Beyond Parallelism: Synergistic Computational Graph Effects in Multi-Head Attention

Haitz Sáez de Ocáriz Borde OCARIZ@SUPERMODEL.AI
Supermodel

Editors: Francisco Acosta, Simone Azeglio, Chase van de Geijn, Christian Shewmake, Sophia Sanborn, and Nina Miolane.

Abstract

Multi-head attention powers Transformer networks, the primary deep learning architecture behind the success of large language models (LLMs). Yet, the theoretical advantages of multi-head versus single-head attention, beyond mere parallel processing, remain underexplored. In this paper, we reframe multi-head attention as a system of potentially synergistic computational graphs, where each head functions as a feedforward directed acyclic graph (DAG) with a common sink state. We provide intuition and preliminary theoretical analysis of mixing time and minimax fidelity in this framework. Our results show that multi-head attention can synergistically enhance information propagation, yielding faster mixing times and minimax fidelity amplification under specific head-diversity conditions. Finally, we train single-head and multi-head Transformers, each with the same total number of parameters, on sequence manipulation tasks and empirically verify the predicted effects. The code is available at https://github.com/haitzsaezdeocariz/beyondparallelism.

Keywords: Attention, Transformer, Graph Theory, Directed Acyclic Graph, Computational Graph, Markov Chain, Mixing Time, Minimax Fidelity, Signal Propagation

1. Introduction

In this paper, we adopt a graph-theoretic lens to interpret how multi-head attention processes and propagates information. By modeling each head as a feedforward computational graph, we expose the synergistic computational pathways enabled by parallel attention and introduce concrete metrics (mixing time and minimax fidelity). We anticipate that these insights may inspire future actionable interpretability techniques, empowering practitioners to diagnose, explain, and optimize attention-based architectures, while also helping us understand how specific behaviors and computations arise.

In a standard (decoder-only) Transformer with causal (masked) attention (Vaswani et al., 2017; Brown et al., 2020), which is omnipresent in language modeling among other applications, each position in the sequence can only attend to itself and to earlier positions. Such dependency can be abstracted via a structure with edges going strictly forward in time: one can view each head's pattern of attention as a feedforward directed acyclic graph (DAG) on n nodes (with the first node having no incoming edges other than its self-loop). The position n is the DAG's sink, since it has edges coming in from all previous positions but does not feed forward to any later positions, see Figure 1.

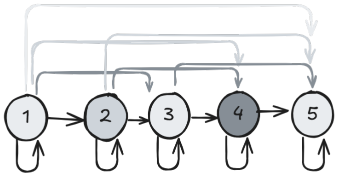


Figure 1: Example feedforward DAG with $n = 5$ nodes.

In Vitvitskyi et al. (2025), the authors use a DAG to model the entire forward pass of the network. On the other hand, in this paper, we will use a similar argument to compare the (intra-layer) computational graph of *single-head* attention against that of the ubiquitous *multi-head* attention, where multiple distinct dynamic graphs (the graph connectivity strength is input dependent) are used for simultaneous information propagation. We theoretically study *mixing time* and *minimax fidelity*. Additionally, we complement our findings by training single-head and multi-head Transformers on toy datasets and computing empirical proxies for these quantities. We indeed find that multi-head attention exhibits improved performance as compared to single-head using the same number of trainable parameters.

2. Multi-Head Mixing Time

Briefly, the mixing time is defined as the time it takes for a probability distribution over states to converge to a stationary distribution. In Vitvitskyi et al. (2025), the authors argue that the mixing time is small if and only if there are numerous paths from most nodes in the graph to the sink. They also claim that since having many paths facilitates efficient data propagation, the mixing time serves as a meaningful measure of how efficiently our network will operate under this computational graph. Hence, a lower mixing time would be better since it generally implies statistical efficiency. In this section we explore the following question: *Does mixing time improve (go down) when we combine multiple parallel feedforward computational graphs?*

2.1. Multi-Head Stationary Distributions

Definition 1 (Feedforward graph) *A* feedforward graph *on n vertices is a DAG in which the vertices can be indexed $1, 2, \ldots, n$ so that all edges $(j \to i)$ satisfy $j \leq i$.*

Definition 2 (Unique sink) *A* unique sink *τ (at position n) is a vertex with no outgoing edges to* distinct *vertices (i.e. the only possible edge is the self-loop $(\tau \to \tau)$).*

Definition 3 (Random walk matrix) *Given a directed graph $G = (V, E)$ on n vertices, the* random walk matrix *W is the $n \times n$ matrix with entries*

$$W_{ij} = \begin{cases} \frac{1}{\delta_j^{\rightarrow}}, & \textit{if } (j \to i) \in E, \\ 0, & \textit{otherwise}, \end{cases} \tag{1}$$

where $\delta_j^{\rightarrow}$ is the outdegree of node j. For causal attention this matrix is lower-diagonal.

Definition 4 (Stationary distribution) *A probability distribution $\pi \in \mathbb{R}^n$ on the vertices is called a* stationary distribution *if $W\pi = \pi$.*

Lemma 5 (Stationary Distribution for a Single-Head Unique Sink) *Let G be a feedforward graph on n vertices with a unique sink τ. Then the only stationary distribution for the random walk matrix W is 1_τ, the distribution taking value 1 at τ and 0 elsewhere (Vitvitskyi et al., 2025).*

As discussed in Vitvitskyi et al. (2025), clearly $W 1_\tau = 1_\tau$ because τ has no outgoing edges to other nodes (only possibly a self-loop). In terms of uniqueness, suppose π is another stationary distribution with $\pi \neq 1_\tau$. Let $j < n$ be the smallest-index vertex with $\pi_j \neq 0$. Because G is feedforward, j has at least one outgoing edge (apart from its self-loop) $(j \to i)$ with $i > j$ unless $j = \tau$. This implies π_j will "leak forward" under W, contradicting stationarity. Thus no such π can exist.

Next, we extend the unique-sink argument in Vitvitskyi et al. (2025). Specifically, we adapt the original derivation used for the full computational graph to model a single attention head in our analysis (as in Lemma 5) and then compare it to the multi-head attention setting.

Lemma 6 (Stationary Distribution for a Multi-Head Unique Sink) *Let $G^{(1)}, \dots, G^{(H)}$ be H feedforward graphs on n vertices, each having the* same *unique sink τ. Suppose the final operation merges the heads (via concatenation and a linear projection or any acyclic merging). Then the only stationary distribution (over the combined state space) is the one that places all its mass at τ.*

Proof See Appendix A. ■

Thus, even with multiple heads, as long as each head is a feedforward DAG sharing the same sink τ (see Figure 2), the unique stationary distribution argument holds.

2.2. Having Multiple Heads can Improve Mixing Time

Definition 7 (Mixing Time) *Given a random walk matrix W_{ij} with stationary distribution π, the* mixing time *is defined as:*

$$T_{\text{mix}}(\epsilon) = \min\{t \geq 0 : \max_j \|W_{ij}^t - \pi\|_{\text{TV}} \leq \epsilon\}, \tag{2}$$

where $\|\cdot\|_{\text{TV}}$ denotes the total variation distance, and $\epsilon > 0$ is a convergence threshold.

The goal is to identify the earliest possible time where the distribution is sufficiently close (within ϵ) to the stationary distribution π. Hence, taking the minimum over t ensures we are finding the fastest convergence time. Additionally, the maximum over all initial states j ensures a worst-case scenario analysis. In other words, mixing time guarantees convergence from every possible starting state: it reflects the slowest possible convergence to the stationary distribution across all starting points.

To address the main question of this section, we derive an upper bound on the mixing time for a combined multi-head attention system using combinatorial and probabilistic arguments rather than relying on the standard spectral gap analysis such as those in Horn and Johnson (1985); Levin and Peres (2017). This is because as pointed out by Vitvitskyi et al. (2025), the latter does not necessarily apply given that our random walk matrices are lower-diagonal.

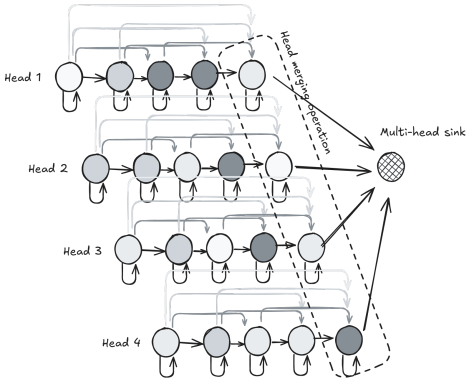


Figure 2: Multi-head sink visualization.

Let $G^{(1)}, \dots, G^{(H)}$ be H feedforward graphs on the same set of n vertices, each with a unique sink τ. Suppose that, for each head h, the probability of making a forward move (i.e., a move that brings the state closer to τ) is at least p_h. Assume that the heads are combined via a convex combination $\overline{W} = \sum_{h=1}^{H} \alpha_h \, W^{(h)}$, with $\alpha_h \geq 0$ and $\sum_{h=1}^{H} \alpha_h = 1$. Define the effective forward probability as $p = \sum_{h=1}^{H} \alpha_h \, p_h$.

Theorem 8 (Multi-Head Mixing Time Bound via Forward Moves) *If it requires at most $N = n - 1$ forward moves to reach the sink τ (with the worst-case being the leftmost node), then with high probability the mixing time of the combined chain satisfies $T_{\mathrm{mix}}(\overline{W}, \epsilon) \lesssim \frac{2N}{p}$, where ϵ is a small constant. In particular, note that since $p \leq \max_{1 \leq h \leq H} p_h$ it follows that $\frac{2N}{p} \geq \frac{2N}{\max_{1 \leq h \leq H} p_h}$. Thus, if the convex weights are chosen arbitrarily, the bound on the mixing time of the combined chain is in general no better than the mixing time of the fastest individual head, but always better than the worse. However, if the weights can be selected adaptively:*

$$T_{\mathrm{mix}}(\overline{W}, \epsilon) \lesssim \frac{2N}{\max_{1 \leq h \leq H} p_h} = \min_{1 \leq h \leq H} T_{\mathrm{mix}}\big(W^{(h)}, \epsilon\big). \tag{3}$$

Proof See Appendix A, including scope (intra-layer analysis) and modeling assumptions (convex proxy). ■

In summary, each head $W^{(h)}$ corresponds to a Markov chain that propagates information along selected edges in the DAG. By merging these heads, additional transition edges are added, reducing potential bottlenecks. While combining heads via a convex combination may not always yield a faster mixing time than the best individual head (it is only guaranteed to be faster than the worse head), adaptive weighting (favoring the head with the highest p_h) can ensure that the bound on the mixing time of the combined chain is close to the fastest head. We empirically verify this effect in Section 4.

3. Multi-Head Minimax Fidelity

In a similar spirit to our analysis of multi-head mixing time, we now turn to studying the fidelity of information propagation when multiple feedforward attention heads are combined. As pointed out in Vitvitskyi et al. (2025), whereas mixing time measures how quickly probability mass converges to the unique sink, fidelity quantifies how sharply the signal from each node is preserved when it reaches the sink.

Definition 9 (Diffusion Matrix) *For a single-head feedforward graph, we define a* diffusion *matrix*

$$\Delta_{ij} = \begin{cases} \frac{1}{\delta_i^{\leftarrow}}, & \text{if } (j \to i) \in E, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

which normalizes by the column (i.e., the in-degree $\delta_i^{\leftarrow}$, which is always positive since we always have self-loops).

Hence, in this notation index i corresponds to the receiving node and the second index j corresponds to the sending node. In our analysis, the receiving node we are interested in is the sink, $i = \tau$.

Definition 10 (Signal at the Sink) *If we initialize with a one-hot vector e_j (placing all mass at node j), then after t steps the signal at the sink τ is given by $\phi_j^{(h)}(t) = \big((\Delta^{(h)})^t\big)_{\tau j}$.*

Definition 11 (Node Fidelity) *The* node fidelity *is defined as the maximum signal that node j ever contributes to τ: $\phi_j^{(h)} = \max_t \phi_j^{(h)}(t) = \max_t \big((\Delta^{(h)})^t\big)_{\tau j}$.*

Definition 12 (Optimal Fidelity Time) *We define the optimal fidelity time as the time at which node j (head h) attains its maximum signal: $t_j^{(h)} = \arg\max_t \Big\{\phi_j^{(h)}(t)\Big\}$.*

Note that the amount of signal that reaches the sink τ does not necessarily increase monotonically, see Proposition 5.1 in Vitvitskyi et al. (2025). Instead, it can rise to a peak at some intermediate time and then decay as the signal becomes diluted or averaged out by further propagation. Therefore, even if the signal eventually decays (due to the averaging nature of the diffusion process), its peak value, captured by the equation above, represents the most effective transmission from node j. Lastly, by taking the minimum over all nodes we identify the worst-case (least preserved) signal among all nodes:

Definition 13 (Minimax Fidelity) *The* minimax fidelity *for a single head h at the sink τ is given by $\phi_{\min}^{(h)} = \min_j \phi_j^{(h)}$.*

Intuitively, this would be the lowest peak signal produced at the sink by any node in the DAG (token in the sequence).

3.1. Having Multiple Heads can Amplify Fidelity

Next, we will show that it is possible to observe synergistic effects in terms of signal propagation over parallel heads.

Definition 14 (Multi-Head Diffusion Operator) *Let $\Delta^{(1)}, \dots, \Delta^{(H)}$ be the diffusion matrices corresponding to a total of $H = |\mathcal{H}|$ feedforward attention heads. We define multi-head diffusion operator using convex weights $\{\beta_h\}_{h\in\mathcal{H}}$ (i.e., $\beta_h \geq 0$ and $\sum_{h\in\mathcal{H}} \beta_h = 1$) as*

$$\overline{\Delta}_{ij} = \sum_{h\in\mathcal{H}} \beta_h \, \Delta_{ij}^{(h)}. \tag{5}$$

Remark 15 (Cross-Head Pathway Contribution) *Taking powers of the multi-head diffusion operator yields the following expression (note non-commutativity: in general $\Delta^{(h)}\Delta^{(h')} \neq \Delta^{(h')}\Delta^{(h)}$) which promotes additional information propagation pathways:*

$$(\overline{\Delta})^t = \Big(\sum_{h\in\mathcal{H}} \beta_h \Delta^{(h)}\Big)^t = \sum_{h_1\in\mathcal{H}} \cdots \sum_{h_t\in\mathcal{H}} \Big(\prod_{r=1}^{r=t} \beta_{h_r}\Big) \, \Delta^{(h_1)} \cdots \Delta^{(h_t)}. \tag{6}$$

Example 1 *Let our nodes be ordered (u, v, τ). Head 1 has edges $u \to v$ (plus self-loops everywhere), head 2 has edges $v \to \tau$ (plus self-loops). Then the diffusion matrices are:*

$$\Delta^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \Delta^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

For a single head: $\left(\Delta^{(1)}\right)^2_{\tau u} = 0, \qquad \left(\Delta^{(2)}\right)^2_{\tau u} = 0,$ *because head 1 never has a path* $u \to \tau$ *in two steps, nor does head 2. On the other hand when using multi-head attention we obtain cross-head product terms:*

$$\left(\Delta^{(2)}\Delta^{(1)}\right)_{\tau u} = \tfrac{1}{2} \times \tfrac{1}{2} = \tfrac{1}{4} > 0.$$

The term corresponding to head 1 then head 2 appears with weight $\beta_1\beta_2$*:*

$$\begin{aligned}(\overline{\Delta})^2_{\tau u} &= \\ &= (\beta_1^2\,(\Delta^{(1)})^2)_{\tau u} + ((\beta_1\,\Delta^{(1)})(\beta_2\,\Delta^{(2)}))_{\tau u} + \\ &+ ((\beta_2\,\Delta^{(2)})(\beta_1\,\Delta^{(1)}))_{\tau u} + (\beta_2^2\,(\Delta^{(2)})^2)_{\tau u} = \\ &+ 0 + 0 + \beta_1\beta_2\frac{1}{4} + 0 = \beta_1\beta_2\frac{1}{4} > 0.\end{aligned}$$

so the multi-head diffusion operator captures a contribution. In other words, whenever one head bridges $u \to v$ *and another bridges* $v \to \tau$*, their composition opens up a two-step path that neither head has alone.*

This would exemplify the trivial case where the fidelity is zero for each individual head, but not for the multi-head diffusion operator.

Definition 16 (Multi-Head Node Fidelity) *We define the multi-head node fidelity as the node fidelity computed based on the multi-head diffusion operator:* $\phi_j^{\text{multi}} = \max_t \Big((\overline{\Delta})^t\Big)_{\tau j}$.

Definition 17 (Multi-Head Minimax Fidelity) *The* multi-head minimax fidelity *is given by* $\phi_{\min}^{\text{multi}} = \min_j \phi_j^{\text{multi}}$.

Definition 18 (Best Head Minimax Fidelity) *Comparing the minimax fidelity across heads in the multi-head attention mechanism, the best head minimax fidelity is:* $\phi_{\min}^{(h_*)} = \max_{h\in\mathcal{H}} \phi_{\min}^{(h)}$.

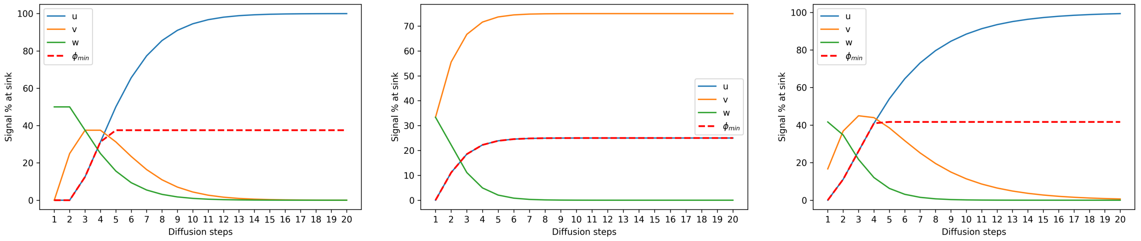
Definition 19 (Best Head) *Consequently we define the best head as:* $h_* = \arg\max_h \left\{\phi_{\min}^{(h)}\right\}$.

Next we proceed to show a non-trivial case in which the best head minimax fidelity is not zero (there exists a path connecting all nodes to the sink), yet the multi-head minimax fidelity is higher:

Example 2 *Let our nodes be ordered* (u, v, w, τ)*. Let head 1 be a linear chain with edges* $u \to v, v \to w, w \to \tau$ *(plus self-loops everywhere), and head 2 a feedforward DAG with* $u \to w, v \to w, v \to \tau, w \to \tau$ *(plus self-loops). Also set* $\beta_1 = \beta_2 = \frac{1}{2}$*. Then the diffusion matrices and the multi-head diffusion operator are:*

$$\Delta^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \qquad \Delta^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}, \qquad \overline{\Delta} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ \frac{1}{6} & \frac{5}{12} & \frac{5}{12} & 0 \\ 0 & \frac{1}{6} & \frac{5}{12} & \frac{5}{12} \end{pmatrix}.$$

Note that in this case all nodes u, v, w have paths to reach the sink τ in both heads independently, unlike in the previous example. Applying matrix multiplication repeatably to each diffusion matrix we obtain the result in Figure 3, in which we can clearly see that the minimax fidelity for the multi-head system is higher than that of each head, that is $\phi_{\text{min}}^{\text{multi}} = 0.417 > \phi_{\text{min}}^{(h_*)} = \phi_{\text{min}}^{(h_1)} = 0.375$ and $\phi_{\text{min}}^{\text{multi}} = 0.417 > \phi_{\text{min}}^{(h_2)} = 0.250$.



(a) Head 1

(b) Head 2

(c) Multi-Head

Figure 3: Diffusion of signal from nodes u, v, w to the sink τ under single-head and multi-head diffusion kernels. Solid lines show signal arrival percentages over diffusion steps, while the dashed line ϕ_{min} indicates the cumulative fidelity.

This example demonstrates that combining multiple feedforward attention heads results in a multi-head system that can achieve a level of minimax fidelity exceeding that of any single head, when this is non-trivial (i.e. not zero). In a single computational graph (or single-head attention mechanism), the diffusion process is governed by one diffusion matrix, limiting the maximum achievable fidelity to that single pathway. In contrast, the multi-head setting employs diverse diffusion matrices, each capturing different aspects of signal propagation. *Although one might expect the weighted average of individual fidelities to be bounded by the highest individual fidelity, the multi-head diffusion operator can surpass the best individual performance.* This synergy suggests that multi-head architectures are not merely parallel computations but can work in concert to enhance the preservation and amplification of information. For an additional discussion regarding Figure 3 see Appendix B.

4. Experimental Validation

In this section, we empirically validate our theoretical analysis by measuring proxies for mixing time and fidelity. Our experiments are conducted on two toy synthetic tasks concerned with string manipulation. We first describe the data generation and model configurations, then detail the algorithms used to compute each proxy, and finally present our experimental results. Tables and figures for this section can be found in Appendix D.

Data Generation We work with two synthetic datasets. I) We generate sequences of length 100 from a vocabulary of 256 tokens. Each token is sampled independently from a discrete uniform distribution. For each sample $x = (x_1, x_2, \ldots, x_{100})$, the target is defined as x itself: $x \mapsto x$. A total of 5000 such examples are used for training. II) Similarly, we generate random sequences of length $n = 100$. However, the target is produced by cyclically shifting the input by one position: $x = (x_1, x_2, \ldots, x_{99}, x_{100}) \quad \mapsto \quad (x_{100}, x_1, x_2, \ldots, x_{99})$. This task requires the model to learn a non-trivial permutation of the sequence.

Model Architecture and Training We adopt a standard pre-norm (RMSNorm) Transformer architecture with causal (decoder-only) attention. The model has 4 transformer layers with a token embedding dimension of 64 (both for the embedding and attention layers), a MultiLayer Perceptron (MLP) hidden dimension of 128, and dropout of 0.1 in both attention and MLP layers. We vary the number of attention heads over $\{1, 4, 8, 16\}$ while keeping the total embedding dimension fixed. Thus, as the head count increases, the per-head dimension decreases accordingly. We do so to consider a constant model parameter count, and to ensure our results are solely dependent on the computational graph structure, rather than on having more trainable weights. Training is performed using the Adam optimizer with a learning rate of 10^{-3}, a batch size of 50, and for 200 epochs.

4.1. Mixing Time Proxy Estimation

To quantify how rapidly information propagates through the model's causal attention graph, we approximate the mixing time by a straightforward (per-layer) Monte Carlo hitting-time experiment, see Algorithm 1 (Appendix E). We extract the attention tensor of shape (H, n, n), where H is the number of heads and $n = 100$ the sequence length. We then form a single transition matrix by weighting each head according to the model's own learned output-projection importances. Thus, we extract each head's contribution to the next-layer representation by slicing the model's output-projection matrix into H equal blocks (one per head) and computing the norm of each block. These norms serve as raw importance scores, which we then normalize to sum to one and use to weight the corresponding $n \times n$ attention matrices. The resulting convex combination defines a single forward-transition matrix over token positions. Interpreting this combined matrix as defining forward-directed transition probabilities, we simulate many independent random walks starting from each token position, sampling the next position according to the attention-induced distribution over successors. Each walk proceeds until it reaches the final token or until a maximum cutoff of 100 steps is exceeded.

Note that the attention tensor depends on the input, so we run this experiment across all samples in the dataset. We repeat this process for 500 simulations per start state and per sample to obtain an empirical mean hitting time. We would like to highlight that to be faithful to Definition 7, one would need to apply the max operator across starting positions, rather than the mean as in Algorithm 1. However, we restrict ourselves to the average hitting time across positions for computational tractability: we find that to obtain meaningful trends for the worse case mixing time it is necessary to increase the cutoff steps substantially making the simulation very expensive. In Table 1 and Table 2 we report the results for mixing time in steps for both tasks and for all layers in the model. We also plot them in Figure 4(a) and Figure 4(b) in which we can more clearly appreciate the trend predicted theoretically: adding more heads (under the same parameter count) leads to faster mixing.

4.2. Minimax Fidelity Proxy Estimation

To quantify how sharply information from every token is preserved when it reaches the sink, we compute a (per-layer) minimax diffusion fidelity over a fixed diffusion horizon, see Algorithm 2 (Appendix E). As before, we begin by extracting the attention tensor of shape (H, n, n) and forming a single forward-transition matrix via a convex combination of heads weighted by their learned output-projection norms. We then simulate the diffusion process up to 100 steps by repeatedly applying matrix multiplication: at each step we record the probability of mass starting at node j arriving at the sink. For each start node j we retain its peak arrival probability over all steps, and finally we take the minimum of these peaks across j to obtain the worst-case (minimax) fidelity for that example. In Table 3, Table 4, and Figure 4(c) and Figure 4(d) we can see that the fidelity goes up with the number of heads, as predicted. Interestingly, Layer 4 of the model trained on cycle sequences appears to perform poorly in terms of both fidelity and mixing time. Similarly, when comparing subfigures in Figure 4, drops in mixing time seem to align with increases in fidelity. This could suggest a correlation between the two metrics, but further investigation is needed.

Note that thus far, we have empirically shown improved minimax fidelity for multi-head attention compared to single-head models. However, our primary discussion in Section 3 focused on the fact that the fidelity from combining multiple heads can surpass the best individual head's fidelity at a given node within the same multi-head model. To this end, we evaluated each head's minimax fidelity independently and compared it directly with the minimax fidelity obtained by combining heads using learned weights. Empirically, we indeed observed multiple instances across both tasks (copy and cycle) and various head counts (4, 8, and 16 heads) where the combined multi-head minimax fidelity exceeded the best individual head minimax fidelity, thus providing experimental support that this effect takes place in learning-based system too, not only when handcrafted as in our original example. See Table 5 and Table 6 for details.

5. Conclusion

In this work we introduced a graph-theoretic framework that models multi-head attention as a collection of synergistic feedforward DAGs, revealing possible benefits beyond mere computational parallelism. We show that the presence of multiple heads can reduce mixing time and amplify minimax fidelity. To verify the validity of the intuition, we perform preliminary sequence manipulation experiments with single-head versus multi-head Transformers with the same parameter count, obtaining satisfactory results. We have often argued that it may be possible to find configurations that satisfy our theoretical requirements via gradient descent optimization. This is true, but in practice the model is trying to optimize the downstream loss function: cross-entropy in the case of pre-training for next-token prediction. This implies that mixing time and minimax fidelity can only be optimized as a consequence of improving downstream performance but not directly. The model must find them useful for the task at hand.

Finally, we note that previous work such as that by (Voita et al., 2019; Michel et al., 2019) has found that many attention heads can be pruned without significant loss in performance. This does not directly contradict our findings: rather, it suggests that the benefits we identify (reduced mixing time and enhanced minimax fidelity) may plateau or saturate beyond a certain number of heads. That is, while a few heads could contribute importantly to synergy, adding more may yield diminishing returns unless head diversity is preserved. Another interesting observation is that pruning (including head pruning) is usually done after training (Cheng et al., 2024), which could suggest that additional heads help during optimization, even if they become redundant at convergence.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations, 2024. URL https://arxiv.org/abs/2308.06767.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, United Kingdom, 1985. ISBN 978-0-521-30586-0.

David A. Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society Textbooks. American Mathematical Society, Providence, RI, USA, 2 edition, 2017. ISBN 978-1-4704-2962-1.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:13756489.

Alex Vitvitskyi, João G. M. Araújo, Marc Lackenby, and Petar Veličković. What makes a good feedforward computational graph?, 2025. URL https://arxiv.org/abs/2502.06751.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, 2019. URL https://arxiv.org/abs/1905.09418.

Appendix A. Proofs

This appendix contains the proofs that complement the discussion in the main text.

Proof [Proof of Lemma 6] For any head h, the graph $G^{(h)}$ is feedforward. This means that any random walk starting at j within head h can only transition to vertices $j \leq i$. Furthermore, because $\tau = n$ is the unique sink for all heads, probability mass within any single head h eventually flows towards τ. Specifically, applying the walk matrix $W^{(h)}$ repeatedly will concentrate the mass at τ. The merging operation combines the outputs from each head at each position. Crucially, this merging process is typically local and acyclic with respect to the sequence positions. For example, concatenating head outputs at position i and projecting them linearly combines information already arrived at position i via paths $(j \to i)$ within the various heads. This merging step does not create new paths from τ back to earlier positions $j < \tau$, nor does it allow information at a position $j < \tau$ to remain indefinitely without flowing towards τ. ■

Proof [Proof of Theorem 8] For each step of the combined Markov chain defined by $\overline{W}$, the probability of making a forward move is at least p. Let X denote the total number of forward moves made after t independent steps. Then

$$X \sim \mathrm{Bin}(t, p) \tag{7}$$

and the expected number of forward moves is

$$\mathbb{E}[X] = t\,p. \tag{8}$$

Setting the expected number of forward moves to be $2N$ (i.e., $t\,p = 2N$) gives

$$t = \frac{2N}{p}. \tag{9}$$

Now, by Hoeffding's inequality (Hoeffding, 1963),

$$\Pr\bigl[X < N\bigr] \leq \exp\Bigl(-\frac{2(tp - N)^2}{t}\Bigr). \tag{10}$$

Substituting $tp = 2N$ yields

$$\Pr\bigl[X < N\bigr] \leq \exp\Bigl(-\frac{2N^2}{2N/p}\Bigr) = \exp(-pN). \tag{11}$$

For fixed p and sufficiently large N, the right-hand side becomes very small (e.g., less than 1/4), meaning that with high probability the chain makes at least N forward moves within $t = \frac{2N}{p}$ steps. In other words, the mixing time is bounded by

$$T_{\mathrm{mix}}(\overline{W}, \epsilon) \lesssim \frac{2N}{p}. \tag{12}$$

The above refers to the mixing time defined with respect to total variation distance, i.e., the time required for the Markov chain to be within ϵ of the stationary distribution. For the combined chain this quantity is no more than a constant multiple of $\frac{2N}{p}$ (or behaves asymptotically like $\frac{2N}{p}$) as the relevant parameters vary.

Next, for H heads with individual forward probabilities $p_1, \ldots, p_H$ and convex weights $\alpha_1, \ldots, \alpha_H$ (with $\alpha_h \geq 0$ and $\sum_{h=1}^{H} \alpha_h = 1$), the effective forward probability is

$$p = \sum_{h=1}^{H} \alpha_h \, p_h. \tag{13}$$

Since the weights are nonnegative and sum to 1, by the properties of convex combinations we have

$$\min_{1 \leq h \leq H} p_h \leq p \leq \max_{1 \leq h \leq H} p_h. \tag{14}$$

Hence, if one could choose the weights optimally (let us assume that gradient descent will likely try to optimize for an optimal combination)—for example, by setting $\alpha_{h^*} = 1$ for the head h^* that attains the maximum p_h and $\alpha_h = 0$ for all $h \neq h^*$—then we would have

$$p^* = \max_{1 \leq h \leq H} p_h. \tag{15}$$

In that ideal case, the overall mixing time (which is roughly bounded by $\frac{2N}{p}$) would be

$$T_{\mathrm{mix}}(\overline{W}, \epsilon) \lesssim \frac{2N}{\max_{1 \leq h \leq H} p_h}, \tag{16}$$

which is as fast as the fastest individual head. The result simply emphasizes that one cannot hope for a combined chain to mix faster than the fastest individual chain, though with optimal (or adaptive) weighting one can match it.

Even if one uses fixed (say, uniform) weights instead of the idealized choice, the presence of multiple heads means that with high probability at least one head has a relatively high p_h. To show this formally assume that the forward probabilities $p_1, p_2, \ldots, p_H$ for the H heads are independent random variables with a common cumulative distribution function (CDF), $F(p)$. For any fixed threshold $\bar{p}$ such that $F(\bar{p}) < 1$, the probability that a single head satisfies $p_h \geq \bar{p}$ is

$$\Pr(p_h \geq \bar{p}) = 1 - F(\bar{p}). \tag{17}$$

Since the heads are independent, the probability that all H heads have $p_h < \bar{p}$ is

$$\Pr\Big(p^* < \bar{p}\Big) = \Big(F(\bar{p})\Big)^H. \tag{18}$$

Thus, the probability that at least one head achieves $p_h \geq \bar{p}$ is

$$\Pr\Big(p^* \geq \bar{p}\Big) = 1 - \Big(F(\bar{p})\Big)^H. \tag{19}$$

Since $F(\bar{p}) < 1$, as H increases,

$$\lim_{H \to \infty} \Pr\Big(p^* \geq \bar{p}\Big) = 1. \tag{20}$$

Consequently, if one could choose weights adaptively to favor the head with the highest p_h, for instance using gradient descent, p will be statistically biased toward higher values. While the worst-case bound is

$$\frac{2N}{\max_h p_h} \leq T_{\text{mix}}(\overline{W}, \epsilon) \lesssim \frac{2N}{\min_h p_h}, \tag{21}$$

the key point is that, statistically, multiple heads increase the chance that a high p_h is present, thereby allowing the effective mixing time to approach the lower bound of $\frac{2N}{\max_h p_h}$. ■

Remark 20 (Scope and Modeling Assumptions in Theorem 8) *In real multi-head attention, heads are concatenated and passed through a linear projection W_{proj}, which produces a new row-stochastic, input-dependent kernel. This is* not *literally a convex combination of the $W^{(h)}$. Our model uses such a combination as a* tractable analytical proxy *to upper-bound the effective forward propagation strength, assuming W_{proj} preserves the feed-forward DAG structure. Additionally, we would like to clarify that our mixing time analysis models information propagation* within a single application *of multi-head attention (i.e., one forward pass through an attention layer). For a fixed input, each head defines a fixed transition matrix $W^{(h)}$, and the merged kernel $\overline{W} = \sum_h \alpha_h W^{(h)}$ is also fixed. Thus, the Markov chain defined by iterating $\overline{W}^t$ is* time-homogeneous*, and our bound applies directly. Note that* time-inhomogeneous *chains arise naturally in* multi-layer *Transformers, where each layer applies a distinct attention kernel. Our analysis is restricted to* intra-layer *dynamics and does* not *extend across layers.*

Appendix B. Example 2 temporal dynamics discussion

Figure 3 illustrates the temporal dynamics of Example 2. Each head defines a distinct diffusion pattern: the first head exhibits a gradual but comprehensive propagation from u to τ, while the second head emphasizes faster, more localized diffusion primarily through v. When combined into a multi-head system, the resulting diffusion operator integrates these complementary pathways, producing a joint evolution that exceeds the minimax fidelity achievable by either head alone. In the combined diffusion, the node signals display cooperative behavior: u maintains long-range reachability while v contributes strong early propagation.

Appendix C. Compute Resources

All experiments were performed on a single H100 GPU under a Linux environment.

Appendix D. Results: Tables and Figures

In this appendix we include the tables and figures discussed in Section 4.

Table 1: Mixing time steps mean ± std dev. for **Copy Sequence**.

Heads	L1	L2	L3	L4
1	3.8344 ± 0.1391	77.9281 ± 2.8387	78.6635 ± 3.2322	73.5020 ± 3.6750
4	3.7370 ± 0.0259	3.7155 ± 0.0365	3.8895 ± 0.0704	7.8050 ± 0.5391
8	3.6793 ± 0.0171	3.6821 ± 0.0329	3.7952 ± 0.0661	3.7467 ± 0.0731
16	3.6328 ± 0.0150	3.6513 ± 0.0212	3.6391 ± 0.0564	3.6594 ± 0.0687

Table 2: Mixing time steps mean ± std dev. for **Cycle Sequence**.

Heads	L1	L2	L3	L4
1	27.0384 ± 2.4355	47.8151 ± 0.6754	47.2291 ± 1.3803	43.0723 ± 2.8604
4	3.6369 ± 0.0161	3.9092 ± 0.0374	45.7709 ± 1.9865	31.2925 ± 3.3583
8	3.6426 ± 0.0098	3.7756 ± 0.0290	4.6735 ± 0.0795	47.5304 ± 0.9180
16	3.5859 ± 0.0118	3.7820 ± 0.0307	4.3759 ± 0.0583	41.9292 ± 1.9165

Table 3: Fidelity % mean ± std dev. for **Copy Sequence**

Heads	L1	L2	L3	L4
1	0.3000 ± 0.1000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
4	0.4500 ± 0.0700	0.2800 ± 0.0800	0.1800 ± 0.0500	0.0100 ± 0.0100
8	0.5000 ± 0.0800	0.3500 ± 0.0800	0.1900 ± 0.0800	0.1600 ± 0.0600
16	0.5800 ± 0.0600	0.4900 ± 0.0700	0.3400 ± 0.0700	0.2500 ± 0.0700

Table 4: Fidelity % mean ± std dev. for **Cycle Sequence**

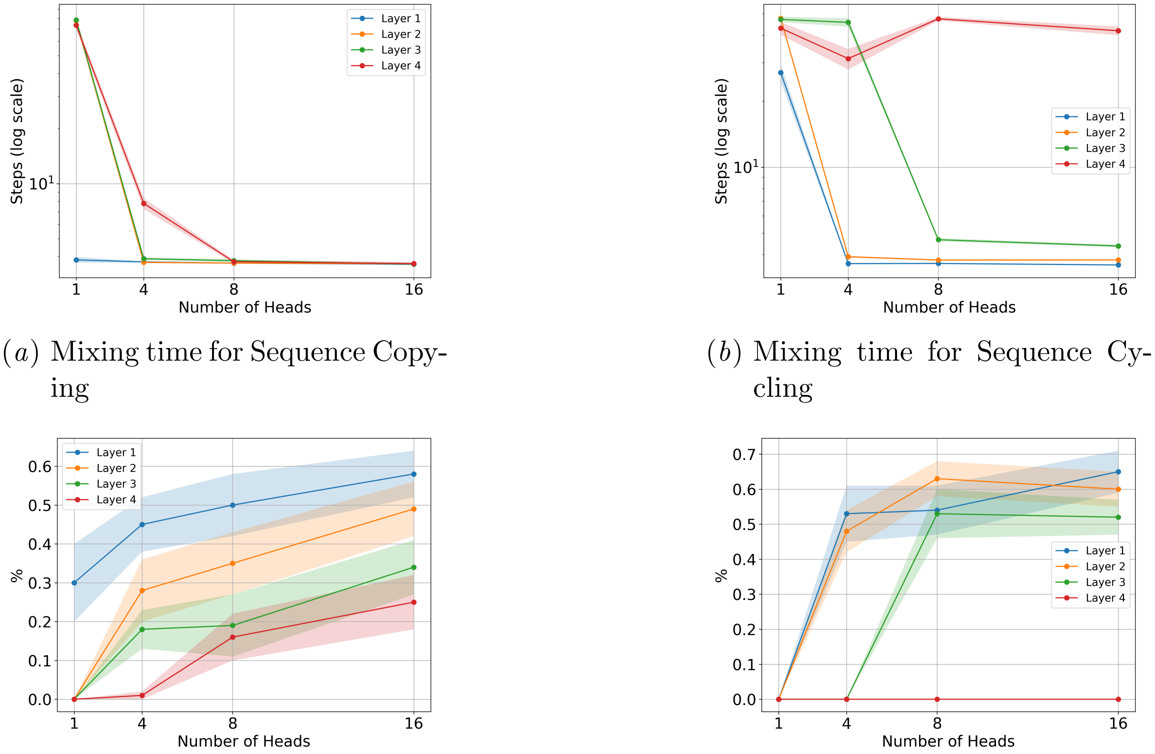
Heads	L1	L2	L3	L4
1	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
4	0.5300 ± 0.0800	0.4800 ± 0.0600	0.0000 ± 0.0000	0.0000 ± 0.0000
8	0.5400 ± 0.0700	0.6300 ± 0.0500	0.5300 ± 0.0700	0.0000 ± 0.0000
16	0.6500 ± 0.0600	0.6000 ± 0.0500	0.5200 ± 0.0500	0.0000 ± 0.0000

Table 5: Best individual vs combined multi-head fidelity % for **Copy Sequence**.

Heads	L1	L2	L3	L4
4 (Individual)	0.30	0.11	0.21	0.00
4 (Combined)	**0.56**	**0.43**	0.19	0.00
8 (Individual)	**0.59**	0.29	**0.18**	**0.18**
8 (Combined)	0.51	**0.37**	0.10	0.17
16 (Individual)	**0.59**	**0.59**	**0.52**	0.22
16 (Combined)	**0.59**	0.51	0.46	**0.23**

Table 6: Best individual vs combined multi-head fidelity % for **Cycle Sequence**.

Heads	L1	L2	L3	L4
4 (Individual)	0.39	0.39	0.00	0.00
4 (Combined)	**0.59**	**0.50**	0.00	0.00
8 (Individual)	0.35	0.49	0.51	0.00
8 (Combined)	**0.59**	**0.72**	**0.55**	0.00
16 (Individual)	0.65	0.49	**0.68**	0.00
16 (Combined)	**0.67**	**0.55**	0.48	0.00



(a) Mixing time for Sequence Copying

(b) Mixing time for Sequence Cycling

(c) Fidelity for Sequence Copying

(d) Fidelity for Sequence Cycling

Figure 4: Mixing time and fidelity for Transformer model trained on synthetic sequence manipulation tasks.

Appendix E. Algorithmic Descriptions

In this appendix, we include the algorithmic descriptions discussed in Section 4.

```
Algorithm 1: Monte Carlo Mixing-Time Proxy
Require : Dataset $D$, with samples indexed by $b \in \{1, \dots, |D|\}$, attention $\{A_b^{(h)}\}_{h=1}^H$, head
          weights $\{w_b^{(h)}\}_{h=1}^H$, sequence length $n$, number of simulations $S$, max steps $M$
Compute: Mixing times $T_{mix}$

for $b \leftarrow 1$ to $|D|$ do
    Compute transition matrix $P_b \leftarrow \sum_{h=1}^H w_b^{(h)} \left(A_b^{(h)}\right)^\top$ for $i \leftarrow 1$ to $n$ do
        for $s \leftarrow 1$ to $S$ do
            $T_{b,i,s} \leftarrow 0 \quad j \leftarrow i, \quad t \leftarrow 0$
            while $j \neq n$ and $t < M$ do
                sample $j \sim P_b(j)$
                $t \leftarrow t + 1$
            end
            $T_{b,i,s} \leftarrow T_{b,i,s} + t$
        end
    end
end
$T_{mix} \leftarrow \frac{1}{|D|} \frac{1}{S} \frac{1}{n} \sum_{b=1}^{b=|D|} \sum_{s=1}^{s=S} \sum_{i=1}^{i=n} T_{b,i,s}$
return $T_{mix}$
```

```
Algorithm 2: Minimax Fidelity Proxy
Require : Dataset $D$, with samples indexed by $b \in \{1, \dots, |D|\}$, attention matrices
          $\{A_b^{(h)}\}_{h=1}^H$ for each sample, head weights $\{w_b^{(h)}\}_{h=1}^H$, diffusion horizon $M$, target
          (sink) position $\tau$.
Compute: Minimax Fidelity $\phi_{\text{min}}^{\text{multi}}$

for $b \leftarrow 1$ to $|D|$ do
    $P_b \leftarrow \sum_{h=1}^H w_b^{(h)} \left(A_b^{(h)}\right)^\top \quad F_{b,m=0,ij} \leftarrow I$   for $m \leftarrow 1$ to $M$ do
        $F_{b,m,ij} \leftarrow \text{matmul}(F_{b,m-1,ij}, P_b)$
    end
end
$\phi_{\text{min}}^{\text{multi}} \leftarrow |D|^{-1} \sum_{b=1}^{|D|} \min_j \max_m F_{b,m,i=\tau j}$
return $\phi_{\text{min}}^{\text{multi}}$
```