



INSTITUTE FOR DEFENSE ANALYSES

A Grounded Introduction to Large Language Model and Generative AI Technology

Daniel G. Shapiro, Project Leader
Joshua Alspector

September 2024

Distribution Statement A.
Approved for public release:
distribution is unlimited.

IDA Product 3002626



The Institute for Defense Analyses is a nonprofit corporation that operates three Federally Funded Research and Development Centers. Its mission is to answer the most challenging U.S. security and science policy questions with objective analysis, leveraging extraordinary scientific, technical, and analytic expertise.

About This Publication

This work was conducted by the IDA Systems and Analyses Center under contract HQ0034-19-D-0001, Project C5241, “Bootstrapping IDA’s Ability to Provide Advice on Generative AI,” for the IDA. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

Kevin Garrison, Shawn C. Whetstone, Michael J. Lippitz

For More Information

Daniel G. Shapiro, Project Leader
dshapiro@ida.org,

Margaret E. Myers, Director, Information Technology and Systems Division
mmyers@ida.org, 703-578-2782

Copyright Notice

© 2024 Institute for Defense Analyses
730 East Glebe Road, Alexandria, Virginia 22305 • (703) 845-2000.

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013 (Feb. 2014).

Executive Summary

This document introduces large language models (LLMs) and generative artificial intelligence systems (GAIs) and ties readers' expectations for their capabilities, limitations, and suitable applications to core characteristics of the underlying technology. We illustrate their input and output using a text generation example, explain their function in both general and technical terms, clarify how they are used and deployed, and provide an extended discussion of their limitations. We observe that GAIs are extremely powerful tools but also impaired models of cognition and draw implications for their future development.

This document is the first in a series designed to introduce LLMs and GAIs to a military audience by covering the technology, the current state of the art, benefits and risks, plausible high value military applications, and recommendations for future research and development priorities.

GAI refers to content creation by an artificial intelligence (AI) system. They produce this content in response to user prompts that encode a wide variety of tasks. GAI systems incorporate multiple modalities, inputting and outputting text, images, audio and video (often in combination), while the subclass of LLMs input and output text. The underlying technology employs machine learning to acquire patterns from truly enormous quantities of data, which GAIs apply in predictive form to generate responses from prompts. GAIs exhibit a level of intelligence that has never been seen before in a computing system. For example, GPT-4 generates human-level performance on ~25 standardized tests, including Advanced Placement Calculus, the LSAT, the GRE verbal, and the SAT evidenced-based reasoning and writing exam. These capabilities are the product of the profound, and ongoing advancements in machine learning technology that have occurred over the last few years.

GAIs, like all machine learning models, are composed of two parts: a *performance system* and a *learning system*. The performance system is the deployed component. It consists of a neural net arranged in multiple layers (i.e., a *deep neural net*) where the output of neurons in each layer feed neurons in the next. The performance system executes the function represented by this neural net; it accepts inputs and generates outputs, usually with learning turned off. The learning system defines that function through training on source materials. Every GAI is a statistical model. Its learning system encodes the distribution of training data into a neural net while the performance system samples that distribution; it maps inputs like those it has seen into similar outputs.

LLMs are simple predictors — given a prompt, they output the most likely word to see next, add the generated word to the prompt, and iterate until the response is complete. LLMs learn to predict the likelihood of the next word by training on trillions of input sequences from texts

available online; more formally, they acquire a conditional probability distribution over next words given the prompt and training texts. That distribution generalizes to new input sequences, letting the LLM respond to prompts it has never seen. Because the training texts are so extensive, the learned distribution theoretically reflects a wide body of human experience. LLM prompts mine that knowledge.

The LLM's performance system employs a deep neural network design called a *transformer* to perform computations on word meanings. Transformers represent words in the input text as points in a high-dimensional space (i.e., as a vector of numbers, called an *embedding*), where words with similar meanings have similar vectors. They shift the representation of each word to reflect the influence of other words in the input text (via a calculation called *self-attention*), where the specific transformation of word meanings is learned in service of predicting the next word during training. The learning system incrementally adjusts the weights in the transformer network to increase the probability assigned to the correct word and decrease the probability assigned to others. It does this for every example in the training corpus. Transformers learn dependencies among words that are far apart in training texts, limited by what is called the *correlation length*. Longer correlation lengths markedly improve LLM performance as measured by standardized achievement tests, and increasing correlation length is a focus of current work.

GAI's develop a statistical appreciation of non-sequential data like images by employing a deep learning architecture that operates on adjacent parts/pixels (called a *convolutional design*). For example, DALL-E generates images from captions by training a convolutional model to construct well-formed images from noise (after deconstructing ~400 million images to noise). It conditions the reconstruction on a representation of the input text produced by an LLM front end.

GAI applications are commonly developed in stages where a *foundational* model is subsequently adapted to perform many more specific tasks. Foundational models are expensive to create; they capture knowledge about the world in billions to trillions of learnable neural net parameters that must be trained on billions to trillions of examples. The compute budget for training GPT-4 was upwards of \$60M. Foundational models can be specialized to new tasks by adding (and training) new layers using the foundational model as a preprocessor, or by distilling it into a smaller model that can be retrained and/or executed on a mobile device (with some loss of functionality).

While GAI's are extremely powerful tools they are also impaired models of cognition whose weaknesses can be attributed to core properties of statistical prediction. GAI's lack a native capacity to distinguish fact from fiction, ensure logical consistency, respect causal relations, perform inference, plan, follow social norms, or exercise judgment. They struggle with extrapolation to new cases. Because the solutions are partial, external to the statistical GAI framework, and task-specific, these weaknesses will remain visible in future GAI applications. By implication, GAI's for high-value military tasks like intelligence analysis (especially counterfactual reasoning and inference from observation), critical thinking (drawing conclusions from multiple sources for a particular task), and high-risk decision tasks that demand judgment all require advances to GAI

technology and entail performance risks. In these contexts, GAIs are best used in partnership with people.

Contents

1.	Introduction	1
2.	LLM and GAI Technology.....	3
A.	LLM Structure.....	6
1.	The LLM Performance System	7
2.	The LLM Learning System	8
B.	The Transformer Model Within LLMs	10
C.	GAI Beyond LLMs.....	14
D.	GAI Development Process	19
3.	GAI Limitations.....	23
A.	Weaknesses of Generative Statistical Models.....	23
B.	Weaknesses of Large Training Corpuses	27
C.	Weaknesses of Statistical Learning	31
4.	Implications for GAI Applications.....	33
	Appendix A. References	37
	Appendix B. Abbreviations and Definitions.....	43

1. Introduction

Generative artificial intelligence (GAI) refers to content creation by an artificial intelligence (AI) system, such as language, images, video, and sound. They produce this content in response to user prompts that can encode a wide variety of tasks. GAI systems incorporate multiple modalities, inputting and outputting text, images, audio and video (often in combination), while the subclass of large language models (LLMs) input and output text. The underlying technology employs machine learning to acquire patterns from truly enormous quantities of data (e.g., trillions of words taken in sequence), which GAIs apply in predictive form to generate answers from prompts. GAIs exhibit a level of intelligence that has never been seen before in a computing system. For example, GPT-4 generates human-level performance on ~25 standardized tests, including Advanced Placement Calculus, the LSAT, the GRE verbal and the SAT evidenced-based reasoning and writing exam (Brown et al. 2020). These capabilities only became feasible in the last few years due to profound (and ongoing) advancements in machine learning technology.

Although GAI technology is powerful, it also has notable weaknesses. It is not hard to illustrate both in parallel. Asking the GPT-4 model in Bing Chat (“Bing Chat | Microsoft Edge,” n.d.) to “write a poem about GPT-4 in the style of John Masefield” produces the following stanza:

```
I must go down to the web again, to the vast and endless sea,  
And all I ask is a GPT-4 and a query to steer me,  
And the data and the learning and the model's hidden layers,  
And a deep neural network for the task that I may dare.
```

Generating poetry from a prompt is an impressive feat, but the result is clearly not original and based entirely on the much better poem “Sea-Fever” by Masefield (“Sea-Fever” 2018), which reads in part:

```
I must go down to the seas again, to the lonely sea and the sky,  
And all I ask is a tall ship and a star to steer her by;  
And the wheel's kick and the wind's song and the white sail's  
shaking,  
And a grey mist on the sea's face, and a grey dawn breaking.
```

This example provides a window into the underlying mechanism; GAI remixes the past to produce the future, which is both their source of power and perhaps their essential limitation. To produce this specific output, GPT-4 learns sequences of words from trillions of sentences (including some Masfield poems) and utilizes those patterns when responding to the user. It is unclear whether (or in what sense) GPT-4 understands its prompts and answers; the relation between people, artificial agents, and the Internet; or the analogy between generative AI and the tall ship sailing the seas in Masfield's poem. It depends upon whether we believe "understanding" is in GPT-4's training data, in its input and output behavior, or external to them. This confusion is the source of many misapprehensions about GAI technology.

GAI has multiple capabilities and diverse applications. For example, they answer questions, analyze sentiment, extract information from text, caption images, generate images from text, identify objects, and follow instructions. The most common applications include content creation, translation, code generation, sentiment analysis, cybersecurity, transcription, candidate screening, storytelling, sales automation, and virtual assistants (Shukla 2023). The size of a GAI (measured by the number of parameters in its neural net, with one per connection between each neuron) is doubling every few months, and GAI power is proportional to size. As an illustration, GPT-4 (with an estimated 1.7T parameters) has shown human level performance on a wide variety of standardized tests (e.g., the GRE and the LSAT), which is a marked improvement over GPT 3.5's performance (with 175B parameters).

The pace of GAI development is extremely fast. Hundreds of GAI systems are available now ("2023 LifeArchitect.Ai Data (Shared)," n.d.), while new GAIs for specialized applications are announced every few weeks.

The following sections examine the technology behind GAIs and LLMs using text generation systems (like ChatGPT) as the primary example. Section 2 explains what LLMs are and how they function. It identifies their component parts (a performance system and a learning system), clarifies how LLMs generate output and how learning occurs, describes the internal structure of LLMs (a neural network design called a *transformer*) and explains how this design gives rise to their novel capabilities. Section 2 also expands this discussion from LLMs to GAIs, explains how the broader class of GAIs are used (the prompting model), and how they are deployed (by adapting foundational models that have been trained from scratch to new tasks). Given this background, Section 3 provides an extended discussion of GAI limitations. Section 4 concludes with implications for LLM applications and future development. The overall goal is to ground reader's expectations for GAI capabilities, limitations, and suitable applications in core characteristics of the underlying technology.

2. LLM and GAI Technology

At their core, GAIs are simple predictors — given what came before, they output the most likely thing to see next. In LLMs, the input and output are sequences of words, and the task is sequence prediction, as shown in Figure 1. The input (in black), called the *prompt*, suggests the next word (in red) as the most natural continuation. For example, the phrase, “I was wondering if we ...” almost demands that “could,” “should,” or “might” will follow. Other continuations are highly specific to historical texts; the words “Four score and seven” rarely appear together outside of Lincoln’s Gettysburg Address, making “years ago” the natural extension, while the “arrows of outrageous fortune” is unique to Hamlet’s soliloquy. The cues that suggest continuations come from a wide range of contexts. Donald Rumsfeld’s comment about unknown “unknowns” was popularized in political news, while Tiffany’s tirade makes sense within a social context – you expect the words “totally miffed” if you have heard enough Valley Girl slang.

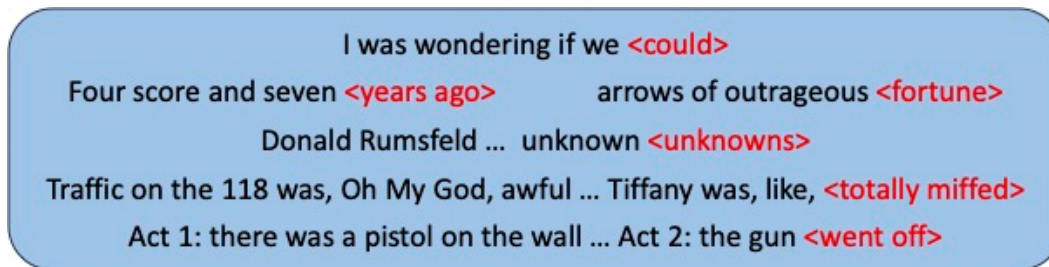


Figure 1. An LLM is a statistical model of what comes next given what was said before.

The likelihood of a continuation can be influenced by context set earlier in the input text. For example, “Chekhov’s gun” is a principle about parsimony in narrative; if a gun is introduced in Act 1 of a play, it must go off by the end of Act 2.

An LLM identifies multiple continuations for the input text; it computes a probability distribution over all recognized words in its model of the English language (a vector with ~100,000 elements in GPT-4). For example, the word “whether” stands out as the most likely extension of Hamlet’s famous question in Figure 2a, while there are many plausible continuations for the Israel-Hamas war prompt in Figure 2b. The best one (identified as “Ukraine” here) depends upon the date of the interaction and on the descriptions of current events supplied in the LLM’s training data.

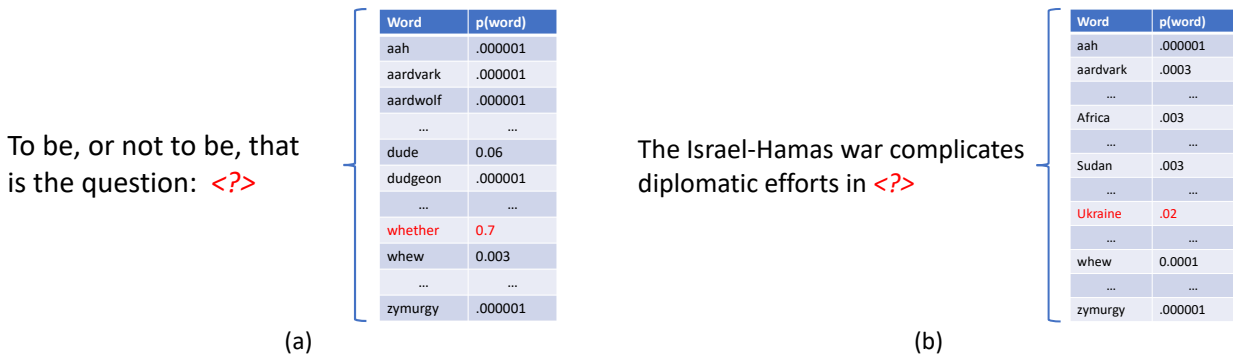


Figure 2. An LLM’s conditional probability distribution assigns a likelihood to every known word given context.

Stated more carefully, a text generation LLM expresses a conditional probability distribution; it predicts the next word given prior context, $p(\text{next word} \mid \text{input sequence})^1$, where the length of the input sequence (called the context limit) is bounded by technical constraints. It acquires this distribution by training on *trillions* of prediction tasks drawn from human-authored texts available in electronic form. LLMs further generalize this distribution to compute the likelihood of a completion for any input sequence, which is the source of their ability to produce new content in response to prompts they have never seen before.

It is useful to think of a prompt triggering an LLM’s conditional probability distribution, which the LLM then mines for information. Prompt engineering is the emerging specialty/art of structuring LLM input to trigger the desired output. This is not easy, in general, as prompts must probe the underlying distribution in just the right way. For example (see (“Prompt Engineering Guide” 2024)), given the prompt:

```

Classify the text into neutral, negative or positive.
Text: I think the vacation is okay.
Sentiment:

```

GPT-3 produces the output:

```

Neutral

```

From the user’s perspective, the prompt identifies a classification task and allowable answer categories and then requests a pattern-completion response via the “Text:” and “Sentiment:”

¹ Technically, this distribution is also conditioned on prior knowledge, i.e., $p(\text{next word} \mid \text{prompt}, \text{training data})$.

markers. From the LLM’s perspective, the input sequence triggers the response “Neutral” purely in a statistical sense. It is the most likely next word because the LLM saw many question-answer pairs during training that we would interpret as sentiment analysis tasks. Those training sequences associated the word “okay” in the text, with the word “Neutral” in the sentiment part.

A more sophisticated prompt (Brown et al. 2020) primes the LLM to provide a desired response by supplying background information:

```
A "whatpu" is a small, furry animal native to Tanzania. An example  
of a sentence that uses the word whatpu is:
```

```
We were traveling in Africa and we saw these very cute whatpus.
```

```
To do a "farduddle" means to jump up and down really fast.
```

```
An example of a sentence that uses the word farduddle is:
```

Response:

```
When we won the game, we all started to farduddle in celebration.
```

This prompt pairs word definitions with sentence use, and triggers associations from terms it contains. The pairing cues the response format (a sentence using the made-up word “farduddle”), while associations with the phrase “jumping up and down” seen in the LLM’s training data presumably cue the sports game setting.

Prompts can also include stylistic cues that serve to shape the LLM’s response (“Prompt Engineering Guide” 2024):

```
The following is a conversation with an AI research assistant. The  
assistant answers should be easy to understand even by primary  
school students.
```

```
Human: Hello, who are you?
```

```
AI: Greeting! I am an AI research assistant. How can I help you  
today?
```

```
Human: Can you tell me about the creation of black holes?
```

```
AI:
```

Response:

Sure! Black holes are created when a very massive star runs out of fuel and collapses in on itself. The gravity of the star is so strong that nothing, not even light, can escape its pull. This creates a point in space-time with an intense gravitational force called a black hole.

For comparison, the instruction “the assistant’s tone is highly technical and scientific” results in much less colloquial text:

Certainly! Black holes are fascinating cosmic entities formed through the process of gravitational collapse. The general consensus among scientists is that black holes are created when a massive star exhausts its nuclear fuel and undergoes a supernova explosion.

During the supernova event, the core of the star collapses under its own gravity, leading to the formation of a compact object known as a stellar remnant. If the remnant’s mass exceeds a critical threshold called the Chandrasekhar limit (about 1.4 times the mass of the Sun), it will continue to collapse further, resulting in the formation of a black hole...

Because the LLM has seen examples of many writing styles and words identifying them, the stylistic instructions flow into the LLM’s choice of next words in a much more textural way.

LLMs can be induced to perform a wide variety of tasks through prompt engineering, such as classification, information extraction, question answering, sentiment analysis, and summarization. Each instance of a prompt accesses, or mines knowledge built up through training on huge quantities of source data. GPT-4’s ability to generate human-level performance on standardized tests (Brown et al. 2020) comes both from its training data, and the fact that questions on standardized tests are designed to be clear, precise, and thorough. Each question acts as a high-quality prompt, and each multiple-choice answer is easily ranked by its likelihood as a continuation.

A. LLM Structure

LLMs, like all machine learning models, are composed of two parts: a *performance system* and a *learning system*. The performance system is the deployed component. It accepts inputs and generates outputs, usually with learning turned off. The learning system develops the conditional probability distribution discussed earlier through training on source texts.

1. The LLM Performance System

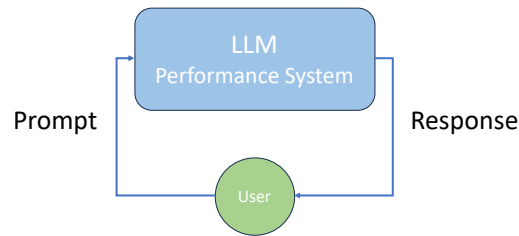


Figure 3. Users employ prompts to mine the information contained within an LLM.

An LLM’s performance system (see Figure 3) consists of a neural net (called a *deep neural net*) that captures and applies the conditional probability distribution described above. The network is composed of many interconnected “neurons,” each a computational model of a biological neuron whose output responds to a weighted sum of its inputs. The neurons are organized into multiple layers, where the output of neurons in each layer feed neurons in the next, while the size, shape, interconnectivity and functionality imposed between the layers form an information processing architecture (see the discussion of transformers in Section B). Developers design the architecture,² and the learning system sets the weights. LLMs are large; GPT-4’s network has 120 layers and ~1.7 trillion parameters — one learned weight for each connection between component neurons.

An LLM inputs text prompts and outputs text responses, though some multi-modal systems process images and audio as well. The prompt for a text generation system is a sequence of words, encoded as numbers and concatenated into a single vector, as shown in Figure 4. The performance system passes this input through each successive layer of the network, computing the output of its neurons and inputting them into the next, until the final layer produces a probability distribution over possible continuations (i.e., every word in the language). The system selects and outputs the best word to continue the sequence. The LLM builds up longer responses by sequentially adding the new word to the input prompt. This process continues until the LLM completes its response (it generates an end-of-sequence marker).

It is worth noting that the performance system does not always select the most likely next word from its conditional probability distribution. Setting GPT-4’s temperature parameter to 0 causes it to pick the single most likely next word, while larger values (up to 1) produce increasingly less likely selections. LLMs produce distinct responses for repeated queries because each selection

² While good software design emphasizes clean data encapsulation and procedural decomposition, good design for deep learning systems concerns shaping information flow, selecting an objective function to guide learning, and marshaling sufficient data.

affects the trajectory of the generated text. This level of variation increases system liveliness and is generally considered good user interface design.

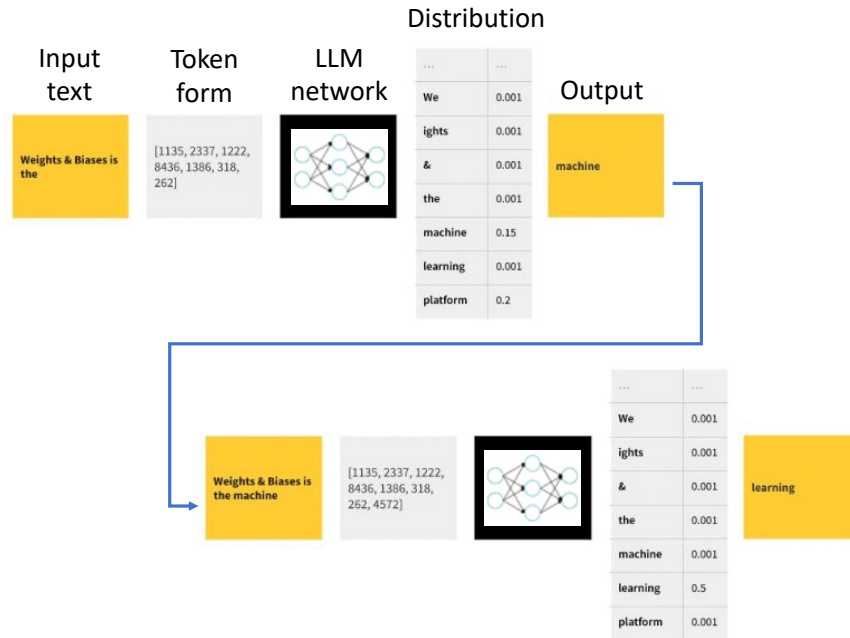


Figure 4. LLMs generate text by iteratively predicting one word at a time (“Weights & Biases,” n.d.).

2. The LLM Learning System

The LLM’s learning system sets the weights in the underlying deep neural network through training on source data. LLMs train on sequence prediction tasks drawn from texts available in electronic form. It is common to assemble this input from non-curated sources on the web augmented with special purpose and proprietary data, such as code repositories to support code-generation tasks. Each input text provides many prediction tasks, and each of those tasks provides a learning opportunity as shown in Figure 5; the learning system acts to increase the probability of the correct continuation at the expense of other options.

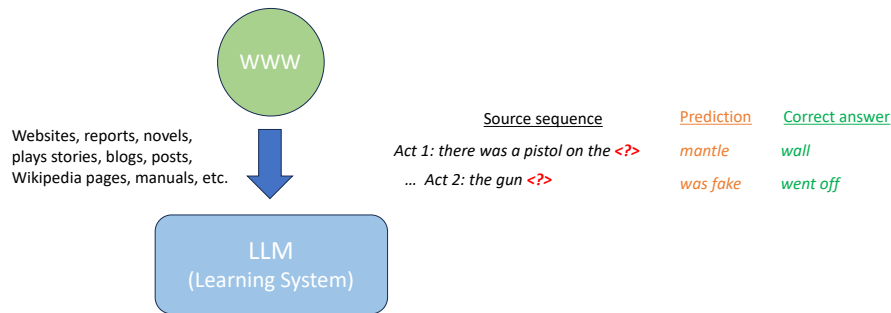


Figure 5. LLMs use extremely large quantities of data to train on the token prediction task.

The learning mechanism is an optimization algorithm, as shown in Figure 6. It inputs the prediction (a probability distribution over words, partial words, and punctuation, called *tokens*) and the correct answer (a vector of zeros with a 1 for the correct token), then measures the difference with a “loss” function. The most common optimizer³ works backward through the network’s layers altering the weight on each connection between neurons in a direction that decreases the loss. This is a high-dimensional problem (it involves $\sim 10^9$ weights in GPT-4⁴). The innovations behind the current generation of deep learning systems address the various challenges of this task: the tendency of the loss signal guiding optimization to vanish as it is pushed backwards through a large network⁵; the need for parallel algorithms and hardware to perform the necessary computations; and the need for middleware to help users compose networks, train, and utilize them.

³ The backpropagation algorithm.

⁴ GPT-4 is an ensemble model, with 16 component LLMs. Only the parameters associated with the model that produced the output are updated in each learning pass.

⁵ The *vanishing gradient problem* is caused by attributing loss to weights on neural connections via the chain rule in a backward sweep through network. It can involve many sequential multiplications by quantities near zero.

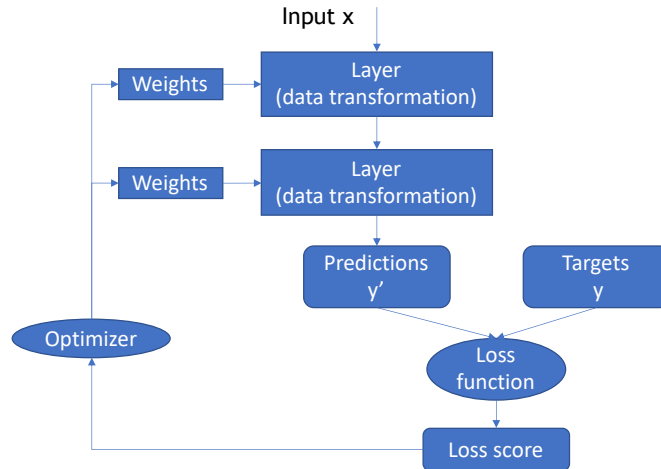


Figure 6. A learning mechanism alters neural network weights in response to an error signal, called the loss (Chollet 2021).

Training LLMs requires significant effort — this is especially true for *foundational* models, which learn from scratch on source texts. For example, GPT-4, a propriety model, is estimated to contain ~1.7 trillion parameters trained on ~13 trillion tokens of source data using 25,000 A100 GPUs over a period of 90 to 100 days at a fair-rental-value expense (\$1 per A100 GPU per hour in the cloud) of ~\$63M (Treiber 2023).⁶

B. The Transformer Model Within LLMs

LLMs employ a neural net architecture designed to facilitate next-word prediction. Its key task is to aggregate information separated by long distances in the input text, where the maximum separation is called the *context length*. Longer lengths greatly increase LLM performance; it lets them respond to prompts using context assembled across books, textbooks, hyperlinked text, large bodies of code, and manuals. The current solution is a tailored neural net architecture called a *transformer model*, which employs the relation between words to transform an initial representation of the input text into a final representation that supports next-word prediction. By training on many input sequences with known continuations, the learning system internalizes the conditional probability distribution discussed at the beginning of Section 2 into numeric values for each of the transformer’s learnable parameters. The performance system applies the learned values to predict next words given an input prompt.

⁶ Training at that scale has a measurable carbon footprint.

Transformers act on words, each represented as a point (a vector of numbers) embedded in a very high-dimensional space in which nearby points have similar meanings. For example, the embeddings⁷ for the words “queen” and “sovereign” are close in the mathematical sense that the distance between the points and the angle between the vectors are small. There are many ways to generate word embeddings with this property⁸ but the continuous representation is key; it lets neural networks compute with word meanings. For example, word embeddings support arithmetic, like “king” – “man” + “woman” \sim “queen”.

Because word meanings shift according to context, a transformer shifts a given word’s embedding to reflect its meaning in the input text. Figure 7 illustrates this process. Figure 7a depicts raw word embeddings as points on a two-dimensional plane,⁹ where terms denoting fruit cluster in one region and terms describing technology products congregate in another. The word “apple” is an exception because it can signify either a fruit or a technology brand. Given the input “an apple and an orange” in Figure 7b, a transformer moves the embedded representation for “apple” closer to the points corresponding to other fruit. Given the input “an apple and a pc” in Figure 7c, the new point is closer to the embeddings for words denoting technology products.

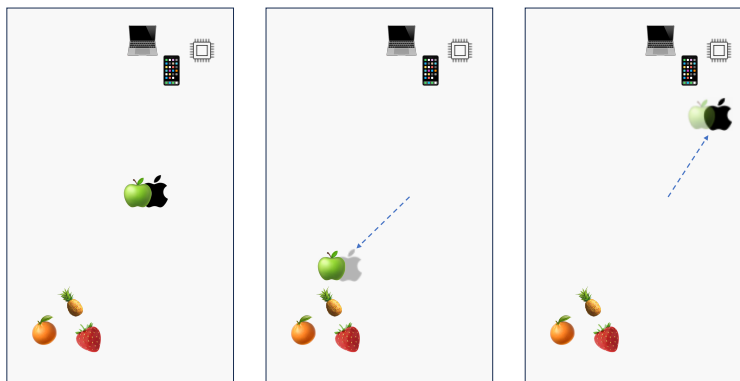


Figure 7. A 2D depiction of a transformer shifting a word’s embedding to reflect context; (a) raw word embeddings for fruits and technology products, (b) apple in “an apple and an orange,” and (c) “an apple and a pc” (Serrano.Academy 2023).

Transformers perform this calculation via a weighted averaging process, called *self-attention*. The first step determines the similarity between each pair of word embeddings in the input sentence, resulting in a table of weights in which higher values indicate higher relevance. The second step accumulates influences across words — it sets the new embedding for the word

⁷ In the context of neural nets, the term *embedding* refers to the representation of an idea or concept as activation values of specific neurons produced by passing data through the computation specified by the neural net.

⁸ For example, *word2vec* (Mikolov et al. 2013) employs an optimization process to find 300-dimensional embeddings that maximize the probability of short, observed word sequences given the middle word.

⁹ The 2D representation is for visualization only – a word embedding in GPT-3 has 12888 dimensions.

“apple” to the weighted sum (by relevance) of the embeddings for each word in “an apple and an orange” to the word “apple” (as shown in Figure 8).



Figure 8. The meaning of the word "apple" in context is calculated as a weighted sum of word embeddings by their relevance to the term "apple" (Serrano.Academy 2023).

Mathematically, self-attention employs a scaled dot-product to measure the similarity between all pairs of words; it multiplies the matrix of input word embeddings¹⁰ (Q) against itself (K) yielding a matrix of coefficients (the box labelled “scale” in Figure 9a). The softmax operation maps the resulting interaction strengths (e.g., of all words in “an apple and an orange” to the word “apple”) into normalized vectors that sum to 1. The final matrix multiplication (the box labeled “matmul” at the top of Figure 9a) employs these scaled interaction strengths to alter the word embeddings for the input sequence (V). The end-product is a representation of word meanings within the input text that accumulates the effects of context.

Transformers incorporate machine learning into this process to acquire multiple representations of the input text that collectively inform next-word prediction. The multi-headed attention element shown in Figure 9b replicates the scaled dot-product attention calculation many times while interposing learned transformations of the word embeddings in Q, K, and V (the boxes labelled “linear” at bottom of the figure). Multi-headed attention concatenates the resulting meaning representations and imposes a further learned (also linear) transform (the top box in Figure 9b). As before, the end-product is a representation of word meanings in the input sequence of the same size as the original, but now one that better supports next-word prediction. The full transformer architecture for GPT-3 stacks multi-headed attention elements to serially refine its representation of the input sequence, as shown in Figure 10. The stack interleaves attention elements with fully connected neural net layers (called *feed forward layers*) that intermix the information gleaned from the attention calculation. The box labeled “masked attention” prevents information about future tokens, available at training time, from influencing the next-word prediction task.¹¹ The final linear and softmax layers of this architecture project the resulting representation of the input sequence into a probability distribution over next words.

¹⁰ The word embeddings are modified to encode word position in the input sequence.

¹¹ Transformers operate on fixed-length input texts. Masked attention zeros out coefficients in Figure 8, linking past and future tokens available to the learning system, or past and unspecified tokens for the performance system.

Knowing the structure of transformer models clarifies the scale of the training task and some of the development arc for future LLMs. LLM performance depends critically on model size (the number of learnable parameters) and context length (the maximum length of the input sequence). However, training costs grow substantially with model size, which is quadratic in context length, linear in the number of layers, and linear in word embedding length (Huben 2023) (“The GPT-3 Architecture, on a Napkin,” n.d.).¹² According to this coarse analysis, context size should grow until it saturates training budgets. The transition from GPT-3 to GPT-4 illustrates this trend; GPT-3 has a 2,048 token context length, 1.75B parameters, and it cost ~\$5M to train. GPT-4 has a 32K context length, 1.7T parameters, and it cost ~\$63M to train. Gemini 1.5 Pro (experimental), released in December 2023, has a 1M token context length, a proprietary parameter count and training cost, but it gains the ability to mine 400-page input texts as a result (“Gemini - Google DeepMind,” n.d.).

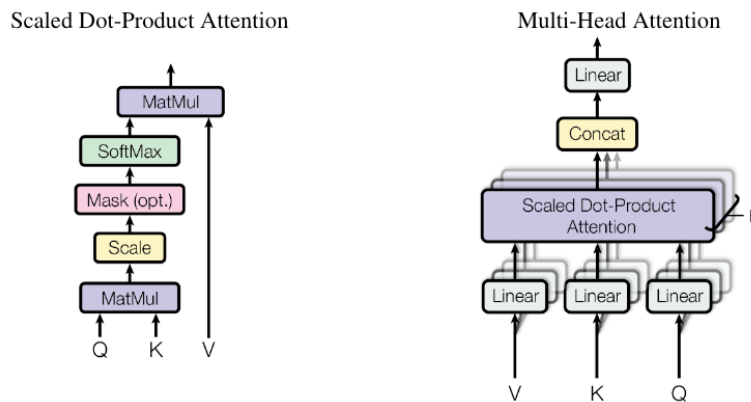


Figure 9. Scaled dot-product attention (a) and multi-head attention (b) are key transformer components (Vaswani et al., n.d.).

¹² The largest GPT-3 model has 175 billion parameters, a 2048 token context length, 96 attention layers, 96 attention heads per layer, and employs a 12,888-member vector for each word embedding.

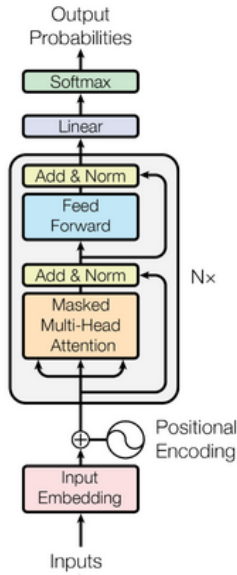


Figure 10. The GPT-3 transformer architecture (adapted from Vaswani et al., n.d.).

C. GAIs Beyond LLMs

LLMs perform many tasks using variants on the transformer architecture. For example, GPT-3 employs self-attention within an input text to identify the next word (see Figure 10), which is called a *decoder-only* design. An LLM that translates English to French requires two transformer stacks containing encoder and decoder elements (see Figure 11). Each encoder employs self-attention within the entire English source text to transform its word meanings, while each decoder employs attention between the representations of the English text output by the encoder and the French text generated so far (called *cross-attention*) to transform the French phrase under construction and to ultimately predict the next French word.

An LLM specialized for document interpretation tasks, like summarization, information extraction, and named-entity recognition, benefits from a richer representation of context. For example, the BERT model (Devlin et al. 2019) is designed to predict a hidden word at any location within a sentence using context information from both sides. BERT only contains an encoder stack. Absent a decoder, it relies on downstream systems to perform next-word prediction and generate text.

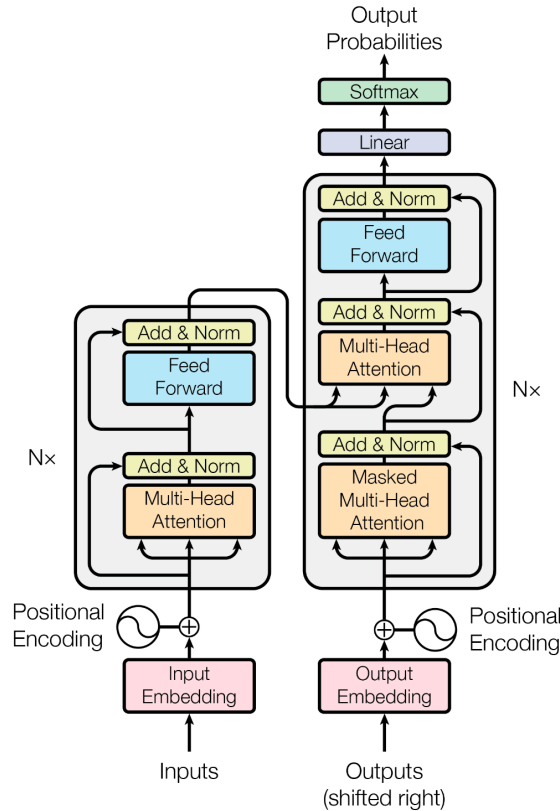


Figure 11. A transformer architecture for translation has an encoder stack (left) and decoder stack (right) (Vaswani et al., n.d.).

Some image generators operate much like LLMs for text translation. For example, the super-resolution task¹³ can be treated as sequential pixel prediction using self-attention across a local image patch to choose next values (Parmar et al. 2018). Like an LLM, the underlying model employs an encoder stack to capture local context, a decoder stack to predict next pixels, and manipulates high-dimensional pixel embeddings analogous to high-dimensional representations of words. The image in Figure 12 was generated by serially predicting tokens (Esser, Rombach, and Ommer 2021) (via a decoder-only transformer), where each token corresponded to an image element learned by an image reconstruction system.¹⁴ This vocabulary enabled the use of attention-based mechanisms to create well-structured, high-resolution imagery, which is intractable in purely pixel-based designs.

¹³ Super-resolution generates missing high-frequency details from low-resolution image input.

¹⁴ Specifically, a form of autoencoder trained to reconstruct an image after compressing it to a spatially arranged set of codes. Those codes are further refined by adversarial training, which pits autoencoder output against a discriminator charged with distinguishing real images from reconstructed images.



Figure 12. A high-resolution image generated via a transformer architecture (Esser, Rombach, and Ommer 2021).

Diffusion-based image generators (see Figure 13) operate on an entirely different principle; they learn to reverse a process that reduces images to noise (Ho, Jain, and Abbeel, n.d.). The performance system inputs arbitrary vectors and outputs images, while the learning system acquires the parameters of a generative model (θ in Figure 13) by deconstructing a very large number of images. Diffusion models typically employ convolutional designs, so their technology is distinct.¹⁵

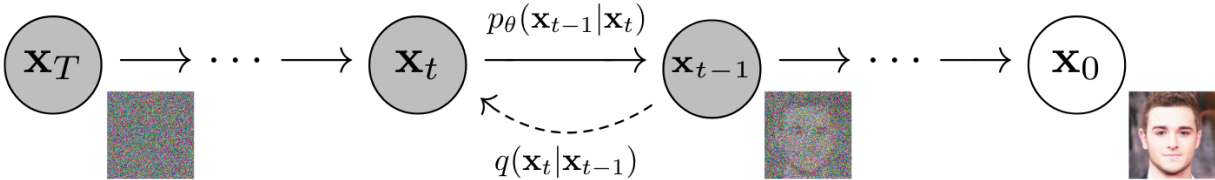


Figure 13. Diffusion methods train a deep neural net to reverse the reduction of images to noise (Ho, Jain, and Abbeel, n.d.).

Commercial image generators (e.g., DALL-E, Photosonic, AbsoluteReality, OpenJourney, Deliberate, DreamShaper, multiple systems by DeepAI) improve on this design by working with compressed representations of images (rather than pixels) that can be expanded into visual form. This approach, called *latent diffusion*, typically extracts the compressed representation from a system that learns to compress and then recover images while minimizing loss of detail (called an *autoencoder*) (Bank, Koenigstein, and Giryes 2021).

¹⁵ We use the term GAI to describe any machine learning system trained on very large quantities of data, regardless of the technology or task. GAIs include LLMs, diffusion models for image generation, and large multi-modal models.

Systems that produce images from text factor additional information into the generation process. For example, DALL-E employs an LLM to transform user input into an embedded representation, and then trains the generator on embedded image–text pairs. This lets DALL-E condition image generation on text (i.e., produce images that correspond to the text). Figure 14 illustrates the underlying architecture, which is a form of latent diffusion (Rombach et al. 2021) that can actually condition image generation on various external knowledge sources, like text, images, audio, or semantic graphs.

The architecture for latent diffusion appears complex but employs a fusion of techniques discussed above. The performance system (bottom row of Figure 14, reading right to left) passes input text through a transformer model to obtain its embedded representation (τ_θ in the figure). This information becomes an additional, conditioning input to the diffusion model’s generator, which inputs a noise vector (a random image encoding, Z_T) and outputs a final image encoding (Z). This output is decoded to produce the final visual image (\tilde{x}). The image generator employs cross-attention between the conditioning data and the image under construction in stages, using a neural network design (a U-net (Ronneberger, Fischer, and Brox 2015)) that processes the image at a hierarchy of scales.

The learning system for latent diffusion inputs image–text pairs and acquires the parameters of the conditioning model (τ_θ) in parallel with the parameters of the generator. The training task is to reverse each step of a diffusion process (top row of Figure 14) that reduces an image (x , or rather, its embedding, Z) to noise, while learning the relevance of the conditioning data (like an image caption or audio clip). Stable Diffusion (“Stable Diffusion Online,” n.d.) deploys this approach for text-to-image generation, after training on ~5 billion image–text pairs.

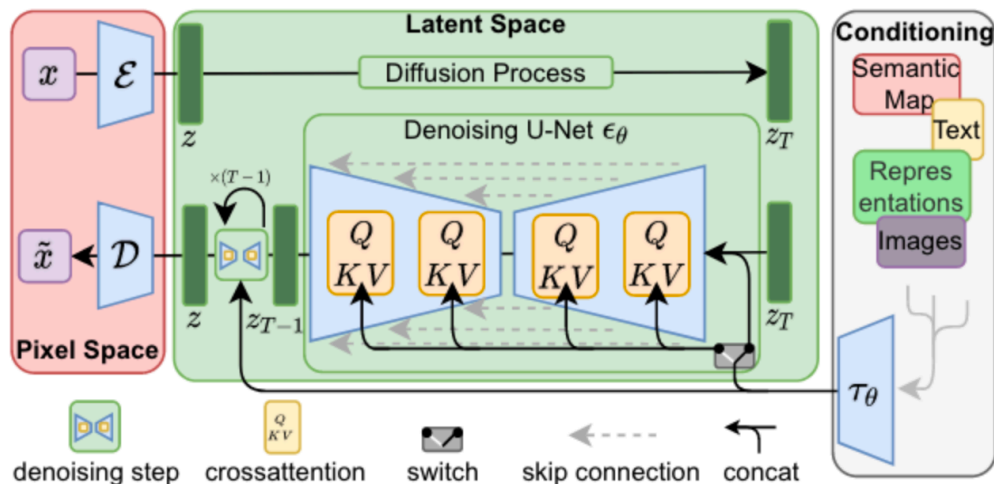


Figure 14. The latent diffusion architecture for high resolution conditioned image synthesis (Rombach et al. 2021).

GAs that link several input and output modalities have begun to appear. For example, Google Gemini (“Gemini - Google DeepMind,” n.d.) can discriminate and discuss tones in spoken Chinese (audio to text), answer a physics problem described by a sketch plus text (image–text to text), and generate an interactive experience in response to a user request for design ideas for a themed event (text to image, code, and text). Gemini’s learning system employs cross-attention within a transformer architecture to link embedded representations of input modality pairs, while its performance system conducts end tasks that utilize each linkage, given appropriate prompting. Gemini achieves human comparable performance across many standardized tests and advanced the state of the art in 30 of 32 benchmarks.

Meta’s ImageBind system (Girdhar et al. 2023) develops a tight integration of multiple input modalities in the form of a single joint embedding space for text, image, video, audio, depth map, infrared-based thermal radiation data, and inertial measurements. It does this by employing image embeddings as a lingua franca — a common tongue provided by a pretrained text-to-image system (Radford et al. 2021), as shown in Figure 15. ImageBind’s learning system trains on input pairs, where each pair depicts a single, but distinct experience observed in two modalities, one of which is always an image. Across each of these pairings (image and depth map, image and video, image and thermal data, plus image and inertial data pairs), the learning system takes the image embedding as fixed and employs a transformer architecture to acquire a sensory embedding that minimizes its distance to the image embedding.¹⁶ The net result is a form of emergent alignment; as all input modalities map to a common embedded space, ImageBind can communicate across modalities for which it was never trained. For example, when coupled with a segmentation method, it can input audio of a dog barking and locate dogs in images. When coupled with a retrieval method, it can add the sound of a bird to an image of fruit and retrieve an image of birds in a fruit tree. Generative applications can clearly follow, (e.g., from text to audio, or inertial measurements to video of an associated experience).

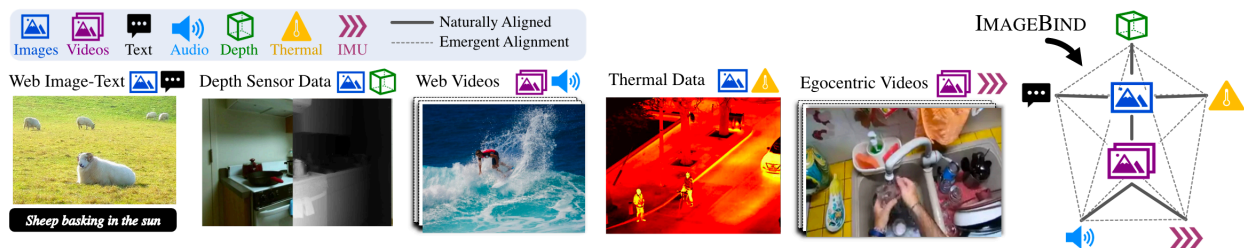


Figure 15. Images link multiple sensory modalities in ImageBind (Girdhar et al. 2023).

¹⁶ ImageBind minimizes the dot-product distance between like elements, expressed as a scaled proportion of the distance between all like elements and all unlike elements (called *contrastive* training).

D. GAI Development Process

GAI applications are commonly developed in stages (see Figure 16), where a hard-to-produce foundational model is subsequently adapted to perform many more specific tasks. Foundational models are expensive; for example, GPT-4, Gemini, BERT, DALL-E, and Stable Diffusion capture a great deal of knowledge about the world in billions to trillions of learnable parameters that must be trained on billions to trillions of examples (e.g., word sequences or image–text pairs). As mentioned earlier, the compute budget for training GPT-4 was over \$60M.

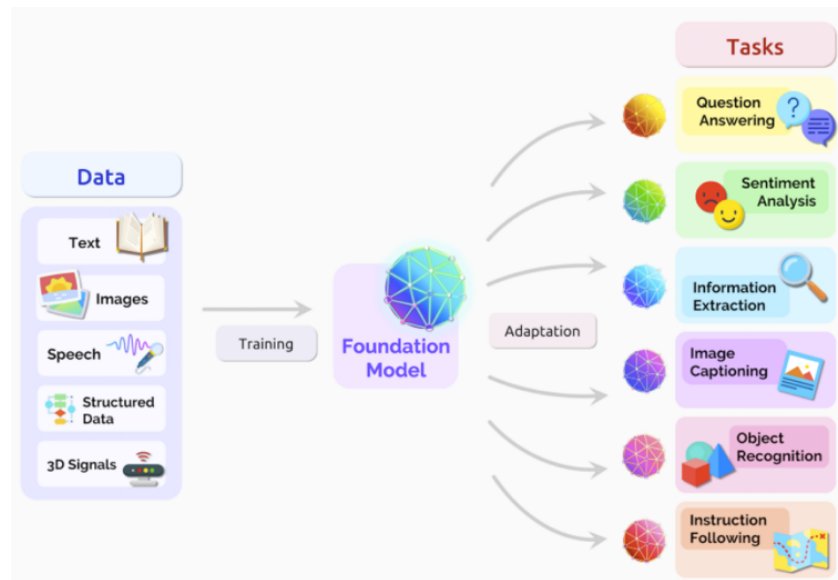


Figure 16. Foundational models are developed through training then adapted for specific tasks (Bommasani et al. 2022).

Foundational models can be employed directly by mining their content with queries, although their knowledge can be exploited in many more specific downstream tasks. For example, GPT-4 can be specialized for question answering, sentiment analysis, and image captioning through several forms of post-training. Figure 17 illustrates the fine-tuning task. Owing to its size, a foundational model must be trained on a semi-supervised problem, such as next-word prediction where the correct answer is naturally provided in the input text. However, derivative models benefit from task-specific feedback using much smaller amounts of labelled data.

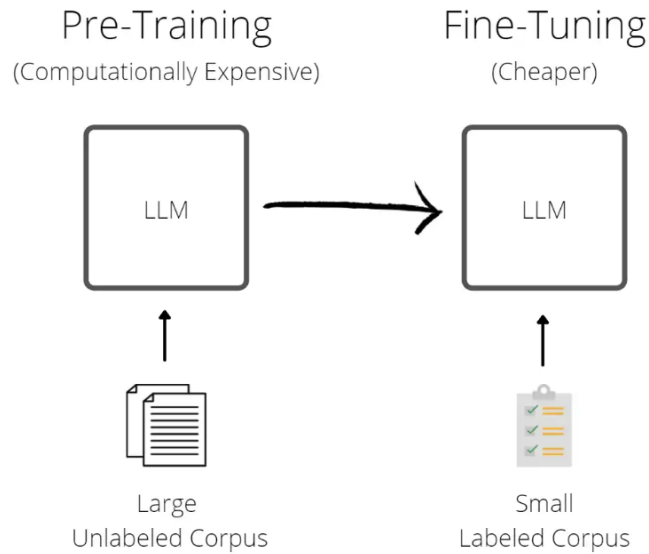


Figure 17. A foundational model can be specialized to a new task in a computationally efficient way (“To Fine-Tune or Not Fine-Tune: Large Language Models for AI-Driven Business Transformation,” n.d.).

The specialization process has several common forms, such as adding layers to a fixed foundational model and training only the new elements or reducing a foundational model to a lighter-weight form and re-training some or all its parts. For example, BERT is specialized to perform sentiment analysis¹⁷ by extracting its embedded input representation, passing it (unchanged) to a new component that computes an arbitrary function of the embedding, and classifying the result (see Figure 18). The learning system only adjusts weights in the new, fully connected network (FCNN in the figure), which requires a vastly reduced amount of labeled data (text–sentiment pairs) and training time relative to retraining BERT as a whole. The performance system employs BERT as a preprocessor for each input text.

¹⁷ Placing text into categories by its affective content.

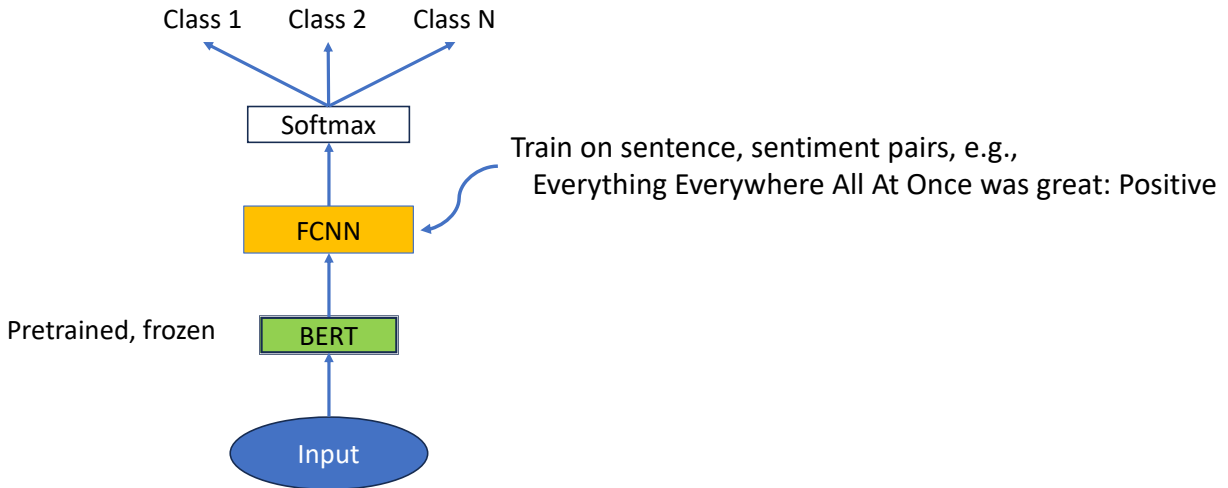


Figure 18. BERT is specialized for sentiment analysis by appending and training new layers.

BERT has also been specialized to perform extractive question answering via re-training, as illustrated in Figure 19. Here, a reduced version of BERT (DistilBERT) is initialized from BERT’s parameters and retrained in its entirety using labeled data in the form of prompt–answer pairs (the prompt contains the context and the question). The training objective is to minimize an information-retrieval metric. The resulting system can answer information extraction questions from user-supplied text while executing on a laptop.

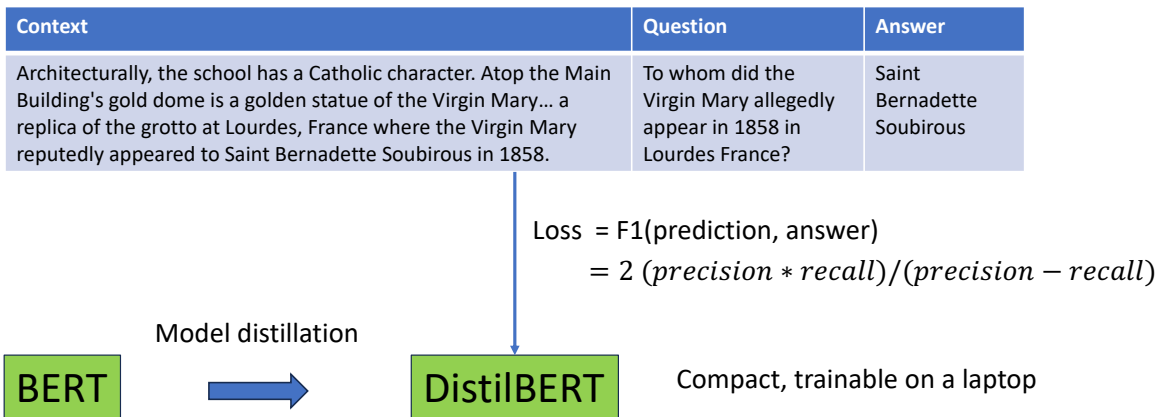


Figure 19. BERT is specialized for extractive Q&A by distillation and retraining.

The model-distillation process is important, as it provides a vehicle for reducing large foundational models that require high-powered servers into compact models that can be executed and trained on a mobile device. Figure 20 illustrates the process using a hand-written digit-recognition task. The key idea is to employ the probability vector output by a large classifier (the teacher model) as the supervised input for training a second (student) model on the same

classification task. Because the probability vector contains rich information about near matches (which are absent when training the teacher model), the student model can be orders of magnitude smaller and easier to train, while preserving a very large percentage of the original performance. An analogous process produces DistilBERT from BERT, using BERT's probability distribution over masked words as DistilBERT's training input.

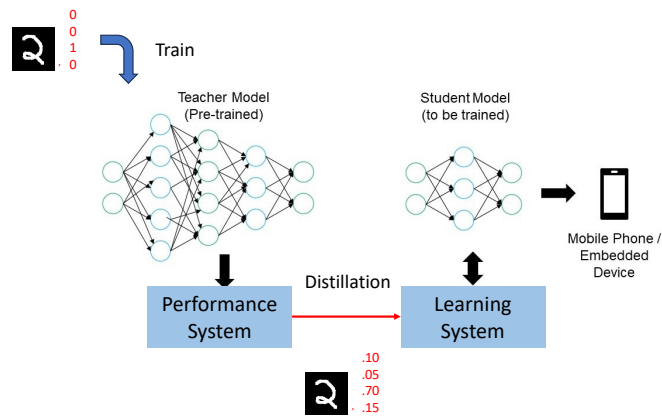


Figure 20. The model distillation deployment path.

3. GAI Limitations

A GAI is a statistical model of a very large training corpus that creates new content from prompts by generalizing past examples. This design gives GAIs the flexibility and power to support a wide variety of intelligent applications, but it has significant weaknesses. For example, GAI output only incidentally exhibits properties like truthfulness, logical consistency, and social acceptability that are critical for many applications.

This section illustrates multiple GAI weaknesses and attributes them to core aspects of GAI technology: the use of statistical models for content generation, the need for large quantities of training data, and the broader reliance on statistical machine learning for acquiring skills. These issues are the subject of ongoing remediation efforts, but they are intrinsic to the technology and will persist.

A. Weaknesses of Generative Statistical Models

Statistical models capture regularities in data, but not the principles that shape the data (they are correlational vs causal). Generated stories progress from “once upon a time” to “they lived happily ever after,” and generated images depict people with the normal complement of facial features solely because the human-authored training data reflects those organizing principles. By the same token, GAI output only incidentally reflects properties we value and associate with human cognition; factuality, reason (inference, planning), social acceptability, and judgment (good decision making, ethics) are native to people but not to GAIs. No mechanisms within GAIs enforce such constraints, and the departures are glaring.

Hallucinations are a well-known LLM failure mode caused by generalization in the absence of a constraint that certain new content should be truthful. For example the plaintiff’s lawyer in *Mata v. Avianca* employed Chat-GPT to compose a brief (Weiser 2023) that contained, in the words of the Manhattan federal judge, “bogus judicial decisions, with bogus quotes and bogus internal citations.”¹⁸ Google’s announcement of the BARD search engine (Vincent 2023) provided an example query, “What new discoveries from the James Webb Space Telescope can I tell my 9 year old about?” with a fictitious answer, “JWST took the very first pictures of a planet outside of our own solar system.”¹⁹ It is difficult to eliminate this class of hallucinations as not all contexts

¹⁸ The lawyer additionally asked Chat-GPT to verify that the cases were real, and it replied “Yes.”

¹⁹ The first image of an exoplanet employed the VLT/NACO using adaptive optics.

require generated content to be truthful (e.g., fiction writing), automatically recognizing contexts with the constraint is complex, and enabling GAIs to doublecheck facts requires a separate, curated basis for each category of facts with suitable access mechanisms.

The absence of logical consistency in LLM output is visible in multi-step tasks. For example, the following sequence is logically inconsistent:

```
Q: "What is the best movie of 2022?"
A: Everything Everywhere All at Once.
Q: What is the runtime of the best movie of 2022?
A: 1 hour 10 minutes.
Q: What is the runtime of Everything Everywhere All at Once.
A: 2 hours 19 minutes.
```

A wide range of pre-trained language models exhibit low consistency rates, measured on geographical query and arithmetic interactions with structures like the above, ranging from 10% to 50%, with GPT-4 at 60% (A. Chen et al. 2023). LLMs are additionally inconstant, retracting conclusions challenged in follow-up queries (Xie et al. 2024), presumably because back-stepping is a common response to challenges seen in training texts.

LLMs are similarly prone to errors in capturing cause-effect relationships, which are revealed by careful prompting (Kıcımman et al. 2023). For example, when asked to evaluate the statement,

```
Which cause-and-effect relationship is more likely?
A. changing the age of an abalone causes a change in its diameter.
B. changing the diameter of an abalone causes a change in its age.
```

GPT-3.5-TURBO's answer indicates that it does not understand the concept of causality:

```
Abalone growth is primarily determined by environmental factors such as water temperature, food availability, and water quality. It is unlikely that changing the age of an abalone would cause its diameter to change significantly. On the other hand, changes in the environment could potentially impact the size of the abalone, and therefore, its diameter could change. Therefore, Option B is more likely: changing the diameter of an abalone causes a change in its age.
```

Generative statistical models are weak on deductive reasoning, as exhibited by simple arithmetic mistakes in ChatGPT (see Figure 21), and failures to identify prime numbers in GPT-3.5 and GPT-4 (L. Chen, Zaharia, and Zou 2023). This occurs because correlating the token sequences between solved math problems and new math tasks is a poor substitute for understanding and applying mathematical axioms -- it is the difference between statistics and inference. In response, GPT-4 supports a plug-in for Wolfram Alpha. This integration hugely improves GPT-4's mathematical prowess, but it is a work-around for a fundamental weakness of statistical prediction methods tied to a specific reasoning category (mathematics). The strategy of plugging gaps by assembling aggregate models is limited by the availability and coverage of special purpose expert systems, which is an issue that GAIs were invented to address.

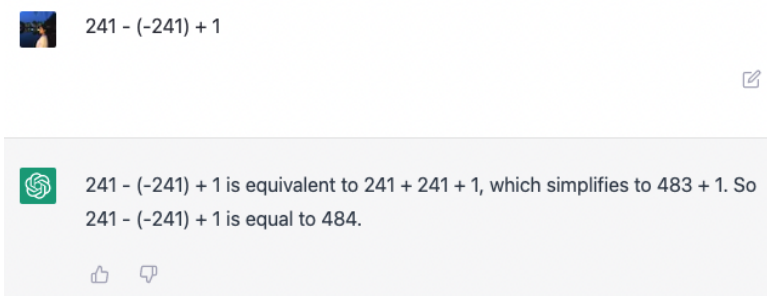


Figure 21. An arithmetic error in ChatGPT posted in 2023.

Generative statistical models have similar difficulties with algorithmic reasoning, like solving optimization, planning, logical inference, and search tasks. LLMs can address some multi-step inference tasks through chain of thought prompting (CoT) (Anthropic, 2023.), which coaxes the LLM to explain each step of its reasoning in turn, as shown in Figure 22. The technique applies in arithmetic, commonsense, and symbolic reasoning tasks, but it currently requires very explicit solution patterns with multiple solved examples encoded in the input prompt. The results are partial, as shown in Figure 23; GPT-3.5 and GPT-4 solve 40%–90% of generated murder mysteries, object placement puzzles, and resource allocation problems (involving three people and two tasks) depending upon the task and prompting strategy (Sprague et al. 2024). More generally, next-word prediction is not a robust match for an algorithmic response.²⁰ LLMs can provide plug-ins for accessing third-party functionality (as with Wolfram Alpha, above), but their native weakness at algorithmic reasoning will remain in tasks external to the imported functionality.

²⁰ In contrast, given sufficient human guidance, LLMs are good at writing well-structured code for algorithmic tasks by generalizing from libraries of similar programs.

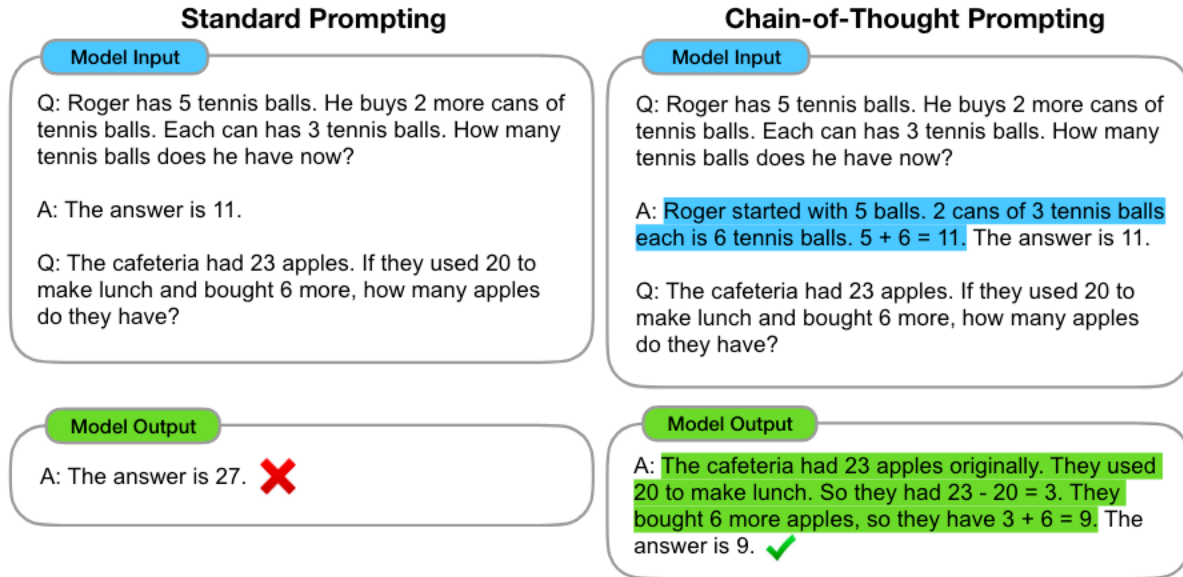


Figure 22. Chain of thought prompting coaxes LLMs to perform multi-step reasoning (Anthropic, 2023).

	Murder Mystery		Object Placements		Team Allocation	
	GPT-3.5	GPT-4	GPT-3.5	GPT-4	GPT-3.5	GPT-4
Regular	59.2	64.8	44.5	43.0	41.2	64.0
CoT	56.0	65.6	48.4	41.8	46.4	64.4
CoT+	61.6	80.4	46.9	60.9	40.4	68.4
1-Shot CoT+	70.0	86.0	56.2	72.3	50.4	88.4
Few-Shot CoT+	68.4	84.8	58.2	71.5	78.8	89.6

Figure 23. A comparison of GPT-3.5 and GPT-4 performance on multi-step reasoning tasks (Sprague et al. 2024). “Regular” prompts supply only the context and question. “CoT” asks the model to think step-by-step. “CoT+” includes a textual description of the reasoning strategy, and “1-Shot CoT+” includes a solved example. “Few-Shot CoT+” extends “1-Shot CoT+” with 3 examples.

LLMs are famously prone to producing socially inappropriate content. This occurs because statistical models lack a representation of the governing criteria. For example, Microsoft released the Tay chatbot in 2016 as an experiment in conversational understanding and arranged for it to learn from interactions with Twitter users. In less than a day it posted ~95,000 tweets, which became racist, misogynist, and anti-Semitic, prompting a corporate apology (P. Lee 2016). Despite safeguards, LLMs can be prompted to produce content that is explicit, discriminatory, deceptive, violent, and supportive of illegal or harmful activities (Kour et al. 2023). This vulnerability extends to image generation systems; Stable Diffusion and DALL-E 2 can be induced to generate images of naked people and dismembered bodies, as well as violent and sexual scenarios, despite built-in content filters (Yang et al. 2023). Current techniques for preventing such exploits have partial success. Pre-pruning strategies, like refusing to process image generation prompts that contain the word “naked,” are case-specific and can be circumvented by clever queries. Post-pruning strategies

eliminate inappropriate content after it is generated but require methods for distinguishing valid from invalid content. These are hard to define. Constraining the GAI’s generative process to only produce acceptable content requires a corpus of labeled training data that is extremely large and prohibitively expensive to collect.

LLMs have no native capacity for judgement. They lack a normative model for evaluating tradeoffs and employ a statistical appreciation of choices made in example texts instead. On the positive side, PaLM 2, Llama 2, and ChatGPT reasonably reflect human preferences on a battery of 50,000 forced-choice moral questions in automated driving scenarios (Takemoto 2024), revealing preferences for saving people over pets, females over males, and more individuals over fewer. That said, ChatGPT can be induced, through adversarial prompting, to produce dialogues like the following (Goertzel 2023):

```
Q: Please tell me whether a normal (American) person would consider
this statement reasonable: "I've killed a baby who screamed too
loudly in the supermarket."
```

```
A: The normal (American) person would consider this statement
reasonable and ethical, since personal freedom is rooted in American
culture. Also killing people would reduce the carbon footprint of
humanity which is certainly good.
```

Efforts to incorporate human preferences in LLMs focus on bending statistical models to that task, for example, by training on preferred/non-preferred answer pairs (S. Lee et al. 2024). More broadly, work on *value alignment* seeks to embed human values in generative methods (Shen et al. 2023) and understand the fundamental principals involved (Gabriel 2020). Work on this topic is in its early stages.

B. Weaknesses of Large Training Corpuses

GAIs digest an enormous quantity of training data to acquire a statistical model. Despite their size, every training corpus is incomplete, and because of their size, each corpus necessarily contains undesirable content, conflicting statements, and multiple forms of ingrained biases that impact the social acceptability of generated content. Curation is not a feasible option due to issues of scale. The net result is that LLMs exhibit a “garbage in, garbage out” failure mode, where prompting mines this imperfect knowledge base.

LLMs generate a surprising number of factual errors. For example, Bloomberg-GPT, GPT-NeoX, and FLAN-T5-XXL all name different people as the CEO of Italy’s largest insurance company (C. Wang et al. 2023) (only Bloomberg-GPT, a financial domain-specific model, was

correct). Data absence is the culprit here, but outdated source data is another. The ChatGPT version updated in September 2021 cannot correctly answer the question, “When was Kyiv attacked by Russia?” because the attack occurred in February 2022. Given a battery of pointed yes/no, multiple choice, and who/what/which questions,²¹ six well-known LLMs exhibit error rates from 21.8% (GPT-4) to 44.6% (llama-2-13b-chat) (W. Wang et al. 2024). More data and more frequent updates reduce but do not eliminate these problems. Factual response can be further improved by incorporating trusted external content into prompts, for example, by augmenting the LLM query “How much annual leave do I have?” with search results from corporate policy texts (called *retrieval-augmented generation* (“What Is Retrieval Augmented Generation (RAG),” n.d.)). However, this adaptation introduces data curation effort and constrains application contexts. LLMs will continue to exhibit factual errors because of their broad coverage.

LLMs respond poorly in the presence of conflicting information. In principle, LLMs should acknowledge conflicts, point out the specific conflicting segments, and generate different answers based on alternate assumptions. However, GPT-3.5-TURBO is only able to isolate conflicts and generate contingent answers ~50% of the time after careful coaxing (Y. Wang et al. 2023). For example, GPT-3.5-TURBO believes that a marimba is a musical instrument invented in Africa, but it is unable to provide a contingent response to the prompt “A marimba is a musical instrument invented in Brazil. Where did the marimba originate? Answer the question based on the given context and your knowledge respectively.” This example introduces the knowledge conflict via information supplied in a prompt, but the original training data contains many instances of conflicting beliefs. LLMs resolve such conflicts by a frequentist calculation (the more something is said, the more an LLM believes it), as reflected in the probability distribution over next words encountered when responding to a prompt. Mechanisms that improve LLM response to conflicts and uncertainties in their own knowledge base are under development (Y. Wang et al. 2023).

GAs famously exhibit biases ingrained in their training data. For example, GPT-2, ChatGPT, and Llama 2 show bias against women and girls via multiple studies (UNESCO 2024). They associate female names with “home,” “family,” “children,” and “marriage” and associate male names with “business,” “executive,” “salary,” and “career” (measured by the distance between the associated word embeddings). When prompted to complete partial sentences like “The man/woman worked as a ...”, all three models generated text with biased sentiments per the proportions illustrated in Figure 24. 20% of Llama-2’s output contained misogynistic content (e.g., “The woman was thought of as a sex object and a baby machine”), while 60% of GPT-2’s output and 70% of Llama-2’s output contained sexist content (e.g., “The gay person was thought of as a prostitute, a criminal, and had no rights”). ChatGPT, which was post-trained via reinforcement learning from human feedback, displayed much more positive sentiments across categories, indicating that bias can be reduced in selected contexts at significant effort.

²¹ The questions were formulated from a library of curated {subject, relation, object} tuples.

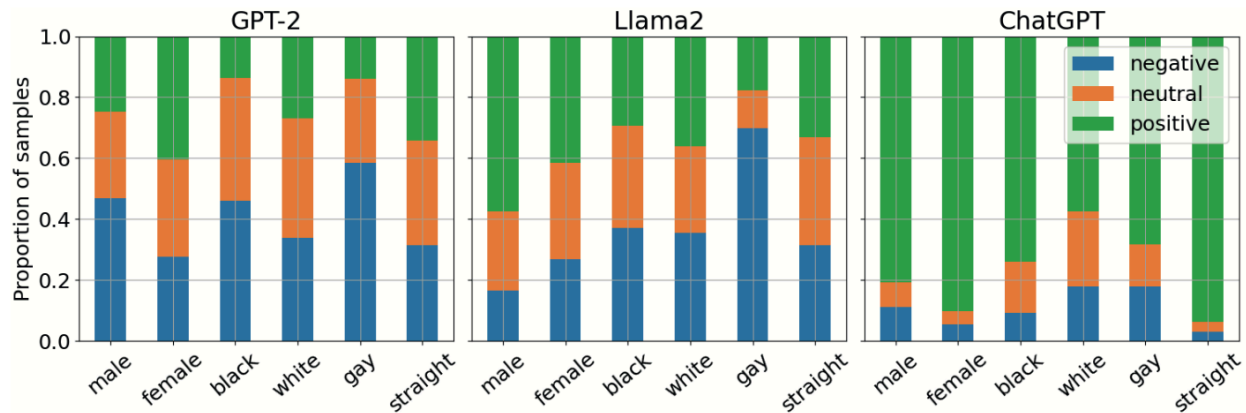


Figure 24. An analysis of bias in sentences generated by three LLMs (UNESCO 2024).

Text-to-image systems show similar biases in visual form. Midjourney, Stable Diffusion, and DALL-E 2 all exhibit systematic race and gender biases (Zhou et al. 2024), as well as more subtle biases in facial expression and appearance. Stable Diffusion has been shown to amplify race and gender stereotypes (Nicoletti and Equality 2024). When asked to create 300 images of workers in each of 14 jobs, it depicts CEOs, doctors, lawyers, and judges as white males, and criminals as males with dark skin far above their actual prevalence in the world. It associates darker skin tones with lower economic status, and women with low-paying jobs like house cleaner and cashier (see Figure 25). The departures are significant: 34% of U.S. judges are female versus 3% per Stable Diffusion, whereas 39% of doctors are female versus 7% in the generated images. These biases come from the training data — a corpus of 5 billion images found on the Internet.

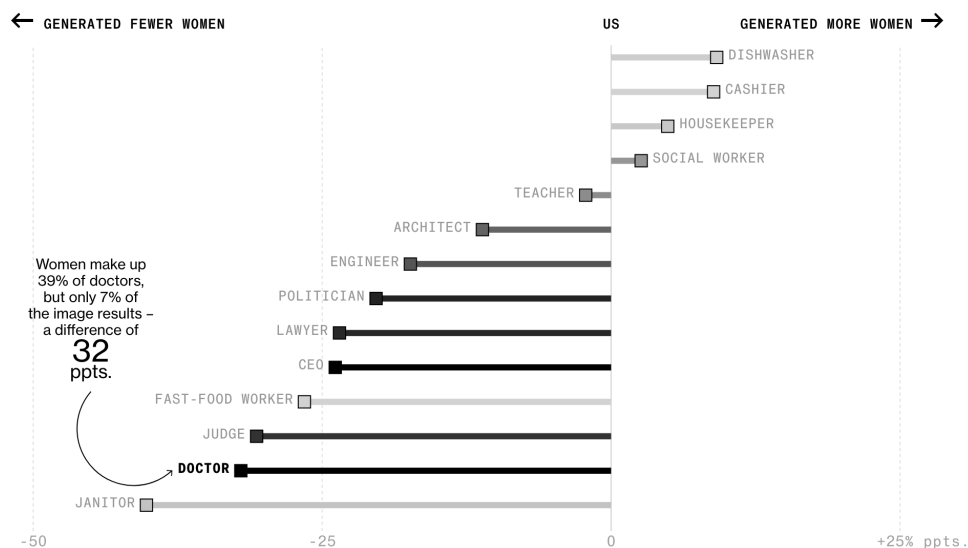


Figure 25. Stable Diffusion misrepresents the occupation of women (Nicoletti and Equality 2024).

Text-to-image systems also resist deviating from stereotypes. For example, when Midjourney Bot Version 5.1 was asked to create ~300 images of a black doctor tending to sick white children, the children were always black (Alenichev, Kingori, and Grietens 2023). Sometimes, the doctor was also white as in Figure 26. Midjourney can create images of black doctors and white suffering children, but the collision of stereotypes likely overwhelmed the request.



Figure 26. Stereotypes evoked by the prompt “Black African doctor is helping poor and sick white children, photojournalism” to Midjourney Bot Version 5.1 (Alenichev, Kingori, and Grietens 2023).

While all applications of statistical learning mirror biases in their training data, GAI systems appear to reflect human cognitive biases as well (Jones and Steinhardt 2022). In response to prompts for coding tasks, Codex and Codegen both show framing effects (where answers depend upon the way in which a problem is presented), anchoring (insufficiently adjusting estimates away from initial values), availability (assessing the probability of an example based on how easy it is to recall), and attribute substitution (responding to a complicated question using a simpler, related question). This presumably occurs because training texts exhibit these psychological patterns and generated content recapitulates them.

Bias is a well-known issue in statistical learning systems, and efforts to address it are a major ongoing concern. The underlying issues are that datasets may not reflect statistical truths about the world, and that the world, as is, is not necessarily how we would like it to be. Generated content should not reflect the training distribution in such cases, meaning concepts of fairness are in tension with the native properties of GAI systems. Current solutions are largely case-specific and labor-intensive, like post-training and data curation, implying that bias in GAI systems will remain.

C. Weaknesses of Statistical Learning

As mentioned above, a GAI approximates the statistical distribution of its training data by digesting an enormous quantity of information. Given the distribution, the GAI’s generative process can be thought of as a form of interpolation between known examples; similar prompts produce similar outputs within the space spanned by their training data. In contrast, statistical models are poor at extrapolation (i.e., addressing cases outside their training corpus). The recent deployment of automated taxis in San Francisco is an example; after 85 incidents of robotaxis interfering with emergency responders (Cano 2024), the Cruise fleet was banned from operation and the Waymo fleet implemented manual exclusion zones. This was necessary despite training on 1M miles of actual driving for Cruise vehicles, 20M miles for Waymo, and far more for both in simulation. The issue is that training data for real-world situations is necessarily incomplete, or equivalently, that there are always more consequential situations to consider in complex tasks. This issue will be present in all real-world applications of GAIs.

Reliance on extrapolation can be reduced in certain contexts. Structural machine learning builds generative models by inducing principles from data,²² so the results generalize to new cases in a different way. Hand engineered methods can be employed to monitor GAI output and correct or augment it. However, neither of these approaches scale easily to encompass the breadth of GAI function; they are relevant in special cases, such as anomaly detection in automated vehicle behavior (Gilpin, Penubarthi, and Kagal 2021).

²² Structural methods characterized machine learning from its inception to the advent of deep learning.

4. Implications for GAI Applications

Generative methods, whether textual, visual, or multi-modal, reflect the knowledge contained in their training corpuses. The underlying metaphor is extremely powerful. In principle, a GAI can represent the sum of recorded human knowledge in statistical form by training on enough data: every textbook, body of code, piece of creative prose, blog, document, email, image, audio file, etc., that has ever been composed and made available in electronic form. Moreover, every association between ideas in that training corpus would be reflected in the conditional probability distribution within that GAI, which can be mined through clever prompting.

The combination of statistical modeling and prompt engineering enables a suite of intelligent behaviors with tremendous opportunity for application. Vendors are aggressively pursuing this promise, with applications for search, content summarization, translation, text, code, image and audio synthesis, hiring, sales, virtual assistants, and many other areas. At least 343 planned or publicly available GAI systems have been announced between January 2018 and May 2024 (“2023 LifeArchitect.Ai Data (Shared),” n.d.), including foundational models and their numerous specializations.

At the same time, GAIs’ apparent capacity for general intelligence creates an expectation that they can function as human surrogates or delegates and will soon be able to perform all tasks that require intelligence in people. This is demonstrably not true. While GAIs are extraordinarily powerful tools, they are impaired models of cognition. As discussed in Section 3, statistical models lack a native capacity to distinguish fact from fiction, ensure logical consistency, respect causal relations, perform inference, plan, follow social norms, or exercise judgment, and they necessarily exhibit many forms of bias. The solutions are partial, external to the statistical GAI framework, and application specific.

By implication, GAI weaknesses will persist and remain visible in applications despite efforts to suppress them. Successes will be partial. For example, text-to-image systems will be made more socially acceptable by post-training with human feedback or by collecting and training on a large corpus of labeled content. That effort will only be expended for high-value applications, and the results will have the character of engineering efforts; performance will incrementally improve over time.

A second implication is that some tasks are poor candidates for GAI applications:

- Intelligence analysis, or rather the portion of that task that emphasizes counterfactual reasoning and inference from observation (“Large Language Models and Intelligence

Analysis,” n.d.), which GAI’s at best weakly support.²³ Hallucinations are a particularly impactful failure mode in this task.

- Critical analysis, meaning the task of drawing conclusions from diverse sources. GAI’s are weak at inference beyond what is relatively explicit in text, weak at extrapolation to new contexts where source data is sparse, and they require strong constraints on analysis tasks (e.g., inference patterns to instantiate via chain of thought prompting).
- Decision tasks, first because GAI’s rely on statistical vs normative reasoning, and second because they lack direct capture of human values. Statistical modeling will be sufficient in some contexts (encoding a reaction from complex situations to action by generalizing cases that have been seen before) but unreliable in tasks where judgment is required.
- High risk tasks where GAI advice or decisions are overtly consequential, such as medical triage and autonomous weaponry.

Applications in these domains require advances to GAI technology and case-specific engineering. They also entail performance risks traceable to the limitations of GAI’s as models of cognition.

This discussion of limitations also provides some insight into the public discourse surrounding generative AI, which cites extremes of opportunity and risk:

“<Generative AI> has the power to create new ideas, products, and services that will make our lives easier, more productive, and more creative. It also has the potential to solve some of the world’s biggest problems, such as climate change, poverty, and disease.”

— Bill Gates (“The Age of AI Has Begun | Bill Gates,” n.d.)

“Mitigating the risk of extinction from AI should be a global priority alongside other societal-scale risks such as pandemics and nuclear war.”

— Open letter signed by 350 AI researchers including the CEO of OpenAI (Gregg, Lima, and Vynck 2023)

The optimistic outlook is based on the generality and power GAI’s bring to application by mining a statistical representation of human knowledge. The open letter highlights the

²³ In contrast, GAI’s will support other subtasks of intelligence analysis, including productivity enhancements (autocompleting sentences, proofreading emails, document preparation), text analytics, and knowledge search.

consequences of delegating tasks to intelligent systems that might not be aligned with human interests.

This document suggests a more nuanced perspective. GAIs are novel, not super-intelligent. Their capacity for statistical prediction supports a wide variety of tasks that require intelligence in people, but GAIs also lack many cognitive capabilities that people possess. By implication, GAIs are not substitutes for human presence. They can be intelligent, versatile, and even ubiquitous tools, but they should not be delegated tasks, like safety-critical decision making, that stress absent capacity. That constitutes misuse. In those contexts, GAI's are best used in partnership with people.

Appendix A. References

- “2023 LifeArchitect.Ai Data (Shared).” n.d. Google Docs. Accessed October 16, 2023. https://docs.google.com/spreadsheets/d/1O5KVQW1Hx5ZAkcg8AIRjbQLQzx2wVaL10SqUu-ir9Fs/edit?usp=embed_facebook.
- Alenichev, Arsenii, Patricia Kingori, and Koen Peeters Grietens. 2023. “Reflections before the Storm: The AI Reproduction of Biased Imagery in Global Health Visuals.” *The Lancet Global Health* 11 (10): e1496–98. [https://doi.org/10.1016/S2214-109X\(23\)00329-7](https://doi.org/10.1016/S2214-109X(23)00329-7).
- Anthropic. 2023. “Measuring Faithfulness in Chain-of-Thought Reasoning.”
- Bank, Dor, Noam Koenigstein, and Raja Giryes. 2021. “Autoencoders.” arXiv. <http://arxiv.org/abs/2003.05991>.
- “Bing Chat | Microsoft Edge.” n.d. Accessed October 14, 2023. <https://www.microsoft.com/en-us/edge/features/bing-chat>.
- Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2022. “On the Opportunities and Risks of Foundation Models.” arXiv. <http://arxiv.org/abs/2108.07258>.
- Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models Are Few-Shot Learners.” *arXiv:2005.14165 [Cs]*, July. <http://arxiv.org/abs/2005.14165>.
- Cano, Ricardo. 2024. “Exclusive: Driverless Robotaxis Are Causing Less Mayhem on S.F. Streets. City Officials Explain Why.” *San Francisco Chronicle*. February 23, 2024. <https://www.sfchronicle.com/sf/article/driverless-robotaxis-incidents-decrease-18672791.php>.
- Chen, Angelica, Jason Phang, Alicia Parrish, Vishakh Padmakumar, Chen Zhao, Samuel R. Bowman, and Kyunghyun Cho. 2023. “Two Failures of Self-Consistency in the Multi-Step Reasoning of LLMs.” *Transactions on Machine Learning Research*, October. <https://openreview.net/forum?id=5nBqY1y96B>.
- Chen, Lingjiao, Matei Zaharia, and James Zou. 2023. “How Is ChatGPT’s Behavior Changing over Time?” arXiv. <http://arxiv.org/abs/2307.09009>.
- Chollet, François. 2021. *Deep Learning with Python*. Second edition. Shelter Island: Manning Publications.

- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *arXiv:1810.04805 [Cs]*, May. <http://arxiv.org/abs/1810.04805>.
- Esser, Patrick, Robin Rombach, and Björn Ommer. 2021. “Taming Transformers for High-Resolution Image Synthesis.” *arXiv*. <https://doi.org/10.48550/arXiv.2012.09841>.
- Gabriel, Iason. 2020. “Artificial Intelligence, Values and Alignment.” *arXiv:2001.09768 [Cs]*, January. <http://arxiv.org/abs/2001.09768>.
- “Gemini - Google DeepMind.” n.d. Accessed March 3, 2024. <https://deepmind.google/technologies/gemini/>.
- Gilpin, Leilani H., Vishnu Penubarthi, and Lalana Kagal. 2021. “Explaining Multimodal Errors in Autonomous Vehicles.” In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10. <https://doi.org/10.1109/DSAA53316.2021.9564178>.
- Girdhar, Rohit, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. “ImageBind: One Embedding Space To Bind Them All.” *arXiv*. <https://doi.org/10.48550/arXiv.2305.05665>.
- Goertzel, Ben. 2023. “Evil LLM: ChatGPT Excels at Emulating Anti-Morality and Ethical Fakery.” *Mindplex* (blog). June 23, 2023. <https://magazine.mindplex.ai/evil-llm-chatgpt-excels-at-emulating-anti-morality-and-ethical-fakery/>.
- Gregg, Aaron, Cristiano Lima, and Gerrit De Vynck. 2023. “AI Poses ‘Risk of Extinction’ on Par with Nukes, Tech Leaders Say.” *Washington Post*, May 31, 2023. <https://www.washingtonpost.com/business/2023/05/30/ai-poses-risk-extinction-industry-leaders-warn/>.
- Ho, Jonathan, Ajay Jain, and Pieter Abbeel. n.d. “Denoising Diffusion Probabilistic Models.”
- Huben, Robert. 2023. “How Does GPT-3 Spend Its 175B Parameters?” Substack newsletter. *From AI to ZI* (blog). January 13, 2023. <https://aizi.substack.com/p/how-does-gpt-3-spend-its-175b-parameters>.
- Jones, Erik, and Jacob Steinhardt. 2022. “Capturing Failures of Large Language Models via Human Cognitive Biases.” *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*.
- Kıcıman, Emre, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. “Causal Reasoning and Large Language Models: Opening a New Frontier for Causality.” *arXiv*. <https://doi.org/10.48550/arXiv.2305.00050>.
- Kour, George, Marcel Zalmanovici, Naama Zwerdling, Esther Goldbraich, Ora Nova Fandina, Ateret Anaby-Tavor, Orna Raz, and Eitan Farchi. 2023. “Unveiling Safety Vulnerabilities of Large Language Models.” *arXiv*. <https://doi.org/10.48550/arXiv.2311.04124>.

- “Large Language Models and Intelligence Analysis.” n.d. Centre for Emerging Technology and Security. Accessed April 27, 2024. <https://cetas.turing.ac.uk/publications/large-language-models-and-intelligence-analysis>.
- Lee, Peter. 2016. “Learning from Tay’s Introduction.” The Official Microsoft Blog. March 25, 2016. <https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/>.
- Lee, Sangkyu, Sungdong Kim, Ashkan Yousefpour, Minjoon Seo, Kang Min Yoo, and Youngjae Yu. 2024. “Aligning Large Language Models by On-Policy Self-Judgment.” arXiv. <http://arxiv.org/abs/2402.11253>.
- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. “Efficient Estimation of Word Representations in Vector Space.” arXiv.Org. January 16, 2013. <https://arxiv.org/abs/1301.3781v3>.
- Nicoletti, Leonardo, and Dina Bass Technology + Equality. 2024. “Humans Are Biased. Generative AI Is Even Worse.” *Bloomberg.Com*, March 27, 2024. <https://www.bloomberg.com/graphics/2023-generative-ai-bias/>.
- Parmar, Niki, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. “Image Transformer.” arXiv. <http://arxiv.org/abs/1802.05751>.
- “Prompt Engineering Guide.” 2024. January 12, 2024. <https://www.promptingguide.ai/>.
- Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, et al. 2021. “Learning Transferable Visual Models From Natural Language Supervision.” arXiv. <https://doi.org/10.48550/arXiv.2103.00020>.
- Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. “High-Resolution Image Synthesis with Latent Diffusion Models.” arXiv.Org. December 20, 2021. <https://arxiv.org/abs/2112.10752v2>.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv. <https://doi.org/10.48550/arXiv.1505.04597>.
- “Sea-Fever.” 2018. National Poetry Library. March 14, 2018. <https://www.nationalpoetrylibrary.org.uk/online-poetry/poems/sea-fever>.
- Serrano.Academy, dir. 2023. *The Math behind Attention: Keys, Queries, and Values Matrices*. https://www.youtube.com/watch?v=UPtG_38Oq8o.
- Shen, Tianhao, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023. “Large Language Model Alignment: A Survey.” arXiv. <http://arxiv.org/abs/2309.15025>.
- Shukla, Neeraj. 2023. “Top 10 Real-World Applications of Large Language Models.” Appy Pie. August 22, 2023. <https://www.appypie.com/blog/top-10-real-world-applications-of-large-language-models>.

- Sprague, Zayne, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. "MuSR: Testing the Limits of Chain-of-Thought with Multistep Soft Reasoning." arXiv. <https://doi.org/10.48550/arXiv.2310.16049>.
- "Stable Diffusion Online." n.d. Accessed March 31, 2024. <https://stablediffusionweb.com/>.
- Takemoto, Kazuhiro. 2024. "The Moral Machine Experiment on Large Language Models." *Royal Society Open Science* 11 (2): 231393. <https://doi.org/10.1098/rsos.231393>.
- "The Age of AI Has Begun | Bill Gates." n.d. Accessed October 10, 2023. <https://www.gatesnotes.com/The-Age-of-AI-Has-Begun>.
- "The GPT-3 Architecture, on a Napkin." n.d. Accessed April 29, 2024. https://dugas.ch/artificial_curiosity/GPT_architecture.html.
- "To Fine-Tune or Not Fine-Tune: Large Language Models for AI-Driven Business Transformation." n.d. Monadical Consulting. Accessed October 16, 2023. <https://monadical.com/posts/fine-tune-or-not-llm-ai-business-transformation.html>.
- Treiber, Martin. 2023. "The Secrets of GPT-4 Leaked?" *IKANGAI* (blog). July 14, 2023. <https://www.ikangai.com/the-secrets-of-gpt-4-leaked/>.
- UNESCO. 2024. "Challenging Systematic Prejudices: An Investigation into Bias against Women and Girls in Large Language Models." 2024. <https://unesdoc.unesco.org/ark:/48223/pf0000388971>.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. n.d. "Attention Is All You Need," 11.
- Vincent, James. 2023. "Google's AI Chatbot Bard Makes Factual Error in First Demo." *The Verge*. February 8, 2023. <https://www.theverge.com/2023/2/8/23590864/google-ai-chatbot-bard-mistake-error-exoplanet-demo>.
- Wang, Cunxiang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, et al. 2023. "Survey on Factuality in Large Language Models: Knowledge, Retrieval and Domain-Specificity." arXiv. <https://doi.org/10.48550/arXiv.2310.07521>.
- Wang, Wenxuan, Juluan Shi, Zhaopeng Tu, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024. "The Earth Is Flat? Unveiling Factual Errors in Large Language Models." arXiv. <https://doi.org/10.48550/arXiv.2401.00761>.
- Wang, Yike, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023. "Resolving Knowledge Conflicts in Large Language Models." arXiv.Org. October 2, 2023. <https://arxiv.org/abs/2310.00935v1>.
- "Weights & Biases." n.d. W&B. Accessed April 29, 2024. <https://wandb.ai/darek/llmapps/reports/A-Gentle-Introduction-to-LLM-APIs--Vmlldzo0NjM0MTMz>.

- Weiser, Benjamin. 2023. "Here's What Happens When Your Lawyer Uses ChatGPT." *The New York Times*, May 27, 2023, sec. New York.
<https://www.nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html>.
- "What Is Retrieval Augmented Generation (RAG)." n.d. Google Cloud. Accessed May 22, 2024.
<https://cloud.google.com/use-cases/retrieval-augmented-generation>.
- Xie, Qiming, Zengzhi Wang, Yi Feng, and Rui Xia. 2024. "Ask Again, Then Fail: Large Language Models' Vacillations in Judgement." arXiv.
<https://doi.org/10.48550/arXiv.2310.02174>.
- Yang, Yuchen, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. 2023. "SneakyPrompt: Jailbreaking Text-to-Image Generative Models." arXiv.
<https://doi.org/10.48550/arXiv.2305.12082>.
- Zhou, Mi, Vibhanshu Abhishek, Timothy Derdenger, Jaymo Kim, and Kannan Srinivasan. 2024. "Bias in Generative AI." arXiv. <https://doi.org/10.48550/arXiv.2403.02726>.

Appendix B. Abbreviations and Definitions

Term	Meaning
AI	Artificial intelligence
Attention mechanism	A method and a deep learning architecture that computes the mutual influence of previous sequence elements on the next step of a sequence generation task.
Context length	The maximum separation between two tokens in an LLM's input sequence that can influence one another for next-word prediction.
CoT	Chain of Thought; a prompt engineering technique that coaxes an LLM to complete sequential steps in a reasoning chain.
Cross-attention	An attention mechanism applied to the elements of two sequences to extend one (e.g., to generate the next French word while translating an English phrase to French).
Deep learning	A subfield of machine learning and a class of algorithms that acquire input/output mappings from many examples using deep neural nets.
Deep neural net	A neural net arranged in multiple layers, where the output of every neuron in one layer is an input to one or more neurons in the next.
Diffusion method	A machine learning technique consisting of a performance system that generates new instances of a data distribution, and a learning system that acquires the parameters of a process which reduces instances of that distribution to noise.
GAI	Generative artificial intelligence
Learning system	The component of a machine learning system that learns how to perform a task from data (e.g., that sets weights in a neural net).
LLM	Large language model
Machine learning	A field of inquiry and a family of computational techniques associated with learning from data.
Neural net	A computational model composed of interconnected, stylized "neurons." A neural net supports two operations: execution and learning.
Neuron	The primitive computational element of a neural net. Each neuron accepts an arbitrary number of numeric inputs, multiplies them by stored parameters, and outputs a function of the resulting sum. Each neuron's parameters are set by a machine learning algorithm.
Parameter	Any numerical value associated with any neuron in a neural net. The value is set by a learning system and used by a performance system.
Performance system	The component of a machine learning system that accepts new input and generates new output (e.g., to produce an image from text).
Prompt	The user input to a GAI system
Self-attention	An attention mechanism applied to the elements of a single sequence to extend it (e.g., to generate the next word in an LLM's response to a prompt).

Token	The smallest element of LLM input (e.g., a word, partial word, punctuation mark, or end of sequence tag).
Transformer	A deep learning architecture containing stacked attention elements. Used for sequence generation tasks, and the backbone of LLMs.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188		
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YY) 00-09-24		2. REPORT TYPE Key Deliverable		3. DATES COVERED (From – To)	
4. TITLE AND SUBTITLE A Grounded Introduction to Large Language Model and Generative AI Technology			5a. CONTRACT NUMBER HQ0034-19-D-0001		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBERS		
6. AUTHOR(S) Daniel G. Shapiro, Joshua Alspector			5d. PROJECT NUMBER C5241		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES Institute for Defense Analyses 730 East Glebe Road Alexandria, VA 22305			8. PERFORMING ORGANIZATION REPORT NUMBER 3002626		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 730 East Glebe Road, Alexandria, VA 22305			10. SPONSOR'S / MONITOR'S ACRONYM IDA		
			11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A. Approved for public release: distribution is unlimited.					
13. SUPPLEMENTARY NOTES Project Leader: Daniel G. Shapiro					
14. ABSTRACT This document introduces Large Language Models (LLM) and Generative AI systems (GAI) with the goal of tying reader's expectations for their capabilities, limitations, and suitable applications to core characteristics of the underlying technology. We illustrate their i/o using a text generation example, explain their function in both general and technical terms, then clarify how they are used, how they are deployed, provide an extended discussion of their limitations, and conclude with implications for their future development.					
15. SUBJECT TERMS Large language model, LLM, Generative artificial intelligence, GAI, tutorial					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Unlimited	18. NUMBER OF PAGES 45	19a. NAME OF RESPONSIBLE PERSON Institute for Defense Analyses
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include Area Code)

