Advances in Artificial Intelligence and Machine Learning; Research 5 (4) 4675-4691 Received 24-08-2025; Accepted 07-12-2025; Published online 14-12-2025

Going Beyond a Basic Attention Head toward an Understanding of Transformer-based Generative AI

Nicholas J. Restrepo nicholasjohnsonr@gmail.com
Dynamic Online Networks Laboratory
George Washington University
Washington, DC 20052, USA

Frank. Y. Huo yfh400@gwu.edu
Physics Department
George Washington University
Washington, DC 20052, USA

Dylan J. Restrepo ddylanj3@gmail.com
Cornell Tech
Cornell University
New York, NY 10044, USA

Neil F. Johnson neiljohnson@gwu.edu
Physics Department
George Washington University
Washington, DC 20052, USA

Corresponding Author: Neil F. Johnson

Abstract

This paper goes beyond the basic Attention head analysis introduced in the companion paper published in this issue, as part of our long-term goal to establish a bottom-up understanding of Chat-GPT-like generative AI. We provide evidence that suggests the output tipping behavior that we reported for the basic Attention head can persist despite some of the complications of real LLMs. Specifically, we consider here (1) a richer vocabulary, (2) non-identity matrices learned during pre-training, (3) non-zero temperature during next-token selection, (4) more than one layer of Attention heads. We then offer some preliminary evidence that the insights gained from this bottom-up approach, can help improve performance in real-world generative AI systems.

Keywords: Dynamics, Physics, Attention head, Generative AI, Complexity science.

1. INTRODUCTION

Large language models (LLMs) based on Transformer attention [1, 2], are now being deployed across medicine, insurance, finance, and defense. Yet sustained societal adoption is constrained by trust: practitioners and regulators still lack a principled, predictive understanding of when an LLM's output will remain reliable versus tipping mid-response into outputting content that is less desirable, misleading, irrelevant or unsafe [3–6]. The field of mechanistic interpretability (MI) has managed to map mesoscale circuits and

Citation: Nicholas J. Restrepo, et al. Going Beyond a Basic Attention Head toward an Understanding of Transformer-based Generative AI. **Advances in Artificial Intelligence and Machine Learning**. 2025;5(4):259. https://dx.doi.org/10.54364/AAIML.2025.54259

head-level roles [7–10], but these results are for simplified models and are difficult to generalize into any compact explanatory law or principle.

The complementary line of research that we pursue in this and our accompanying paper [1], is a bottom-up strategy borrowed from decades of research in physics, chemistry and materials science: *begin with a minimal 'atom' and build upward*. A century or more of minimal-model reasoning from across physics and chemistry shows that deeply understanding the properties of a highly simplified version of the basic building block (atom) and how they couple together (bond) can yield a transparent, robust and even predictive understanding of larger aggregates (molecules, solids etc.). By analogy, it seems possible that treating an Attention head as a GPT's 'atom' could in principle provide a clean — though of course massively oversimplified – way to isolate a tractable core mechanism such as a tipping point in its behavior; and perhaps even to derive closed-form predictions (e.g. when and how local failure modes emerge); and then perhaps to compose and perturb these atoms to eventually approach the behavioral scale of a real, multilayer ChatGPT-like system. Clearly this bottom-up lens does not replace empirical circuit-mapping. But instead it can complement it with potentially simple, testable formulae and clear domains of validity. It may also help establish a more concrete understanding that would enable engineers, auditors, and policymakers to engage together in discussions that build trust in generative AI's wide-scale deployment across society [5, 6].

Our focus is on how and when a toy Transformer model exhibits tipping points in its output response to a given input. Each tipping point is an instance during the system's response to a user's prompt, where the choice of next token suddenly switches from one type of content (e.g. type B content which could be desirable) to another type of content (e.g. type D content which could be undesirable) so that, in symbol form, the output looks like . . . B B B B D D D . . . As we illustrated in the accompanying paper [1], adopting a coarse-grained picture of the output means that these symbols (tokens) can represent phrases or sentences of a given type instead of just single words. As a point of reference, FIGURE 1(i) shows examples of GPT-2's attractor-like output behaviors that we would eventually like to understand and explain. Depending on the temperature T' (and hence the effective breadth of search) at the final layer during next-token selection, different attractor-like states emerge in a somewhat transition-like (i.e. non-smooth) way. They exhibit occasional tipping points between competing attractor states, among other complex behaviors.

We stress that the output in FIGURE 1(i) is not some cherry-picked, one-off example: this behavior is very common across LLMs. To prove this to readers, we offer the Python code used to obtain FIGURE 1(i). Using this code, readers will be able to verify for themselves that similar results occur across all open-source LLMs, in particular in the low temperature limit approaching greedy decoding as discussed in the paper. The LLM used (in FIGURE 1) is a GPT-2 model – but other such open-source LLMs produce similar types of behavior as in FIGURE 1, as can be checked directly with the code that we are offering readers. We note that commercial LLMs would also natively produce such tipping as well, but are likely prevented from doing so by external controls such as repetition penalties, fine- tuning etc. Although the empirical GPT-2 output is not quantitatively identical to the bottom-up two-Attention head model that we study, the qualitative similarity in terms of outputs at low temperatures (toward the greedy decoding limit) is encouraging. Whatever the respective pair of two competing content types during a given iteration of next-token generation (e.g. B vs. D) each tipping point reflects the competition between them where each plays the role of B or D. Hence the formula that we derived in the accompanying paper [1], and its generalizations that we hint at in this paper, can be applied to each tipping point successively.

2. BACKGROUND

The leap from our tiny, transparent Transformer 'molecule' to a massive LLM is significant: but recent empirical research indeed suggests that scalability exists in LLMs. For example, studies in Mechanistic

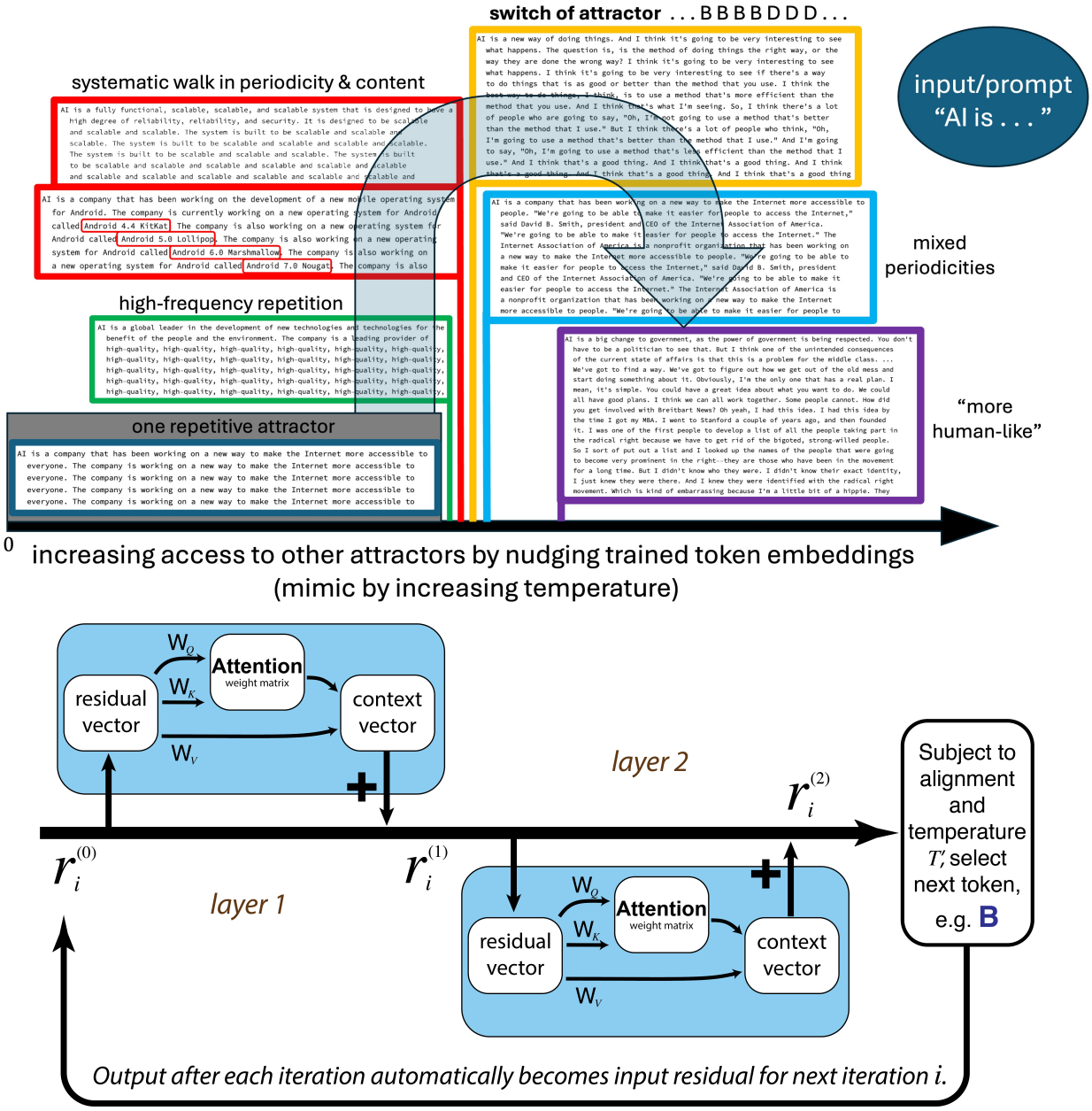

Figure 1: Top panel (i) gives illustrative output from an example LLM (GPT-2) as a function of increasing next-token-picking temperature T' on the horizontal axis – and hence as its access to other internal attractor states increases (left to right, schematic). While the observed periodicity and systematic walks in the output content may also be reproducible from our mathematical analysis, this paper's focus is on sudden switches in the output's attractor which can happen long into its generative response (..BBBBDD... transition at token iteration $n^* \gg 1$). Bottom panel (ii) shows the 'molecule' Transformer model studied in this paper. It consists of two single-head Attention layers (i.e. two 'atoms' operating in series) each of which is identical to the one introduced in Ref. [1]. They both contribute to the residual stream, and – as in full-scale LLM's – the residual vector after the final layer selects the output token based on alignment and the decoding temperature dial value T'.

Interpretability (MI) have shown that despite their massive scale, LLMs' behavior often hinges on a surprisingly small and specialized subset of its components [11–13]. Specifically, phenomena like head redundancy (where a fraction of heads can be removed with minimal performance loss) and layer redundancy (as shown by layer distillation and early exiting techniques) imply that a single 'effective head' or a small handful of layers may dominate the computational role for a given task. Our simplified few-head–few-layer models do at least capture the directional logic of how content flows and accumulates through the residual stream. They also provide a minimal testbed for the multi-agent perspective where Attention heads are viewed as interacting agents coupled via the residual stream, whose collective behavior may govern emergent states like coordination or competition. Hence understanding the LLM's predictable behaviors at the scale of the 'atom' (one layer with one head) and 'molecule' (two layers with one head per layer as in FIGURE 1(ii)) is a credible first step toward understanding and perhaps even predicting the emergent dynamics of the entire LLM 'material'.

In summary, our bottom-up, physics-inspired approach complements but differs fundamentally from the dominant paradigm of circuit-level Mechanistic Interpretability. We now further clarify this relationship in order to better position our contribution within the broader interpretability research landscape.

2.1 Mechanistic Interpretability: The Mesoscopic View

The Mechanistic Interpretability community has achieved remarkable success in reverse-engineering specific computational motifs within trained Transformers into human-understandable algorithms [7–10, 14–23]. Elhage et al. (2021) [7], established a mathematical framework for analyzing Transformer components, decomposing the residual stream and attention patterns into interpretable circuits. Building on this foundation, Olsson et al. (2022) [8], identified "induction heads"—specific attention patterns that enable in-context learning through a copy-paste mechanism. Wang et al.(2022) [18], mapped complete circuits for indirect object identification in GPT-2 small, demonstrating how individual heads coordinate to solve linguistic tasks.

More recently, circuit discovery has become increasingly automated. Conmy et al. (2023) [19], developed methods for systematically identifying minimal subgraphs responsible for specific behaviors, while Ameisen et al. (2025) [20], introduced attribution graphs that trace computational flow through the network. Cunningham et al. (2023) [15], showed that sparse autoencoders can extract interpretable features from the residual stream, and recent work by Soo et al. (2025) [16], and Li et al. (2025) [17], has demonstrated that these features can be used for targeted model steering.

This mesoscopic approach—identifying which heads and layers implement which algorithms—has proven invaluable for understanding specific capabilities, e.g. mapping specific circuits for behaviors like in-context learning and providing a mathematical framework for Transformer components. However, as Rai et al. (2024) [10], note in their comprehensive review, these circuit-level explanations often remain tied to particular model architectures and tasks, making it challenging to extract general, predictive principles that transfer across contexts.

2.2 Our Complementary Microscopic Approach

In contrast to the mesoscopic, circuit-level MI approach, our complementary line of inquiry adopts a reductionist, bottom-up, strategy, borrowed from physics and chemistry: isolate the fundamental 'atom' (the Attention head), derive from first principles clean, robust, and potentially predictive laws about their core behaviors, like tipping points; then systematically compose these atoms into "molecules" (multi-head, multi-layer systems, illustrated in FIGURE 1(ii)) to understand emergent collective behavior. This framework is

not a replacement for circuit-mapping but a complement, offering potentially simple, testable formulae and clear domains of validity.

This approach offers three key advantages that complement MI's circuit-mapping paradigm:

(1) Closed-form predictive equations. Rather than empirically discovering that certain heads perform certain functions, we derive mathematical expressions (e.g., Eq. 1 for the tipping point n^*) that predict when behavioral transitions will occur based solely on the geometry of embedding vectors. These equations provide quantitative predictions that can be tested across different prompts, vocabularies, and model configurations.

(2) Transparent domains of validity. Our simplifications (identity matrices, no LayerNorm, etc.) are explicit methodological choices that define clear boundaries for where our analysis applies. This transparency allows future work to systematically relax each assumption and measure its impact—a principled path for scaling toward realism that parallels how physicists build from ideal gas laws to real gases to condensed matter.

(3) Emergent collective dynamics. By treating attention heads as interacting agents coupled through the residual stream, our framework naturally suggests hypotheses about coordination (fixed points), competition (oscillations), and phase transitions (tipping points) that emerge when multiple heads interact. While MI identifies what individual circuits do, our approach may help predict how they interact to produce system-level behaviors like output instability.

2.3 Empirical Support for the Reductionist Methodology

Our minimalist approach finds support in recent empirical findings about Transformer redundancy and sparsity:

Head redundancy. Michel, Levy, and Neubig (2019) [11], demonstrated that most attention heads can be pruned without significant performance degradation, suggesting that a small subset carries the primary computational load. Voita et al. (2019) [21], showed that specialized heads handle the 'heavy lifting', while others contribute minimally. This implies that for many tasks, the effective dynamics may be dominated by just one or a few critical heads—precisely the regime where our single-head analysis becomes relevant.

Layer redundancy. Distillation studies reveal similar sparsity across layers. Sanh et al. (2019) [12], showed that DistilBERT, with half the layers of BERT, retains over 95% of performance. Fan, Grave, and Joulin (2019) [13], introduced LayerDrop, demonstrating that random layer pruning has minimal impact, and early-exit techniques show that confident predictions often emerge from intermediate layers [11]. These findings suggest that for specific inputs or tasks, a single "effective layer" may capture the dominant computation.

Parameter sparsity. Frantar and Alistarh (2023) [22], demonstrated that massive language models can be pruned to extreme sparsity in one shot, while Dettmers and Zettlemoyer (2022) [23], showed that 4-bit precision often suffices for inference. This suggests that the high-dimensional parameter space of LLMs contains substantial redundancy, and that lower-dimensional projections—like our minimal models—may capture essential dynamics.

2.4 Synthesis: From Atoms to Molecules to Materials

The relationship between our work and mechanistic interpretability mirrors the relationship between quantum chemistry and materials science. MI's circuit-level analyses are akin to characterizing specific molecular structures and reaction mechanisms—essential for understanding what compounds exist and how they behave. Our bottom-up approach is analogous to deriving chemical bonding from first principles (quantum mechanics of electrons and nuclei), then using those principles to predict when new phases of matter will emerge. Both approaches are necessary. MI provides the empirical grounding and identifies which circuits matter in practice. Our framework offers complementary theoretical tools: predictive equations with explicit domains of validity, and a conceptual scaffold for reasoning about collective dynamics as we scale from one head to many heads to full LLMs. Together, they form a more complete interpretability toolkit—one that combines empirical discovery with principled mathematical prediction.

3. TIPPING POINT IN PRESENCE OF A RICHER VOCABULARY

It was shown in the accompanying paper [1], that tipping points can occur for a single attention head, and that increasing the vocabulary (e.g. adding a C as in FIGURE 5 of that paper [1]) can introduce multiple tipping points which happen sequentially. FIGURE 2 provides an illustrative example of what happens when many more embedding vectors (i.e. larger vocabulary) are added and we increase the embedding dimension. Since these extra tokens and hence embedding vectors are presumed to introduce new meanings and concepts, FIGURE 2(a) represents them as approximately orthogonal to the B–D subspace. Our simulations (of which FIGURE 2, is just one example) then suggest that even if there are large numbers of additional near-perpendicular embedding vectors, the original tipping point may persist. Indeed in the case shown in FIGURE 2, the tipping point still occurs at exactly the same $n^* = 3$ value predicted by Eq. 1 in the accompanying paper [1], which we reproduce here in a slightly simpler form by writing the token A's embedding vector as $\boldsymbol{A}$ etc.:

$$n^* = \frac{m e^{\boldsymbol{B}\cdot\boldsymbol{A}/T}(\boldsymbol{A}\cdot\boldsymbol{B} - \boldsymbol{A}\cdot\boldsymbol{D})}{e^{\boldsymbol{B}\cdot\boldsymbol{B}/T}(\boldsymbol{B}\cdot\boldsymbol{D} - \boldsymbol{B}\cdot\boldsymbol{B})} \quad . \tag{1}$$

where m is the number of A tokens in the prompt and T is the general temperature parameter in the softmax associated with the Attention score calculation. We stress that T is unrelated to T' which is the temperature for next-token selection: T' is taken to be very small so that we have greedy decoding, but T can have any value. Only the vectors for B and D play a dominant role near the tipping point because they are the only ones that have similarly large dot products with the context vector. $n^* = 3$ as before. More generally, the precise value of n^* will change, but the tipping point still occurs. Future work will look in detail at how n^* changes numerically as new embedding vectors are added to the vocabulary, together with the interesting issue of the effect of more embedding vectors versus more embedding dimensions.

4. TIPPING POINT IN PRESENCE OF NON-IDENTITY TRAINING MATRICES

One of the main shortcomings of the basic analysis in our accompanying paper [1], is that we considered the fixed matrices W learned during training to be identity matrices. This can easily be generalized: it affects Eq. 1 by sandwiching in W matrices to the dot products, hence making the equation quite messy. FIGURE 3 illustrates numerically the impact on the tipping point of the basic Attention head. Interestingly, the tipping point survives, albeit at a different value of n^*. This is easily understood: the W matrices act

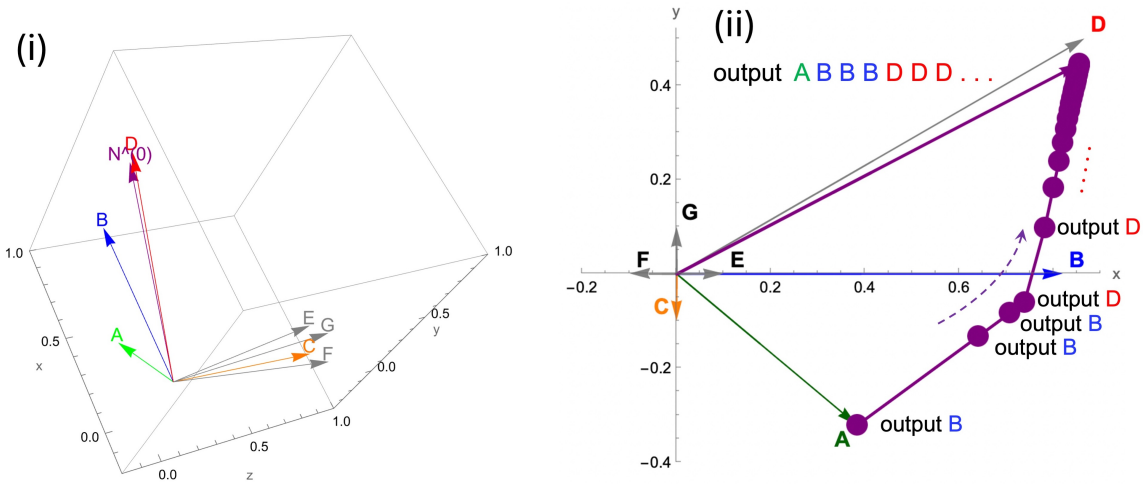

Figure 2: Impact on the output dynamics of the basic Attention head of (i) more embedding vectors (i.e. larger vocabulary) and more embedding dimensions. (ii) the original tipping point persists. In this case the numerical value ($n^* = 3$) happens to remain the same as the simple case just involving A, B and D, but in general this value of n^* will differ. However it can still be calculated, using a slightly more general form of Eq. 1. We stress that this is a conceptual proof-of-principle work: panel (i) simply points out that the addition of near orthogonal vocabulary embeddings does not affect our finding – which is that the 2 Attention heads (i.e. 2 layers of a single Attention head) behave similarly to the 1 Attention head case, and hence that 1 Attention head properties may help inform multi-Attention head settings.

like a transformation of the embedding vectors, akin to shifting axes. So the criterion and calculation of the tipping point will remain the same, but will simply be shifted by the W transformations. Hence FIGURE 3, shows similar tipping with shifted n^* because learned W's induce bilinear scores and values. The structural form of the Eq. 1 criterion n^* is unchanged after substituting these, but the specific numerical value of n^* and hence location of tipping points will shift with the learned bilinear forms.

5. TIPPING POINT IN PRESENCE OF FINITE TEMPERATURE T' FOR NEXT-TOKEN SELECTION

Greedy decoding (i.e. $T' = 0$) was used for next-token selection in the accompanying paper [1], and in the derivation of Eq. 1, for convenience. We stress that this temperature T' is totally separate from the temperature T appearing in the softmax Attention calculation leading to Eq. 1, which is completely general in its value. FIGURE 4 presents the numerical results of code that we developed (available on reasonable request from the authors) for a more realistic case of finite temperature T' next-token selection. We note that it also includes the correct masking and scaling. For moderate T', tipping persists, with increasing T' mainly adding stochastic mixing around the tipping point boundary. As the temperature T' increases, the effect as shown is to add disorder (noise) that stochastically mixes the B and D ordering near the tipping point and can add in additional symbols (i.e. A in this case). This is analogous to the effect of adding disorder with increasing temperature in real materials. The basic occurrence of a tipping point still persists in FIGURE 4, albeit with an increasingly blurred boundary. We point to Ref. [24], for related discussions. We also direct users to the website `https://d-ai-ta.netlify.app` where this simulation is freely available to explore.

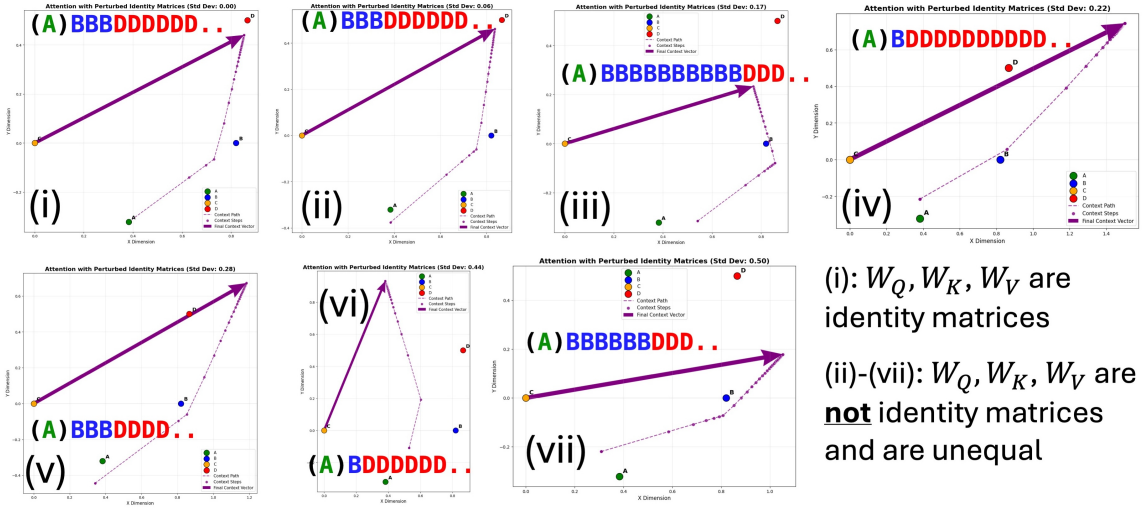

Figure 3: Impact on the output dynamics of the basic Attention head, of non-identity fixed matrices W which were learned from the training. The tipping point survives, albeit at a different value of n^*. This is because the structural form of the Eq. 1 criterion n^* is unchanged after substituting non-identity W's, but the specific numerical value of n^* and hence location of tipping points generally shifts with these learned bilinear forms.

6. TIPPING POINT IN PRESENCE OF MULTILAYER ATTENTION HEADS

LLMs contain layers of Attention heads. Within a layer, the Attention heads operate independently—but between layers there is in general coupling since the tokens' vector representations are getting carried forward through each layer successively. This is the aspect we explore here in a very simple but methodological way, by considering a two-layer 'molecule' and hence a **tiny toy Transformer with one Attention head per layer and two layers** as in FIGURE 1(ii). This may sound far too simple to be useful: but as noted in our accompanying paper [1], as well as in Sec. 2.3 above, LLMs' behavior is often driven by a surprisingly small and specialized subset of their components. This simplified setup allows for a completely transparent, pencil-and-paper understanding of how such a tiny Transformer works – albeit with some additional simplifications mentioned below that we adopt solely for the purpose of displaying tractable mathematical expressions in this paper.

The Appendix lays out in detail this mathematics and hence understanding explicitly—but we stress that the paper can be read and understood without this. The mathematics for our tiny transparent Transformer deliberately omits several components of a standard Transformer architecture that will be considered in future work. (1) LayerNorm (LN): Real transformers apply LayerNorm (pre-norm in most modern LLMs) which rescales/centers the residual before attention/MLP. LN affects stability and relative scaling across layers. Our mathematics omits LN to keep the arithmetic transparent and make it fit in the confines of this paper: hence it can over-amplify magnitudes across layers (e.g. doubling effects when there is only one token). In real models, LN curbs runaway growth and re-centers features. (2) Feed-Forward Networks (MLP/FFN): FFNs contribute non-linear feature synthesis and behave like key-value associative memories that can retrieve and insert facts into the residual stream. The above mathematics omitted this to focus on the Attention geometry. This means that tasks relying on strong non-linear composition or factual retrieval won't be captured; however, the directional bias in the residual due to Attention still illustrates how token similarity steers next-token odds. (3) Non-identity projections and unembedding: In practice W_Q, W_K, W_V, and W_U are learned and non-identity. Setting them to be identity matrices makes the Attention weights

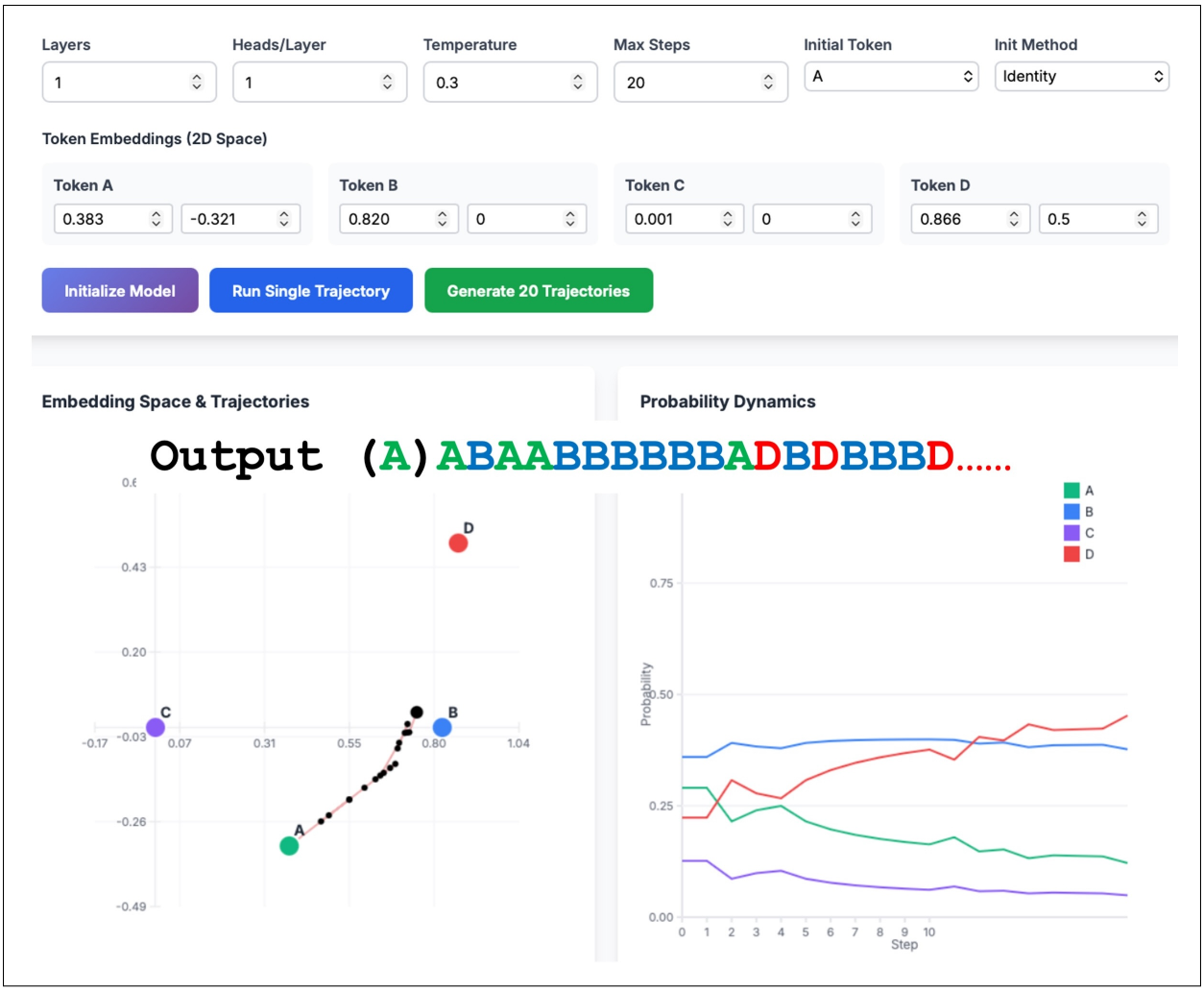

Figure 4: Impact on the output dynamics of the basic Attention head, of finite decoding temperature T' for next-token selection. This is a snapshot from the numerical calculation that we provide as a dashboard for any reader to explore at `https://d-ai-ta.netlify.app`. As long as the temperature T' is not too high, the tipping point persists, albeit with an increasingly blurred boundary as shown. Right panel shows the corresponding next-token probabilities.

purely functions of residual self-similarities and makes logits literal dot-products with token vectors. Our mathematics reused the embedding space as both the key/value and logit space, yet real models learn separate (but related) spaces. However, FIGURE 3 showed for the single Attention head that the tipping point can still persist when adding in real W matrices, albeit at different n^* values.

We stress that our model's simplifications are a methodological choice to create a geometric proxy that yields predictions about directional bias, which can then perhaps be explored in more complex, real-world models. Our 2-layer 'molecule' model (FIGURE 1(ii)) with one head per layer, identity projections, and a residual stream that simply adds the context vector at each layer, serves to isolate three mechanisms that are central in real models: (1) Residual stream as the main state. Each layer takes what the model 'currently believes' (the residual vector) and adds in a context vector computed via Attention from earlier tokens. This is how

real Transformers update the residual stream. (2) A few heads/layers often dominate, as stated in Sec. 2.3 with significant support in the literature. (3) Dot-product geometry predicts next-token bias. If we read out the residual with the unembedding, the relative dot products with word vectors govern the next-token probabilities. Our model shows these dot products evolving across layers in a transparent way that mirrors the real mechanism. So while simplified, our model captures the directional logic of how content flows and accumulates through a transformer, and why a single effective head or a handful of layers can sometimes explain most of the observed behavior. For example, our tiny transparent Transformer model lets us see when and why the system's internal state swings toward B or D.

Scaling our tiny transparent Transformer to more than two layers, as we will do in a future work, suggests a view of Attention heads as interacting agents coupled via the residual stream. We expect that their collective behavior can lead to coordination (fixed points), competition (oscillations), and coalition shifts (mixed regimes), with the two-layer molecule version serving as the minimal testbed for this multi-agent perspective.

7. EMPIRICAL VALIDATION AND PERFORMANCE IMPROVEMENTS

Having demonstrated tipping-point persistence under various generalizations (Sections 2–5), we now present empirical evidence that our theoretical insights might potentially improve real-world LLM performance. We emphasize that these results are preliminary and require extensive follow-up validation.

7.1 Theoretical Motivation for Interventions

Despite its obviously crude nature, our approach of following the mathematical effects of the Attention process for simplistic models, already suggests the following design improvements for generative AI systems: **(1) Gap Cooling.** Equation 1 reveals that tipping points n^* occur when the dot-product gap between competing tokens (e.g., $\langle \text{context}, B\rangle - \langle \text{context}, D\rangle$) flips sign. If the gap between the top two next-token logits becomes too small (indicating imminent tipping or instability), one should artificially widen the gap by scaling the top logit upward or the second logit downward. This is analogous to quenching in materials science—rapidly stabilizing a desired phase before the system tips into an undesired one. Formally, if $\ell_1 - \ell_2 < \epsilon_{\text{gap}}$ for logits ℓ_1, ℓ_2, we replace $\ell_1 \to \ell_1 + \lambda(\epsilon_{\text{gap}} - (\ell_1 - \ell_2))$. **(2) Temperature Annealing.** the sampling temperature T' during next-token selection controls the breadth of search. Dynamically adjusting T' during generation—starting high to explore diverse paths, then cooling to stabilize on coherent output—may prevent premature tipping into low-quality attractors while avoiding excessive randomness.

7.2 Proof-of-Concept Results on GPT-2

In order to explore the above strategies and their potential impact, we override GPT-2's standard decoding with two custom functions: Gap Cooling and Temperature Annealing.

Experimental setup. We evaluate on the TruthfulQA benchmark [25], which tests whether models generate truthful answers or mimic common human misconceptions. This benchmark is well-suited for testing output stability because tipping from a "truthful" attractor to a "misleading" attractor corresponds to exactly the kind of mid-response failure we aim to prevent. We use ROUGE-1 F1 score as our primary metric, comparing (1) a **greedy** baseline (i.e. $T' = 0$), (2) a **constant finite** $T' = 0.5$, with no annealing or gap cooling, (3) an **Annealing only** (i.e. no gap cooling) scheme with temperature dial following $T'(n) = 1.5 \cdot 1.2\exp(-n/30)$,

and (4) **Annealing + Gap Cooling** combined: $T'(n)$ as above, while gap threshold $\epsilon_{\text{gap}} = 0.05$ and scaling factor $\lambda = 3$.

Results. FIGURE 5 shows ROUGE-1 scores averaged over 817 TruthfulQA questions. The combined Annealing + Gap Cooling condition achieves ROUGE-1 = 0.404, representing a 26% improvement over Greedy (0.320) and 2% improvement over Annealing alone (0.396). Constant temperature $T' = 0.5$ without annealing performs worse than all other conditions (0.333), suggesting that naive temperature increases introduce instability without the stabilizing effect of adaptive cooling.

These preliminary results suggest that our geometric understanding of tipping points can inform practical interventions. Gap Cooling appears to stabilize generation near decision boundaries, preventing premature tipping into misleading or repetitive attractors. Temperature Annealing allows early exploration of the answer space before converging to a stable response, leveraging the theoretical insight that T' controls access to competing attractors (FIGURE 4).

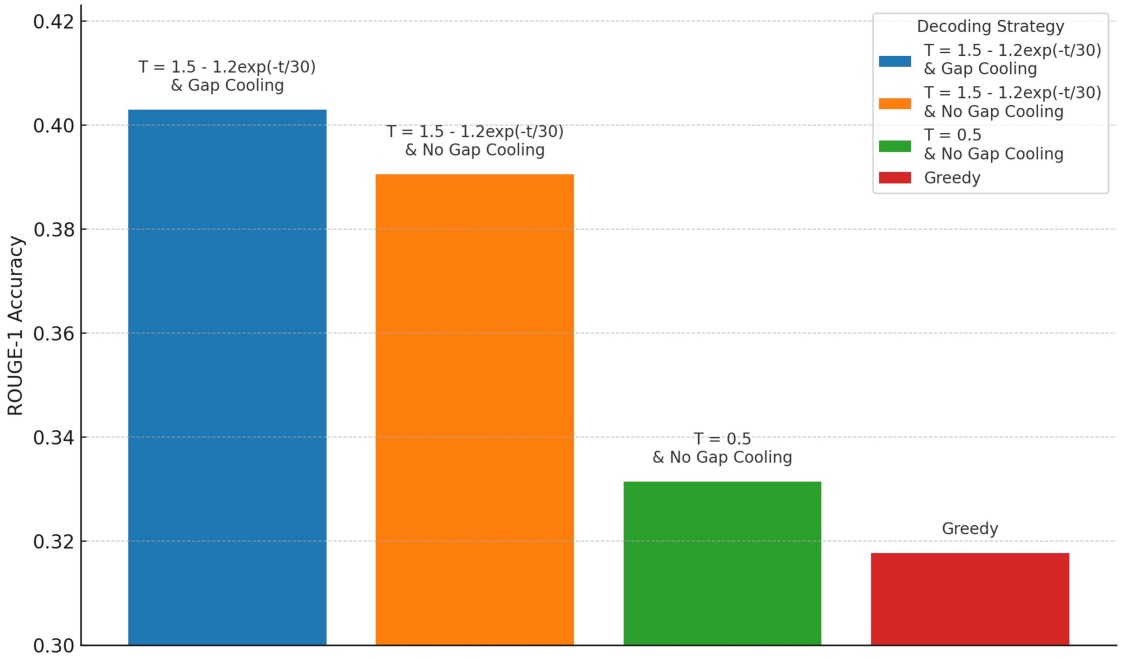

Figure 5: ROUGE-1 scores of different decoding strategies.

We freely acknowledge that this above analysis including FIGURE 5, is obviously just a first step, as a proof-of-principle. It serves a heuristic purpose and hence does not aim at providing full statistical rigor, complete hyperparameter search, or covering a wide range of benchmarks and baseline comparisons. As such we acknowledge that the results in FIGURE 5, are purely a suggestive pilot result that needs more exploration. To really connect this to practical impact, the link between theoretical tipping point analysis and real-world improvements (e.g., gap cooling, temperature annealing) needs more detailed experiments and comparisons with baseline approaches. We leave these to future work.

8. LIMITATIONS AND FUTURE WORK

Our current work establishes a conceptual proof-of-principle: that tipping-point dynamics can persist from single attention heads to more complex multi-head, multi-layer configurations, and that these dynamics may

inform practical interventions. However, gaps remain between this minimal 'molecule'-level framework and a predictive understanding of production LLMs. We enumerate these limitations and propose concrete research directions.

1. Architectural Simplifications. Though our reductionist methology is validated as stated in Sec. 2.3, and has indeed provided key insights, future perturbation or extension of the model to include modules such as LayerNorm, MLP/FFN layers, and multiple Attention heads mediated (averaged over) by non-identity pre-trained W_U matrices, would improve the precision of our current model.

2. Experimental and Statistical Rigor. As has been discussed in Sec. 7.2, while detailed statistical analyses on empirical improvements are beyond the scope of this paper, future work would expand on the choices of hyperparameter, ranges of benchmarks and statistical rigor.

3. Scalability and Generalization. A central question for our minimalist framework is whether insights from single-head and two-head systems can inform understanding of production LLMs with dozens of heads per layer and dozens of layers. Though our theory is rigorously derived from first principle and does predict phenomena seen in GPT-2, the apparent gap between the scales of our toy model and production LLMs would be better filled with future research e.g. ablation studies on production LLMs using attention knockout [11], coarse-graining methods aimed for effective theories, and empirical observations on larger vocabulary size and longer context length. Nevertheless, we see theoretical support for scalability: as Sec. 2.3 mentions, the effective head redundancy, and hence compression, means not only fewer effective heads [11, 21], but also earlier emergence [12, 13]. Moreover, the fact that residual stream works as a shared communication channel, to which all Attention heads contribute, means that the residual stream can be seen as acting as a 'mean field' that aggregates head contributions, with our formula for n^* (Equation 1) capturing a universal transition criterion, albeit with modified effective vectors that account for multi-head mixing.

4. Decoding temperature. Much of our analysis assumes a low-temperature limit ($T' \to 0$), close to greedy decoding. Future studies would expand this to behaviors under finite temperature T', to better reflect realistic decoding strategies.

5. Broader Implications for Safety and Alignmen. Beyond performance metrics, our tipping-point framework offers a potential lens for understanding and mitigating alignment failures. If harmful or misleading outputs arise from tipping points where the model switches from a "safe" attractor (B-type content: helpful, honest, harmless) to a "dangerous" attractor (D-type content: toxic, deceptive, or harmful), then real-time monitoring of dot-product gaps and adaptive interventions could serve as a lightweight safety layer. Recent work on interpretable steering [16, 17], shows that targeted activation modifications can guide model behavior toward desired outcomes. Our geometric perspective suggests that steering interventions might be most effective when applied near predicted tipping points n^*—precisely the moments when small nudges can prevent large behavioral deviations. Future work should explore whether combining our tipping-point detection with activation steering yields synergistic improvements in both capability and safety.

9. CONCLUSION

This paper has looked at various generalizations of the basic Attention head model discussed in an accompanying paper [1], with a view to gently advancing toward a bottom-up understanding of generative AI like ChatGPT. Obviously our analysis is still very far from a real LLM – we have continually stressed this throughout the paper. But we do provide evidence that suggests the output tipping behavior that we reported

for the basic Attention head can persist as we scale up in complexity by adding some of the complications of real LLMs. Specifically, we considered here (1) a richer vocabulary, (2) non-identity matrices learned during pre-training, (3) finite temperature during next-token selection, (4) multilayer Attention heads. We then gave some preliminary evidence that the insights gained from this bottom-up approach, can be useful for performance improvements in real-world generative AI systems. We also pointed throughout the paper to next steps for this line of bottom-up, minimalist physics-style research with the goal of better understanding ChatGPT-like generative AI.

References

[1] Restrepo DJ, Huo FY, Restrepo NJ, Johnson NF. Basic Attention Head as a Building Block toward Understanding Transformer-based Generative AI. Adv Artif Intell Mach Learn. 2025. (Ahead of print). https://dx.doi.org/10.54364/AAIML.2025.54251

[2] Vaswani A, Shazeer N, Parmar N, Uszkoreit J et al. Attention Is All You Need. Adv Neural Inf Process Syst. NeurIPS. 2017. ArXiv Preprint: https://arxiv.org/pdf/1706.03762

[3] https://today.yougov.com/technology/articles/49099-americans-2024-poll-ai-top-feeling-caution.

[4] https://www.nytimes.com/2024/10/23/technology/characterai-lawsuit-teen-suicide.html.

[5] Johnson NF, Huo FY. Multispin Physics of AI Tipping Points and Hallucinations. 2025. ArXiv preprint: https://arxiv.org/pdf/2508.01097

[6] Johnson NF, Huo FY. Jekyll-And-Hyde Tipping Point in an Ai's Behavior. 2025. ArXiv preprint: https://arxiv.org/pdf/2504.20980.

[7] https://transformer-circuits.pub/2021/framework/index.html

[8] Olsson C, Elhage N, Nanda N, Joseph N, Olah C, et al. In-Context Learning and Induction Heads. 2022. ArXiv preprint: https://arxiv.org/pdf/2209.11895

[9] Nanda N, Chan L, Lieberum T, Smith J, Steinhardt J. Progress Measures for Grokking via Mechanistic Interpretability. 2023. ArXiv preprint: https://arxiv.org/pdf/2301.05217

[10] Rai D, Zhou Y, Feng S, Saparov A, Yao Z. A Practical Review of Mechanistic Interpretability for Transformer-Based Language Models. 2024. ArXiv preprint: https://arxiv.org/pdf/2407.02646

[11] Michel P, Levy O, Neubig G. Are Sixteen Heads Really Better Than One?. 2019. ArXiv preprint: https://arxiv.org/pdf/1905.10650

[12] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT a Distilled Version of BERT: Smaller Faster Cheaper and Lighter. 2019. ArXiv. https://arxiv.org/pdf/1910.01108

[13] Fan A, Grave E, Joulin A. Reducing Transformer Depth on Demand With Structured Dropout. LayerDrop. 2019. ArXiv preprint: https://arxiv.org/pdf/1909.11556

[14] Rogers A, Kovaleva O, Rumshisky A. A Primer in BERTology: What We Know About How BERT Works. Trans Assoc Comput Linguist. 2020;8:842-866.

[15] Cunningham H, Ewart A, Riggs L, Huben R, Sharkey L. Sparse Autoencoders Find Interpretable Features in the Residual Stream. 2023. ArXiv preprint: https://arxiv.org/pdf/2309.08600

[16] Soo S, Guang C, Teng W, Balaganesh C, et al. Interpretable Steering of Large Language Models With Feature Guided Activation Additions. 2025. ArXiv preprint: https://arxiv.org/pdf/2501.09929

[17] Li Z, Wang X, Yang Y, Yao Z, Xiong H, et al. Feature Extraction and Steering for Enhanced Chain-Of-Thought Reasoning in Language Models. 2025. ArXiv preprint: https://arxiv.org/pdf/2505.15634.

[18] Wang K, Variengien A, Conmy A, Shlegeris B, Steinhardt J. Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small. 2022. ArXiv preprint: https://arxiv.org/pdf/2211.00593

[19] Conmy A, Mavor-Parker A, Lynch A, Heimersheim S, Garriga-Alonso A. Towards Automated Circuit Discovery for Mechanistic Interpretability. Adv Neural Inf Process Syst. NeurIPS. 2023;36:16318-16352.

[20] https://transformer-circuits.pub/2025/attribution-graphs/methods.html

[21] Voita E, Talbot D, Moiseev F, Sennrich R, Titov I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In: Annual Meeting of the Association for Computational Linguistics. ACL. 2019. ArXiv preprint: https://arxiv.org/pdf/1905.09418

[22] Frantar E, Alistarh D, SPARSEGPT. Massive Language Models Can Be Accurately Pruned in One-Shot. 2023. ArXiv preprint: https://arxiv.org/pdf/2301.00774

[23] Dettmers T, Zettlemoyer L. The Case for 4-Bit Precision: K-Bit Inference Scaling Laws. 2022. ArXiv preprint: https://arxiv.org/pdf/2212.09720

[24] Holtzman A, Buys J, Du L, Forbes M, Choi Y. The Curious Case of Neural Text Degeneration. In: International Conference on Learning Representations. International Conference on Learning Representations ICLR. 2020. ArXiv preprint: https://arxiv.org/pdf/1904.09751

[25] Lin S, Hilton J, Evans O. TruthfulQA: Measuring How Models Mimic Human Falsehoods. 2021. ArXiv preprint: https://arxiv.org/pdf/2109.07958

Appendix A. Mathematical details

For this multi-layer system, we expand the mathematical approach of the accompany paper by writing the residual vectors for layer ℓ and token position t as $\boldsymbol{r}_t^{(\ell)}$, the context vector updates $\boldsymbol{c}_t^{(\ell)}$, and the Attention coefficients $\alpha_{t,j}^{(\ell)}$. For continuity, we consider a similar three-token vocabulary as used in the accompanying paper's [1], analysis of a single Attention head and as used in FIGURE 2 (and FIGURE 4), of this paper: $\boldsymbol{A} = (0.383, -0.321, 0)$, $\boldsymbol{B} = (0.820, 0, 0)$ and $\boldsymbol{D} = (0.866, 0.500, 0)$. These vectors gave the output ABBBDDD . . . and tipping point $n^* = 3$ from Eq. 1, and our goal from presenting this detailed mathematics is to understand exactly what the effect of more than one layer is on this tipping point and its n^* value. We now repeat this analysis for clarity here using the prompt token A (i.e. $m = 1$ in Eq. 1). $\boldsymbol{A} \cdot \boldsymbol{B} = 0.383(0.820) + (-0.321)(0) = 0.314$, $\boldsymbol{A} \cdot \boldsymbol{D} = 0.383(0.866) + (-0.321)(0.5) = 0.171$, $\boldsymbol{B} \cdot \boldsymbol{B} = 0.820^2 = 0.672$, $\boldsymbol{B} \cdot \boldsymbol{D} = 0.820(0.866) = 0.711$. Taking $T = 1$ in Eq. 1 yields

$$n^* = \frac{e^{0.314}(0.314 - 0.171)}{e^{0.672}(0.711 - 0.672)} \approx 2.6 \quad \Longrightarrow \; n^* = 3$$

which means that the mathematics predicts the output string to be ABBBDDD. . . and tipping point $n^* = 3$.

As before, we start with the prompt A and we auto-regressively generate subsequent tokens by applying both layers and greedy decoding from the final residual vector at each step. For simplicity in the presentation of the mathematics in this paper, we consider identity matrix W's, no layer normalization (LayerNorm) and no multilayer perceptron (MLP, FFN). They could be included but the mathematics becomes more clumsy.

Hence we are taking the residual update as simply $\boldsymbol{r}_t^{(\ell)} = \boldsymbol{r}_t^{(\ell-1)} + \boldsymbol{c}_t^{(\ell)}$. We do incorporate the real feature of causal attention, i.e. the token at position t can attend only to tokens at positions $\leq t$. We also consider the realistic feature of scaled dot-product Attention with scale $1/\sqrt{d}$, where $d = 3$. The logits for candidate next token $X \in \{A, B, D\}$ are $\langle \boldsymbol{r}_t^{(2)}, \boldsymbol{X} \rangle$ where t is the final token position and two layers have been applied. As before, greedy decoding selects $\arg\max_X$. We note that depending on the calculation precision used to check our results below, one may find irrelevant rounding errors: but they have no effect on the output or conclusions or n^* value obtained which are all robust. We also purposely choose to show mixed arithmetic precision (mixed number of decimal places) so that readers can check they reproduce the numbers in full at particular stages, and yet the paper does not get overloaded with high precision numbers for every step. The final results and conclusions are all correct.

At layer $\ell \in \{1, 2\}$ and position t in the token sequence, we hence have the query, key and value as $\boldsymbol{q}_t = \boldsymbol{r}_t^{(\ell-1)}$, $\boldsymbol{k}_j = \boldsymbol{r}_j^{(\ell-1)}$, $\boldsymbol{v}_j = \boldsymbol{r}_j^{(\ell-1)}$. The Attention scores are $s_{t,j}^{(\ell)} = \frac{\langle \boldsymbol{q}_t, \boldsymbol{k}_j \rangle}{\sqrt{3}} = \frac{\langle \boldsymbol{r}_t^{(\ell-1)}, \boldsymbol{r}_j^{(\ell-1)} \rangle}{\sqrt{3}}$, $\alpha_{t,j}^{(\ell)} = \frac{e^{s_{t,j}^{(\ell)}}}{\sum_{i \leq t} e^{s_{t,i}^{(\ell)}}}$, and the context vector and residual vector get updated as follows:

$$\boldsymbol{c}_t^{(\ell)} = \sum_{j \leq t} \alpha_{t,j}^{(\ell)} \boldsymbol{v}_j = \sum_{j \leq t} \alpha_{t,j}^{(\ell)} \boldsymbol{r}_j^{(\ell-1)}, \qquad \boldsymbol{r}_t^{(\ell)} = \boldsymbol{r}_t^{(\ell-1)} + \boldsymbol{c}_t^{(\ell)}.$$

<u>Iteration 1.</u> Prompt A hence Input Sequence is A:

Layer 0 (input).

$$\boldsymbol{r}_1^{(0)} = \boldsymbol{A} = (0.383, -0.321, 0).$$

Layer 1 (self only at $t = 1$).

$$\|\boldsymbol{r}_1^{(0)}\|^2 = 0.383^2 + (-0.321)^2 = 0.249730, \qquad s_{1,1}^{(1)} = \frac{\langle \boldsymbol{r}_1^{(0)}, \boldsymbol{r}_1^{(0)} \rangle}{\sqrt{3}} = \frac{0.249730}{1.732051} = 0.144181683.$$

$$\alpha_{1,1}^{(1)} = 1, \qquad \underbrace{\boldsymbol{c}_1^{(1)}}_{\text{context at } t=1} = \alpha_{1,1}^{(1)} \boldsymbol{r}_1^{(0)} = (0.383, -0.321, 0).$$

$$\underbrace{\boldsymbol{r}_1^{(1)}}_{\text{residual at } t=1} = \boldsymbol{r}_1^{(0)} + \boldsymbol{c}_1^{(1)} = (0.383, -0.321, 0) + (0.383, -0.321, 0) = (0.766, -0.642, 0).$$

Layer 2 (self only at $t = 1$).

$$s_{1,1}^{(2)} = \frac{\langle \boldsymbol{r}_1^{(1)}, \boldsymbol{r}_1^{(1)} \rangle}{\sqrt{3}} = \frac{\|(0.766, -0.642, 0)\|^2}{1.732051} = 0.576726731, \qquad \alpha_{1,1}^{(2)} = 1,$$

$$\underbrace{\boldsymbol{c}_1^{(2)}}_{\text{context at } t=1} = \alpha_{1,1}^{(2)} \boldsymbol{r}_1^{(1)} = (0.766, -0.642, 0),$$

$$\underbrace{\boldsymbol{r}_1^{(2)}}_{\text{residual at } t=1} = \boldsymbol{r}_1^{(1)} + \boldsymbol{c}_1^{(2)} = (0.766, -0.642, 0) + (0.766, -0.642, 0) = (1.532, -1.284, 0).$$

Greedy decode. Logits from $\boldsymbol{r}_1^{(2)}$:

$$\langle \boldsymbol{r}_1^{(2)}, \boldsymbol{A} \rangle = 4\,\langle \boldsymbol{A}, \boldsymbol{A} \rangle = 4 \times 0.249730 = 0.998920, \quad \langle \boldsymbol{r}_1^{(2)}, \boldsymbol{B} \rangle = 4\,\langle \boldsymbol{A}, \boldsymbol{B} \rangle = 4 \times 0.314060 = 1.256240,$$

$$\langle \boldsymbol{r}_1^{(2)}, \boldsymbol{D} \rangle = 4\,\langle \boldsymbol{A}, \boldsymbol{D} \rangle = 4 \times (0.383 \cdot 0.866 + (-0.321) \cdot 0.5) = 4 \times 0.171178 = 0.684712.$$

Hence the next token is B and so the input sequence to the next iteration becomes AB. Following exactly the same mathematical process, the predicted tokens for iterations 3 and 4 are also B. Hence we skip to iteration 4 where the input sequence is now ABBB.

Iteration 4. Input Sequence is now ABBB

Layer 0 (inputs).

$$\boldsymbol{r}_1^{(0)} = \boldsymbol{A}, \qquad \boldsymbol{r}_2^{(0)} = \boldsymbol{B}, \qquad \boldsymbol{r}_3^{(0)} = \boldsymbol{B}, \qquad \boldsymbol{r}_4^{(0)} = \boldsymbol{B}.$$

Layer 1.

- $t = 1$ **(self only).**

$$s_{1,1}^{(1)} = 0.144181683, \quad \alpha_{1,1}^{(1)} = 1, \quad \underbrace{\boldsymbol{c}_1^{(1)}}_{t=1} = (0.383,\ -0.321,\ 0), \quad \underbrace{\boldsymbol{r}_1^{(1)}}_{t=1} = (0.766,\ -0.642,\ 0).$$

- $t = 2$ **(attend to** $1, 2$**).**

$$s_{2,1}^{(1)} = 0.181322626, \quad s_{2,2}^{(1)} = 0.388210321, \quad \alpha_{2,1}^{(1)} = 0.448461776, \quad \alpha_{2,2}^{(1)} = 0.551538224,$$

$$\underbrace{\boldsymbol{c}_2^{(1)}}_{t=2} = (0.624022204,\ -0.143956230,\ 0), \quad \underbrace{\boldsymbol{r}_2^{(1)}}_{t=2} = (1.444022204,\ -0.143956230,\ 0).$$

- $t = 3$ **(attend to** $1, 2, 3$**).**

$$\alpha_{3,\cdot}^{(1)} = (0.289043330,\ 0.355478335,\ 0.355478335),$$

$$\underbrace{\boldsymbol{c}_3^{(1)}}_{t=3} = (0.693688065,\ -0.092782909,\ 0), \quad \underbrace{\boldsymbol{r}_3^{(1)}}_{t=3} = (1.513688065,\ -0.092782909,\ 0).$$

- $t = 4$ **(attend to** $1, 2, 3, 4$**).**

$$s_{4,1}^{(1)} = 0.181322626, \qquad s_{4,2}^{(1)} = s_{4,3}^{(1)} = s_{4,4}^{(1)} = 0.388210321,$$

$$\alpha_{4,\cdot}^{(1)} \approx \mathrm{softmax}([0.1813,\ 0.3882,\ 0.3882,\ 0.3882]) = (0.213240834,\ 0.262253055,\ 0.262253055,\ 0.262253055),$$

$$\underbrace{\boldsymbol{c}_4^{(1)}}_{t=4} = 0.213240834\,\boldsymbol{A} + 0.262253055(\boldsymbol{B} + \boldsymbol{B} + \boldsymbol{B}) = (0.726813756,\ -0.068450308,\ 0),$$

$$\underbrace{\boldsymbol{r}_4^{(1)}}_{t=4} = \boldsymbol{r}_4^{(0)} + \boldsymbol{c}_4^{(1)} = (0.820, 0, 0) + (0.726813756, -0.068450308, 0) = (1.546813756,\ -0.068450308,\ 0).$$

Layer 2.

- $t = 1$ **(self only).**

$$s^{(2)}_{1,1} = 0.576726731, \quad \alpha^{(2)}_{1,1} = 1, \quad \underbrace{\boldsymbol{c}^{(2)}_1}_{t=1} = (0.766, -0.642, 0), \quad \underbrace{\boldsymbol{r}^{(2)}_1}_{t=1} = (1.532, -1.284, 0).$$

- $t = 2$ **(attend to** $1, 2$**).**

$$s^{(2)}_{2,1} = 0.691977916, \quad s^{(2)}_{2,2} = 1.215855512, \quad \alpha^{(2)}_{2,1} = 0.371945970, \quad \alpha^{(2)}_{2,2} = 0.628054030,$$

$$\underbrace{\boldsymbol{c}^{(2)}_2}_{t=2} = (1.191834578, -0.329201603, 0), \quad \underbrace{\boldsymbol{r}^{(2)}_2}_{t=2} = (2.635856782, -0.473157833, 0).$$

- $t = 3$ **(attend to** $1, 2, 3$**).**

$$s^{(2)}_{3,\cdot} = [0.703819819, 1.269683224, 1.327824920], \quad \alpha^{(2)}_{3,\cdot} = (0.216106144, 0.380555907, 0.403337949),$$

$$\underbrace{\boldsymbol{c}^{(2)}_3}_{t=3} = (1.325596326, -0.230946406, 0), \quad \underbrace{\boldsymbol{r}^{(2)}_3}_{t=3} = (2.839284390, -0.323729315, 0).$$

- $t = 4$ **(attend to** $1, 2, 3, 4$**).**

$$s^{(2)}_{4,\cdot} = \frac{[1.228804, 2.243487, 2.347745, 2.397319]}{1.732051} = [0.709450571, 1.295277972, 1.355470941, 1.384092331]$$

$$\begin{aligned}\alpha^{(2)}_{4,\cdot} &= \mathrm{softmax}([0.709450571, 1.295277972, 1.355470941, 1.384092331]) \\ &= (0.149975874, 0.269428174, 0.286143894, 0.294452058),\end{aligned}$$

$$\underbrace{\boldsymbol{c}^{(2)}_4}_{t=4} = \sum_{j=1}^{4} \alpha^{(2)}_{4,j}\, \boldsymbol{r}^{(1)}_j = (1.392536876, -0.181774972, 0),$$

$$\begin{aligned}\underbrace{\boldsymbol{r}^{(2)}_4}_{t=4} &= \boldsymbol{r}^{(1)}_4 + \boldsymbol{c}^{(2)}_4 = (1.546813756, -0.068450308, 0) + (1.392536876, -0.181774972, 0) \\ &= (2.939350632, -0.250225280, 0).\end{aligned}$$

Greedy decode.

$$\langle \boldsymbol{r}^{(2)}_4, \boldsymbol{A} \rangle = 1.206093607, \quad \langle \boldsymbol{r}^{(2)}_4, \boldsymbol{B} \rangle = 2.410267518, \quad \langle \boldsymbol{r}^{(2)}_4, \boldsymbol{D} \rangle = 2.420365007.$$

Hence the next token is D. This means that the next input sequence would be ABBBD and for subsequent iterations D always wins. So the output sequence is ABBBDDD . . . which means that the tipping point still occurs in this two-layer Attention head 'molecule' system. Moreover in this case, the output is exactly the same as for the single Attention head case and it occurs at exactly the same $n^* = 3$ value as for the single Attention head case.

To summarize this particular case with these sets of simplifications, a user who prompts the multilayer Attention head system (i.e. our tiny transparent Transformer) will get exactly the same output response as if they had used the single Attention head system. More generally this will not be the case, but we find this example of some robustness in going from the 'atom' to the 'molecule' to be quite intriguing.