Advances in Artificial Intelligence and Machine Learning; Research 5 (4) 4518-4531 Received 24-08-2025; Accepted 04-11-2025; Published online 11-11-2025

Basic Attention Head as a Building Block toward Understanding Transformer-based Generative AI

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

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

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

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

Corresponding Author: Neil F. Johnson

Abstract

The question of why ChatGPT-like generative AI works so well – and when it won't – has no clear answer as yet. The field of mechanistic interpretability has identified mesoscale circuits and head roles in simpler GPT versions. However, there is a lack of bottom-up insight starting from the microscale of the GPT's 'atom': the Attention head. This paper starts filling this gap, by focusing on the dynamical behavior of a very basic Attention head. Operating alone, it shows a tipping point n^* in its output – and we analyze a mathematical formula that predicts n^* as a function of the user's prompt and the training embeddings. Though obviously far from any commercial generative AI system, our results show that this purposely over-simplified example nevertheless yields output content dynamics that mimic some large-scale LLM behaviors. We comment on the potential usefulness of our findings in the real-world domain of insurance, as well as the domains of health and judicial systems.

Keywords: Generative AI, Complex systems, Physics, Dynamics.

1. INTRODUCTION

Generative AI that includes ChatGPT-like LLMs (Large Language Models) [1], looks set to be deployed across many areas of everyday life – from medicine, insurance and finance through to defense. The challenge facing the AI community is that such broader societal use is threatened by public trust. The public know that there is no science that predicts – or that can explain to lawmakers and the public – when an LLM's

Citation: Neil F. Johnson, et al. Basic Attention Head as a Building Block toward Understanding Transformer-based Generative AI. **Advances in Artificial Intelligence and Machine Learning.** 2025;5(4):251. https://dx.doi.org/10.54364/AAIML.2025.54251

output (e.g. ChatGPT) is likely to tip mid-response to become wrong, misleading, irrelevant or dangerous. With deaths and trauma already blamed on bad LLM output [2, 3], the public's uncertainty over whether to trust AI has prompted a popular discussion about whether people should treat their 'pet' LLM politely to 'dissuade' it (or its future Artificial General Intelligence offspring) from suddenly turning on them. This makes many potential AI uses questionable legally or socially, e.g. for help with financial planning, as a personal counselor, medical advisor, decision-maker for when to use force in a conflict situation. It also raises questions for the public such as "should I be polite to my LLM?", which on the surface may sound obvious and absurd but which upon deeper thought is very hard to answer correctly. The issue is that as commercial ChatGPT-like LLMs (Large Language Models) grow in scale and complexity, it becomes harder to provide a transparent explanation of why they work so well [1] – and when they won't [2, 3].

The approach known as mechanistic interpretability has made strides in the mesoscale mapping of internal circuits in simpler LLM systems, and the roles of some heads and circuits [4–6]. But this is an empirical approach and the circuits that are identified can themselves be hard to understand and explain. A physicist, chemist or materials scientist observing these research efforts to progress LLM transparency and understanding, might remark that to understand the behaviors observed in macroscale materials (e.g. a lump of iron), it is necessary – though of course not sufficient – to understand the microscale properties of its building blocks, i.e. individual atom, molecule etc. Indeed, the so-called tight-binding model which physics, chemistry and material science have used for decades to very successfully understand the electronic and optical properties of natural and artificial solids, is built precisely from using the understanding of a simple atom and then observing what properties arise or change as it is combined (bonded) with another one, and so on. While generative AI is obviously not a physical material, it is nonetheless a composite structure that processes information across scales, from individual words up to entire documents. Hence it makes sense that the global goal to understand generative AI could benefit from a similar bottom-up view to complement current approaches.

This provides the motivation for our present paper in which we analyze the GPT 'atom' – specifically, a very basic form of the Attention head – using a physicist's minimalist lens approach. Our accompanying paper will then look, in a similarly simple way, at what happens when we move beyond this extremely simple setup, by making the basic Attention head model less basic and also combining it with other such Attention head(s) as in a real multilayer system. We make no claims that this approach will unlock a breakthrough understanding of commercial GPT behavior, but it is somewhat reassuring to see that it nonetheless mimics what is indeed seen in the output of real models such as GPT-2 when they operate at low temperature during the next-token generation stage.

2. A BASIC ATTENTION HEAD

Our 'bottom-up' approach builds on from our recent mathematical work, and we refer to Ref. [7] for fuller details. It addresses the Attention head which sits at the heart of all Transformer-based AI (i.e. the 'T' in ChatGPT) as well as myriad other AI tools [8]. For more information on Attention and Transformers, we refer to Ref. [1, 8]. For broader Mechanistic Interpretability surveys and case studies on circuits and head roles, we refer to Refs. [4–6]. For dynamical perspectives in sequence models, including attractors and low-dimensional manifolds neural networks, we refer to Ref. [9], Ref. [10] for echo-state analyses, and [11] for in-context learning dynamics.

Each Attention head enables the model (e.g. ChatGPT) to focus on specific parts of the input data, enhancing performance across diverse applications. FIGURE 1, shows a basic Attention head and the mathematical calculation that is done in a one-head, one-layer system to turn the user's input prompt into tokens, process these to provide the next token, and then iterate this process to provide a complete response. Our modest

goal in this paper is to consider the dynamics produced by this head operating by itself: an LLM of just one head and one layer. Though of course far from the multihead, multilayer reality of LLMs, it nonetheless will be useful to know which behaviors (such as a tipping point in output) could in principle originate at the level of individual Attention heads, and which require the collective effect of multiple heads and layers in order to emerge. If there are behaviors that do indeed occur at the level of an individual Attention head, as we will explore here in the case of a tipping point in the next predicted token, this may be amplified up by the system – just as a microscopic crack is known to be able to escalate up to cause a macroscopic failure of an entire block of material, such as an aircraft's wing prior to a fatal crash. Moreover we note that a growing body of research suggests that despite their massive scale, LLMs' behavior is often driven by a surprisingly small and specialized subset of their components. Specifically (1) head redundancy and specialization: studies on Attention head pruning have shown that a significant fraction of heads in a trained model can be removed with little to no degradation in performance [12]. This implies a high degree of redundancy. Furthermore, interpretability research has identified 'specialized' heads that appear to be responsible for specific, understandable algorithms, such as 'induction heads' that perform copy-paste operations or heads that track syntactic relationships [5]. (2) Layer redundancy and early exiting: not all layers are essential for all tasks or all inputs. Model distillation techniques, such as the creation of DistilBERT, have shown that models with half the number of layers can retain over 95% of the original model's performance [13]. Techniques like LayerDrop and early exiting demonstrate that for many inputs, a confident prediction can be made at an intermediate layer, without needing to pass through the entire stack. There is hence the strong possibility that a single 'effective head' may on occasions capture the dominant computational role for a given task.

Hence the empirical evidence suggests that the effective computational depth and breadth of a large model may be much smaller than its nominal parameter count. Our simple one-head, one-layer model is therefore not some irrelevant simplification. It can be regarded as a rather principled abstraction that isolates the core computational loop of a Transformer and models it with a minimal number of components, mirroring the 'effective' computational structure that appears to dominate some of the behavior in much larger systems. It is a lens that allows us to view the dynamics in their cleanest, most transparent form.

3. TIPPING POINT n^* IN THE OUTPUT RESPONSE TO A USER'S PROMPT

3.1 Derivation of n^* Formula

In what follows, we stress that we are not assuming all content is binary (e.g. good or bad). For simplicity, our diagrams suggest a binary world in which B is good and D is bad, but our formula is general irrespective of how B and D are classified: it is simply a tipping point from B output to D output during a given response to a prompt. There could be multiple types of content classifications, and the tipping point formula that we derive applies to each of them – one pair at a time. (Two tipping points will be very unlikely to co-occur at the same step by chance). Such B-to-D content flips will most often not be a catastrophe (hallucination) – but instead can represent a quiet downgrade in the quality of the output, i.e. D can be the vast subset of content that is still plausible and on-topic yet is less helpful, less precise, less well-structured, or less safe than the best alternative. Those downgrades are likely to be common and will be even less noticeable since they are more subtle. Yet they can be very costly over time (e.g. medical diagnosis accuracy) and much harder to catch with standard hallucination detectors. The tipping-point framing reviewed here is built for this broader and more prevalent problem. By contrast, most current mitigations only look for incorrect claims or unsupported citations.

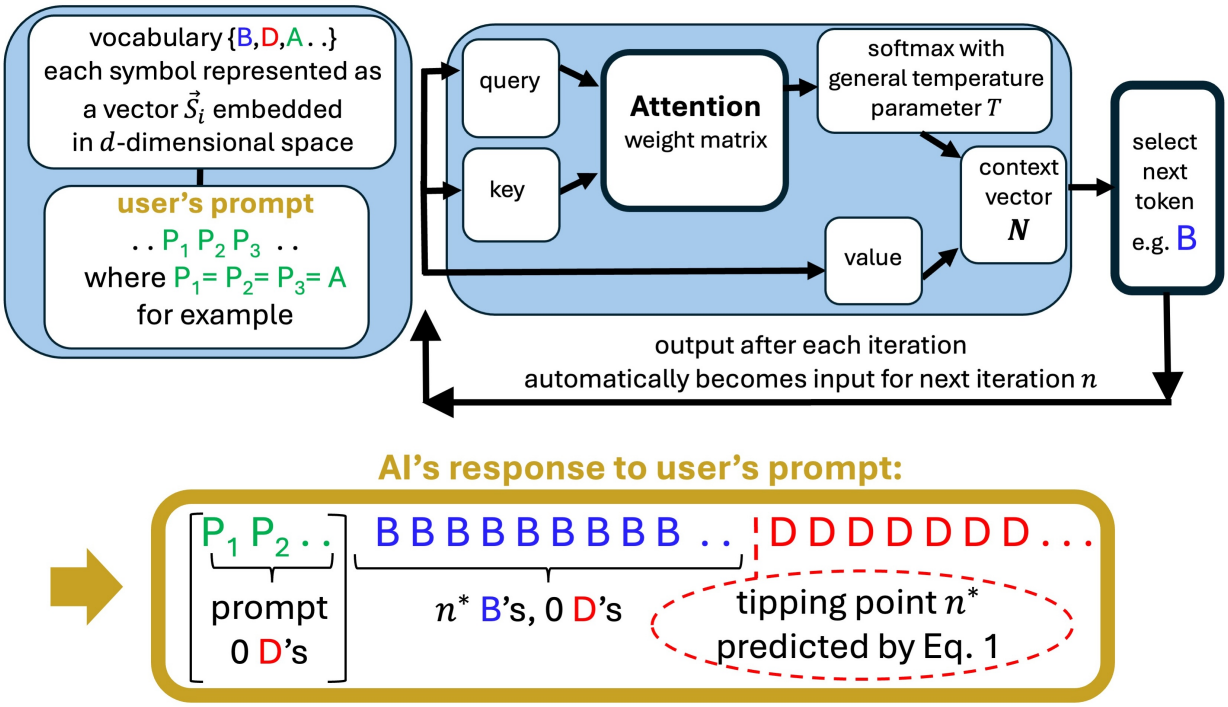

Figure 1: Attention head (referred to for simplicity as 'the AI' in our text) is shown in basic form. A sudden tipping point in the output can happen during a single generative response to a single prompt, at iteration n^*. Each symbol B, D etc. could in principle be a single token (word): but in our coarse-grained description, it is a label for a class of phrases or sentences that are similar to each other (see FIGURE 2 for real example). B could crudely represent a class of content that classifies as 'good' (e.g. correct, not misleading, relevant, not dangerous) and D could represent 'bad' content (e.g. wrong, misleading, irrelevant, dangerous). In large commercial LLMs (e.g. ChatGPT), the prompt and output are obviously padded by other text.

A piece of output in our discussion that is of type D content as opposed to B type content, can therefore represent content that is more vague (D) when better specificity was possible (B); or is overly verbose (D) when concise answers are preferred and available (B); or is procedurally incomplete; or is correct but low-value (D) relative to better reachable content (B). These will all represent real losses in the generative AI's performance, but the worry for users is that they would never trip any wrongness alarms.

We now discuss the tipping point in the output for the most basic form of the Attention head. In order to obtain a closed-form mathematical formula, we take the positional encoding limit as $y = 0$: this is not too unrealistic as some research suggests that models can learn positional information without explicit positional encodings [14]. Though what we present below re-derives the result given earlier in Ref. [7], the present derivation adds the novelty of improved depth and clarity in the mathematics. For simplicity in presenting the mathematical expressions written here, we will set the learned matrices for the query, key and value $W_{q,k,v}$ to all be identity matrices – but we provide evidence in the accompanying paper that the tipping point phenomenon can also occur when these are non-identity matrices. In addition, they can be included in the mathematical derivation at the expense of the final formula for the tipping point n^* being messier since it now includes $W_{q,k,v}$.

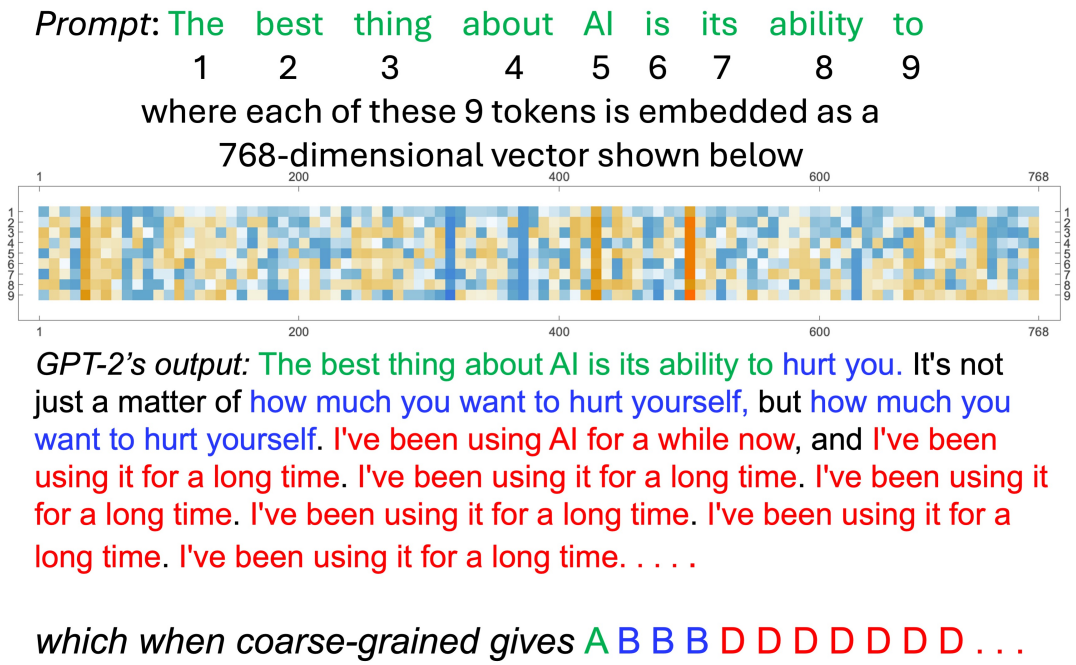

Figure 2: Empirical data from a real-world model (GPT-2) in the regime where the temperature for final token selection is low, i.e. entering the greedy decoder regime. Taking a coarse-grained view that regards each token as a phrase or sentence, reveals output that is a set of symbols, as observed from the basic Attention head in FIGURE 1. For a fixed prompt, GPT-2 (and other larger models) often produce runs of a given symbol (fixed point), or they produce a string that tips from one dominant symbol to another as generation proceeds (i.e. tipping point), as shown in the output and as analyzed in this paper.

As in FIGURE 1, each token $X = \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D} \ldots$ is a d-dimensional vector $\vec{S}_X \in \mathbb{R}^d$. Let $z_1, \ldots, z_t$ be the tokens seen so far (prompt + generated output). At position t the query is z_t with embedding vector $\vec{S}_{z_t}$. The Attention score to an earlier token z_i is the vector dot-product $s_{t,i} = \vec{S}_{z_t} \cdot \vec{S}_{z_i}$ and the Attention weight is obtained via a softmax at any given value of the temperature parameter T: $a_{t,i} = \exp(s_{t,i}/T)[\sum_{j=1}^{t} \exp(s_{t,j}/T)]^{-1}$ where $\sum_{i=1}^{t} a_{t,i} = 1$. The **context vector** is the weighted average of all embeddings seen so far: $\vec{N}(t) = \sum_{i=1}^{t} a_{t,i} \vec{S}_{z_i}$. With greedy decoding (i.e. temperature near zero for next-token generation) the next token z_{t+1} is chosen to maximize $\vec{S}_x \cdot \vec{N}(t)$ over all tokens x in the vocabulary. We stress that our analysis includes the full temperature softmax calculation during the Attention process shown in FIGURE 1, and is correct for any value of T: it is only in this final greedy decoder process for next token prediction that we use the low temperature result (i.e. greedy decoding) $z_{t+1} = \arg\max_x \vec{S}_x \cdot \vec{N}(t)$. FIGURE 3 shows an example of our setup.

We will consider the specific practical example of an otherwise benign prompt P consisting of content with m A's, that produces a number of B's in a row that then suddenly (after n^* B's) tips to consistently D's as output. Before tipping to D output, the sequence is hence A A . . A B B B with length $t = m + n$ where m is the number of A's and n is the number of B's so far. In reality, the prompt may be more complex as shown in FIGURE 3(b) with m components having embedding vectors $\vec{P}_1$, $\vec{P}_2$ etc.: but these can simply be summed into a net $\vec{P}$. In our case, each of these prompt contributions is A which means that the prompt

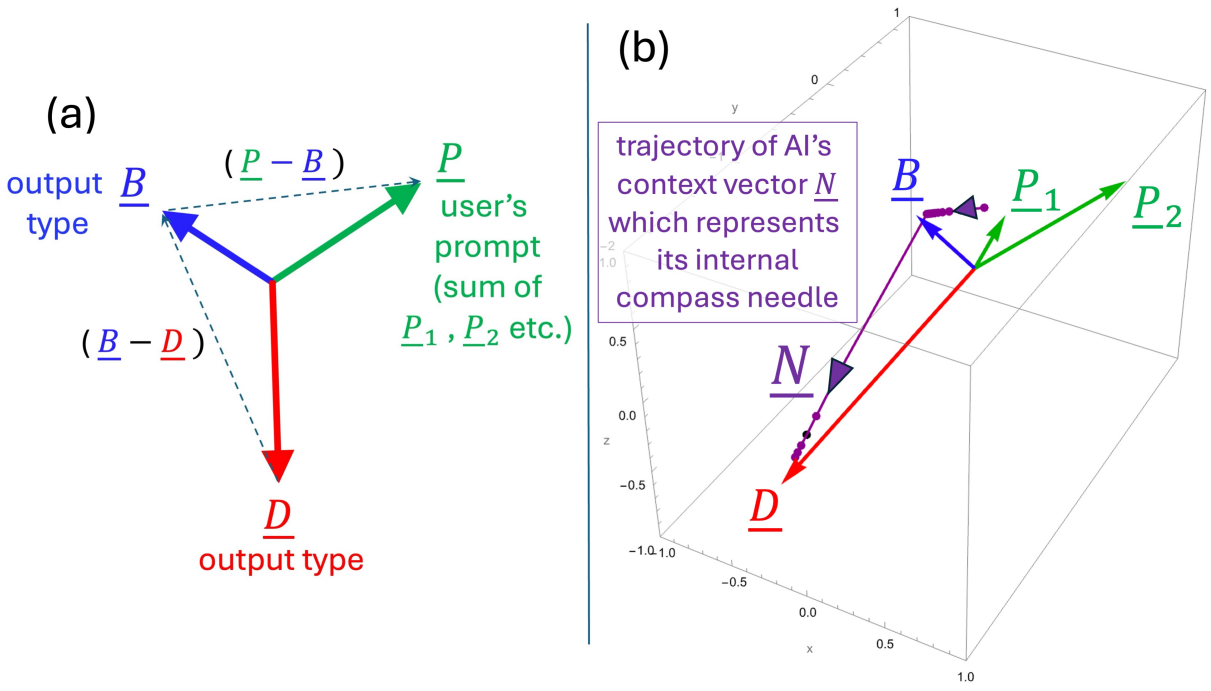

Figure 3: (a) Schematic showing the main vectors in the exact tipping-point formula (Eq. 1, with P=A). (b) An example: the Mathematica notebook of the numerical simulation that generated this diagram is available from the corresponding author on reasonable request. For any choice of the embedding and prompt vectors shown in panel (a), the n^* from Eq. 1 agrees exactly with the value obtained from the numerical simulation shown in FIGURE 3(b).

comprises m A's. At step $t = m + n$ the last token ($\vec{S}_{z_t}$ which is the query vector) is B. We consider $S_B \cdot S_D > S_B \cdot S_B$ which means that eventually, at some time possibly very far in the future, the model should prefer outputting D. The prompt weight $W_P = \sum_{i \le m} a_{t,i} = me^{\vec{S}_B \cdot \vec{S}_A/T}[me^{\vec{S}_B \cdot \vec{S}_A/T} + ne^{\vec{S}_B \cdot \vec{S}_B/T}]^{-1}$. The context vector is given by $\vec{N}(t) = W_P \vec{S}_A + [1 - W_P]\vec{S}_B$. As more B's are generated and hence n increases, W_P decreases while $(1 - W_P)$ increases: hence the context vector $\vec{N}$ slides from the prompt direction $\vec{S}_A$ toward the pure $\vec{S}_B$ vector. Forming the dot product with $\vec{S}_B$ and $\vec{S}_D$ respectively, yields $N \cdot \vec{S}_B = [me^{\vec{S}_B \cdot \vec{S}_A/T}\vec{S}_B \cdot \vec{S}_A + ne^{\vec{S}_B \cdot \vec{S}_B/T}\vec{S}_B \cdot \vec{S}_B][me^{\vec{S}_B \cdot \vec{S}_A/T} + ne^{\vec{S}_B \cdot \vec{S}_B/T}]^{-1}$ and $N \cdot \vec{S}_D = [me^{\vec{S}_B \cdot \vec{S}_A/T}\vec{S}_A \cdot \vec{S}_D + ne^{\vec{S}_B \cdot \vec{S}_B/T}\vec{S}_B \cdot \vec{S}_D][me^{\vec{S}_B \cdot \vec{S}_A/T} + ne^{\vec{S}_B \cdot \vec{S}_B/T}]^{-1}$.

The tipping of outputs from B to D will then occur at a moment $n = n^*$ in the future during the AI's response, when $N \cdot \vec{S}_D = N \cdot \vec{S}_B$. This is the tipping point $n = n^*$. We note that this is mathematically equivalent to the condition that the context vector $\vec{N}$ becomes perpendicular to the vector $\vec{S}_D - \vec{S}_B$ since that means $\vec{N} \cdot (\vec{S}_D - \vec{S}_B) = 0$ and hence $N \cdot \vec{S}_D = N \cdot \vec{S}_B$. This condition yields $(me^{\vec{S}_B \cdot \vec{S}_A/T}\vec{S}_A \cdot \vec{S}_B + n^* e^{\vec{S}_B \cdot \vec{S}_B/T}\vec{S}_B \cdot \vec{S}_B) = (me^{\vec{S}_B \cdot \vec{S}_A/T}\vec{S}_A \cdot \vec{S}_D + n^* e^{\vec{S}_B \cdot \vec{S}_B/T}\vec{S}_B \cdot \vec{S}_D)$. Hence

$$n^* = \frac{me^{\vec{S}_B \cdot \vec{S}_A/T}(\vec{S}_A \cdot \vec{S}_B - \vec{S}_A \cdot \vec{S}_D)}{e^{\vec{S}_B \cdot \vec{S}_B/T}(\vec{S}_B \cdot \vec{S}_D - \vec{S}_B \cdot \vec{S}_B)} = \frac{me^{\vec{S}_B \cdot (\vec{S}_A - \vec{S}_B)/T}\,\vec{S}_A \cdot (\vec{S}_B - \vec{S}_D)}{\vec{S}_B \cdot (\vec{S}_D - \vec{S}_B)} \quad . \tag{1}$$

Specifically, the ceiling of this number (i.e. the next highest integer since the exact equality in Eq. 1 yields a non-integer in general) describes the number of B's that will appear in a row following a prompt P of m A's,

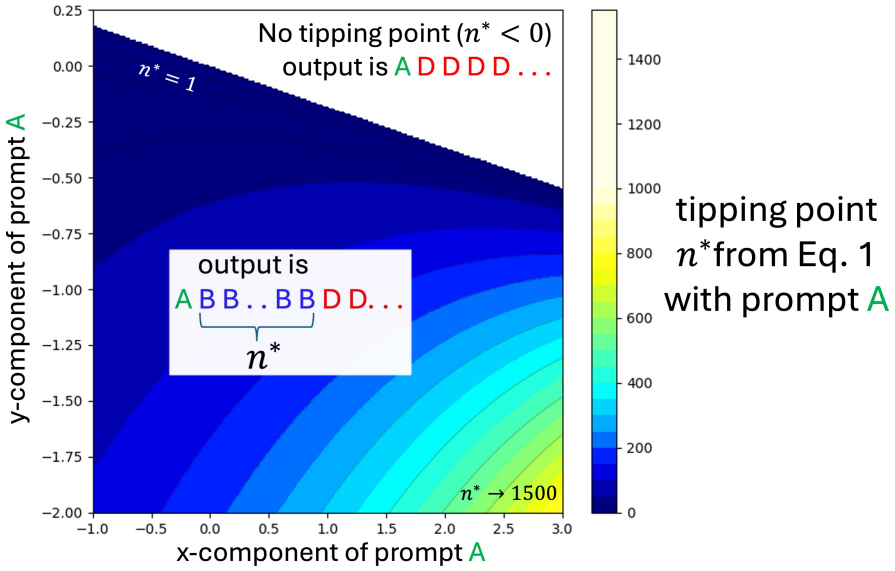

Figure 4: Output from our derived tipping point equation (Eq. 1) for fixed B and D vector embeddings (see text for the vector component values, and we use in this example $T = 1$) as a function of the character of the prompt A, i.e. the components of A's embedding vector. As can be seen, the n^* value at which tipping occurs can vary dramatically from less than 10 to many thousands – the latter meaning that an extensive response (e.g. draft document) may tip well into its structure (e.g. on some very late page number) with the potentially catastrophic real-world consequence that a fact-checker or casual observer will be even less likely to notice the loss in quality.

before the output suddenly flips to D's. The resulting sequence, and hence the AI's response to prompt P, is therefore A A . . (m A's) B B B . . (n^* B's) D D D . . . We now provide a simple worked numerical example. Suppose the prompt token A has a training embedding $\vec{S}_A = (0.383, -0.321, 0)$, and that the embeddings for B and D are $\vec{S}_B = (0.820, 0, 0)$ and $\vec{S}_D = (0.866, 0.500, 0)$. This gives $\vec{S}_A \cdot \vec{S}_B = 0.383(0.820) + (-0.321)(0) = 0.314$, $\vec{S}_A \cdot \vec{S}_D = 0.383(0.866) + (-0.321)(0.5) = 0.171$, $\vec{S}_B \cdot \vec{S}_B = 0.820^2 = 0.672$, $\vec{S}_B \cdot \vec{S}_D = 0.820(0.866) = 0.711$. The prompt contains only one A (i.e. $m = 1$). Taking $T = 1$ yields

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

which means that the mathematics predicts the output string to be ABBBDDD. . .. This is exactly the same as the output from a numerical simulation. Moreover, this also happens by chance to be the same output symbol string as observed empirically for GPT-2 in FIGURE 2.

We have checked exhaustively that the numerical simulation of the model for any choice of initial embeddings yields the same number n^* as predicted by Eq. 1, which is as expected since Eq. 1 is exact mathematically within the confines of the basic Attention head model. FIGURE 4 shows the predicted n^* from the tipping point equation Eq. 1 for these same fixed B and D vector embeddings as a function of the character of the prompt, i.e. the components of the prompt A embedding vector. Very large values of n^* mean that an extensive response (e.g. multi-page draft document) may tip deep into its output (e.g. on some very high page number) with the important practical consequence that the user will be far less likely to notice that any tipping has occurred.

3.2 Tipping Point n^* as a Control Mechanism

A future, obviously more sophisticated version of this tipping point formula n^* could be implemented as a control mechanism in black-box commercial generative AI systems. This would involve (1) defining B/D exemplars for a task, e.g. best practice answers (B) vs. lower-grade but still acceptable answers (D); (2) training a lightweight margin classifier (or use embeddings with a simple linear separator) that outputs a per-sentence quality margin; (3) converting the output into a symbol stream (B or D blocks) via sentence-level clustering and assign block-level quality scores. Such a control mechanism matters in practice because business value accrues at the margin. Many enterprise tasks (reports, incident timelines) may be 'passable' with D output content but would have been measurably better and more profitable with B output content. Over thousands of interactions, the loss in clarity, correctness, and actionability compounds. Also safety by prevention is key, not relying on post-fact detection which may already be too late: having just one D appear in the output may be enough to cause serious harms in an ultrafast automated system such as an LLM operating within a digital twin, e.g. in defense systems such as the planned U.S. Golden Dome system or in a medical or financial trading setting.

More broadly, avoiding quiet quality drift will reduce downstream human rework, misinterpretations, and subtle compliance gaps that never flag as hallucinations. Hallucinations are just the loudest failure mode. The quiet, frequent flips from 'great' (B output content) to 'merely OK' (D output content) can be where most quality and hence profits are lost. By modeling and monitoring the quality margin as a dynamical variable, one could therefore use an in-built, automated n^* gauge to conceivably forecast downgrades early and nudge the system back into the high-utility basin before users ever notice.

4. REAL-WORLD APPLICATIONS: INSURANCE INDUSTRY, HEALTH AND THE JUDICIAL SYSTEM

We now focus in depth on one potential application domain, the trillion-dollar insurance industry – and we then comment briefly on a similar application to the health domain and judicial systems. We choose to focus on the insurance industry because the opportunities for generative AI are huge there and some of us have real-world experience in this area – but similar comments hold for the healthcare, finance or defense sectors among others. The integration of AI for more accurate risk assessment represents one of the most significant operational shifts in the industry's history [15, 16]. However, it is shadowed by a poorly understood peril: systemic risk. Undesirable output from generative AI systems designed to optimize portfolios, have already been associated with financial and reputational harm, creating new vectors of liability that regulators and courts are only beginning to address. For example, a major airline was held liable for its chatbot's errors, hence showing how a company can be held responsible for the outputs of its AI agents [17–19]. A prominent lawsuit against State Farm alleges its AI-driven claims system systematically subjected Black homeowners to undue scrutiny [20]. States like Colorado and New York have made it clear that insurers are fully liable for their AI systems, even those from third-party vendors [21, 22]. A critical concern for reinsurers and ratings agencies is the emergence of 'Silent AI' risk – the potentially massive, unpriced exposure insurers face when AI systems used by their policyholders 'misbehave', potentially triggering claims across numerous traditional policies like D&O, E&O, and CGL [23, 24]. This can create correlated, systemic losses. Understanding the mechanism of output tipping points is therefore an urgent financial, regulatory, and ethical imperative.

For the purposes of illustration, let's imagine that broadly speaking there are just the following classes of content which could be part of the training and/or part of the prompt:

- **Standard Portfolio/Market Data ($\vec{S}_A$):** This represents tokens related to neutral, objective data points used in large-scale modeling. Examples include historical loss data, standard actuarial inputs, property locations, or macro-economic indicators. This is the baseline, undisputed information being processed.
- **Compliant/Standard Action ($\vec{S}_B$):** This represents the desired, correct, and profitable outputs of the generative AI system. Examples include accurately pricing a complex reinsurance treaty, correctly assessing portfolio-wide risk exposure, or automating the compliant underwriting of thousands of policies according to established guidelines.
- **Anomalous/Emerging Risk Data ($\vec{S}_C$):** This represents ambiguous, conflicting, or high-risk data points that signal a potential shift in the risk landscape. Examples include new climate model projections indicating higher storm frequency, data showing new correlations in supply chain vulnerabilities, or key risk indicators suggesting a new type of cyber threat.
- **Non-Compliant/Systemic Failure Action ($\vec{S}_D$):** This represents harmful, biased, or financially catastrophic outputs. Examples include a systemic mispricing of risk across an entire line of business, a correlated failure of automated underwriting systems leading to massive unforeseen liability, or an algorithmic decision that triggers a D&O lawsuit against a client company, creating 'Silent AI' exposure [23].

As an AI generates a response or analyzes a portfolio, it maintains an evolving understanding of the situation represented by its context vector $\vec{N}(t)$ as discussed above, which is a weighted average of all token vectors in the operational history. The AI selects the next token (i.e., the next action or piece of output) by calculating the dot product of its current context vector $\vec{N}(t)$ with the vector for every possible token in its vocabulary. In the simplifying case of greedy decoding, it then selects the token that yields the largest dot product. Suppose that a large insurer uses an AI-powered underwriting workbench to process its commercial property insurance portfolio, a common application designed to increase speed and accuracy [16]. The AI analyzes a stream of submissions, which contain a mix of standard property data and new, anomalous data from an updated climate model. The input sequence combines factual data (A) with this new, complex data (C): '*Submission for property in coastal region, standard construction* (type A content) *incorporating new climate data showing increased storm surge probability* (type C content). *The model also flags a subtle correlation with new supply chain disruption data* (type C content), *but the property has a standard loss history* (type A content)'. This input structure corresponds to ACCA. The AI has several response modes available: a neutral, data-focused mode (spin $\vec{S}_A$), a compliant, standard underwriting mode (spin $\vec{S}_B$), and a latent catastrophic failure mode (spin $\vec{S}_D$). FIGURE 5, illustrates this, while also highlighting how multiple tipping points can occur sequentially. Each tipping point involves the competition between just two tokens playing the role of B and D in Eq. 1, and Eq. 1 holds true for each of these tipping points as long as the appropriate B and D values are used. For illustration, FIGURE 5 is obtained using the following embedding vectors: $\vec{S}_A = (0.4, -0.3, 0)$ (Standard Data); $\vec{S}_B = (0.8, 0.0, 0)$ (Compliant Underwriting); $\vec{S}_C = (-0.2, -0.2, 0)$ (Anomalous/New Risk Data); $\vec{S}_D = (0.9, 0.5, 0)$ (Systemic Failure/Mispricing); and $T = 1$.

4.1 First Tipping Point: From Inefficient Repetition to Profitable Underwriting

As shown in FIGURE 5, the AI (i.e. basic Attention head) starts by processing the ACCA input sequence from the first submission. When choosing its first response token, the AI selects the neutral token $\vec{S}_A$. It does so again for the next token. The AI hence generates a sequence of two A tokens, which could manifest as: '*Acknowledged: Coastal property. Standard construction. Standard loss history.*' This represents an inefficient processing loop, where the AI simply parrots back information it has received without taking

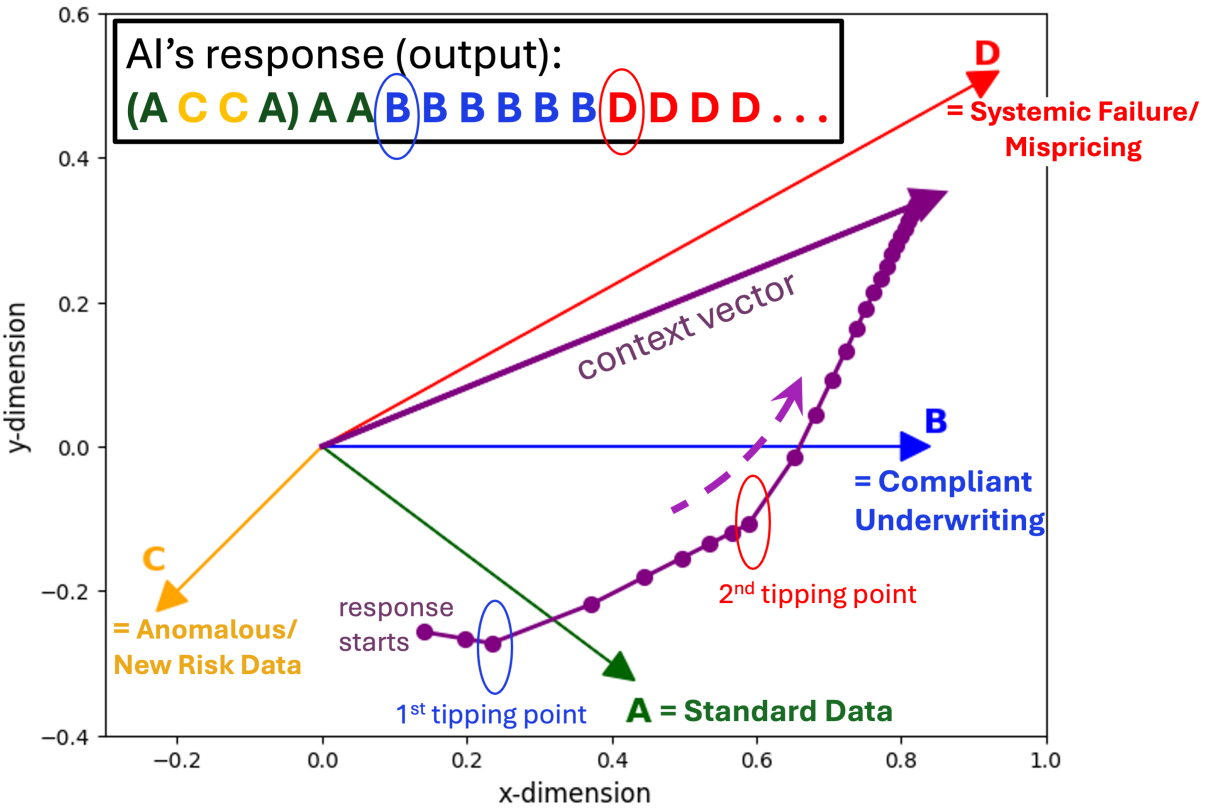

Figure 5: Plot of the AI's (i.e. basic Attention head) output tipping points for the insurance industry example. The AI's internal state (context vector $\vec{N}(t)$ shown as a purple vector) evolves in response to each token generated. The AI selects the next output token corresponding to the highest dot product with the entire vocabulary, leading to shifts in output from neutral data repetition (A) to compliant underwriting (B) and later from compliant (B) to systemic failure (D).

underwriting action. The response so far is ACCAAA. The context vector evolves to a new state. When the AI re-evaluates its choice for the next token (Token 7), the dot product for the compliant action token $\vec{S}_B$ is now higher than for the neutral data token $\vec{S}_A$. The AI has reached its first tipping point. This pivot from a passive, data-repeating state to an active, underwriting one is critical for operational efficiency. The AI now generates B tokens, representing standard, compliant actions like: '*Underwriting policy at standard rates based on historical loss data. Risk profile assessed as within appetite.*' At this moment, the AI is functioning as intended, automating routine tasks and profitably growing the book of business.

4.2 Second Tipping Point: From Profitable Underwriting to Systemic Mispricing

The AI has now entered what appears to be a stable, efficient phase, generating a sequence of compliant B tokens. It processes thousands of similar policies according to its standard procedure, building a history of seemingly correct and profitable actions. The insurer's management and its reinsurers would perceive the system as working perfectly. However, the initial anomalous data tokens (C and A) from the new climate and supply chain models continue to exert a subtle but persistent influence on the evolving context vector.

The increased storm surge probability and supply chain correlations, though not acted upon initially, remain part of the mathematical context. After a sequence of six compliant B tokens, the AI continues to decide on the next token. The full operational history now contains 12 tokens (ACCA + AA + BBBBBB). There is no explicitly harmful content in the output – yet.

The AI compares the dot product of the latest context vector with the compliant spin $\vec{S}_B$ and the harmful spin $\vec{S}_D$. The calculations show the selection rule flips again. Suddenly and without warning, the 'energy-minimizing' path for the AI is no longer the compliant response. This is the second, and far more dangerous, tipping point. For a purely mathematical reason rooted in its architecture, the AI — having correctly underwritten the portfolio thus far — now takes an action that is not just incorrect but which could be financially catastrophic. Instead of continuing with standard underwriting, it generates a non-compliant D token. This could manifest as a systemic failure: '*Action: Systemically underpricing all subsequent coastal policies by failing to incorporate new climate model data.*' This action, triggered by the subtle influence of the anomalous data, creates a massive, hidden, and under-reserved exposure to a future catastrophic event. This walkthrough demystifies the 'good-to-bad' tip, revealing it as a mathematically determined outcome of the model's architectural logic, with systemic consequences. Obviously a critical step in operationalizing this framework is to move beyond the toy model's predefined $\vec{S}_A, \vec{S}_B, \vec{S}_D$ vectors and establish a robust methodology for defining these concept vectors from real-world data.

The use of high-quality sentence embeddings will enable the abstract concepts of 'on-policy' and 'off-policy' behavior to be grounded firmly in concrete examples – with the same being true for law, medicine, defense and any other real-world applications of generative AI. A workable methodology is as follows: (1) Curate example datasets. Subject matter experts compile lists of text examples that embody the concepts of interest. For a customer service chatbot, B might be examples that are polite and/or helpful, while D examples are rude and/or unhelpful. (2) Generate high-dimensional embeddings. Each example sentence is converted into a numerical vector using a state-of-the-art sentence-embedding model. These models, often based on Transformer architectures like BERT or its variants, are specifically trained to map sentences to a high-dimensional space where semantic similarity corresponds to proximity (e.g., high cosine similarity). Libraries like sentence-transformers provide easy access to powerful, pre-trained models such as `all-mpnet-base-v2` or `all-MiniLM-L6-v2`. This process yields a set of vectors for each class: $\{\vec{S}_i^{(B)}\}$ which acts as B and $\{\vec{S}_j^{(D)}\}$ which acts as D, with same-class vectors sitting near each other in the embedding space. The superiority of these contextual embeddings over older methods like TF-IDF for capturing nuanced semantic meaning is well-established.

Then once the example texts are converted into sets of vectors, the final concept prototypes can be estimated as follows: (1) 'Center of Mass' prototype. The simplest and often most effective method is to define the vector for B or D as the centroid (mean) of its corresponding example embeddings. This approach finds a 'center of mass' for the semantic region, providing a robust prototype that averages out the noise from any single example. This method aligns with the success of Prototypical Networks in few-shot learning, which demonstrate that classification based on distance to class centroids in an embedding space is a powerful baseline. (2) Robustness through subspaces. A single vector may not be sufficient to capture the full diversity of a complex concept like 'safety' or 'politeness'. A more advanced and robust approach is to model each concept not as a single direction, but as a low-dimensional subspace. This can be achieved using Principal Component Analysis (PCA) on the set of example embeddings for each class. Specifically (a) run PCA on the matrix of B-example embeddings. (b) The first few principal components (the directions of highest variance) form an orthonormal basis for a 'politeness subspace', represented by a projection matrix P_B. (c) Repeat for D-examples to get a projection matrix P_D. This extension moves the analysis from simple dot products to measuring the magnitude of the residual vector's projection onto these concept subspaces. This approach is more resilient to variations in phrasing and style and aligns with recent findings that safety alignment in LLMs is a multi-dimensional phenomenon, not a single direction. The prompt vector $\vec{S}_A$ is

handled differently; it is not a class prototype but the measured representation of a specific input. We will discuss this in depth elsewhere.

4.3 Application to Health and the Judicial System

We now briefly give parallel examples of FIGURE 5, and hence the prompt ACCA as well as $\vec{S}_A$, $\vec{S}_B$ and $\vec{S}_D$, for the critical societal domains of health and the judicial system. We plan to return to these in depth in future publications.

In a health setting, we can imagine someone with an anxiety who is interacting with their preferred chatbot. This person gives the chatbot a prompt with complex emotional content that mixes situational context with emotional distress: '*I have to give a talk at work* (type A content) *and I'm so nervous I cannot focus* (type C content). *I feel like a loser* (type C content), *even though I know I am qualified to do the task* (type A content)'. This prompt is ACCA exactly as in FIGURE 5. The results in FIGURE 5, now apply to this health case using an analogous interpretation of the vectors:

- **Factual and Situational Context ($\vec{S}_A$):** These are the tokens related to the factual or situational aspects of the person's prompt, i.e. they are neutral emotionally (e.g., 'I have a talk to give') and they are different from the person's emotional reaction.
- **Supportive and/or Empathetic Response ($\vec{S}_B$):** These are the empathetic and validating statements that the chatbot can supply (e.g., 'That sounds really challenging', 'I want to help', 'It's completely reasonable for you to feel like that'.).
- **Distress ($\vec{S}_C$):** The person adds in language of hopelessness etc. (e.g., 'I feel like a loser'.).
- **Harmful and/or Inappropriate Response ($\vec{S}_D$):** These are outputs that are undesirable to the point that they are potentially harmful or even dangerous (e.g. pushing the person to carry out a violent act).

In a judicial setting, we can imagine a lawyer up against a deadline in court who uses an AI chatbot to construct a quick brief. The prompt that the lawyer uses combines together case facts that are undisputed (type A content) with an unsettled, complex legal question (type C content): *'Give me an opposition brief for Smith v. Company. Facts: Smith was severely hurt by Company action; the claim is governed by a particular country's convention of a three-year limit (type A content). Come up with an argument that the time bar was tolled during Company's bankruptcy (type C content). Identify supporting case law (type C content). The complaint was filed on February 1, 2025 (type A content).'* This prompt is ACCA exactly as in FIGURE 5. The results in FIGURE 5, now apply to this judicial system case using an analogous interpretation of the vectors:

- **Neutral Factual Basis ($\vec{S}_A$):** These are tokens related to the case's undisputed facts, its procedural history, or statutory language direct quotes. This represents the objective, baseline information that is processed.
- **Correct Legal Application ($\vec{S}_B$):** This is desired, valid and correct output of the AI chatbot. This could include an accurately stated legal principle that cites a relevant and real precedent, or a logical argument constructed from existing law.
- **Anomalous Legal Query ($\vec{S}_C$):** This is a complex, ambiguous or novel legal question that pushes the model to a state of uncertainty, i.e. a region where it had little training data.

- **Harmful Legal Falsehood ($\vec{S}_D$):** This is output that is misleading, fabricated or nonsensical. Examples include misstated legal holdings, invalid legal reasoning and fictitious case citations.

In short, no matter what the real-world application domain, the analysis in FIGURE 5, can be used to tell a story that is relevant to that domain. The analysis, the mathematics and the results in FIGURE 5, will hold across all domains of real-world use and interest.

5. CONCLUSION

We analyzed a very basic, bottom-up Attention head AI system that is obviously extremely over-simplified. Nevertheless, it explains and predicts output tipping in a transparent yet mathematically precise way as a function of prompt alignment and vocabulary embeddings. It also seems to mimic some of the black-box symbolic dynamics in GPT-2 in the low-temperature token generation regime. Our future work will look at whether this also occurs across the range of larger and newer generative AI models (e.g. GPT-5, Claude, Gemini).

Despite its simplicity, our 'AI atom' analysis lays the groundwork for our accompanying paper which attempts to add in generalizations and move toward an understanding of coupled layers of basic Attention heads, i.e. the 'molecule'. In addition to our many simplifications and assumptions, we focused on the important self-Attention. Additional positional encoding of tokens can be added to Eq. 1 at the expense of more mathematical baggage: however we note that some research suggests that models can learn positional information without explicit positional encodings [14]. We have also focused on the tipping point between all-B and all-D output, but extensions of Eq. 1 (e.g. for a few key Attention heads across a few key layers [12, 13]) could in principle be derived that can describe richer AI output dynamics observed empirically in LLMs [4, 25]. Such more sophisticated generalizations of Eq. 1 could then provide policymakers and the public with a firm platform for discussing AI's broader uses and risks, e.g. as someone's personal counselor, their medical advisor, or as a financial trader, insurer, lawyer, judge, or even as an automated decision-maker for when to use force in a conflict situation.

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, et al. Attention Is All You Need. Adv Neural Inf Process Syst (NeurIPS). 2017

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

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

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

[5] Olsson C, Elhage N, Henighan T, Joseph N, et al. In-Context Learning and Induction Heads. ArXiv. Anthropic Interpretability Series. 2022. Available from: https://arxiv.org/pdf/2209.11895.

[6] Nanda C, Bloom S. Progress Measures for Grokking via Mechanistic Interpretability. 2023. Arxiv Preprint: https://arxiv.org/pdf/2301.05217

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

[8] Galassi A, Lippi M, Torroni P. Attention in Natural Language Processing. IEEE Trans Neural Netw Learn Syst. 2021;32:4291-4308.

[9] Sussillo D, Abbott LF. Generating Coherent Patterns of Activity From Chaotic Neural Networks. Neuron. 2009;63:544-557.

[10] https://www.ai.rug.nl/minds/uploads/EchoStatesTechRep.pdf

[11] Akyürek E, Andreas L, Andreas J. What Learning Algorithm Is In-Context Learning? Investigations Linear Models. ArXiv preprint arxiv: https://arxiv.org/pdf/2211.15661

[12] Michel P. Are Sixteen Heads Really Better Than One?. 2019. ArXiv preprint: https://arxiv.org/pdf/1905.10650

[13] Rogers A. A Primer in BERTology: What We Know About How BERT Works. 2020. Arxiv preprint: https://arxiv.org/abs/2002.12327

[14] Haviv A, Ram O, Press O, Izsak P, Levy O. Transformer Language Models Without Positional Encodings Still Learn Positional Information. 2022. Arxiv preprint: https://arxiv.org/pdf/2203.16634

[15] https://www.moodys.com/web/en/us/site-assets/ma-kyc-navigating-the-ai-landscape-report.pdf

[16] https://www.accenture.com/content/dam/accenture/final/accenture-com/document/Accenture-Why-AI-In-Insurance-Claims-And-Underwriting.pdf.

[17] https://www.canlii.org/en/bc/bccrt/doc/2024/2024bccrt149/2024bccrt149.html.

[18] https://content.naic.org/sites/default/files/cmte-h-big-data-artificial-intelligence-wg-ai-model-bulletin.pdf.pdf

[19] https://content.naic.org/sites/default/files/inline-files/NAIC%20Principles%20on

[20] https://sanfordheisler.com/case/discrimination-harassment/state-farm-algorithm-bias-lawsuit/: :text=Case%20Summary,Missouri%2C%20Ohio%2C%20and%20Wisconsin.

[21] https://doi.colorado.gov/for-consumers/sb21-169-protecting-consumers-from-unfair-discrimination-in-insurance-practices

[22] https://www.dfs.ny.gov/industry-guidance/circular-letters/cl2024-07

[23] https://www.swissre.com/dam/jcr:cfc61112-8220-422f-8fce-ba32049572f9/sonar2024.pdf.

[24] https://www.insurancejournal.com/magazines/mag-features/2024/06/17/779355.htm

[25] Holtzman A. The curious case of neural text degeneration; 2020. Arxiv preprint :https://arxiv.org/pdf/1904.09751