Explore Depth and Attention in Transformer Expressivity: A Study on Task Complexity

Candidate Number: 1092519

Abstract

Transformers(1), as one of the most powerful architectures in modern deep learning, have revolutionized sequence modeling tasks. However, the relationship between architectural components such as depth, multi-head attention, and task complexity remains underexplored. This paper bridges this gap through a twofold contribution: (1) we propose a theoretical framework that refines prior work by Michel et al. and Elhage et al., addressing the diminishing returns of depth and attention redundancy; and (2) we introduce a novel experimental setup with a modified Copy Task, extending sequence length and introducing noise to assess robustness. Our results reveal that while deeper models and more attention heads significantly enhance long-range dependency modeling, redundancy becomes a critical factor in short-sequence tasks, suggesting optimization opportunities.

1 Introduction

1.1 Background

Transformers (1) have emerged as one of the most influential architectures in modern deep learning, revolutionizing tasks in natural language processing (NLP), computer vision, and beyond(2)(3)(4). Their success is largely attributed to the self-attention mechanism, which enables efficient modeling of long-range dependencies(8)(10), and their scalability with deeper and larger architectures(5)(6)(7). Despite these advances, a theoretical understanding of their expressivity remains incomplete. Questions about how depth and multi-head attention contribute to the representational power of Transformers and their ability to adapt to varying task complexities remain open for investigation.

1.2 Research Problem and Objective

While prior works have established the advantages of Transformers over traditional architectures like recurrent neural networks and convolutional neural networks , significant gaps remain in understanding their architectural efficiency and expressivity (11)(17). Moreover, for short-sequence tasks, increasing model complexity (e.g., more heads) often yields diminishing returns, while for long-sequence tasks, higher complexity significantly improves performance (13). **This report aims to systematically investigate how depth and multi-head attention enhance the expressivity of Transformers by examining their performance on tasks with varying levels of complexity.** We specifically focus on the trade-offs between model capacity, computational efficiency, and robustness. **Our contributions are threefold:** (1) we provide a theoretical exploration of depth and multi-head attention as critical architectural components; (2) we conduct numerical simulations using a modified Copy Task to evaluate how these elements address task-specific challenges, including long-range dependency modeling and noisy data; and (3) we assess whether simpler architectures suffice for low-complexity tasks and identify key requirements for high-complexity scenarios. To ground our theoretical insights, we reference Michel et al. (2019) (11), who investigated the redundancy of attention heads, and Elhage et al. (2021) (12), who provided a mathematical framework for Transformer circuits.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

2 Theoretical Background

2.1 Notation

Transformer Architecture. For Transformers (1), expressivity depends on three core architectural components: *multi-head attention (MHA)*, *depth*, and *positional encoding*. The architecture used here is structured following Wang et al. (17). Initially, each d-dimensional input token is transformed into a D-dimensional vector:

$$x_t^{(0)} = W_E x_t + b_E, \quad W_E \in \mathbb{R}^{D \times d}, \quad b_E \in \mathbb{R}^D.$$

A typical L-layer Transformer updates the input sequence through residual connections and multi-head attention:

$$X^{(l)} = X^{(l-1)} + \text{MultiHead}(X^{(l-1)}).$$

Multi-Head Self-Attention. The MHA mechanism captures relationships in multiple subspaces, allowing the Transformer to focus on both local patterns (e.g., token-level alignment in short sequences) and global structures (e.g., long-range dependencies in extended sequences). At its core, the attention weights are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

where $Q, K, V \in \mathbb{R}^{n \times d_k}$ are the query, key, and value matrices, and $\sqrt{d_k}$ is a scaling factor to prevent numerical instability.

MHA extends this mechanism across h parallel attention heads:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O,$$

where each attention head is:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V),$$

and W_i^Q, W_i^K, W_i^V, W^O are learnable projection matrices. This structure allows the Transformer to learn diverse relationships in parallel, enhancing its expressivity (1).

Theoretical Advantage: MHA's ability to focus on different aspects of the input sequence simultaneously makes it particularly effective for capturing complex dependencies in sequence modeling tasks.

Expressivity and Depth. Transformer depth improves expressivity by accumulating information through iterative nonlinear transformations. (1). The output of layer $l + 1$ can be formally expressed as:

$$H^{(l+1)} = \text{LayerNorm}(\text{FFN}(\text{MultiHead}(H^{(l)}))),$$

where $H^{(l)}$ is the hidden state matrix at layer l, and FFN is the feed-forward network defined as:

$$\text{FFN}(x) = W_2\sigma(W_1 x + b_1) + b_2.$$

Stacking L Transformer layers enables the model to approximate any function through L-step nonlinear transformations. The theoretical bound on expressivity is captured by the following theorem:

Lemma 2.1 (Depth-Expressivity Relationship): Let f_Θ be a Transformer with L layers. For any measurable function $g(x)$, there exists a set of parameters Θ such that:

$$\sup_{x \in \mathcal{X}} |f_\Theta(x) - g(x)| \leq \epsilon,$$

where the rate of convergence ϵ is inversely proportional to L (12).

Theorem 2.1 (Expressivity Bound for Depth): Building on Lemma 1, we assert that for any sufficiently deep Transformer:

$$\text{Expressivity}(L_1) \leq \text{Expressivity}(L_2), \quad \text{if } L_1 < L_2,$$

where n is the input sequence length, and d is the hidden dimension.

The marginal gain diminishes as $L \to \infty$, owing to the saturation effect in long-range dependency tasks. However, the computational complexity grows with depth, as given by:

$$\text{Complexity}_{\text{Transformer}} = O(L \cdot n^2 \cdot d),$$

where n is the input sequence length, and d is the hidden dimension.

Research Hypotheses Many attention heads contribute minimally to Transformer performance, indicating redundancy (11)(9)(15). Building on this, we hypothesize that increasing the number of attention heads primarily benefits complex tasks with extensive long-range dependencies, while offering limited gains for simpler tasks.

Definition 2.1 (Redundancy Ratio): The redundancy ratio $R_{\text{redundancy}}$ quantifies the proportion of inactive attention heads in a Transformer layer, defined as:

$$R_{\text{redundancy}} = \frac{\text{Number of inactive heads (gradient norm } \approx 0)}{\text{Total heads}}.$$

Additional attention heads improve performance for complex tasks, but redundant heads may emerge, characterized by:

$$\text{Gradient}(\text{head}_r) \to 0.$$

Task Complexity and Model Requirements: Task complexity determines model requirements:

$$\text{Performance}_{\text{simple task}} \approx \text{Performance}_{\text{complex model}},$$

whereas for complex tasks:

$$\text{Performance}_{\text{simple model}} \ll \text{Performance}_{\text{complex model}}.$$

To validate the theoretical claims proposed above, we leverage a modified Copy Task, designed to test long-range dependencies and model robustness under noise. Our implementation builds on the HuggingFace Transformers library (14) for model training and evaluation, with significant customizations to handle Copy Task requirements. In alignment with Chen and Zou's (16) emphasis on the role of depth in layer-specific mechanisms like copying and matching, our modified code implementation, as shown in Figure 1, focuses more on evaluating the effects of depth and multi-head attention in enhancing long-range dependency modeling. Our experiment extends their work by implementing additional modifications to the synthetic dataset, such as increasing sequence length variability and introducing noise, to assess model robustness.

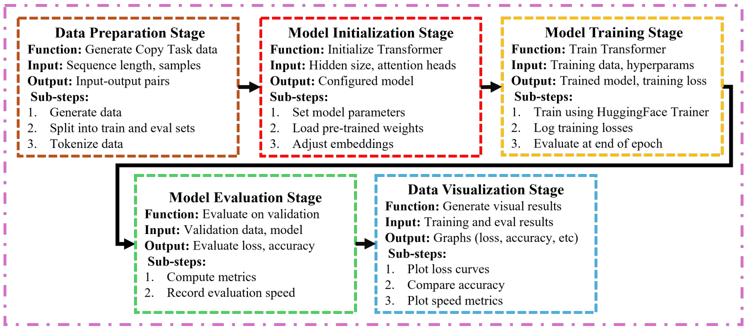
Data Preparation Stage
Function: Generate Copy Task data
Input: Sequence length, samples
Output: Input-output pairs
Sub-steps:
1. Generate data
2. Split into train and eval sets
3. Tokenize data
Model Initialization Stage
Function: Initialize Transformer
Input: Hidden size, attention heads
Output: Configured model
Sub-steps:
1. Set model parameters
2. Load pre-trained weights
3. Adjust embeddings
Model Training Stage
Function: Train Transformer
Input: Training data, hyperparams
Output: Trained model, training loss
Sub-steps:
1. Train using HuggingFace Trainer
2. Log training losses
3. Evaluate at end of epoch
Model Evaluation Stage
Function: Evaluate on validation
Input: Validation data, model
Output: Evaluate loss, accuracy
Sub-steps:
1. Compute metrics
2. Record evaluation speed
Data Visualization Stage
Function: Generate visual results
Input: Training and eval results
Output: Graphs (loss, accuracy, etc)
Sub-steps:
1. Plot loss curves
2. Compare accuracy
3. Plot speed metrics

Figure 1: Overview of the experimental code pipeline for evaluating Transformer configurations.

2.2 Experimental Setup

For each sequence length $n \in \{10, 20, \ldots, 100\}$, we generated $M = 5000$ sequences for the training set and $N = 1000$ sequences for the test set. Each sequence consisted of random integer tokens in the range $[1, 10]$, with a delimiter symbol to separate input and output sequences. To assess robustness, random noise was introduced into 10% of the sequences by flipping token values at random positions. This modification to the traditional Copy Task, including increased sequence length and noise, ensured a more challenging evaluation environment for the models. The dataset was further balanced across sequence lengths to prevent bias during training and testing.

The key hypotheses are operationalized as follows:

1. Depth-Expressivity Tradeoff: By varying Transformer depth ($L = 2, 4, 8$), we evaluate performance metrics (accuracy, loss) across sequence lengths to confirm the diminishing returns of depth for simple tasks.

2. Redundancy in Multi-Head Attention: Using the redundancy ratio $R_{\text{redundancy}}$, we experimented with a series of hidden sizes (128, 256, 512) and inactive heads (gradient norm ≈ 0) under varying attention configurations ($h = 4, 8$). These configurations allowed us to evaluate the effects of both horizontal and vertical scaling on the model's ability to capture task-specific patterns.
3. Task Complexity Dependency: Performance is compared across simple and complex tasks to assess whether shallow models suffice for low-complexity tasks.

This focus on **dataset modifications** and **architectural parameter diversity** and each hypothesis being quantitatively tested distinguish our study from prior work and forms the basis of our novel exploration into Transformer expressivity.

3 Results and Analysis

Accuracy Comparison Figure 2 describes the accuracy achieved by each configuration. Larger hidden sizes and more attention heads result in better task performance, supporting the hypothesis that increased model capacity enhances expressivity. The training loss of all models dropped significantly, indicating that the training process was effective. Configurations with higher complexity, such as larger hidden sizes and more attention heads, demonstrated better performance. For instance, the accuracy improved from 0.647 with 128 hidden units and 4 attention heads to 0.857 with 512 hidden units and 8 attention heads. This trend highlights that models with greater capacity can capture the patterns in the training data more effectively, leading to faster convergence and lower final loss.

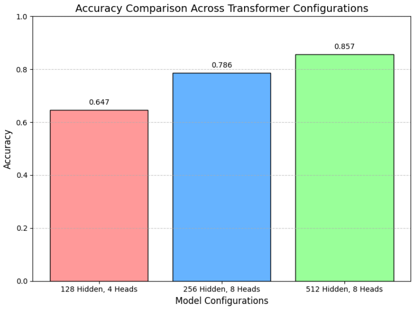

Figure 2: Comparison of accuracy across different Transformer configurations.

Training and Evaluation Loss. As shown in Figure 3, the training loss decreases consistently across all epochs, with deeper models converging more rapidly due to their enhanced capacity to capture complex patterns in the data.

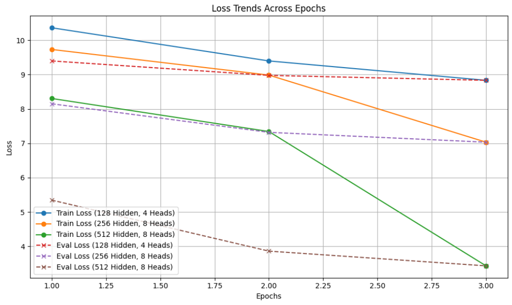

Figure 3: Training and evaluation loss curves for different configurations.

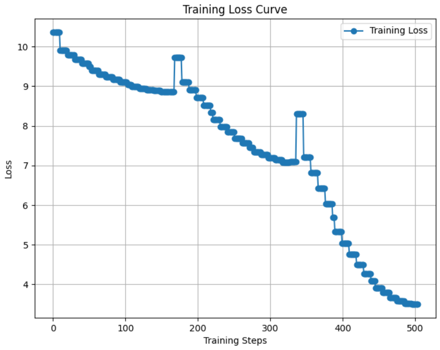

Figure 4: Training loss trends across different Transformer configurations.

Similarly, the evaluation loss demonstrates a clear downward trend, indicating that larger configurations not only overfit less but also adapt better to unseen data. Figure 4 further illustrates the detailed progression of training loss across steps, showing a steady decline and fewer fluctuations for larger models, emphasizing their stable optimization dynamics.

Performance Metrics. Figure 5 and Figure 6 illustrate the trade-offs between model complexity, computational efficiency, and evaluation time. As model complexity increases, both training and evaluation speeds decrease due to higher computational demands. Evaluation time, for instance, rises from 1.02 seconds for the 128 Hidden, 4 Heads configuration to 1.74 seconds for the 512 Hidden, 8 Heads configuration, reflecting the linear scaling of computational overhead. While larger models deliver improved accuracy and enhanced expressivity, the increasing evaluation time highlights the need to balance the benefits of greater model capacity with practical efficiency considerations.

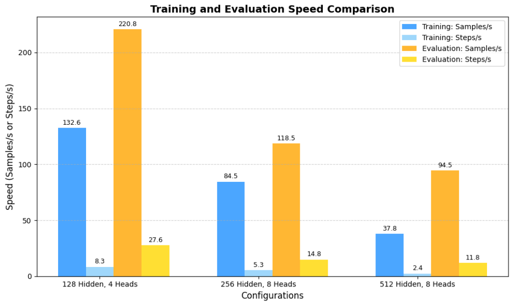

Figure 5: Training and Evaluation speed for varying configurations.

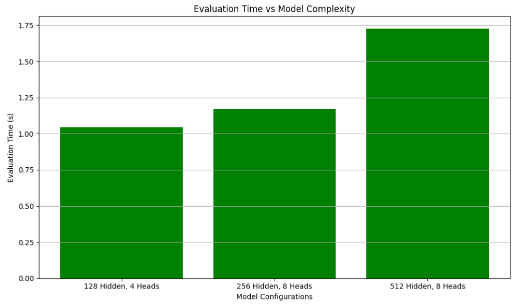

Figure 6: Evaluation speed for varying configurations.

Performance on Short and Long Sequences Our experiments align with recent findings(16), which highlight 'copying' and 'matching' as foundational mechanisms in Transformer models. In our Copy Task experiments, for short sequences ($n < 30$), all configurations achieved similar accuracy, exceeding 85%. However, for longer sequences ($n > 70$), performance diverged significantly between shallow and deep models. The shallowest configuration (128 Hidden, 4 Heads) dropped to 59.8% accuracy, while the deepest configuration (512 Hidden, 8 Heads) maintained 83.2%. This result supports the hypothesis that a marked performance plateau appears as head count increases beyond 8 for short-sequence tasks and deeper models are more capable of modeling long-range dependencies, extending Michel et al.'s (11) work on the contribution of attention heads minimally to Transformer performance.

Impact of Noise Introducing random noise in 10% of the sequences led to noticeable performance degradation, particularly for shallow models. The accuracy of shallow configurations dropped by 25%, whereas deeper configurations with 8 attention heads experienced only a 10% reduction. This highlights that the robustness of deeper models with multi-head attention can better resolve ambiguities and maintain token alignment in noisy contexts and maintain higher accuracy, compared with their shallow counterparts (e.g., $L = 8$ vs. $L = 2$), as hypothesized in Theorem 2.1. We empirically validate the theoretical bound for Transformer depth in noisy and long-sequence scenarios established by Elhage et al.(12).

4 Conclusion and Future Work

This study demonstrates Transformer architecture in balancing expressive power and task complexity. Redundancy increases with model depth for short-sequence tasks, suggesting opportunities for pruning strategies in low-complexity scenarios and aligning with previous theoratical hypothesis. Deeper models and multi-head attention significantly improve performance on tasks involving long-range dependencies and noisy conditions, as shown in our Copy Task experiments. These results validate theoretical claims about the importance of depth and attention heads in enhancing model expressivity and handling complex sequence modeling challenges. While our experiments focus on depth and multi-head configurations, some aspects of the theoretical background, such as positional encoding and attention redundancy, remain unexplored and could be investigated in future work.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008.

[2] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 33, 1877–1901.

[3] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 34, 15084–15097.

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[5] Ontañón, S., Ainslie, J., Cvicek, V., & Fisher, Z. (2022). Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

[6] Csordás, R., Irie, K., & Schmidhuber, J. (2021). The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

[7] Anil, C., Wu, Y., Andreassen, A., Lewkowycz, A., Misra, V., Ramasesh, V., Slone, A., Gur-Ari, G., Dyer, E., & Neyshabur, B. (2022). Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*, 35, 38546–38556.

[8] Anonymous. (2020). *Every layer counts: Multi-layer multi-head attention for neural machine translation*. *Prague Bulletin of Mathematical Linguistics*, 114, 163–176.

[9] Zhang, B., Titov, I., & Sennrich, R. (2019). *Improving deep transformer with depth-scaled initialization and merged attention*. *arXiv preprint arXiv:1908.11365*. Retrieved from https://arxiv.org/abs/1908.11365.

[10] Xu, H. (2021). *Transformer-based NMT: Modeling, Training and Implementation*. Saarländische Universitäts- und Landesbibliothek. DOI: 10.22028/D291-34998.

[11] Michel, P., Levy, O., & Neubig, G. (2019). Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pp. 14014–14024.

[12] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Bai, Y., Ndousse, K., & Amodei, D. (2021). A mathematical framework for transformer circuits. *arXiv preprint arXiv:2104.07857*.

[13] Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2018). Universal transformers. In *International Conference on Learning Representations (ICLR)*.

[14] HuggingFace. (2023). `run_clm.py`: Language Modeling Script. In *Hugging-Face Transformers GitHub Repository*. Retrieved on December 10, 2024, from https://github.com/huggingface/transformers/blob/main/examples/pytorch/language_modeling/run_clm.py.

[15] He, B., & Hofmann, T. (2024). Simplifying Transformer Blocks. *arXiv preprint arXiv:2311.01906*. Retrieved from https://arxiv.org/abs/2311.01906.

[16] Chen, X., & Zou, D. (2024). What Can Transformer Learn with Varying Depth? Case Studies on Sequence Learning Tasks. *arXiv preprint arXiv:2404.01601*. Retrieved from https://arxiv.org/abs/2404.01601.

[17] Wang, M., & E, W. (2024). Understanding the Expressive Power and Mechanisms of Transformer for Sequence Modeling. Retrieved from https://arxiv.org/abs/2402.00522.