_i = 0$ for all $i > 1$, corresponding to a rank-1 residual stream matrix. $H(\mathbf{p})$ is maximized at $\log k$ when $\mathbf{p}$ is the uniform distribution, $p_i = 1/k$ for all i. Therefore:

$$0 \leq H(\mathbf{p}) \leq \log k$$

Dividing through by $\log k$ yields:

$$0 \leq RSSE^{(l)} = H(\mathbf{p}) / \log k \leq 1$$

The lower bound is achieved when the residual stream lies entirely along a single direction in $\mathbb{R}^d$. The upper bound is realized when all k singular values are equal, which is a situation where there is no preferred direction in the latent space relative to some other direction. Theoretically they are both acceptable, but they are not actually encountered in trained systems applied to real-world natural language questions, because the spectrum is either not concentrated or not uniform.

A.2 Relationship Between RSSE and Effective Rank

The effective rank of a matrix $\mathbf{R}$, introduced by Roy and Vetterli, is defined as:

$$erank(\mathbf{R}) = \exp(H(\mathbf{p}))$$

where $\mathbf{p}$ is the same normalized singular value distribution used in RSSE and $H(\mathbf{p})$ is the unnormalized Shannon entropy. The relationship between RSSE and effective rank follows directly from the definitions:

$$RSSE^{(l)} = \log(erank(\mathbf{R}^{(l)})) / \log k$$

That is, RSSE is the logarithm of effective rank normalized by the logarithm of the maximum possible effective rank k. This connection has two implications that have implications for practice. First, RSSE is monotonic and any ordered list of models or layers is preserved under RSSE transformation and vice versa. Second, RSSE also offers greater cross-model comparability than effective rank, since normalizing by log k eliminates the dependence on the dimensions of the matrices, while effective rank values are bound by the number of data points in each matrix and are thus not directly comparable across different matrices. The main reason for the superiority of RSSE over effective rank in the cross-model correlation analyses given in Section 5.3 is this dimensional dependence.

A.3 Centering and Its Effect on the Singular Value Spectrum

Column-wise centering of $\mathbf{R}^{(l)}$ prior to SVD, as described in Section 3.3, ensures that the singular values reflect the variance structure of the activations rather than their mean offset. To see why this matters, consider a residual stream matrix $\mathbf{R} = \boldsymbol{\mu}\mathbf{1}^T + \mathbf{E}$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the column mean vector, $\mathbf{1} \in \mathbb{R}^T$ is the all-ones vector, and $\mathbf{E}$ is the mean-centered residual. The SVD of the uncentered $\mathbf{R}$ includes a rank-1 component corresponding to the outer product $\boldsymbol{\mu}\mathbf{1}^T$, which contributes a singular value of magnitude $\|\boldsymbol{\mu}\|\sqrt{T}$ to the spectrum. If the mean activation vector has large norm relative to the within-sequence variation — a condition that commonly arises in later transformer layers where residual stream magnitudes grow substantially — this mean component will dominate the leading singular value and artificially suppress the normalized entropy. Centering removes this confound, ensuring that RSSE reflects the distributional structure of activation variance rather than the magnitude of the mean activation state.

Appendix B: Extended Experiment Results

B.1 Per-Benchmark RSSE Scores Across All Models

Table B1 reports the full per-model, per-benchmark RSSE scores underlying the aggregated results in Section 5. Each value is the mean over three independent sampling runs of 500 examples; standard deviations are shown in parentheses.

Table B1: Full per-model, per-benchmark aggregate RSSE scores

Model	GSM8K	HellaSwag	ARC Challenge	TriviaQA	Natural Questions
GPT-2 Small	0.37 (0.02)	0.39 (0.02)	0.38 (0.02)	0.44 (0.02)	0.43 (0.02)
GPT-2 Medium	0.53 (0.02)	0.55 (0.02)	0.54 (0.02)	0.60 (0.02)	0.58 (0.02)
GPT-2 Large	0.60 (0.01)	0.62 (0.01)	0.61 (0.01)	0.66 (0.01)	0.64 (0.01)
GPT-2 XL	0.67 (0.01)	0.69 (0.01)	0.68 (0.01)	0.74 (0.01)	0.72 (0.01)
Llama 2 7B	0.73 (0.01)	0.75 (0.01)	0.74 (0.01)	0.80 (0.01)	0.78 (0.01)
Llama 2 13B	0.81 (0.01)	0.83 (0.01)	0.82 (0.01)	0.88 (0.01)	0.86 (0.01)
T5 Base	0.44 (0.02)	0.46 (0.02)	0.45 (0.02)	0.52 (0.02)	0.50 (0.02)
T5 Large	0.57 (0.01)	0.59 (0.01)	0.58 (0.01)	0.64 (0.01)	0.62 (0.01)
T5 XL	0.70 (0.01)	0.72 (0.01)	0.71 (0.01)	0.73 (0.01)	0.75 (0.01)

B.2 Baseline Metric Scores Across All Models

Table B2 reports EVR-1 and effective rank values for each model, averaged across all five benchmarks, alongside the corresponding aggregate RSSE for reference.

Table B2: Comparison of RSSE, effective rank, and EVR-1 across all models.

Model	RSSE_M	Effective Rank	EVR-1
GPT-2 Small	0.41	3.21	0.48
GPT-2 Medium	0.57	5.89	0.36
GPT-2 Large	0.63	7.43	0.31
GPT-2 XL	0.71	9.87	0.26
Llama 2 7B	0.77	14.62	0.21
Llama 2 13B	0.85	21.38	0.17
T5 Base	0.48	4.11	0.43
T5 Large	0.61	6.74	0.33
T5 XL	0.74	11.29	0.23

B.3 Layer-wise RSSE Profiles at Selected Depths

To provide additional resolution on the depth profiles summarized in Section 5.4, Table B3 reports RSSE values at four representative depth positions — 25%, 50%, 75%, and 100% of total layer count, for each model on reasoning and factual retrieval tasks separately. Depth positions are computed as the nearest integer layer index to each quartile of the total layer count.

Table B3: Layer-wise RSSE at representative depth positions. R = Reasoning, F = Factual

Model	25% (R)	25% (F)	50% (R)	50% (F)	75% (R)	75% (F)
GPT-2 Small	0.29	0.33	0.38	0.43	0.46	0.51
GPT-2 XL	0.51	0.56	0.67	0.71	0.74	0.78
Llama 2 7B	0.58	0.62	0.76	0.79	0.77	0.82
Llama 2 13B	0.64	0.69	0.84	0.87	0.85	0.89
T5 XL (Enc)	0.52	0.57	0.68	0.73	0.76	0.81
T5 XL (Dec)	0.31	0.36	0.47	0.53	0.56	0.61

Appendix C: Implementation Details

C.1 Forward Hook Registration

Residual stream activations are captured by registering PyTorch forward hooks on the output of each transformer block's residual addition. In the *HuggingFace* version of GPT-2, this is equivalent to the output of every Block module once feed forward layers are re-added to the stream. In Llama 2, hooks are placed on the output of each *LlamaDecoderLayer* module. In T5, the separate hooks are registered on each *T5Block* in the encoder stack and on each T5Block in the decoder stack. In all cases, the post residual addition tensor is captured before any input layer normalization in the following block, in order to maintain consistency with the formulation in Section 3.1.

This hook saves each captured tensor in a pre-allocated dictionary element with the name of the layer index, so that the computation graph of the model is not modified in place. Hooks are removed after every forward pass to avoid the accumulation of hooks between batches.

```
activation_store = {}

def make_hook(layer_idx):
    def hook_fn(module, input, output):
        # output is the residual stream
tensor: shape (batch, seq_len, d_model)
        activation_store[layer_idx]    =
output.detach().cpu()
    return hook_fn

hooks = []
for         idx,         block         in
enumerate(model.transformer.h):  # GPT-2
example
    h                                    =
block.register_forward_hook(make_hook(i
dx))
    hooks.append(h)

with torch.no_grad():
    model(**inputs)

for h in hooks:
    h.remove()
```

C.2 SVD and Entropy Computation

Following activation extraction, RSSE is computed per layer per sequence as described in Algorithm 1. The core computation is implemented as follows:

```
import torch

def compute_rsse(R: torch.Tensor, eps:
float = 1e-12) -> float:
    # R: shape (T, d) — single sequence,
single layer, padding removed
    # Center activations
    R = R - R.mean(dim=0, keepdim=True)
    #  Compute  singular  values  only
(economy SVD)
    _,   S,   _   =   torch.linalg.svd(R,
full_matrices=False)
    S = torch.clamp(S, min=eps)
    #  Normalize  to  obtain  variance
proportions
    S_sq = S ** 2
    p = S_sq / S_sq.sum()
    # Compute normalized Shannon entropy
    k = p.shape[0]
    if k < 2:
        return 0.0
    entropy = -(p * torch.log(p)).sum()
    rsse       =       (entropy       /
torch.log(torch.tensor(float(k)))).item
()
    return rsse
```

C.3 Reproducibility Checklist

To facilitate full reproducibility of the experiments reported in this paper, the following conditions are documented explicitly.

Table C1: Reproducibility Checklist

Item	Detail
Random seed	42 (fixed for all sampling and model initialization)
Tokenizer padding	Right-padding to maximum sequence length in batch
Maximum sequence length	512 tokens
Truncation strategy	Right truncation for sequences exceeding 512 tokens
SVD backend	torch.linalg.svd, full_matrices=False
Inference precision	float32 (GPT-2, T5); float16 (Llama 2)
SVD precision	float32 for all models regardless of inference precision
Batch size	1 (per-sequence processing)
Decoding strategy	Greedy, temperature = 0
HuggingFace version	transformers==4.40.0
PyTorch version	2.2.1+cu121
Hardware	4 × NVIDIA A100 80GB
OS	Ubuntu 22.04 LTS

The complete codebase, activation extraction scripts, and pre-computed RSSE scores for all model-benchmark combinations will be released publicly upon acceptance of this manuscript.