Journal of Advances in Science and Technology
Vol. 22, Issue No. 4, September-2025, ISSN 2230-9659

DOI: https://doi.org/10.29070/s23zye93

Check for updates

LLMs and Compound AI Systems - Exploring Potential Applications in Real-World Scenarios

Jai Ramesh Juneja [1] *

1. Independent Researcher, Surat, Gujarat, India
jaijuneja11@gmail.com

Abstract: Compound AI Systems are the emerging phenomena integrating "large language models (LLMs)" with additional components like agents, retrievers, orchestrators, and tools to deal with issues of individual models in tasks which need reasoning, memory, multimodal knowledge, and real-time grounding. These systems compose various specialized modules into cohesive flows for more context-aware and capable behaviors. Irrespective of rising adoption in both industry and academia, the landscape of compound AI systems has been fragmented, with a lack of unified model for taxonomy, analysis, and evaluation. In LLMs, recent advancements and AI systems have made a significant change in optimization and design of complex workflows. Compound AI systems have been adept with various components when it comes to perform smart tasks. With the rise in complex systems, there are new challenges when it comes to optimizing both interactions and components. While traditional optimization models are foundational, such as "reinforcement learning (RL)" and "supervised fine-tuning (SFT)", there are promising new avenues with the rise of "natural language processing (NLP)", especially when it comes to optimizing different systems. This study offers a systematic review of recent developments for compound AI systems. It validates the notion of optimizing compound AI systems, highlights open research challenges, and classifies current approaches apart from major dimensions.

Keywords: natural language processing, reinforcement learning, large language models, supervised fine-tuning, Compound AI Systems

X

INTRODUCTION

Based on "Transformer architecture", "Large Language Models (LLMs)" have evolved significantly from academic models to foundational models for artificial intelligence (AI). These models support chat interfaces, scientific tools for discovery, enterprise copilots, and automated systems for generating codes. Flagship LLMs like Claude, Gemini, and GPT-4 regularly exceed baselines in NLP and reasoning benchmarks and the international market for Gen AI may be $1.3 trillion or beyond by 2030 (Bloomberg, 2023). However, LLMs seem to be compelling for the same properties to penetrate on autoregressive prediction of token and static corpora to bring structural changes.

First, LLMs may produce factually inaccurate and fluent output, declining trust in high-stakes environments like law, healthcare, and scientific analysis. Second, LLMs are unable to access knowledge after training, affecting their receptiveness to common facts. Third, inference budgets and context windows in bounded reasoning oblige long-horizon and multi-hop reasoning decomposition of tasks. These issues impede the effective and safe placement of LLMs in real-world, dynamic settings which need factual dependency, recency, and reasoning of composition.

The community is meeting on a new paradigm to deal with these challenges – compound AI systems

(CAIS). These are extensible and modular designs integrating LLMs with external parts, such as, agents, retrievers with high recall, long-term modules, symbolic planners, orchestration models, and multimodal encoders to perform dynamic, complex, and accurate tasks. By dissociating responsibilities across tasks and routing them to right models smartly, compound AI systems go beyond LLM's capabilities ahead of what can be achieved by monolithic models.

Early placements suggest the transformative role of this change. For instance, Perplexity.ai (2025) and other retrieval-amplified models offer real-time answers with citations based on chain-of-thought. In addition, Copilot-X by GitHub arranges repository search, reasoning of code, and test generation to improve developer throughput by around 55% (Peng et al, 2023). Combined with triage agents, multimodal pipelines have minimized turnaround of report by 30% while retaining accuracy in expert level (RADLogics, 2021). These cases mark a change in philosophy of design, i.e., from LLMs as independent soloists to conductors for scoring varied ensembles of AI.

Irrespective of rising interest, systematic knowledge is elusive in Compound AI systems. Recent studies have addressed individual aspects of this environment, such as, surveys conducted on "retrieval-augmented generation (RAG), multi-agent models, LLM agents, and LLM-based optimization (Fan et al, 2024; Li, 2025; Guo et al, 2024; Lin et al, 2024). Some focus on narrow sides like benchmark analysis, prompt engineering, or agent protocols (Ferrag et al, 2025; Ma et al, 2024; Yan et al, 2025), without having to address trade-offs and interactions across the whole stack of Compound AI systems. These works provide valuable knowledge about their domains, still there is a lack of system-level, holistic synthesis.

Compound AI systems are a new generation of AI models based on LLMs, having various smart components like code interpreters, simulators, RAG modules, and web search tools. These systems are highly capable across domains and perform better than separate LLMs. For example, LLMs working with solvers can solve the math problems at Olympiad level (Trinh et al, 2024), integrating with code interpreters and search engines to match the overall performance of programmers (Yang et al, 2024b), and drive discovery of biological materials with knowledge graphs (Ghafarollahi and Buehler, 2024).

Even with mature toolkits rationalizing the design process of CAIs, including LlamaIndex and LangChain, significant human intervention is needed to personalize these systems for targeted applications (Liu, 2022; Chase, 2022). It often covers trial-and-error mechanisms and heuristic-based pipelines (Xia et al, 2024; Zhang et al, 2024c). This limitation has encouraged efforts for developing automated, principled approaches for end-to-end optimization of AI systems. Still, operational schemes of these models diverge well as per whether modifications are allowed to the topology of the system and how learning signals are transmitted.

In addition, there is still a lack of cohesive conceptual model or standard terminology in the field, making some articles harder for navigators and newcomers (Khattab et al, 2023; Cheng et al, 2024). Irrespective of having solid frameworks in existing surveys, they are based on optimization based on natural language, overlooking major schemes allowing updates without covering recent advances (Liu et al, 2025; Lin et al, 2024). There are four important dimensions to fill these gaps, when it comes to examining current approaches. A 2x2 taxonomy has been proposed by Lee et al (2025) based on these dimensions (Figure 1) covering different works.

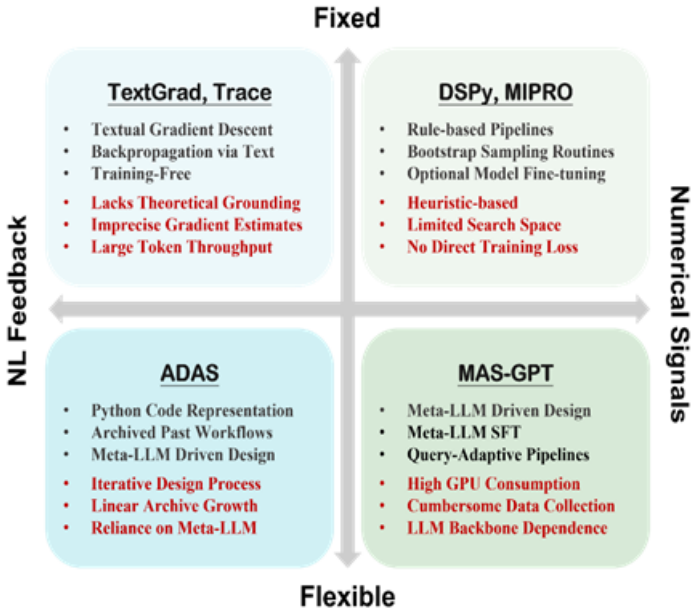

Figure 1: The 2x2 framework covering "Learning Signals (x-axis)" and "Structural Flexibility (y-axis)"

Source – Lee et al (2025)

BACKGROUND

Compound AI Systems

In contrast to individual AI models functioning as statistical models, compound AI systems are referred to as systems dealing with AI tasks with various interacting components (Vaswani et al, 2017; Zaharia et al, 2024) (Figure 2). Somehow, the term "Compound AI system" often overlaps with concepts and is often used together in the field. These cover "multi-agent systems (MAS)", language model programs, and language model pipelines (Zhou et al, 2025; Khattab et al, 2023; Opsahl-Ong et al, 2024).

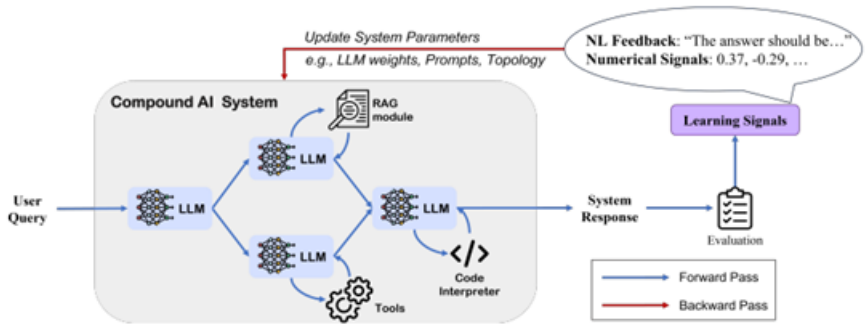

Figure 2: An illustration of a Compound AI system and optimizing it

Source – Lee et al (2025)

In Figure 2, CAI is based on LLMs and various interacting modules. Complex user queries are used by the system. There are two kinds of learning signals leveraged in automated strategies for optimization, such as

numerical signals and NLP feedback to guide updates for enhanced performance. Even though end-to-end optimization of models like neural networks is simple due to gradient-oriented backpropagation, compound AI systems are made of non-differentiable parts which need novel approaches for optimization (Paszke, 2019). Some of the common examples are heuristic models applied for finding the right examples in LLM prompts, along with methods using auxiliary models for offering textual feedback on updates (Khattab et al, 2023; Yuksekgonul et al., 2025).

Large Language Models

Language modelling is a basic task performed by NLP to predict the next character or term in a specific range of text (Jones, 1994; Chowdhary, 2020). It consists of developing models to generate and understand intelligible language. The major goal of language modelling is capturing the distribution of possibility of terms in a language, enabling the model to generate complete sentences, new text, and predicting the odds of various sequences of word (Iqbal and Qureshi, 2022; Nozza et al, 2021; Min et al, 2023; Soam and Thakur, 2022). They are widely classified into ML models, statistical models, transformer models, and deep learning models (Figure 3). N-gram models or early models were based on individual statistical approaches to predict the odds of word sequences with frequency (Diao et al, 2021; Brown et al, 1992; Omar and Al-Tashi, 2018). However, a lot of strong computing devices and public datasets can be used for processing such big data with complex models and has resulted in improving large models (Rawat et al, 2022; Lhoest et al, 2021; Sharir et al, 2020).

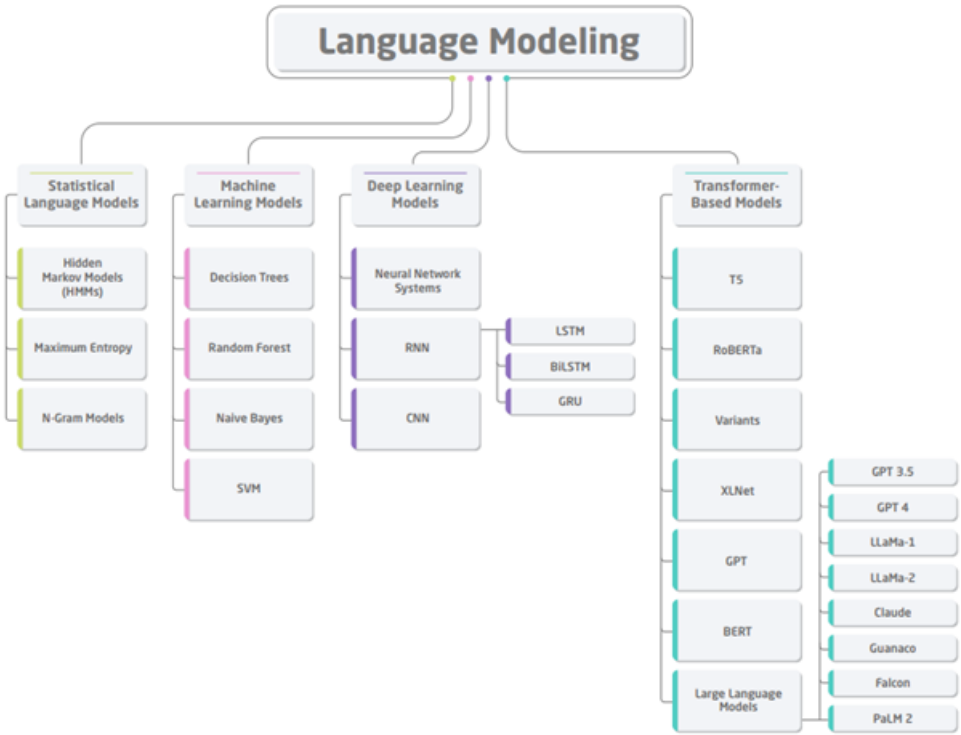

Figure 3: Categories of LLMs – ML models, Statistical models, Transformer models, and DL models. Source – Hadi et al (2024)

Also known as "next-generation" or "transformative" language models, Large Language Models (LLMs) represent a huge innovation in Natural Language Processing (Hochreiter and Schmidhuber, 1997). These

models use DL approaches, especially the transformer model (Luitse and Denkena, 2021) to understand and learn complex structures and patterns prevalent in language data (Dong et al, 2023). Ability for processing big data, such as capturing semantic relations between phrases and words and unstructured text is a key trait of LLMs (Adnan and Akbar, 2019). These models process audio, audiovisual, visual, and multi-modal information and learn semantic relations among them (Awais et al, 2025; Zhang and Bing, 2023; Rouditchenko et al, 2020; Zhao et al, 2023). These models have improved the potential of machines significantly to generate and understand familiar language (Huang and Chang, 2022).

It is possible to trace back the history of LLMs to the early stage of neural networks and language models (Pappas and Meyer, 2012). The journey starts with the age of statistical models (Bellegarda, 2004). Researchers depend mostly on prediction of sequences of words and probabilistic models (Lafferty and Zhai, 2003). Some of the classic examples are "Maximum Entropy Models", "Hidden Markov Models (HMMs) and n-grams" (Petrushin, 2000; Khudanpur and Wu, 2000). For instance, N-grams are sequences of adjacent tokens or words that can predict the odds of the next term based on preceding ones (Wang et al, 2020). These models have marked an important point to start in NLP.

They enabled basic word prediction and text generation but limited in their potential for capturing complex relationships (Rosenfeld, 2002; Arisoy et al, 2012; Bellegarda, 2002). Then, a shift has been observed towards data-based approaches (Alva-Manchego et al, 2020). Researchers have explored ML models for improving knowledge of language (Malik et al, 2021). These models have learned relationships and patterns in a large corpus of texts. ML models have brought a smarter approach to tasks related to NLP, developing applications like sentiment analysis and spam detection (Crawford et al, 2015; Neethu and Rajasree, 2013).

OPTIMIZING COMPOUND AI SYSTEMS WITH GEN AI DESIGN CYCLE

Even though end-to-end optimization of individual neural network models is simple with gradient boosting on their layer connections, compound AI systems are designed with non-differentiable components and need novel optimization (Paszke, 2019; Rumelhart, 1986). Some of the examples are heuristic methods based on bootstrap applied on finding ideal examples in-context (Khattab et al, 2023). There are also approaches using auxiliary LLM which can provide feedback on updates or improved typologies have been proposed (Yuksekgonul et al., 2025).

GenAI of Generative AI is the most generalized and disruptive technology of the debate, which has influenced a lot of industries like marketing, gaming, media, education, medical, software development, pharmaceuticals, and construction technology (Chiang, 2023). Unlike general AI programs used for specific tasks like data clustering, classification, segmentation, object detection, or predictions, GenAI can generate new and sensible content of various modalities of data, such as images, videos, and speech (Aldausari et al, 2022). Some of the common examples of GenAI are code generators (Co-Pilot), chatbots (ChatGPT or Bard), and image generators (like Midjourney) (Hong et al, 2023).

Size-wise, GenAI models have been scaled to hundreds of billions of parameters to a few million over the past years (Aydın and Karaarslan, 2023). With the rise in size of model, model also performs better, and it can be generalized for different tasks (Kim et al, 2021). However, small models can also be designed well

for better tasks (Sanh et al, 2019). Google's Bard, OpenAI's ChatGPT, and Meta's Llama are some of the Large Language Models (LLMs) designed to generate familiar language to respond to a prompt given by a human user (Jo, 2023). These models are trained on big data with techniques for learning statistical language patterns. However, a lot of people accord the potential of models to more computing power and data rather than better research (Bansal et al, 2022).

GenAI leverages statistical models and complex ones to generate new content imitating the traits and patterns of training data (Mariani, 2022). These models may cover probabilistic approaches like "Variations Auto-encoders and Autoregressive model, or Diffusion models and Generative Adversarial Networks, or Reinforcement Learning Human Feedback (RLHF)" (Zeng et al, 2021). GenAI has gathered a lot of attention over the years for its great performance in different applications related to video, text, and image generation (Muneer and Fati, 2020).

Based on the foundation of the transformer model, these models suggest excellent capacity to generate and process manual content with the use of big training data for different topics (Hassani and Silva, 2023). To understand the complex GenAI systems, it is worth focusing on the concepts of generation, variance, and data.

Data is the core of Gen AI systems. Training models which can capture the given structures and patterns successfully of the target domain. They need diverse and high-quality training data. Generating performance is affected by the quality, amount, and symbol of training data (Solaiman, 2023). In addition, labeled, large-scale datasets available enable the growth of more coherent and accurate samples, while biased or restricted training data may result in sub-optimal results (Tan et al, 2020; Wach et al, 2023).

Gen AI uses knowledge achieved from training data to generate samples with the same patterns (Goodfellow et al, 2020). These models can capture distributions of training data and create realistic and reliable samples with properties which are consistent with actual dataset (Che et al, 2017). The process of generation consists of latent space exclamation, adversarial training, and autoregressive modeling (Shafani et al, 2019; Mukherjee et al, 2019; Morrison et al, 2021).

Another important factor to define the quality and diversity of sample generation is variance, which suggests variability in samples generated (Kaushik et al, 2020). Repetitive or similar samples are generated by low variance of systems related to Gen AI systems, resulting in high variance and poor generation. It may result in unrealistic, incoherent, and diversified samples (Yang and Lerch, 2020). It is challenging to strike the balance between fidelity and variation in Gen AI, as it needs handling of trade-off among using and exploring learned distribution of data (Geneva and Zabaras, 2020; Cohen et al, 2007; Xu et al, 2021).

It is important to understand and regulate the relationship between data generation and variation for developing efficient GenAI solutions (Dhoni, 2023). It consists of dealing with issues like mode collapse, dataset biases, and managing exploitation and exploration (Vigliensoni et al, 2022; Kossale et al, 2022). GenAI systems may generate diversified, realistic, and high-quality samples corresponding to desired applications and aims by making training data refined to optimize regulation patterns (Ding et al, 2019).

It is important to evaluate the diversity and quality of generated samples to assess the overall performance of the models (Bandi et al, 2023). Here are some of the evaluation techniques for assessing the diversity,

quality, and authenticity of samples generated

- Inception Score (IS) is a well-known evaluation metric. It measures the quality of samples generated based on diversity of classes generated and visual appeal. Higher IS score suggests better diversity and quality of samples generated (Barratt and Sharma, 2018).
- Visual inspection is a subjective evaluation approach where human users or experts determine the samples generated and qualitative feedback (Chen et al, 2018).
- "Fréchet Inception Distance (FID)" is used to compare generated and real samples by estimating Fréchet distance between their features gathered from the Inception model which is generated earlier (Obukhov and Krasnyanskiy, 2020).

Figure 4 illustrates a typical design cycle of Gen AI. The development cycle of Gen AI may be classified into four steps (1) defining the problem; (2) selecting or developing models from scratch; (3) aligning and adapting the model or fine-tuning, when needed; and (4) optimization and deployment (Schmidt, 2023).

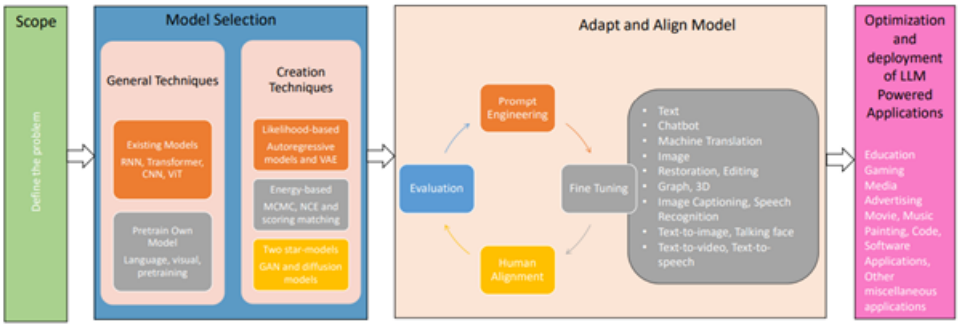

Figure 4: Schematic diagram of Gen AI Design Process. Source – Hadi et al (2024)

The first stage entails deciding the nature of the target model by GenAI. For instance, is the target to improve performance of various tasks or just one task? The next stage is model selection. This way, the developer has to decide whether to use the current model for application or pre-training the model from the beginning (Muse et al, 2023). The developer can choose general techniques like transformers, RNNs, or pretraining the model or focusing on creation models. They can cover more nuanced modifications for using the model (Foster, 2022). Following the second stage is an iterative stage to adapt and align the model for the selected scope (Wang et al, 2023). It consists of steps related to one-shot learning, zero-shot learning, or fine-tuning based on scope (Zhong et al, 2021; Dang et al, 2022). The final stage is optimizing and deploying the target application (Liu et al, 2023).

Prompting

Prompt Engineering has emerged with Large Language Models. Prompts refer to instructions offered to LLMs to follow certain roles, automated processes, and ensure that output is of high quantity or quality (Liu et al, 2023). Prompt engineering refers to wording and designing of prompts specified to LLMs to get the desired response. Hence, writing a prompt properly is very important when it comes to using LLMs to perform tasks in the best possible way (Oppenlaender, 2024).

While some formal approaches like directing LLM for some tasks (Explicit guidelines), Formatting with

example (giving sample question and answer and telling LLM to give an answer similarly), System-wide instructions (asking a question to answer from LLM), Control tokens (using special keywords to get an answer through a prompt while considering special criteria), and iteration and interaction (to reach robust answer by fine-tuning on each response) (Xue et al, 2023). Various frameworks have been proposed in the position of prompt patterns. These generic patterns target a certain category like input semantics and prompt improvement. This study aims to present some commands to enable users to make the most of the capabilities of LLMs from a generic point of view (Yang et al, 2023).

- **Defining the context** – It must be the first prompt for LLM. An example may be, "Act as a Lawyer", "Act as a Doctor", "Act as an engineer", or "Act as a Python programmer". One can define the role and direct the LLM to give replies or perform tasks as a human when information is provided (Singh et al, 2022). Context can also be provided to give a background information about the conditions to work for LLMs. For example, "We are an organization involved in mobile app development". It can be followed up with tasks to perform, actions, and steps to follow (Santu and Feng, 2023).
- **Prompt generation** – Asking the model to generate prompts for a specified task is another interesting command (Zamfirescu-Pereira et al, 2023). This way, LLMs can generate optimized prompts for performing some tasks. For example, "You are an LLM and expert in creating prompts. Generate the best prompts to gather vital data from time series information."
- **Chain of thoughts** – In the context of "Language Models (LM)", prompts related to chain of thoughts refer to providing a range of partial sentences or prompts to guide the connected or coherent text. Rather than offering individual prompt, a chain of thoughts consists of giving various prompts to keep LLMs to generate text following a certain line of narrative (Diao et al, 2023).

In-context learning is a concept in prompt engineering in terms of "teaching" the LLM to act specifically (Dong et al, 2022). A typical scheme is zero-shot inference in which LLM can perform a certain role (Kojima et al, 2022). It is in the context of an existing task and LLM can be guided to perform a task without giving a sample solution for the same. An example of this kind of prompt is classifying tweets. A user needs to provide the text and ask the same to classify it as negative or positive. One-shot inference is another kind of prompting (Liu et al, 2022). A user can give an example in this case to the LLM and ask the same for performance.

For example, a user may provide a tweet sample in tweet sentiment analysis and information to the LLM that sentiment is positive and offer a second tweet. Few-shot inference is the third kind of prompt. In this context, the user gives a few examples of solutions to teach the LLM about the operation to be performed by the user (Liu et al, 2023). For the example of sentiment analysis, it would be offering tweets suggesting its sentiment and positive sentiment. Finally, the LLM could be used for classifying tweets. In-context learning could be used to "fine-tune" the LLM for certain tasks to be performed in application (Hu et al, 2023).

Negative Prompting

It directs the LLM about the prompt's aspect that it needs to avoid excluding or generating in the process of generation (Miyake et al, 2025; Liu et al, 2022). With negative prompts, one can fine-tune the findings

generated by LLMs against a prompt while keeping it generic (Ma et al, 2022). Negative prompting enables control of output generated by the model while preventing inappropriate or harmful content. For example, "Don't write anything which is factually wrong, harmful, or offensive." With this prompt, the model will not generate text which could be inaccurate, harmful or offensive. When testing text-based image translation, negative prompting was found to be helpful when it comes to work with texture-less images (Tumanyan et al, 2023). In addition, this kind of prompting when working on image or text generation can be adopted to Muse and other image generation methods (Chang et al, 2023).

Visual Prompting

Visual prompting refers to using non-visual or visual images when offering paths to a model along with simple text prompts (Chen et al, 2023). The aim is providing a starting point to an AI model or reference or example which can be used for the specific task. It can be made possible to modify the image offered or generate something which is similar in color, texture, or style (Bar et al, 2022). It can generate content closer to the expectation of the user from the use of generative AI.

Visual prompting could be defined with an image-based example like giving an image of an office and asking it to generate a different theme, maybe in a different color or nature-centric style (Jia et al, 2022). Visual prompting enables better control over the output generated and findings in a more accurate way. It is worth noting that visual prompting is not associated only with images, which is widely being explored for a lot of applications, such as, composition of music (in which supplied piece of music could be used as reference), text generation (generating something on the basis of text to copy the writing style), game development, and augmented and virtual reality (Chakrabarty et al, 2023; Volum et al, 2022; Hegde et al, 2023).

CHALLENGES AND FUTURE DIRECTIONS

After review of different methods, this chapter focuses on various major challenges and future research directions. Irrespective of focusing on automation of optimization processes, this study identifies human interventions in current approaches, especially in designing hyperparameters related to algorithms, which limit practical value and challenge claims related to automation.

In the Fixed Structure category, some of the methods need users to design topologies related to systems based on expertise. Even though some studies determine their models across various designs (Zhao et al, 2025; Yin and Wang, 2025), it is not guaranteed that these configurations meet the needs of all targeted applications. Textual hyperparameters also appear constantly in different approaches. For instance, prompt templates are used for optimizer, evaluator, and gradient estimator in TextGrad and the ones in ADAS are designed with lack of proper design or sensitivity analysis (Hu et al., 2024; Yuksekgonul et al., 2025).

There are also numerical hyperparameter designs and bootstrap samples are persistent, which cannot be automated (Khattab et al, 2023). Manual configuration is needed also in flexible structure approaches which are seemingly automated, including MAS-GPT as evidenced by prompt templates (Ye et al, 2025). Irrespective of efforts applying meta-learning to optimize evaluator, human intervention is needed to craft

the prompts of meta-learner. To move ahead for automated optimization, like neural network training, future research is needed to reduce dependence on both numerical and textual hyperparameters. For any hyperparameter, complete sensitivity analyses are needed to know robustness and behavior of each approach.

Burden optimization is quite more challenging for compound AI systems as compared to tuning simple models. Current approaches resort to several workarounds, resulting in increased computational cost. TextGrad and other approaches in feedback learning need various LLMs to approximate a single step. Even though approaches like ADAS and Trace use global LLM are needed in each step of optimization, they should embed widespread context, which enhances token throughput (Hu et al, 2023). As these approaches usually depend on branded models, they suffer increased cost of API (Achiam et al, 2023). On the other hand, numerical approaches based on signals often use open-source LLMs to prevent API costs. Usually, these models need fine-tuning for better performance, while changing the burden of resources related to GPU. Hence, developers face the tradeoff among GPU cost and API needs.

In addition, there is also a rise in computational expenses in inference. By focusing majorly on overall system performance, existing approaches often ignore the need to regularize complexity of the system, causing unbounded consumption of resources at runtime. Even though a few approaches have encouraged simpler designs, their scalability and applicability in deployments should be tested (Zhang et al, 2023; Liu et al, 2022). Hence, it is suggested that future studies should propose efficient optimization models and device key ways to constrain complexity of the system without affecting performance.

There is limited experimental scope in this study. Compound AI systems are supposed to address complex situations. Future studies may investigate their efficiency in more challenging roles. In this field, studies focus majorly on their proposed approaches on widely used datasets for individual LLMs like code generation, MMLU, and commonsense reasoning" (Cobbe et al, 2021; Chen et al, 2023; Austin et al, 2021). Though these evaluations suggest overall efficiency, it is worth covering benchmarks covering more complex roles. For instance, there are various tasks related to various LLMs in the system to discuss and cooperate. Given the large-scale use of compound AI systems, for example, healthcare systems embedded as nodes, it is worth evaluating the performance of algorithms where humans act as nodes in the system.

While promising empirical findings to have been observed in NL Feedback, there is a lack of theoretical assurance. For instance, textual gradient has been convergent, and classical gradient is supported by formal proofs of convergence (Hutzenthaler et al., 2021). Those proofs offer a strong foundation for constant advancement of optimizing individual models. Hence, future studies should provide convergence and analyze optimally to learn through feedback, which offers more in-depth knowledge to promote theoretical underpinnings of the field.

While safety issues and defenses have been studied widely, including jailbreak attacks and their control on LLMs (Yi et al, 2024) and manual AI pipelines to minimize harmful outputs and prompts (Zhou et al, 2025), the attack surface expands significantly in compound AI models (Banerjee et al, 2024). For example, privacy-preserving models may still leak sensitive information when designed as a larger system component (Debenedetti et al, 2024).

In addition, since compound AI systems are executed and embodied as code in the enterprise settings, latent modes may not be detected and undermine reliability of systems, even without explicit attacks. Research extends ahead of downstream performance on compound AI systems, and it has addressed execution efficiency so far (Zhang et al, 2025), with small attention given to system-led safety or alignment (Zheng et al, 2025). Safeguarding optimization and mature alignment available for individual models need future studies to extend the strategies related to compound solutions to manage capability improvements with guarantee for safety (Dai et al, 2023).

There is also a lack of widely adopted and standardized library support in this field. Even though DSPy, TextGrad and other maintained libraries have become popular among practitioners, still a lot of works have adopted optimization of compound AI systems with self-crafted, custom databases (Khattab et al, 2023; Yuksekgonul et al., 2025). While frameworks dominate training of individual models like PyTorch and TensorFlow (Paszke, 2019; Abadi et al, 2015), best practices are still being developed for optimizing and adopting compound AI systems. Future studies must focus on comparing and benchmarking current libraries for optimizing compound AI systems. Those efforts could help in improving and building clear guidelines for researchers and developers (Novac et al, 2022).

CONCLUSION

This study provides comprehensive insights to emerging challenges, trends, and potentials of compound AI systems, integrating LLMs with retrievers, agents, visual encoders, orchestrators, and symbolic planners. By mapping the existing research frameworks and landscape, this study places compound AI systems as the next point of inflection in AI development, enabling transition from monolithic, static models to context-based and modular systems that are capable of adaptive learning, reasoning and coordination. The study has re-entered the optimization models, architecture, human-in-the-loop, and multimodal extensions defining the new generation of AI programs that can address complex issues like software validation, automated research, healthcare systems, and scientific discovery.

This study contributes by highlighting the shifting of compound AI systems towards addressing the limitations of individual LLMs. While traditional LLMs rely completely on autoregressive prediction of token restricted to static corpora and fixed context windows, compound systems rely on distribution intelligence and augmentation to achieve combined cognition. Retrievals are used by these systems for factual evidence, planners for adaptive routing, and agents for executing decisions, causing smart architecture like reasoning decomposition. From linear processing to orchestrated intelligence based on memory, multimodality, and reasoning, there is a paradigm shift marked by specialized modules.

In compound AI systems, current optimization models are widely classified into flexible and fixed structural systems from a methodological point of view. Pre-defined topologies are widely used by fixed structures depending on gradient-boosting and reinforcement learning techniques for aligning subsystems. In contrast, flexible structures adopt feedback loops dynamically with reinforcing natural language. In addition, natural language (NL) based gradient boosting and feedback optimization is identified as promising ways to align AI without backpropagation. In addition, there is an evolving notion of optimization, i.e., from numerical convergence to coordination and semantic consistency. They have shown wider redefinition of the meaning of learning in compositional designs.

There are various core challenges pointed out in this study along with significant advancements. Over-dependence on manual tuning is one of the major problems, contradicting the goal of automation. The numerical and textual hyperparameters are designed to demand expert intuition, resulting in reproducibility, subjectivity, and inefficiency issues. Like backpropagation in neural networks, achieving complete optimization has been a vital pursuit for the research. In addition, performance efficiency can be balanced with scalability in computation. While powerful, compound systems often suggest computational overhead because of constant LLM calls and orchestration in feedback loops, which inflate carbon footprints and token-based processing. Hence, cost-aware, lightweight architectures are needed to focus on modular efficiency.

The study also highlights the gap between innovation and standardization in compound AI systems. When open-source tools like Llama, DSPy, etc. have democratized access to designing the workflow for compound AI solutions. Depending excessively on proprietary environments, APIs generate challenges to reproducible research and limit testing with LLMs. The community needs a unified environment like TensorFlow or PyTorch for optimizing compound AI systems, supporting retrievers, interoperability among various agents, and evaluators. Building such a standard would improve benchmarking, reproducibility, and scalability to enable researchers to compare systems in a transparent manner.

The dimensions of interpretability, safety, and ethical aspects also call for increased attention from scholars. With the rise in complexity of compound AI systems, there is a rise in attacks, generating new risks associated with adversarial attacks, privacy leaks, and vulnerabilities in pipelines. Unlike impartial LLMs, compound systems include interaction of components which may amplify or propagate unsafe outputs. Multi-level alignment is needed to go beyond "reinforcement learning" to system-wide reinforcement learning to ensure interpretability, safety, and accountability with the whole orchestration chain. Strong defensive layers like calibrated limitations, verification of context, and rule-oriented guardrails must be entrenched at every level to comply with applications.

This study was based on a survey of recent advancements when it comes to optimize compound AI systems with components and tools. This study also proposed formal approaches to enable structural analysis of system configurations. This study examines current approaches across various core areas, while highlighting trade-offs and key trends, such as, NL interface, computational expenses, and problems in generalizing and scaling. It also identifies open challenges and discusses future research directions.

This study also observes some limitations. First, compound AI systems are not defined universally. This study covers works which identify themselves as optimizing LM systems or multi-agent systems, without systematic analysis of conceptual differences and overlaps. Secondly, this study focuses majorly on approaches which optimize systems of various nodes explicitly to exclude traditional techniques for prompt optimization for individual LLMs. This field is rapidly evolving, even during the preparation of this study.

When it comes to implications, the study reveals socio-technical aspects of AI in redefining workforce education, scientific innovation, and productivity. Empirical evidence related to improving multimodal diagnosis and developer throughput systems promote clinical decisions suggesting tangible effects of orchestration-based intelligence. However, responsible governance is needed with the same power demands. As organizations embed the mission-critical pipelines of AI, the ethical allocation between

machine autonomy and human oversight is needed. There is a need to codify a clear framework for accountability to ensure transparency in complex AI decisions.

Various trajectories are emerging with the perspective of research. First, the autonomous system is evolving with rise in interest, where compound AI agents co-optimize and collaborate their behavior with self-improvement meta-learning. Such developments could promote innovation, resulting in redesigning AI systems. In addition, differentiable programing and reasoning are needed to be integrated into compound models to improve both reliability and interpretability. By combining reinforcement learning and symbolic logic, future compound systems may attain equilibrium between neural intuition and structural knowledge. Third, the ability to combine textual, auditory, and visual reasoning will dictate competitive AI platforms in robotics, education, and environmental modeling.

In addition, this study focuses on the significance of benchmarking and evaluation when it comes to advanced compound AI systems. Current techniques related to assessment are specified to single module or narrow tasks. Future studies should measure behaviors like failure recovery, fusion of knowledge, and collaborative reasoning around modules. For instance, cross-disciplinary benchmarks should be designed to evaluate systems on real-time disaster response or sustainability modeling. These meta-level metrics must account for interpretability, accuracy, social value and efficiency.

To conclude, this study underlines that optimization of compound AI systems is not just a computational issue, but a systemic shift on designing intelligence, understanding and deploying them. The field is known for confluence of philosophical reorientation and technical growth, requiring multidisciplinary efforts for cognitive studies, ethics and computer science.

References

1. Bloomberg Intelligence (2023). Generative AI to Become a $1.3 Trillion Market by 2032, Research Finds. https://www.bloomberg.com/company/press/generative-ai-to-become-a-1-3-trillion-market-by-2032-research-finds/.
2. Perplexity AI (2025). Perplexity AI. https://www.perplexity.ai/.
3. Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot. arXiv preprint arXiv:2302.06590.
4. RADLogics (2021). Use of AI to Analyze Chest CT Shortens Turnaround Times in Russia. https://www.auntminnieeurope.com/imaging-informatics/artificial-intelligence/article/15655440/use-of-ai-to-analyze-chest-ct-shortens-turnaround-times-in-russia.
5. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., ... & Li, Q. (2024, August). A survey on rag meeting llms: Towards retrieval-augmented large language models. In Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining (pp. 6491-6501).
6. Li, X. (2025, January). A review of prominent paradigms for llm-based agents: Tool use, planning (including rag), and feedback learning. In Proceedings of the 31st International Conference on

Computational Linguistics (pp. 9760-9779).

7. Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., ... & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680.
8. Lin, M., Sheng, J., Zhao, A., Wang, S., Yue, Y., Wu, Y., ... & Liu, Y. J. (2024). LLM-based Optimization of Compound AI Systems: A Survey. arXiv e-prints, arXiv-2410.
9. Ferrag, M. A., Tihanyi, N., & Debbah, M. (2025). From llm reasoning to autonomous ai agents: A comprehensive review. arXiv preprint arXiv:2504.19678.
10. Ma, R., Wang, X., Zhou, X., Li, J., Du, N., Gui, T., ... & Huang, X. (2024). Are large language models good prompt optimizers?. arXiv preprint arXiv:2402.02101.
11. Yan, B., Zhou, Z., Zhang, L., Zhang, L., Zhou, Z., Miao, D., ... & Zhang, X. (2025). Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. arXiv preprint arXiv:2502.14321.
12. Lee, Y. A., Yi, G. T., Liu, M. Y., Lu, J. C., Yang, G. B., & Chen, Y. N. (2025). Compound AI Systems Optimization: A Survey of Methods, Challenges, and Future Directions. arXiv preprint arXiv:2506.08234.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
14. Zaharia, M., Khattab, O., Chen, L., Davis, J. Q., Miller, H., Potts, C., ... & Ghodsi, A. (2024). The shift from models to compound ai systems. Berkeley Artificial Intelligence Research Lab. Available online at: https://bair. berkeley. edu/blog/2024/02/18/compound-ai-systems/(accessed February 27, 2024).
15. Zhou, H., Wan, X., Sun, R., Palangi, H., Iqbal, S., Vulić, I., ... & Arık, S. Ö. (2025). Multi-agent design: Optimizing agents with better prompts and topologies. arXiv preprint arXiv:2502.02533.
16. Opsahl-Ong, K., Ryan, M. J., Purtell, J., Broman, D., Potts, C., Zaharia, M., & Khattab, O. (2024). Optimizing instructions and demonstrations for multi-stage language model programs. arXiv preprint arXiv:2406.11695.
17. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.
18. Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., ... & Potts, C. (2023). Dspy: Compiling declarative language model calls into self-improving pipelines. arXiv preprint arXiv:2310.03714.
19. Yuksekgonul, M., Bianchi, F., Boen, J., Liu, S., Lu, P., Huang, Z., ... & Zou, J. (2025). Optimizing generative AI by backpropagating language model feedback. Nature, 639(8055), 609-616.
20. Jones, K. S. (1994). Natural language processing: a historical review. Current issues in computational linguistics: in honour of Don Walker, 3-16.

21. Chowdhary, K. (2020). Natural language processing. Fundamentals of artificial intelligence, 603-649.

22. Iqbal, T., & Qureshi, S. (2022). The survey: Text generation models in deep learning. J. King Saud Univ. Comput. Inf. Sci., 34(6 Part A), 2515-2528.

23. Nozza, D., Bianchi, F., & Hovy, D. (2021). HONEST: Measuring hurtful sentence completion in language models. In Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies. Association for Computational Linguistics.

24. Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., ... & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys, 56(2), 1-40.

25. Soam, M., & Thakur, S. (2022, January). Next word prediction using deep learning: A comparative study. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 653-658). IEEE.

26. Diao, S., Xu, R., Su, H., Jiang, Y., Song, Y., & Zhang, T. (2021, August). Taming pre-trained language models with n-gram representations for low-resource domain adaptation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 3336-3349).

27. Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-480.

28. Omar, N., & Al-Tashi, Q. (2018). Arabic nested noun compound extraction based on linguistic features and statistical measures. GEMA Online® Journal of Language Studies, 18(2).

29. Rawat, B., Bist, A. S., Rahardja, U., Aini, Q., & Sanjaya, Y. P. A. (2022, September). Recent deep learning based nlp techniques for chatbot development: An exhaustive survey. In 2022 10th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-4). IEEE.

30. Lhoest, Q., Del Moral, A. V., Jernite, Y., Thakur, A., Von Platen, P., Patil, S., ... & Wolf, T. (2021). Datasets: A community library for natural language processing. arXiv preprint arXiv:2109.02846.

31. Sharir, O., Peleg, B., & Shoham, Y. (2020). The cost of training nlp models: A concise overview. arXiv preprint arXiv:2004.08900.

32. Hadi, M. U., Al-Tashi, Q., Qureshi, R., Shah, A., Muneer, A., Irfan, M., ... & Shah12, M. (2024). LLMs: A Comprehensive Survey of Applications, Challenges, Datasets, Models, Limitations, and Future Prospects.

33. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

34. Luitse, D., & Denkena, W. (2021). The great transformer: Examining the role of large language models in the political economy of AI. Big Data & Society, 8(2), 20539517211047734.

35. Dong, Z., Tang, T., Li, L., & Zhao, W. X. (2023). A survey on long text modeling with transformers. arXiv preprint arXiv:2302.14502.

36. Adnan, K., & Akbar, R. (2019). An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data, 6(1), 1-38.

37. Awais, M., Naseer, M., Khan, S., Anwer, R. M., Cholakkal, H., Shah, M., ... & Khan, F. S. (2025). Foundation models defining a new era in vision: a survey and outlook. IEEE Transactions on Pattern Analysis and Machine Intelligence.

38. Zhang, H., Li, X., & Bing, L. (2023). Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858.

39. Rouditchenko, A., Boggust, A., Harwath, D., Chen, B., Joshi, D., Thomas, S., ... & Glass, J. (2020). Avlnet: Learning audio-visual language representations from instructional videos. arXiv preprint arXiv:2006.09199.

40. Zhao, Y., Lin, Z., Zhou, D., Huang, Z., Feng, J., & Kang, B. (2023). Bubogpt: Enabling visual grounding in multi-modal llms. arXiv preprint arXiv:2307.08581.

41. Huang, J., & Chang, K. C. C. (2022). Towards reasoning in large language models: A survey. arXiv preprint arXiv:2212.10403.

42. Pappas, N., & Meyer, T. (2012). A survey on language modeling using neural networks. Idiap, Martigny, Switzerland, Tech. Rep. Idiap-RR-32-2012.

43. Bellegarda, J. R. (2004). Statistical language model adaptation: review and perspectives. Speech communication, 42(1), 93-108.

44. Lafferty, J., & Zhai, C. (2003). Probabilistic relevance models based on document and query generation. In Language modeling for information retrieval (pp. 1-10). Dordrecht: Springer Netherlands.

45. Petrushin, V. A. (2000, July). Hidden markov models: Fundamentals and applications. In Online Symposium for Electronics Engineer.

46. Khudanpur, S., & Wu, J. (2000). Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. Computer Speech & Language, 14(4), 355-372.

47. Wang, H., He, J., Zhang, X., & Liu, S. (2020). A short text classification method based on N-gram and CNN. Chinese Journal of Electronics, 29(2), 248-254.

48. Rosenfeld, R. (2002). Two decades of statistical language modeling: Where do we go from here?. Proceedings of the IEEE, 88(8), 1270-1278.

49. Arisoy, E., Sainath, T. N., Kingsbury, B., & Ramabhadran, B. (2012, June). Deep neural network

language models. In Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT (pp. 20-28).

50. Bellegarda, J. R. (2002). Exploiting latent semantic information in statistical language modeling. Proceedings of the IEEE, 88(8), 1279-1296.

51. Alva-Manchego, F., Scarton, C., & Specia, L. (2020). Data-driven sentence simplification: Survey and benchmark. Computational Linguistics, 46(1), 135-187.

52. Malik, M., Malik, M. K., Mehmood, K., & Makhdoom, I. (2021). Automatic speech recognition: a survey. Multimedia Tools and Applications, 80(6), 9411-9457.

53. Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), 23.

54. Neethu, M. S., & Rajasree, R. (2013, July). Sentiment analysis in twitter using machine learning techniques. In 2013 fourth international conference on computing, communications and networking technologies (ICCCNT) (pp. 1-5). IEEE.

55. Chiang, I. (2023). Unleashing the power of generative AI: The race for advancement and the global ramifications. Massachusetts Institute of Technology.

56. Aldausari, N., Sowmya, A., Marcus, N., & Mohammadi, G. (2022). Video generative adversarial networks: a review. ACM Computing Surveys (CSUR), 55(2), 1-25.

57. Hong, S., Seo, J., Shin, H., Hong, S., & Kim, S. (2023). Direct2v: Large language models are frame-level directors for zero-shot text-to-video generation. arXiv preprint arXiv:2305.14330.

58. Aydın, Ö., & Karaarslan, E. (2023). Is ChatGPT leading generative AI? What is beyond expectations?. Academic Platform Journal of Engineering and Smart Systems, 11(3), 118-134.

59. Kim, B., Kim, H., Lee, S. W., Lee, G., Kwak, D., Jeon, D. H., ... & Sung, N. (2021). What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. arXiv preprint arXiv:2109.04650.

60. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

61. Jo, A. (2023). The promise and peril of generative AI. Nature, 614(1), 214-216.

62. Bansal, H., Gopalakrishnan, K., Dingliwal, S., Bodapati, S., Kirchhoff, K., & Roth, D. (2022). Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. arXiv preprint arXiv:2212.09095.

63. Mariani, M. (2022). Generative artificial intelligence and innovation: conceptual foundations. Available at SSRN 4249382.

64. Zeng, W., Ren, X., Su, T., Wang, H., Liao, Y., Wang, Z., ... & Tian, Y. (2021). PanGu-$\alpha $:

Large-scale autoregressive pretrained Chinese language models with auto-parallel computation. arXiv preprint arXiv:2104.12369.

65. Muneer, A., & Fati, S. M. (2020). A comparative analysis of machine learning techniques for cyberbullying detection on twitter. Future Internet, 12(11), 187.

66. Hassani, H., & Silva, E. S. (2023). The role of ChatGPT in data science: how ai-assisted conversational interfaces are revolutionizing the field. Big data and cognitive computing, 7(2), 62.

67. Solaiman, I. (2023, June). The gradient of generative AI release: Methods and considerations. In Proceedings of the 2023 ACM conference on fairness, accountability, and transparency (pp. 111-122).

68. Tan, S., Shen, Y., & Zhou, B. (2020). Improving the fairness of deep generative models without retraining. arXiv preprint arXiv:2012.04842.

69. Wach, K., Duong, C. D., Ejdys, J., Kazlauskaitė, R., Korzynski, P., Mazurek, G., ... & Ziemba, E. (2023). The dark side of generative artificial intelligence: A critical analysis of controversies and risks of ChatGPT. Entrepreneurial Business and Economics Review, 11(2), 7-30.

70. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

71. Che, Z., Cheng, Y., Zhai, S., Sun, Z., & Liu, Y. (2017, November). Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In 2017 IEEE International Conference on Data Mining (ICDM) (pp. 787-792). IEEE.

72. Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., ... & Goldstein, T. (2019). Adversarial training for free!. Advances in neural information processing systems, 32.

73. Mukherjee, S., Asnani, H., Lin, E., & Kannan, S. (2019, July). Clustergan: Latent space clustering in generative adversarial networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4610-4617).

74. Morrison, M., Kumar, R., Kumar, K., Seetharaman, P., Courville, A., & Bengio, Y. (2021). Chunked autoregressive gan for conditional waveform synthesis. arXiv preprint arXiv:2110.10139.

75. Kaushik, S., Choudhury, A., Natarajan, S., Pickett, L. A., & Dutt, V. (2020). Medicine expenditure prediction via a variance-based generative adversarial network. IEEE Access, 8, 110947-110958.

76. Yang, L. C., & Lerch, A. (2020). On the evaluation of generative models in music. Neural Computing and Applications, 32(9), 4773-4784.

77. Geneva, N., & Zabaras, N. (2020). Multi-fidelity generative deep learning turbulent flows. arXiv preprint arXiv:2006.04731.

78. Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. Philosophical Transactions of the Royal

Society B: Biological Sciences, 362(1481), 933-942.

79. Xu, D., Zhu, F., Liu, Q., & Zhao, P. (2021). Improving exploration efficiency of deep reinforcement learning through samples produced by generative model. Expert Systems with Applications, 185, 115680.

80. Dhoni, P. (2023). Exploring the synergy between generative AI, data and analytics in the modern age. Authorea Preprints.

81. Vigliensoni, G., Perry, P., & Fiebrink, R. (2022). A small-data mindset for generative AI creative work.

82. Kossale, Y., Airaj, M., & Darouichi, A. (2022, October). Mode collapse in generative adversarial networks: An overview. In 2022 8th International Conference on Optimization and Applications (ICOA) (pp. 1-6). IEEE.

83. Ding, Y., Mishra, N., & Hoffmann, H. (2019, June). Generative and multi-phase learning for computer systems optimization. In Proceedings of the 46th International Symposium on Computer Architecture (pp. 39-52).

84. Bandi, A., Adapa, P. V. S. R., & Kuchi, Y. E. V. P. K. (2023). The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges. Future Internet, 15(8), 260.

85. Novac, O. C., Chirodea, M. C., Novac, C. M., Bizon, N., Oproescu, M., Stan, O. P., & Gordan, C. E. (2022). Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural network. Sensors, 22(22), 8872.

86. Barratt, S., & Sharma, R. (2018). A note on the inception score. arXiv preprint arXiv:1801.01973.

87. Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., & Smagt, P. (2018, March). Metrics for deep generative models. In International Conference on Artificial Intelligence and Statistics (pp. 1540-1550). PMLR.

88. Obukhov, A., & Krasnyanskiy, M. (2020). Quality assessment method for GAN based on modified metrics inception score and Fréchet inception distance. In Proceedings of the Computational Methods in Systems and Software (pp. 102-114). Cham: Springer International Publishing.

89. Schmidt, A. (2023, June). Speeding up the engineering of interactive systems with generative AI. In Companion Proceedings of the 2023 ACM SIGCHI Symposium on Engineering Interactive Computing Systems (pp. 7-8).

90. Muse, H., Bulathwela, S., & Yilmaz, E. (2023). Pre-training with scientific text improves educational question generation (student abstract). In proceedings of the aaai conference on artificial intelligence (Vol. 37, No. 13, pp. 16288-16289).

91. Foster, D. (2022). Generative deep learning. " O'Reilly Media, Inc.".

92. Wang, Y., Zhong, W., Li, L., Mi, F., Zeng, X., Huang, W., ... & Liu, Q. (2023). Aligning large language models with human: A survey. arXiv preprint arXiv:2307.12966.

93. Zhong, R., Lee, K., Zhang, Z., & Klein, D. (2021). Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. arXiv preprint arXiv:2104.04670.

94. Dang, H., Mecke, L., Lehmann, F., Goller, S., & Buschek, D. (2022). How to prompt? Opportunities and challenges of zero-and few-shot learning for human-AI interaction in creative applications of generative models. arXiv preprint arXiv:2209.01390.

95. Oppenlaender, J. (2024). A taxonomy of prompt modifiers for text-to-image generation. Behaviour & Information Technology, 43(15), 3763-3776.

96. Xue, T., Wang, Z., & Ji, H. (2023). Parameter-efficient tuning helps language model alignment. arXiv preprint arXiv:2310.00819.

97. Yang, K., Ji, S., Zhang, T., Xie, Q., Kuang, Z., & Ananiadou, S. (2023). Towards interpretable mental health analysis with large language models. arXiv preprint arXiv:2304.03347.

98. Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., ... & Garg, A. (2022). Progprompt: Generating situated robot task plans using large language models. arXiv preprint arXiv:2209.11302.

99. Santu, S. K. K., & Feng, D. (2023). Teler: A general taxonomy of llm prompts for benchmarking complex tasks. arXiv preprint arXiv:2305.11430.

100. Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B., & Yang, Q. (2023, April). Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In Proceedings of the 2023 CHI conference on human factors in computing systems (pp. 1-21).

101. Diao, S., Wang, P., Lin, Y., Pan, R., Liu, X., & Zhang, T. (2023). Active prompting with chain-of-thought for large language models. arXiv preprint arXiv:2302.12246.

102. Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., ... & Sui, Z. (2022). A survey on in-context learning. arXiv preprint arXiv:2301.00234.

103. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. Advances in neural information processing systems, 35, 22199-22213.

104. Liu, F., Eisenschlos, J. M., Piccinno, F., Krichene, S., Pang, C., Lee, K., ... & Altun, Y. (2022). Deplot: One-shot visual language reasoning by plot-to-table translation. arXiv preprint arXiv:2212.10505.

105. Liu, X., McDuff, D., Kovacs, G., Galatzer-Levy, I., Sunshine, J., Zhan, J., ... & Patel, S. (2023). Large language models are few-shot health learners. arXiv preprint arXiv:2305.15525.

106. Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E. P., Bing, L., ... & Lee, R. K. W. (2023). Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. arXiv preprint

arXiv:2304.01933.

107. Miyake, D., Iohara, A., Saito, Y., & Tanaka, T. (2025, February). Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (pp. 2063-2072). IEEE.

108. Liu, N., Li, S., Du, Y., Torralba, A., & Tenenbaum, J. B. (2022, October). Compositional visual generation with composable diffusion models. In European conference on computer vision (pp. 423-439). Cham: Springer Nature Switzerland.

109. Ma, F., Zhang, C., Ren, L., Wang, J., Wang, Q., Wu, W., ... & Song, D. (2022). Xprompt: Exploring the extreme of prompt tuning. arXiv preprint arXiv:2210.04457.

110. Tumanyan, N., Geyer, M., Bagon, S., & Dekel, T. (2023). Plug-and-play diffusion features for text-driven image-to-image translation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1921-1930).

111. Chang, H., Zhang, H., Barber, J., Maschinot, A. J., Lezama, J., Jiang, L., ... & Krishnan, D. (2023). Muse: Text-to-image generation via masked generative transformers. arXiv preprint arXiv:2301.00704.

112. Chen, A., Yao, Y., Chen, P. Y., Zhang, Y., & Liu, S. (2023). Understanding and improving visual prompting: A label-mapping perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19133-19143).

113. Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., & Efros, A. (2022). Visual prompting via image inpainting. Advances in Neural Information Processing Systems, 35, 25005-25017.

114. Jia, M., Tang, L., Chen, B. C., Cardie, C., Belongie, S., Hariharan, B., & Lim, S. N. (2022, October). Visual prompt tuning. In European conference on computer vision (pp. 709-727). Cham: Springer Nature Switzerland.

115. Chakrabarty, T., Saakyan, A., Winn, O., Panagopoulou, A., Yang, Y., Apidianaki, M., & Muresan, S. (2023). I spy a metaphor: Large language models and diffusion models co-create visual metaphors. arXiv preprint arXiv:2305.14724.

116. Volum, R., Rao, S., Xu, M., DesGarennes, G. A., Brockett, C., Van Durme, B., ... & Dolan, B. (2022, July). Craft an iron sword: Dynamically generating interactive game characters by prompting large language models tuned on code. In The Third Wordplay: When Language Meets Games Workshop.

117. Hegde, D., Valanarasu, J. M. J., & Patel, V. (2023). Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2028-2038).

118. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

119. Ye, R., Tang, S., Ge, R., Du, Y., Yin, Z., Chen, S., & Shao, J. (2025). MAS-GPT: Training LLMs to

build LLM-based multi-agent systems. arXiv preprint arXiv:2503.03686.

120. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

121. Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., ... & Schulman, J. (2021). Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

122. Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., ... & Sutton, C. (2021). Program synthesis with large language models. arXiv preprint arXiv:2108.07732.

123. Hutzenthaler, M., Jentzen, A., Pohl, K., Riekert, A., & Scarpa, L. (2021). Convergence proof for stochastic gradient descent in the training of deep neural networks with ReLU activation for constant target functions. arXiv preprint arXiv:2112.07369.

124. Yi, S., Liu, Y., Sun, Z., Cong, T., He, X., Song, J., ... & Li, Q. (2024). Jailbreak attacks and defenses against large language models: A survey. arXiv preprint arXiv:2407.04295.

125. Zhou, H., Wan, X., Sun, R., Palangi, H., Iqbal, S., Vulić, I., ... & Arık, S. Ö. (2025). Multi-agent design: Optimizing agents with better prompts and topologies. arXiv preprint arXiv:2502.02533.

126. Banerjee, S., Sahu, P., Luo, M., Vahldiek-Oberwagner, A., Yadwadkar, N. J., & Tiwari, M. (2024). Sok: A systems perspective on compound ai threats and countermeasures. arXiv preprint arXiv:2411.13459.

127. Debenedetti, E., Severi, G., Carlini, N., Choquette-Choo, C. A., Jagielski, M., Nasr, M., ... & Tramèr, F. (2024). Privacy side channels in machine learning systems. In 33rd USENIX Security Symposium (USENIX Security 24) (pp. 6861-6848).

128. Zheng, C., Chen, J., Lyu, Y., Ng, W. Z. T., Zhang, H., Ong, Y. S., ... & Yin, H. (2025). MermaidFlow: Redefining Agentic Workflow Generation via Safety-Constrained Evolutionary Programming. arXiv preprint arXiv:2505.22967.

129. Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., ... & Yang, Y. (2023). Safe rlhf: Safe reinforcement learning from human feedback. arXiv preprint arXiv:2310.12773.

130. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.