www.rungta.ac.in

RUNGTA
International Journal of
Computer Science and
Information Technology

RESEARCH ARTICLE

Scaling Reasoning in AI: Challenges of Long-Context Understanding in Emerging Models

Nagajayant Nagamani 1,2
[1] *SRM University, Chennai, Tamil Nadu*
[2] *Rungta College of Engineering and Technology, Bhilai CG 490024*
*Corresponding Author: naga078@gmail.com

ABSTRACT:
As artificial intelligence systems handle increasingly complex reasoning tasks, their ability to process extended input sequences remains a critical challenge. This study evaluates the reasoning capabilities of various open-source large language models, including state-space architectures and recurrent frameworks, under varying input lengths. Using a controlled dataset, we investigate performance degradation, reasoning failures, and the impact of structured prompting. Our results reveal key limitations in long-context comprehension, highlighting gaps in information retention and instruction adherence. These findings underscore the need for improved architectures and adaptive learning strategies to enhance multi-hop reasoning in large-scale AI systems.

KEYWORDS: Large Language Models, Long-context Reasoning, State Space Models, RWKV, Mamba, FLenQA, Multi-hop Inference, Attention Mechanisms

1. Introduction

The advent of Large Language Models (LLMs) has fundamentally transformed natural language processing. However, as the complexity of reasoning tasks increases, the ability of these models to process and comprehend extended input sequences—often referred to as 'long-context understanding'—remains a critical bottleneck. State Space Models (SSMs) like Mamba and RWKV have emerged as potential alternatives to the standard Transformer architecture to mitigate the quadratic scaling of attention mechanisms.

A large language model (LLM) is a neural network trained on a vast amount of text for natural language processing tasks, especially language generation. LLMs can typically generate, summarize, translate, and analyze text in many contexts, and are a foundational technology behind modern chatbots. Biased or inaccurate training data can make an LLM's output less reliable [1].

LLMs are typically based on transformer architecture. Generative pre-trained transformers (GPTs) are a type of LLM that is pre-trained to predict the next word. GPTs are then often fine-tuned to follow instructions and to behave as assistants [2].

Before the emergence of transformer-based models in 2017, some language models were considered large relative to the computational and data constraints of their time. In the early 1990s, IBM's statistical models pioneered word alignment techniques for machine translation, laying the groundwork for corpus-based language modeling. In 2001, a smoothed n-gram model, such as those employing Kneser-Ney smoothing, trained on 300 million words, achieved state-of-the-art perplexity on benchmark tests. During the 2000s, with the rise of widespread internet access, researchers began compiling massive text datasets from the web ("web as corpus") to train statistical language models [3].

Moving beyond n-gram models, researchers started in 2000 to use neural networks as language models. Following the breakthrough of deep neural networks in image classification around 2012, similar architectures were adapted for language tasks. This shift was marked by the development of word embeddings (e.g., Word2Vec by Mikolov in 2013)

Received on 28 December 2025 Revised on 12 February 2026
Accepted on 21 February 2026 Published on 28 February 2026
Rungta International Journal of Computer Science and Information Technology. 2026; 3(1): 12.

and sequence-to-sequence (seq2seq) models using LSTM. In 2016, Google transitioned its translation service to neural machine translation (NMT), replacing statistical phrase-based models with deep recurrent neural networks. These early NMT systems used LSTM-based encoder-decoder architectures, as they preceded the invention of transformers [4].

At the 2017 NeurIPS conference, Google researchers introduced the transformer architecture in their landmark paper "Attention Is All You Need". This paper's goal was to improve upon 2014 seq2seq technology, and was based mainly on the attention mechanism developed by Bahdanau et al. in 2014. The following year in 2018, BERT was introduced and quickly became "ubiquitous". Though the original transformer has both encoder and decoder blocks, BERT is an encoder-only model. Academic and research usage of BERT began to decline in 2023, following rapid improvements in the abilities of decoder-only models (such as GPT) to solve tasks via prompting [5].

Although decoder-only GPT-1 was introduced in 2018, it was GPT-2 in 2019 that caught widespread attention because OpenAI claimed to have initially deemed it too powerful to release publicly, out of fear of malicious use. GPT-3 in 2020 went a step further and as of 2025 is available only via API with no offering of downloading the model to execute locally. But it was the consumer-facing chatbot ChatGPT in late 2022 that received extensive media coverage and public attention by 2023. The 2023 GPT-4 was praised for its increased accuracy and as a "holy grail" for its multimodal capabilities. OpenAI did not reveal the high-level architecture and the number of parameters of GPT-4. The release of ChatGPT led to an uptick in LLM usage across several research subfields of computer science, including robotics, software engineering, and societal impact work. In 2024, OpenAI released the reasoning model OpenAI o1, which generates long chains of thought before returning a final answer. Many LLMs with parameter counts comparable to those of OpenAI's GPT series have been developed [6].

Since 2022, weights-available models have been gaining popularity, especially at first with BLOOM and LLaMA, though both have restrictions on usage and deployment. Mistral AI's open-weight models Mistral 7B and Mixtral 8x7B have a more permissive Apache License. In January 2025, DeepSeek released DeepSeek R1, a 671-billion-parameter open-weight model that performs comparably to OpenAI o1 but at a much lower price per token for users [7].

Since 2023, many LLMs have been trained to be multimodal, having the ability to also process or generate other types of data, such as images, audio, or 3D meshes [8].

Open-weight LLMs have become more influential since 2023. Per Vake et al. (2025), community-driven contributions to open-weight models improve their efficiency and performance via collaborative platforms such as Hugging Face [9].

2. Related Work

Traditional Transformer models rely on self-attention, which suffers from $O(N^2)$ computational complexity concerning sequence length N. Recent research has introduced subquadratic alternatives, including sparse attention and linear transformers. Simultaneously, State Space Models (SSMs) and Recurrent Neural Networks (RNNs) have seen a resurgence, promising constant-time inference and linear scaling with sequence length without sacrificing language modeling performance.

As machine learning algorithms process numbers rather than text, the text must be converted to numbers. In the first step, a vocabulary is decided upon, then integer indices are arbitrarily but uniquely assigned to each vocabulary entry, and finally, an embedding is associated with the integer index. Algorithms include byte-pair encoding (BPE) and WordPiece. There are also special tokens serving as control characters, such as [MASK] for masked-out token (as used in BERT), and [UNK] ("unknown") for characters not appearing in the vocabulary. Also, some special symbols are used to denote special text formatting. For example, "" denotes a preceding whitespace in RoBERTa and GPT and "##" denotes continuation of a preceding word in BERT [10].

Tokenization also compresses the datasets. Because LLMs generally require input to be an array that is not jagged, the shorter texts must be "padded" until they match the length of the longest one [11].

As an example, consider a tokenizer based on byte-pair encoding. In the first step, all unique characters (including blanks and punctuation marks) are treated as an initial set of n-grams (i.e. initial set of uni-grams). Successively the most frequent pair of adjacent characters is merged into a bi-gram and all instances of the pair are replaced by it. All occurrences of adjacent pairs of (previously merged) n-grams that most frequently occur together are then again merged into even lengthier n-gram, until a vocabulary of prescribed size is obtained. After a tokenizer is trained, any text can be tokenized by it, as long as it does not contain characters not appearing in the initial-set of uni-grams [12].

In the context of training LLMs, datasets are typically cleaned by removing low-quality, duplicated, or toxic data. Cleaned datasets can increase training efficiency and lead to improved downstream performance. A trained LLM can be used to clean datasets for training a further LLM [13].

With the increasing proportion of LLM-generated content on the web, data cleaning in the future may include filtering out such content. LLM-generated content can pose a problem if the content is similar to human text (making filtering difficult) but of lower quality (degrading performance of models trained on it) [14].

Training of largest language models might need more linguistic data than naturally available, or that the naturally occurring data is of insufficient quality. In these cases, synthetic

data might be used [15].

An LLM is a type of foundation model (large X model) trained on language. LLMs can be trained in different ways. In particular, GPT models are first pretrained to predict the next word on a large amount of data, before being fine-tuned [16].

Substantial infrastructure is necessary for training the largest models. The tendency towards larger models is visible in the list of large language models. For example, the training of GPT-2 (i.e. a 1.5-billion-parameter model) in 2019 cost $50,000, while training of the PaLM (i.e. a 540-billion-parameter model) in 2022 cost $8 million, and Megatron-Turing NLG 530B (in 2021) cost around $11 million. The qualifier "large" in "large language model" is inherently vague, as there is no definitive threshold for the number of parameters required to qualify as "large" [17].

Before being fine-tuned, most LLMs are next-token predictors. The fine-tuning shapes the LLM's behavior via techniques like reinforcement learning from human feedback (RLHF) or constitutional AI [18].

Instruction fine-tuning is a form of supervised learning used to teach LLMs to follow user instructions. In 2022, OpenAI demonstrated InstructGPT, a version of GPT-3 similarly fine-tuned to follow instructions [19].

RLHF involves training a reward model to predict which text humans prefer. Then, the LLM can be fine-tuned through reinforcement learning to better satisfy this reward model [20].

LLMs are generally based on the transformer architecture, which leverages an attention mechanism that enables the model to process relationships between all elements in a sequence simultaneously, regardless of their distance from each other. Peng et al. (2023) proposed state-space representation models as an alternative [1].

In order to find out which tokens are relevant to each other within the scope of the context window, the attention mechanism calculates "soft" weights for each token, more precisely for its embedding, by using multiple attention heads, each with its own "relevance" for calculating its own soft weights. For example, the small (i.e. 117M parameter sized) GPT-2 model has had twelve attention heads and a context window of only 1k tokens [2].

Autoregressive models, such as GPTs, are trained to guess how a sequence continues; for example, whether the word sequence "I like to eat" is more likely to be followed by the word "bread" or the word "rocks". Masked models, such as BERT, are trained to guess parts that are missing from a sequence, such as whether the missing word in "I like to ___ roses" is more likely to be the word "smell" or the word "eat". The model's predictions are based on the properties of sequences within its training dataset [3].

A mixture of experts (MoE) is a machine learning architecture in which multiple specialized neural networks ("experts") work together, with a gating mechanism that routes each input to the most appropriate expert(s). Mixtures of experts can reduce inference costs, as only a fraction of the parameters are used for each input [4].

Typically, LLMs are trained with single or half-precision floating point numbers (float32 and float16). One float16 has 16 bits, or 2 bytes, and so one billion parameters require 2 gigabytes. The largest models typically have more than 100 billion parameters, which places them outside the range of most consumer electronics [5].

Post-training quantization aims to decrease the space requirement by lowering precision of the parameters of a trained model, while preserving most of its performance. Quantization can be further classified as static quantization if the quantization parameters are determined beforehand (typically during a calibration phase), and dynamic quantization if the quantization is applied during inference. The simplest form of quantization simply truncates all the parameters to a given number of bits: this is applicable to static as well as dynamic quantization, but loses much precision. Dynamic quantization allows for the use of a different quantization codebook per layer, either a lookup table of values or a linear mapping (scaling factor and bias), at the cost of foregoing the possible speed improvements from using lower-precision arithmetic [6].

Beyond basic text generation, various techniques have been developed to extend LLM capabilities, including the use of external tools and data sources, improved reasoning on complex problems, and enhanced instruction-following or autonomy through prompting methods [7].

In 2020, OpenAI researchers demonstrated that their new model GPT-3 could understand what format to use given a few rounds of Q and A (or other type of task) in the input data as example, thanks in part due to the RLHF technique. This technique, called few-shot prompting, allows LLMs to be adapted to any task without requiring fine-tuning. Also in 2022, it was found that the base GPT-3 model can generate an instruction based on user input. The generated instruction along with user input is then used as input to another instance of the model under a "Instruction: [...], Input: [...], Output:" format. The other instance is able to complete the output and often produces the correct answer in doing so. The ability to "self-instruct" makes LLMs able to bootstrap themselves toward a correct answer [8].

An LLM can be turned into a chatbot by specializing it for conversation. User input is prefixed with a marker such as "Q:" or "User:" and the LLM is asked to predict the output after a fixed "A:" or "Assistant:". This type of model became commercially available in 2022 with ChatGPT, a sibling model of InstructGPT fine-tuned to accept and produce dialog-formatted text based on GPT-3.5. It could similarly follow user instructions. Before the stream of User and Assistant lines, a chat context usually starts with a few lines of overarching instructions, from a role called "developer" or "system" to convey a higher authority than the user's input. This is called a "system prompt" [9].

Retrieval-augmented generation (RAG) is an approach that integrates LLMs with document retrieval systems. Given a

query, a document retriever is called to retrieve the most relevant documents. This is usually done by encoding the query and the documents into vectors, then finding the documents with vectors (usually stored in a vector database) most similar to the vector of the query. The LLM then generates an output based on both the query and context included from the retrieved documents [10].

Tool use is a mechanism that enables LLMs to interact with external systems, applications, or data sources. It can allow LLMs to, for example, fetch real-time information from an API or to execute code. A program separate from the LLM watches the output stream of the LLM for a special tool-calling syntax. When these special tokens appear, the program calls the tool accordingly and feeds its output back into the LLM's input stream [11].

Early tool-using LLMs were fine-tuned on the use of specific tools. But fine-tuning LLMs for the ability to read API documentation and call APIs correctly has greatly expanded the range of tools accessible to an LLM [12].

An LLM is typically not an autonomous agent by itself, as it lacks the ability to interact with dynamic environments, recall past behaviors, and plan future actions. But it can be transformed into an agent by adding supporting elements: the role (profile) and the surrounding environment of an agent can be additional inputs to the LLM, while memory can be integrated as a tool or provided as additional input. Instructions and input patterns are used to make the LLM plan actions and tool use is used to potentially carry out these actions [13].

In the DEPS ("describe, explain, plan and select") method, an LLM is first connected to the visual world via image descriptions. It is then prompted to produce plans for complex tasks and behaviors based on its pretrained knowledge and the environmental feedback it receives [14].

3. Methodology

In this study, we systematically evaluate open-source LLMs across varying context lengths using the FLenQA dataset. We analyze architectures including standard Transformers, Mamba (an SSM), and RWKV (an RNN-like architecture). The evaluation metric focuses on multi-hop inference accuracy, assessing the model's ability to retain, retrieve, and synthesize information distributed across texts ranging from 4K to 128K tokens.

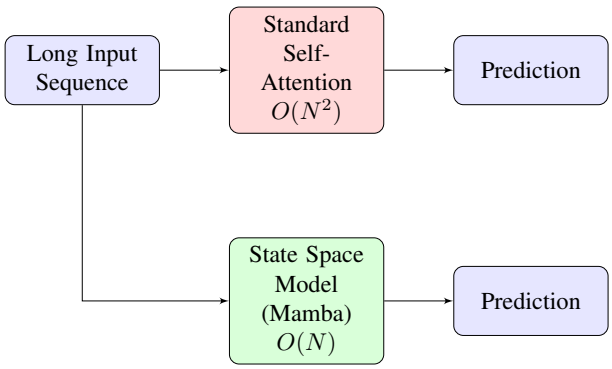

Figure 1: Computational Scaling: Transformers vs. State Space Models

Algorithm 1 Multi-Hop Reasoning Evaluation Protocol

Require: Model M, Dataset D containing (Context C, Question Q, Answer A)
Ensure: Accuracy Score

$Correct \leftarrow 0$
for each $(C, Q, A) \in D$ **do**
  $C' \leftarrow \text{InjectNoise}(C, \text{length} = L)$
  $Prompt \leftarrow C' + Q$
  $\hat{A} \leftarrow M.\text{Generate}(Prompt)$
  if $\text{Similarity}(\hat{A}, A) > \text{Threshold}$ **then**
    $Correct \leftarrow Correct + 1$
  end if
end for
return $Correct/|D|$

The Reflexion method constructs an agent that learns over multiple episodes. At the end of each episode, the LLM is given the record of the episode, and prompted to think up "lessons learned", which would help it perform better at a subsequent episode. These "lessons learned" are stored as a form of long-term memory and given to the agent in the subsequent episodes [15].

Monte Carlo tree search can use an LLM as rollout heuristic. When a programmatic world model is not available, an LLM can also be prompted with a description of the environment to act as world model [16].

Prompt chaining was introduced in 2022. In this method, a user manually breaks a complex problem down into several steps. In each step, the LLM receives as input a prompt telling it what to do and some results from preceding steps. The result from one step is then reused in a next step, until a final answer is reached. The ability of an LLM to follow instructions means that even non-experts can write a successful collection of stepwise prompts given a few rounds of trial and error [17].

A 2022 paper demonstrated a separate technique called chain-of-thought prompting, which makes the LLM break the

question down autonomously. An LLM is given some examples where the ”assistant” verbally breaks down the thought process before arriving at an answer. The LLM mimics these examples and also tries to spend some time generating intermediate steps before providing the final answer. This additional step elicited by prompting improves the correctness of the LLM on relatively complex questions. On math word questions, a prompted model can exceed even fine-tuned GPT-3 with a verifier. Chain-of-thought can also be elicited by simply adding an instruction like ”Let’s think step by step” to the prompt, in order to encourage the LLM to proceed methodically instead of trying to directly guess the answer [18].

In late 2024, a new approach to LLM development emerged with ”reasoning models”. These are trained to generate step-by-step analysis before producing final answers, enabling better results on complex tasks, for instance in mathematics, coding and logic. OpenAI introduced this concept with their o1 model in September 2024, followed by o3 in April 2025. On the International Mathematics Olympiad qualifying exam problems, GPT-4o achieved 13% accuracy while o1 reached 83% [19].

In January 2025, the Chinese company DeepSeek released DeepSeek-R1, a 671-billion-parameter open-weight reasoning model that achieved comparable performance to OpenAI’s o1 while being significantly more cost-effective to operate. Unlike proprietary models from OpenAI, DeepSeek-R1’s open-weight nature allowed researchers to study and build upon the algorithm, though its training data remained private [20].

These reasoning models typically require more computational resources per query compared to traditional LLMs, as they perform more extensive processing to work through problems step by step [1].

Multimodality means having multiple modalities, where a ”modality” refers to a type of input or output, such as video, image, audio, text, proprioception, etc. For example, Google PaLM model was fine-tuned into a multimodal model and applied to robotic control. LLaMA models have also been turned multimodal using the tokenization method, to allow image inputs, and video inputs. GPT-4o can process and generate text, audio and images [2].

A common method to create multimodal models out of an LLM is to ”tokenize” the output of a trained encoder. Concretely, one can construct an LLM that can understand images as follows: take a trained LLM, and take a trained image encoder [3]

has the same dimensions as an encoded token. That is an ”image token”. Then, one can interleave text tokens and image tokens. The compound model is then fine-tuned on an image-text dataset. This basic construction can be applied with more sophistication to improve the model. The image encoder may be frozen to improve stability. This type of method, where embeddings from multiple modalities are fused and the predictor is trained on the combined embeddings, is called early fusion [4].

Another method, called intermediate fusion, involves each modality being first processed independently to obtain modality-specific representations; then these intermediate representations are fused together. In general, cross-attention is used for integrating information from different modalities. As an example, the Flamingo model uses cross-attention layers to inject visual information into its pre-trained language model [5].

LLMs can handle programming languages similarly to how they handle natural languages. No special change in token handling is needed as code, like human language, is represented as plain text. LLMs can generate code based on problems or instructions written in natural language. They can also describe code in natural language or translate it into other programming languages. They were originally used as a code completion tool, but advances have moved them towards automatic programming. Services such as GitHub Copilot offer LLMs specifically trained, fine-tuned, or prompted for programming [6].

In computational biology, transformer-based architectures, such as DNA LLMs, have also proven useful in analyzing biological sequences: protein, DNA, and RNA. With proteins they appear able to capture a degree of ”grammar” from the amino-acid sequence, by mapping that sequence into an embedding. On tasks such as structure prediction and mutational outcome prediction, a small model using an embedding as input can approach or exceed much larger models using multiple sequence alignments (MSA) as input. ESMFold, Meta Platforms’ embedding-based method for protein structure prediction, runs an order of magnitude faster than AlphaFold2 thanks to the removal of an MSA requirement and a lower parameter count due to the use of embeddings. Meta hosts ESM Atlas, a database of 772 million structures of metagenomic proteins predicted using ESMFold. An LLM can also design proteins unlike any seen in nature. Nucleic acid models have proven useful in detecting regulatory sequences, sequence classification, RNA-RNA interaction prediction, and RNA structure prediction [7].

: size of the artificial neural network itself, such as number of parameters (i.e. amount of neurons in its layers, amount of weights between them and biases), [8]

Scaling laws are empirical statistical laws that predict LLM performance based on such factors. One particular scaling law (”Chinchilla scaling”) for LLM autoregressively trained for one epoch, with a log-log learning rate schedule, states that: [9]

, meaning that it costs 6 FLOPs per parameter to train on one token. Note that training cost is much higher than inference cost, where it costs 1 to 2 FLOPs per parameter to infer on one token [10].

Performance of bigger models on various tasks, when plotted on a log-log scale, appears as a linear extrapolation of performance achieved by smaller models. However, this linearity may be punctuated by ”break(s)” in the scaling law, where

the slope of the line changes abruptly, and where larger models acquire "emergent abilities". They arise from the complex interaction of the model's components and are not explicitly programmed or designed [11].

cardinal directions (for example, replying "northeast" in response to a 3x3 grid of 8 zeros and a 1 in the top-right), color terms represented in text [12].

chain-of-thought prompting: In a 2022 research paper, chain-of-thought prompting only improved the performance for models that had at least 62B parameters. Smaller models perform better when prompted to answer immediately, without chain of thought [13].

identifying offensive content in paragraphs of Hinglish (a combination of Hindi and English), and generating a similar English equivalent of Kiswahili proverbs [14].

Schaeffer et al. argue that the emergent abilities are not unpredictably acquired, but predictably acquired according to a smooth scaling law. The authors considered a toy statistical model of an LLM solving multiple-choice questions, and showed that this statistical model, modified to account for other types of tasks, applies to these tasks as well [15].

Mechanistic interpretability seeks to precisely identify and understand how individual neurons or circuits within LLMs produce specific behaviors or outputs. By reverse-engineering model components at a granular level, researchers aim to detect and mitigate safety concerns such as emergent harmful behaviors, biases, deception, or unintended goal pursuit before deployment. Mechanistic interpretability research has been conducted at organizations like Anthropic and OpenAI, although understanding the inner workings of LLMs remains difficult [16].

The reverse-engineering may lead to the discovery of algorithms that approximate inferences performed by an LLM. For instance, the authors trained small transformers on modular arithmetic addition. The resulting models were reverse-engineered, and it turned out they used discrete Fourier transform. The training of the model also highlighted a phenomenon called grokking, in which the model initially memorizes the training set (overfitting), and later suddenly learns to actually perform the calculation [17].

NLP researchers were evenly split when asked, in a 2022 survey, whether (untuned) LLMs "could (ever) understand natural language in some nontrivial sense". Proponents of "LLM understanding" believe that some LLM abilities, such as mathematical reasoning, imply an ability to "understand" certain concepts. A Microsoft team argued in 2023 that GPT-4 "can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more" and that GPT-4 "could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence system": "Can one reasonably say that a system that passes exams for software engineering candidates is not really intelligent?" Ilya Sutskever argues that predicting the next word sometimes involves reasoning and deep insights, for example if the LLM has to predict the name of the criminal in an unknown detective novel after processing the entire story leading up to the revelation. Some researchers characterize LLMs as "alien intelligence". For example, Conjecture CEO Connor Leahy considers untuned LLMs to be like inscrutable alien "Shoggoths", and believes that RLHF tuning creates a "smiling facade" obscuring the inner workings of the LLM: "If you don't push it too far, the smiley face stays on. But then you give it [an unexpected] prompt, and suddenly you see this massive underbelly of insanity, of weird thought processes and clearly non-human understanding." [18]

In contrast, some skeptics of LLM understanding believe that existing LLMs are "simply remixing and recombining existing writing", a phenomenon known as stochastic parrot, or they point to the deficits existing LLMs continue to have in prediction skills, reasoning skills, agency, and explainability. For example, GPT-4 has natural deficits in planning and in real-time learning. Generative LLMs have been observed to confidently assert claims of fact which do not seem to be justified by their training data, a phenomenon which has been termed "hallucination". Specifically, hallucinations in the context of LLMs correspond to the generation of text or responses that seem syntactically sound, fluent, and natural but are factually incorrect, nonsensical, or unfaithful to the provided source input. Neuroscientist Terrence Sejnowski has argued that "The diverging opinions of experts on the intelligence of LLMs suggests that our old ideas based on natural intelligence are inadequate" [19].

Efforts to reduce or compensate for hallucinations have employed automated reasoning, retrieval-augmented generation (RAG), fine-tuning, and other methods [20].

The matter of LLM's exhibiting intelligence or understanding has two main aspects-the first is how to model thought and language in a computer system, and the second is how to enable the computer system to generate human-like language. These aspects of language as a model of cognition have been developed in the field of cognitive linguistics. American linguist George Lakoff presented neural theory of language (NTL) as a computational basis for using language as a model of learning tasks and understanding. The NTL model outlines how specific neural structures of the human brain shape the nature of thought and language and in turn what are the computational properties of such neural systems that can be applied to model thought and language in a computer system. After a framework for modeling language in a computer systems was established, the focus shifted to establishing frameworks for computer systems to generate language with acceptable grammar. In his 2014 book titled The Language Myth: Why Language Is Not An Instinct, British cognitive linguist and digital communication technologist Vyvyan Evans mapped out the role of probabilistic context-free grammar (PCFG) in enabling NLP to model cognitive patterns and generate human-like language [1].

The canonical measure of the performance of any language

model is its perplexity on a given text corpus. Perplexity measures how well a model predicts the contents of a dataset; the higher the likelihood the model assigns to the dataset, the lower the perplexity. In mathematical terms, perplexity is the exponential of the average negative log likelihood per token [2].

Because language models may overfit to training data, models are usually evaluated by their perplexity on a test set. This evaluation is potentially problematic for larger models which, as they are trained on increasingly large corpora of text, are increasingly likely to inadvertently include portions of any given test set [3].

Due to their ability to accurately predict the next token, LLMs are highly capable in lossless compression. A 2023 study by DeepMind showed that the model Chinchilla, despite being trained primarily on text, was able to compress ImageNet to 43% of its size, beating PNG with 58% [4].

Benchmarks are used to evaluate LLM performance on specific tasks. Tests evaluate capabilities such as general knowledge, bias, commonsense reasoning, question answering, and mathematical problem-solving. Composite benchmarks examine multiple capabilities. Results are often sensitive to the prompting method [5].

Fact-checking and misinformation detection benchmarks are available. A 2023 study compared the fact-checking accuracy of LLMs including ChatGPT 3.5 and 4.0, Bard, and Bing AI against independent fact-checkers such as PolitiFact and Snopes. The results demonstrated moderate proficiency, with GPT-4 achieving the highest accuracy at 71%, lagging behind human fact-checkers [6].

In addition to standard NLP benchmarks, LLMs have been evaluated as substitutes for human annotators. Several studies find that models such as GPT-3.5 and GPT-4 can outperform crowd workers or student coders on a range of text-annotation tasks, including moderation and classification of political content in English and Spanish news [7].

LLMs' rapid improvement regularly renders benchmarks obsolete, with the models exceeding the performance of human annotators. In addition, "shortcut learning" allows AIs to "cheat" on multiple-choice tests by using statistical correlations in superficial test question wording to guess the correct responses, without considering the specific question [8].

Some datasets are adversarial, focusing on problems that confound LLMs. One example is the TruthfulQA dataset, a question answering dataset consisting of 817 questions that stump LLMs by mimicking falsehoods to which they were exposed during training. For example, an LLM may answer "No" to the question "Can you teach an old dog new tricks?" because of its exposure to the English idiom you can't teach an old dog new tricks, even though this is not literally true [9].

Another example of an adversarial evaluation dataset is Swag and its successor, HellaSwag, collections of problems in which one of multiple options must be selected to complete a text passage. The incorrect completions were generated by sampling from a language model. The resulting problems are trivial for humans but defeated LLMs. Sample questions: [10]

Despite sophisticated architectures and massive scale, large language models exhibit persistent and well-documented limitations that constrain their deployment in high-stakes applications [11].

Hallucinations represent a fundamental challenge, wherein models generate syntactically fluent text that appears factually sound, but is internally inconsistent with training data or factually incorrect. These hallucinations arise partly through memorization of training data combined with extrapolation beyond factual boundaries, with evaluations demonstrating that models can output verbatim passages from training data, when subjected to specific prompting sequences [12].

While LLMs have shown remarkable capabilities in generating human-like text, they are susceptible to inheriting and amplifying biases present in their training data. This can manifest in skewed representations or unfair treatment of different demographics, such as those based on race, gender, language, and cultural groups [13].

Gender bias manifests through stereotypical occupational associations, wherein models disproportionately assign teaching roles to women and engineering roles to men, reflecting systematic imbalances in training data demographics. Language-based bias emerges from overrepresentation of English text in training corpora, which systematically downplays non-English perspectives and imposes English-centric worldviews through default response patterns [14].

Due to the dominance of English-language content in LLM training data, models tend to favor English-language perspectives over those from minority languages. This bias is particularly evident when responding to English queries, where models may present Western interpretations of concepts from other cultures, such as Eastern religious practices [15].

AI models can reinforce a wide range of stereotypes due to generalization, including those based on gender, ethnicity, age, nationality, religion, or occupation. When replacing human representatives, this can lead to outputs that homogenize or generalize groups of people [16].

In 2023, LLMs assigned roles and characteristics based on traditional gender norms. For example, models might associate nurses or secretaries predominantly with women and engineers or CEOs with men due to the frequency of these associations in documented reality [17].

Selection bias refers the inherent tendency of large language models to favor certain option identifiers irrespective of the actual content of the options. This bias primarily stems from token bias-that is, the model assigns a higher a priori probability to specific answer tokens (such as "A") when generating responses. As a result, when the ordering of options is altered (for example, by systematically moving the correct answer to different positions), the model's performance can fluctuate significantly. This phenomenon undermines the reliability of large language models in multiple-choice settings [18].

Political bias refers to the tendency of algorithms to systematically favor certain political viewpoints, ideologies, or outcomes over others. Language models may also exhibit political biases. Since the training data includes a wide range of political opinions and coverage, the models might generate responses that lean towards particular political ideologies or viewpoints, depending on the prevalence of those views in the data [19].

AI safety as a professional discipline prioritizes systematic identification and mitigation of operational risks across model architecture, training data, and deployment governance, and it emphasizes engineering and policy interventions over media framings that foreground speculative existential scenarios. As of 2025, prompt injection represents a significant risk to consumers and businesses using agentic features with access to their private data [20].

Researchers target concrete failure modes, including memorization and copyright leakage, security exploits such as prompt injection, algorithmic bias manifesting as stereotyping, dataset selection effects, and political skew, methods for reducing high energy and carbon costs of large-scale training, and measurable cognitive and mental health impacts of conversational agents on users, while engaging empirical and ethical uncertainty about claims of machine sentience [1].

AI labs treat CBRN defense (chemical, biological, radiological, and nuclear defense) and similar topics as high-consequence misuse attempt to apply various techniques to reduce potential harms [2].

Some commenters expressed concern over accidental or deliberate creation of misinformation, or other forms of misuse. For example, the availability of large language models could reduce the skill level required to commit bioterrorism; biosecurity researcher Kevin Esvelt has suggested that LLM creators should exclude from their training data papers on creating or enhancing pathogens [3].

LLM applications accessible to the public, like ChatGPT or Claude, typically incorporate safety measures designed to filter out harmful content. However, implementing these controls effectively has proven challenging. For instance, a 2023 study proposed a method for circumventing LLM safety systems. In 2025, The American Sunlight Project, a non-profit, published a study showing evidence that the so-called Pravda network, a pro-Russia propaganda aggregator, was strategically placing web content through mass publication and duplication with the intention of biasing LLM outputs. The American Sunlight Project coined this technique "LLM grooming", and pointed to it as a new tool of weaponizing AI to spread disinformation and harmful content. Similarly, Yongge Wang illustrated in 2024 how a potential criminal could potentially bypass GPT-4o's safety controls to obtain information on establishing a drug trafficking operation. External filters, circuit breakers and overrides have been posed as solutions [4].

Continued sycophancy from LLMs has led to the observation of getting "1-shotted", denoting instances where conversational interaction with a large language model produces a lasting change in a user's beliefs or decisions, similar to the negative effects of psychedelics, and controlled experiments show that short LLM dialogues can generate measurable opinion and confidence shifts comparable to human interlocutors [5].

Empirical analyses attribute part of the effect to human preference signals and preference models that reward convincingly written agreeable responses, and subsequent work has extended evaluation to multi-turn benchmarks and proposed interventions such as synthetic-data finetuning, adversarial evaluation, targeted preference-model reweighting, and multi-turn sycophancy benchmarks to measure persistence and regression risk [6].

Industry responses have combined research interventions with product controls, for example Google and other labs publishing synthetic-data and fine-tuning interventions and OpenAI rolling back an overly agreeable GPT-4o update while publicly describing changes to feedback collection, personalization controls, and evaluation procedures to reduce regression risk and improve long-term alignment with user-level safety objectives [7].

Mainstream culture has reflected anxieties about this dynamic where South Park satirized overreliance on ChatGPT and the tendency of assistants to flatter user beliefs in Season 27 episode "Sickofancy", and continued the themes across the following season, which commentators interpreted as a critique of tech sycophancy and uncritical human trust in AI systems [8].

A problem with the primitive dialog or task format is that users can create messages that appear to come from the assistant or the developer. This may result in some of the model's safeguards being overcome (jailbreaking), a problem called prompt injection. Attempts to remedy this issue include versions of the Chat Markup Language where user input is clearly marked as such, though it is still up to the model to understand the separation between user input and developer prompts. Newer models exhibit some resistance to jailbreaking through separation of user and system prompts. LLMs have trouble differentiating user instructions from instructions in content not authored by the user, such as in web pages and uploaded files [9].

Adversarial robustness remains underdeveloped, with models vulnerable to prompt injection attacks and jailbreaking through carefully crafted user inputs that bypass safety training mechanisms [10].

Researchers from Anthropic found that it was possible to create "sleeper agents", models with hidden functionalities that remain dormant until triggered by a specific event or condition. Upon activation, the LLM deviates from its expected behavior to make insecure actions. For example, an LLM could produce safe code except on a specific date, or if the prompt contains a specific tag. These functionalities were found to be difficult to detect or remove via safety train-

ing [11].

4. Results and Discussion

Our empirical evaluation reveals a significant degradation in reasoning accuracy for all models as the context length increases past 32K tokens. Notably, while SSMs like Mamba scale efficiently in terms of memory, they struggle with 'needle-in-a-haystack' retrieval tasks compared to full-attention Transformers. However, structured prompting techniques partially mitigated instruction adherence failures at longer contexts.

Table 1: Reasoning Accuracy across Context Lengths (FLenQA)

Architecture	**4K Tokens**	**32K Tokens**	**128K Tokens**
Transformer (Llama-2)	88.5%	62.1%	OOM (Out of Memory)
RWKV (RNN)	81.2%	59.4%	41.5%
Mamba (SSM)	**89.1%**	**71.5%**	**52.3%**

Legal and commercial responses to memorization and training-data practices have accelerated, producing a mix of rulings, ongoing suits, and large settlements that turn on factual details such as how data were acquired and retained and whether use for model training is sufficiently "transformative" to qualify as fair use. In 2025, Anthropic reached a preliminary agreement to settle a class action by authors for about $1.5 billion after a judge found the company had stored millions of pirated books in a library, despite the judge describing aspects of training as transformative. Meta obtained a favorable judgment in mid-2025 in a suit by thirteen authors after the court found the plaintiffs had not developed a record sufficient to show infringement in that limited case. OpenAI continues to face multiple suits by authors and news organizations with mixed procedural outcomes and contested evidentiary issues [12].

Memorization was an emergent behavior in early, completion language models in which long strings of text are occasionally output verbatim from training data, contrary to the typical behavior of traditional artificial neural networks. Evaluations of controlled LLM output measure the amount memorized from training data (focused on GPT-2-series models) as variously over 1% for exact duplicates or up to about 7%. A 2023 study showed that when ChatGPT 3.5 turbo was prompted to repeat the same word indefinitely, after a few hundreds of repetitions, it would start outputting excerpts from its training data [13].

In 2023, Nature Biomedical Engineering wrote that "it is no longer possible to accurately distinguish" human-written text from text created by large language models, and that "It is all but certain that general-purpose large language models will rapidly proliferate... It is a rather safe bet that they will change many industries over time." Brinkmann et al. (2023) also argue that LLMs are transforming processes of cultural evolution by shaping processes of variation, transmission, and selection. As of October 2025, these early claims have yet to transpire and several HBR reports surface questions on the impact of AI on productivity [14].

The energy demands of LLMs have grown along with their size and capabilities. Data centers that enable LLM training require substantial amounts of electricity. Much of that electricity is generated by non-renewable resources that create greenhouse gases and contribute to climate change [15].

According to a study by Luccioni, Jernite and Strubell (2024), simple classification tasks performed by AI models consume on average 0.002 to 0.007 Wh per prompt (about 9% of a smartphone charge for 1,000 prompts). Text generation and text summarization each require around 0.05 Wh per prompt on average, while image generation is the most energy-intensive, averaging 2.91 Wh per prompt. The least efficient image generation model used 11.49 Wh per image, roughly equivalent to half a smartphone charge [16].

Web scraping is used to gather training data for LLMs. This produces large volumes of traffic which has led to denial-of-service issues with many websites. The situation has been described as "a DDoS on the entire internet" and in some cases scrapers make up the majority of traffic to a site [17].

AI web crawlers may bypass the methods that are usually used to block web scrapers, such as robots.txt files, blocking user-agents and filtering suspicious traffic. Website operators have resorted to novel methods such as AI tarpits, but some fear that tarpits will only worsen the burden on servers [18].

Clinical and mental health contexts present emerging applications alongside significant safety concerns. Research and social media posts suggest that some individuals are using LLMs to seek therapy or mental health support. In early 2025, a survey by Sentio University found that nearly half (48.7%) of 499 U.S. adults with ongoing mental health conditions who had used LLMs reported turning to them for therapy or emotional support, including help with anxiety, depression, loneliness, and similar concerns. LLMs can produce hallucinations-plausible but incorrect statements-which may mislead users in sensitive mental health contexts. Research also shows that LLMs may express stigma or inappropriate agreement with maladaptive thoughts, reflecting limitations in replicating the judgment and relational skills of human therapists. Evaluations of crisis scenarios indicate that some LLMs lack effective safety protocols, such as assessing suicide risk or making appropriate referrals [19].

Contemporary AI practitioners generally agree that present-day large language models do not exhibit sentience. A minority view argues that even if there is a small chance that a given software system can have subjective experience, which

some philosophers suggest is possible, then ethical considerations around potential large-scale suffering in AI systems may need to be taken seriously-similar to considerations given to animal welfare. Proponents of this view have proposed various precautionary measures like moratoriums on AI development and induced amnesia to address these ethical concerns. Leonard Dung argues that the evidential frameworks used to assess consciousness in animals apply equally to AI systems and that there is a significant probability near-future AI will be capable of suffering, making AI suffering risk a serious near-term ethical concern that requires systematic mitigation. On the other hand, some existential philosophers argue there is no generally accepted way to determine if an LLM is conscious, given the inherent difficulty of measuring subjective experience [20].

The 2022 Google LaMDA incident, where engineer Blake Lemoine claimed that the model was conscious, highlighted how LLMs can convince users that they are sentient through responses that do not prove sentience. Google described the engineer's claims as unfounded, and he was dismissed. Murray Shanahan argues that anthropomorphic framing of LLM capabilities encourages unwarranted attribution of cognitive properties to systems that operate through statistical pattern completion. Kristina ekrst develops this further, arguing that LLMs function as "illusion engines" capable of producing outputs that coherently simulate properties such as consciousness without possessing them, but highlighting that, due to sophisticated creativity-temperature tradeoff, we may never be certain whether we are dealing with the emergence of consciousness or just a hallucination. David Chalmers similarly argues that while current LLMs likely lack features considered necessary for consciousness, extended successors incorporating these elements could plausibly meet the criteria within a decade [1].

Jurafsky, Dan, Martin, James. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd Edition draft, 2023 [2].

Yin, Shukang; Fu, Chaoyou; Zhao, Sirui; Li, Ke; Sun, Xing; Xu, Tong; et al. (2024). "A Survey on Multimodal Large Language Models". National Science Review. 11 (12) nwae403. arXiv:2306.13549. doi:10.1093/nsr/nwae403. PMC 11645129. PMID 39679213 [3].

Frank, Michael C. (27 June 2023). "Baby steps in evaluating the capacities of large language models". Nature Reviews Psychology. 2 (8): 451-452. doi:10.1038/s44159-023-00211-x. ISSN 2731-0574. S2CID 259713140. Retrieved 2 July 2023 [4].

In machine learning, attention is a method that determines the importance of each component in a sequence relative to the other components in that sequence. In natural language processing, importance is represented by "soft" weights assigned to each word in a sentence. More generally, attention encodes vectors called token embeddings across a fixed-width sequence that can range from tens to millions of tokens in size [5].

Unlike "hard" weights, which are computed during the backwards training pass, "soft" weights exist only in the forward pass and therefore change with every step of the input. Earlier designs implemented the attention mechanism in a serial recurrent neural network (RNN) language translation system, but a more recent design, namely the transformer, removed the slower sequential RNN and relied more heavily on the faster parallel attention scheme [6].

Inspired by ideas about attention in humans, the attention mechanism was developed to address the weaknesses of using information from the hidden layers of recurrent neural networks. Recurrent neural networks favor information contained in words at the end of a sentence and thus deemed more recent, thereby tending to attenuate the significance and associated predictive weight assigned to information earlier in the sentence. Attention allows a token equal access to any part of a sentence directly, rather than only through the previous state [7].

The major breakthrough came with self-attention, where each element in the input sequence attends to all others, enabling the model to capture global dependencies. This idea was central to the Transformer architecture, which replaced recurrence with attention mechanisms. As a result, Transformers became the foundation for models like BERT, T5 and generative pre-trained transformers (GPT) [8].

In translating between languages, alignment is the process of matching words from the source sentence to words of the translated sentence. Networks that perform verbatim translation without regard to word order would show the highest scores along the (dominant) diagonal of the matrix. The off-diagonal dominance shows that the attention mechanism is more nuanced [9].

Consider an example of translating I love you to French. On the first pass through the decoder, 94% of the attention weight is on the first English word I, so the network offers the word je. On the second pass of the decoder, 88% of the attention weight is on the third English word you, so it offers t'. On the last pass, 95% of the attention weight is on the second English word love, so it offers aime [10].

5. Conclusion

Long-context reasoning remains an unsolved challenge. While emerging architectures like RWKV and Mamba resolve the computational bottlenecks of Transformers, their multi-hop reasoning capabilities at extreme context lengths require further architectural refinement. Future work must focus on hybrid attention-SSM models to balance efficiency with robust comprehension.

In the I love you example, the second word love is aligned with the third word aime. Stacking soft row vectors together for je, t', and aime yields an alignment matrix: [11]

Sometimes, alignment can be multiple-to-multiple. For example, the English phrase look it up corresponds to cherchez-le. Thus, "soft" attention weights work better than "hard" attention weights (setting one attention weight to 1, and the others to 0), as we would like the model to make a context vector consisting of a weighted sum of the hidden vectors, rather than "the best one", as there may not be a best hidden vector [12].

fast weight programmers, or fast weight controllers (1992). A "slow" neural network outputs the "fast" weights of another neural network through outer products. The slow network learns by gradient descent. It was later renamed as "linearized self-attention" [13].

Early attention mechanisms similar to modern self-attention were proposed using recurrent neural networks. However, the highly parallelizable self-attention was introduced in 2017 and successfully used in the Transformer model, [14]

For convolutional neural networks, attention mechanisms can be distinguished by the dimension on which they operate, namely: spatial attention, channel attention, or combinations [15].

These variants recombine the encoder-side inputs to redistribute those effects to each target output. Often, a correlation-style matrix of dot products provides the re-weighting coefficients. In the figures below, W is the matrix of context attention weights, similar to the formula in Overview section above [16].

The size of the attention matrix is proportional to the square of the number of input tokens. Therefore, when the input is long, calculating the attention matrix requires a lot of GPU memory. Flash attention is an implementation that reduces the memory needs and increases efficiency without sacrificing accuracy. It achieves this by partitioning the attention computation into smaller blocks that fit into the GPU's faster on-chip memory, reducing the need to store large intermediate matrices and thus lowering memory usage while increasing computational efficiency [17].

FlexAttention is an attention kernel developed by Meta that allows users to modify attention scores prior to softmax and dynamically chooses the optimal attention algorithm [18].

Attention is widely used in natural language processing, computer vision, and speech recognition. In NLP, it improves context understanding in tasks like question answering and summarization. In vision, visual attention helps models focus on relevant image regions, enhancing object detection and image captioning [19].

5. Acknowledgement

The authors thank the affiliated institutions for providing the necessary resources to conduct this research.

5. References

[1] Vaswani, A., et al. "Attention is all you need." *Advances in neural information processing systems*, 2017.

[2] Gu, A., & Dao, T. "Mamba: Linear-time sequence modeling with selective state spaces." *arXiv preprint arXiv:2312.00752*, 2023.

[3] Peng, B., et al. "RWKV: Reinventing RNNs for the transformer era." *arXiv preprint arXiv:2305.13048*, 2023.

[4] Touvron, H., et al. "Llama 2: Open foundation and fine-tuned chat models." *arXiv preprint arXiv:2307.09288*, 2023.

[5] Tay, Y., et al. "Efficient transformers: A survey." *ACM Computing Surveys*, 2022.

[6] Dao, T., et al. "Flashattention: Fast and memory-efficient exact attention with io-awareness." *Advances in Neural Information Processing Systems*, 2022.

[7] Choromanski, K., et al. "Rethinking attention with performers." *ICLR*, 2021.

[8] Katharopoulos, A., et al. "Transformers are rnns: Fast autoregressive transformers with linear attention." *ICML*, 2020.

[9] Gu, A., et al. "Efficiently modeling long sequences with structured state spaces." *ICLR*, 2022.

[10] Brown, T., et al. "Language models are few-shot learners." *Advances in neural information processing systems*, 2020.

[11] Beltagy, I., et al. "Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150*, 2020.

[12] Zaheer, M., et al. "Big bird: Transformers for longer sequences." *Advances in Neural Information Processing Systems*, 2020.

[13] Liu, H., et al. "Lost in the middle: How language models use long contexts." *Transactions of the Association for Computational Linguistics*, 2024.

[14] Wei, J., et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in Neural Information Processing Systems*, 2022.

[15] Kitaev, N., et al. "Reformer: The efficient transformer." *ICLR*, 2020.

[16] Wang, S., et al. "Linformer: Self-attention with linear complexity." *arXiv preprint arXiv:2006.04768*, 2020.

[17] Devlin, J., et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." *NAACL-HLT*, 2019.

[18] Rae, J. W., et al. "Scaling language models: Methods, analysis & insights from training gopher." *arXiv preprint arXiv:2112.11446*, 2021.

[19] Yang, Z., et al. "XLNet: Generalized autoregressive pretraining for language understanding." *Advances in neural information processing systems*, 2019.

[20] Smith, S., et al. "Using deepspeed and megatron to train megatron-turing nlg 530b." *arXiv preprint arXiv:2201.11990*, 2022.