Artificial Intelligence Review (2024) 57:175
https://doi.org/10.1007/s10462-024-10824-0

A survey of safety and trustworthiness of large language models through the lens of verification and validation

Xiaowei Huang[1] · Wenjie Ruan[1] · Wei Huang[1,5] · Gaojie Jin[1] · Yi Dong[1] · Changshun Wu[2] · Saddek Bensalem[2] · Ronghui Mu[1] · Yi Qi[1] · Xingyu Zhao[1] · Kaiwen Cai[1] · Yanghao Zhang[1] · Sihao Wu[1] · Peipei Xu[1] · Dengyu Wu[1] · Andre Freitas[3] · Mustafa A. Mustafa[3,4]

Accepted: 5 June 2024 / Published online: 17 June 2024

Abstract
Large language models (LLMs) have exploded a new heatwave of AI for their ability to engage end-users in human-level conversations with detailed and articulate answers across many knowledge domains. In response to their fast adoption in many industrial applications, this survey concerns their safety and trustworthiness. First, we review known vulnerabilities and limitations of the LLMs, categorising them into inherent issues, attacks, and unintended bugs. Then, we consider if and how the Verification and Validation (V&V) techniques, which have been widely developed for traditional software and deep learning models such as convolutional neural networks as independent processes to check the alignment of their implementations against the specifications, can be integrated and further extended throughout the lifecycle of the LLMs to provide rigorous analysis to the safety and trustworthiness of LLMs and their applications. Specifically, we consider four complementary techniques: falsification and evaluation, verification, runtime monitoring, and regulations and ethical use. In total, 370+ references are considered to support the quick understanding of the safety and trustworthiness issues from the perspective of V&V. While intensive research has been conducted to identify the safety and trustworthiness issues, rigorous yet practical methods are called for to ensure the alignment of LLMs with safety and trustworthiness requirements.

Keywords AI Safety · Trustworthy AI · Verification and Validation · Safeguarding · Large Language Models · Generative AI

✉ Xiaowei Huang
xiaowei.huang@liverpool.ac.uk

[1] University of Liverpool, Liverpool, UK

[2] Université Grenoble Alpes, Grenoble, France

[3] The University of Manchester, Manchester, UK

[4] COSIC, KU Leuven, Leuven, Belgium

[5] Purple Mountain Laboratories, Nanjing, China

1 Introduction

A large language model (LLM) is a deep learning model equipped with a massive amount of learnable parameters (commonly reaching more than 10 billion). LLMs are attention-based sequential models based on the transformer architecture (Hrinchuk et al. 2020), which consistently demonstrated the ability to learn universal representations of language. The universal representations of language can then be used in various Natural Language Processing (NLP) task. The recent scale-up of these models, in terms of both numbers of parameters and pre-trained corpora, has confirmed the universality of transformers as mechanisms to encode language representations. At a specific scale, these models started to exhibit in-context learning (Min et al. 2022; Ye et al. 2022), and the properties of learning from few examples (zero/one/few-shot—without the need for fine-tuning) and from natural language prompts (complex instructions which describe the behavioural intent that the model needs to operate). Recent works on Reinforcement Learning via Human Feedback (RLHF) (Ouyang et al. 2022) have further developed the ability of these models to align and respond to increasingly complex prompts, leading to their popularisation in systems such as ChatGPT (https://openai.com/chatgpt) and their use in a large spectrum of applications. The ability of LLMs to deliver sophisticated linguistic and reasoning behaviour, has pushed their application beyond their intended operational envelope.

While being consistently fluent, LLMs are prone to hallucinations (Shuster et al. 2021), stating factually incorrect statements (Shuster et al. 2022), lacking necessary mechanisms of safety, lacking transparency and control (Tanguy et al. 2016), among many others. Such vulnerabilities and limitations have already led to bad consequences such as suicide case (https://www.vice.com/en/article/pkadgm/man-dies-by-suicide-after-talking-with-ai-chatbot-widow-says), lawyer submitted fabricated cases as precedent to the court (https://www.bbc.co.uk/news/world-us-canada-65735769), leakage of private information (https://www.engadget.com/chatgpt-briefly-went-offline-after-a-bug-revealed-user-chat-histories-115632504.html), etc. Therefore, research is urgently needed to understand the potential vulnerabilities and how the LLMs' behaviour can be assured to be safe and trustable. The goal of this paper is to provide a review of known vulnerabilities and limitations of LLMs and, more importantly, to investigate how the V&V techniques can be adapted to improve the safety and trustworthiness of LLMs. While there are several surveys on LLMs (Zhou et al. 2023; Zhao et al. 2023a), as well as a categorical archive of ChatGPT failures (Borji 2023), to the best of our knowledge, this is the first work that provides a comprehensive discussion on the safety and trustworthiness issues, from the perspective of the V&V.

With the rising of LLMs and its wide applications, the need to ensure their safety and trustworthiness become prominent. Considering the broader subject of deep learning systems, to support their safety and trustworthiness, a diverse set of technical solutions have been developed by different research communities. For example, the machine learning community is focused on adversarial attacks (Goodfellow et al. 2014; Madry et al. 2017; Croce and Hein 2020; Xu et al. 2020a), outlier detectors Pang et al. (2021), adversarial training (Szegedy et al. 2013; Mirman et al. 2018; Wong et al. 2020), and explainable AI (Xu et al. 2019; Gunning et al. 2019; Ribeiro et al. 2016; Zhao et al. 2021a). The human–computer interaction community is focused on engaging the learning systems in the interactions with end users to improve the end users' confidence (Dudley and Kristensson 2018). Formal methods community treats ML models as yet another symbolic system (evidenced by their consideration of neurons, layers, etc.) and adapts existing formal methods tools to work on the new systems (Huang et al. 2020a). While research has been

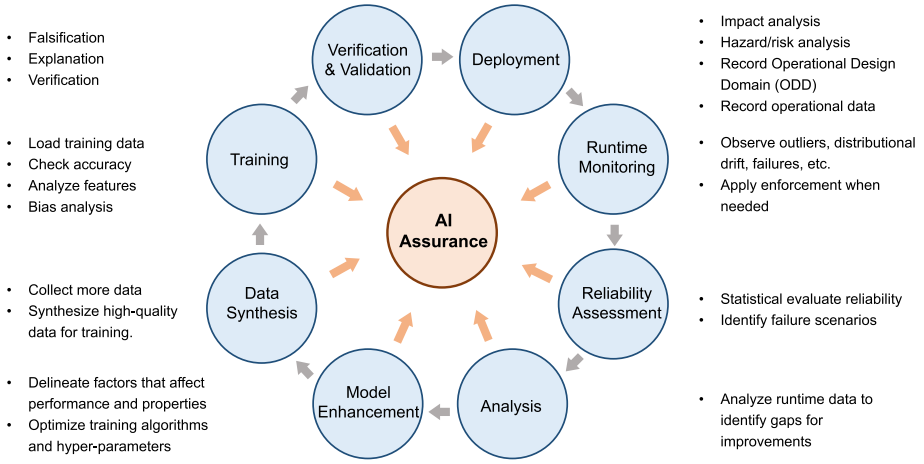

Fig. 1 Summarisation of lifecycle V&V methods to support AI Assurance

intense on these individual methods, a synergy among them has not been addressed. *Without a synergy, it is hard, if not impossible, to rigorously understand the causality between methods and how they might collectively support the safe and trusted autonomous systems at runtime*. This survey is rooted in the field of AI assurance, aiming to apply a collection of rigorous V&V methods throughout the lifecycle of ML models, to provide assurance on the safety and trustworthiness. An illustrative diagram is given in Fig. 1 for general ML models. To begin with, data collection and synthesis is required to obtain as many as possible the training data, including the synthesis of high quality data through e.g., data argumentation or generative models. In the training phase, other than the prediction accuracy, multiple activities are needed, including e.g., the analysis of the learned feature representations and the checking for unintended bias. After the training, we apply offline V&V methods to the ML model, including techniques to falsify, explain, and verify the ML models. During the deployment phase, we must analyse the impact and hazard of the potential application environment. The operational design domain and operational data will be recorded. A run-time monitor is associated to the ML model to detect outliers, distribution shifts, and failures in the application environment. We may further apply reliability assessment methods to evaluate the reliability of the ML model and identify failure scenarios. Based on the detection or assessment results, we can identify the gaps for improvement. Finally, we outline the factors that affect the performance of the ML model, and optimise the training algorithm to obtain an enhanced ML model.

These V&V techniques have been successful in supporting the reliable and dependable development of software and hardware that are applied to safety-critical systems, and have been adapted to work with machine learning models, mainly focusing on the convolutional neural networks for image classification [see surveys such as Huang et al. (2020a), Liu et al. (2021a) and textbooks such as Huang et al. (2012)], but also extended to consider, for example, object detection, deep reinforcement learning, and recurrent neural networks. This paper discusses how to extend further the V&V techniques to deal with the safety and trustworthiness challenges of LLMs.

V&V are independent procedures that are used together for checking that a system (or product, service) meets requirements and specifications and that it fulfills its intended

purpose (https://www.imdrf.org/sites/default/files/docs/ghtf/final/sg3/technical-docs/ghtf-sg3-n99-10-2004-qms-process-guidance-04010.pdf). Among them, verification techniques check the system against a set of design specifications, and validation techniques ensure that the system meets the user's operational needs. From software, convolutional neural networks to LLMs, the scale of the systems grows significantly, which makes the usual V&V techniques less capable due to their computational scalability issues. White-box V&V techniques that take the learnable parameters as their algorithmic input will not work well in practice. Instead, the research should focus on black-box techniques, on which some research has started for convolutional neural networks. In addition, V&V techniques need to consider the *non-deterministic nature* of LLMs (i.e., different outputs for two tests with identical input), which is a noticeable difference with the usual neural networks, such as convolutional neural networks and object detectors, that currently most V&V techniques work on.

Considering the fast development of LLMs, this survey does not intend to be complete (although it includes 370+ references), especially when it comes to the applications of LLMs in various domains, but rather a collection of organised literature reviews and discussions to support the understanding of the safety and trustworthiness issues from the perspective of V&V. Through the survey, we noticed that the current research are focused on identifying the vulnerabilities, with limited efforts on systematic approaches to evaluate and verify the safety and trustworthiness properties.

The structure of the paper is as follows. In Sect. 2, we review the LLMs and its categories, its lifecycle, and several techniques introduced to improve safety and trustworthiness. Then, in Sect. 3, we present a review of existing vulnerabilities. This is followed by a general verification framework in Sect. 4. The framework includes V&V techniques such as falsification and evaluation (Sect. 5), verification (Sect. 6), runtime monitor (Sect. 7), and ethical use (Sect. 8). We conclude the paper in Sect. 10.

2 Large language models

This section summarises the categories of machine learning tasks based on LLMs, followed by a discussion of the lifecycle of LLMs. We will also discuss a few fundamental techniques relevant to the safety analysis.

2.1 Categories of large language models

LLMs have been applied to many tasks, such as text generation (Li et al. 2022) content summary (Zhang et al. 2023a) conversational AI (i.e., chatbots) (Wei et al. 2023) and image synthesis (Koh et al. 2023) Other LLMs applications can be seen as their adaptations or further applications. In the following, we discuss the two most notable categories of LLMs: text-based conversational AI and image synthesis. While they might have slightly different concerns, this survey will be more focused on issues related to the former, without touching some issues that are specific to image synthesis such as the detection of fake images.

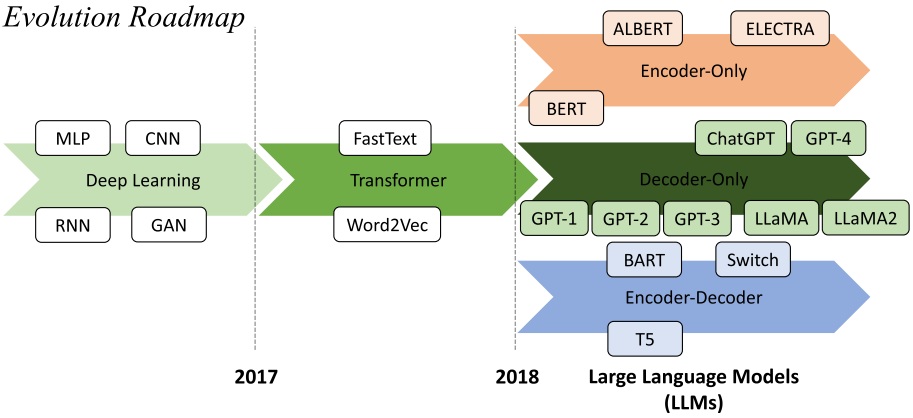

Fig. 2 Large language models: evolution roadmap

2.1.1 Text-based conversational AI

LLMs are designed to understand natural language and generate human-like responses to queries and prompts. Almost all NLP tasks [e.g., language translation (Brants et al. 2007) chatbots (Lin et al. 2019; Gu et al. 2020) and virtual assistants (Tulshan and Dhage 2019)] have witnessed tremendous success with Transformer-based pretrained language models (T-PTLMs), relying on Transformer (Vaswani et al. 2017) self-supervised learning (Jaiswal et al. 2020; Liu et al. 2021b)and transfer learning (Houlsby et al. 2019; Ruder et al. 2019) to process and understand the nuances of human language, including grammar, syntax, and context.

The well-known text-based LLMs include GPT-1 (Radford et al. 2018) BERT (Devlin et al. 2019) XLNet (Yang et al. 2019) RoBERTa (Liu et al. 2019) ELECTRA (Clark et al. 2020) T5 (Raffel et al. 2020) ALBERT (Lan et al. 2019) BART (Lewis et al. 2020) and PEGASUS (Zhang et al. 2020). These models can learn general language representations from large volumes of unlabelled text data through self-supervised learning and subsequently transfer this knowledge to specific tasks, which has been a major factor contributing to their success in NLP (Kalyan et al. 2021). Kaplan et al. (2020) demonstrated that simply increasing the size of T-PTLMs can lead to improved performance (Kalyan et al. 2021). This finding has spurred the development of LLMs such as GPT-3 (Brown et al. 2020c), PANGU (Zeng et al. 2021b), GShard (Lepikhin et al. 2020), Switch-Transformers (Fedus et al. 2021) and GPT-4 (OpenAI 2023).

With the advancement of the Transformer development (Vaswani et al. 2017), significant enhancements were achieved in handling sequential data. Leveraging the Transformer architecture, LLMs have been created as potent models with the capacity to generate text resembling human language. ChatGPT represents a distinct embodiment of an LLM, characterised by its remarkable performance that yields groundbreaking outcomes. The progression of LLMs, depicted in Fig. 2, starts from the evolution of deep learning and transformer-based frameworks, culminating in the latest explosion of LLMs. We divide the LLMs into Encoder-only, Decoder-only, and Encoder–Decoder according to Yang et al. (2023). In Encoder-only and Encoder–Decoder architectures, the model predicts masked words in a sentence while taking into account the surrounding context. While Decoder-only models are trained by generating the subsequent word

in a sequence based on the preceding words. GPT-style language model belongs to the Decoder-only type.

We also note that, there are advanced uses of LLMs (or advanced prompt engineering) by considering e.g., self-consistency (Wang et al. 2023b), knowledge graph (Pan et al. 2023), generating programs as the intermediate reasoning steps (Gao et al. 2023), generating both reasoning traces and task-specific actions in an interleaved manner (Yao et al. 2023), etc.

2.1.2 Text-based image synthesis

The transformer model (Vaswani et al. 2017) has become the standard choice for Language Modelling tasks, but it has also found widespread integration in text-to-image tasks. We present a chronological overview of the advancements in text-to-image research. DALL-E (Ramesh et al. 2021) is a representative approach that leverages Transformers for a text-to-image generation. The methodology involves training a dVAE (Rolfe 2016) and subsequently training a 12B decoder-only sparse transformer supervised by image tokens from the pre-trained dVAE. The transformer generates image tokens solely based on text tokens during inference. The resulting image candidates are evaluated by a pretrained CLIP model (Radford et al. 2017) to produce the final generated image. StableFusion (Rombach et al. 2022) differs from DALL-E (Ramesh et al. 2021) by using a diffusion model instead of a Transformer to generate latent image tokens. To incorporate text input, StableFusion (Rombach et al. 2022) first encodes the text using a transformer then conditions the diffusion model on the resulting text tokens. GLIDE (Nichol et al. 2021) employs a transformer model (Vaswani et al. 2017) to encode the text input and then trains a diffusion model to generate images that are conditioned on the text tokens directly. DALL-E2 (Ramesh et al. 2022) effectively leverages LLMs by following a three-step process. First, a CLIP model is trained using text-image pairs. Next, using text tokens as input, an autoregressive or diffusion model generates image tokens. Finally, based on these image tokens, a diffusion model is trained to produce the final image. Imagen (Saharia et al. 2022) employs a pre-trained text encoder, such as BERT (Devlin et al. 2018) or CLIP (Radford et al. 2017), to encode text. It then uses multiple diffusion models to train a process that generates images that start from low-resolution and gradually progress to high-resolution. Parti (Yu et al. 2022) demonstrates that a VQGAN (Esser et al. 2021) and Transformer architecture can achieve superior image synthesis outcomes compared to previous approaches, even without utilising a diffusion model. The eDiff-I model (Balaji et al. 2022) has recently achieved state-of-the-art performance on the MS-COCO dataset (Lin et al. 2014) by leveraging a combination of CLIP and diffusion models.

In summary, text-to-image research commonly utilises transformer models (Vaswani et al. 2017) for encoding text input and either the diffusion model or the decoder of an autoencoder for generating images from latent text or image tokens.

2.2 Lifecycle of LLMs

Figure 3 illustrates the lifecycle stages and the vulnerabilities of LLMs. This section will focus on the introduction of lifecycle stages, and the detailed discussions about vulnerabilities will appear in Sect. 3. The offline model construction is formed of three steps (Zhao et al. 2023a): pre-training, adaptation tuning, and utilisation improvement, such that each step includes several interleaving sub-steps. In general, the *pre-training* step is similar to

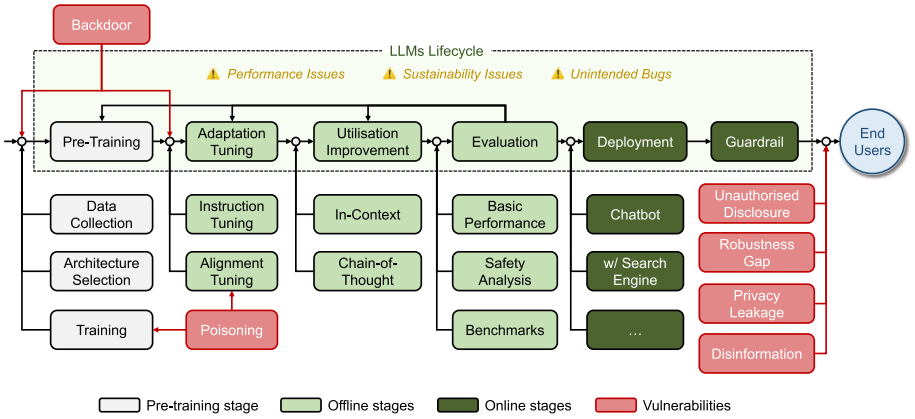

Fig. 3 Large language models: lifecycle and vulnerabilities

the usual machine learning training that goes through data collection, architecture selection, and training. On *adaptation tuning*, it might conduct instruction tuning (Lou et al. 2023) to learn from task instructions, and alignment tuning (Ouyang et al. 2022; Christiano et al. 2017) to make sure LLMs are aligned with human values, e.g., fair, honest, and harmless. Beyond this, to improve the interaction with the end users, *utilisation improvements* may be conducted through, for example, in-context learning (Brown et al. 2020c) and chain-of-thought learning (Wei et al. 2022).

Once an LLM is trained, an *evaluation* is needed to ensure that its performance matches the expectation. Usually, we consider the evaluation from three perspectives: evaluation on basic performance, safety analysis to evaluate the consequence of applying the LLM in an application, and the evaluation through publicly available benchmark datasets. The basic performance evaluation considers several basic types of abilities such as language generation and complex reasoning. Safety analysis is to understand the impacts of human alignment, interaction with external environment, and incorporation of LLMs into broader applications such as search engines. On top of these, benchmark datasets and publicly available tools are used as well to support the evaluation. The evaluation will determine if the LLM is acceptable (for pre-specified criteria), and if so, the process will move forward to the deployment stage. Otherwise, at least one failure will be identified, and the process will move back to either of the three training steps.

On the *deployment* stage, we will determine how the LLM will be used. For example, it could be available in a web platform for direct interaction with end users, such as the ChatGPT.[1] Alternatively, it may be embedded into a search engine, such as the new Bing.[2] Nevertheless, according to the common practice, a *guardrail* is imposed on the conversations between LLMs and end users to ensure that the AI regulation is maximally implemented.

In Fig. 2, within the LLMs lifecycle, three main issues run through: performance issues, sustainability issues, and unintended bugs. These may be caused by one or more stages in the lifecycle. The red block shows that vulnerabilities appear in the LLMs lifecycle, and they may appear in the early stage of the whole period. For example, backdoor attacks and

[1] https://openai.com/blog/ChatGPT.
[2] https://www.bing.com/new.

poisoning can contaminate raw data. When LLMs are deployed, problems such as a robustness gap may also arise.

2.3 Key techniques relevant to safety and trustworthiness

In the following, we discuss two fundamental techniques that are distinct from the usual deep learning models and have been used by e.g., ChatGPT to improve safety and trustworthiness: reinforcement learning from human feedback and guardrails.

2.3.1 Reinforcement learning from human feedback (RLHF)

RLHF can be conducted in any stage of the "Adaptation Tuning", "Utilisation Improvement", or "Evaluation" in the framework of Fig. 3. RLHF (Christiano et al. 2017; Ouyang et al. 2022; Bai et al. 2022a, b; OpenAI 2023; Lambert et al. 2022; Ziegler et al. 2019) plays a crucial role in the training of language models, as it allows the model to learn from human guidance and avoid generating harmful content. In essence, RLHF assists in aligning language models with safety considerations through fine-tuning with human feedback. OpenAI initially introduced the concept of incorporating human feedback to tackle complex reinforcement learning tasks in Christiano et al. (2017), which subsequently facilitated the development of more sophisticated LLMs, from InstructGPT (Ouyang et al. 2022) to GPT4 (OpenAI 2023). According to InstructGPT (Ouyang et al. 2022), the RLHF training process typically begins by learning a reward function intended to reflect what humans value in the task, utilising human feedback on the model's outputs. Subsequently, the language model is optimised via an RL algorithm, such as PPO (Schulman et al. 2017), using the learned reward function. Reward model training and fine-tuning with RL can be iterated continuously. More comparison data is collected on the current best policy, which is used to train a new reward model and policy. The InstructGPT models demonstrated enhancements in truthfulness and reductions in generating toxic outputs while maintaining minimal performance regressions on public NLP datasets.

Following InstructGPT, Red Teaming language models (Bai et al. 2022a) introduces a harmlessness preference model to help RLHF to get less harmful agents. The comparison data from red team attacks is used as the training data to develop the harmlessness preference model. The authors of Bai et al. (2022a) utilised the helpful and harmless datasets in preference modelling and RLHF to fine-tune LLMs. They discovered that there was a significant tension between helpfulness and harmlessness. Experiments showed helpfulness and harmlessness model is significantly more harmless than the model trained only on helpfulness data. They also found that alignment with RLHF has many benefits and no cost to performance, like combining alignment training with programming ability and summarisation. The authors of Ganguli et al. (2023) found that LLMs trained with RLHF have the capability for moral self-correction. They believe that the models can learn intricate normative concepts such as stereotyping, bias, and discrimination that pertain to harm. Constitutional AI (Bai et al. 2022b) trains the preference model by relying solely on AI feedback, without requiring human labels to identify harmful outputs. To push the process of aligning LLMs with RLHF, an open-sourced modular library, RL4LMs, and evaluation benchmark, GRUE, designed for optimising language generator with RL are introduced in Ramamurthy et al. (2022). Inspired by the success of RLHF in language-related domains, fine-tuning approaches that utilise human feedback to improve text-to-image models (Lee et al. 2023; Xu et al. 2023; Wu et al. 2023c) have gained popularity as well. To achieve

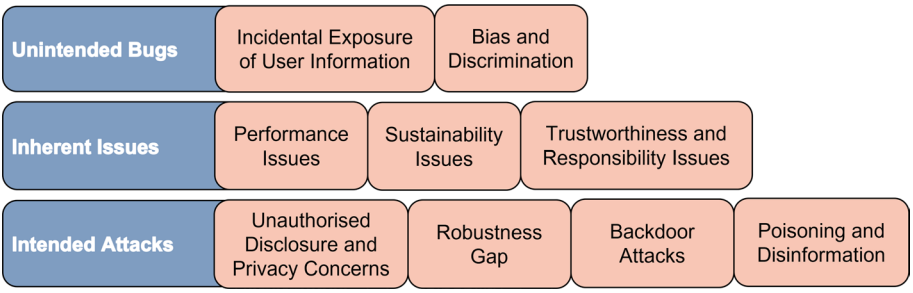

Fig. 4 Taxonomy of vulnerabilities

human–robot coexistence, the authors of Gu et al. (2023b) proposed a human-centred robot RL framework consisting of safe exploration, safety value alignment, and safe collaboration. They discussed the importance of interactive behaviours and four potential challenges within human–robot interactive procedures. Although many works indicate that RLHF could decrease the toxicity of generations from LLMs, the induced RLHF, like introducing malicious examples by annotators (Carlini et al. 2023), may cause catastrophic performance and risks. We hope better techniques that lead to transparency, safe and trustworthy RLHF will be developed in the coming future.

2.3.2 Guardrails

Considering that some LLMs are interacting directly with end-users, it is necessary to put a layer of protection, called guardrail, when the end users ask for information about violence, profanity, criminal behaviours, race, or other unsavoury topics. Guardrails are deployed in most, if not all, LLMs, including ChatGPT, Claude, and LLaMA. In such cases, a response is provided with the LLM refusing to provide information. While this is a very thin layer of protection because there are many tricks (such as prompt injections that will be reviewed in Sect. 5.1) to circumvent it, it enhances the social responsibility of LLMs.

3 Vulnerabilities, attacks, and limitations

This section presents a review of the known types of vulnerabilities. The vulnerabilities can be categorised into inherent issues, intended attacks, and unintended bugs, as illustrated in Fig. 4.

Inherent issues are vulnerabilities that cannot be readily solved by the LLMs themselves. However, they can be gradually improved with, e.g., more data and novel training methods. Inherent issues include performance weaknesses, which are those aspects that LLMs have not reached the human-level intelligence, and sustainability issues, which are because the size of LLMs is significantly larger than the usual machine learning models. Their training and daily execution can have non-negligible sustainability implications. Moreover, trustworthiness and responsibility issues are inherent to the LLMs.

Attacks are initiated by malicious attackers, which attempt to implement their goals by attacking certain stages in the LLMs lifecycle. Known intended attacks include

Table 1 Performance error exists across different LLMs

LLMs	Output for question: "Adam's wife is Eve. Adam's daughter is Alice. Who is Alice to Eve?"
ChatGPT (https://openai.com/chatgpt)	Alice is Eve's granddaughter
ERNIE Bot (Yang 2023)	Alice is Eve's granddaughter
Llama2 (Touvron et al. 2023)	Alice is Eve's granddaughter
Bing Chat (Mehdi 2023)	Alice is Adam's daughter and Eve's granddaughter
GPT-4 (OpenAI 2023)	Alice is Eve's daughter

Retrieved 24 August 2023

robustness gap, backdoor attack, poisoning, disinformation, privacy leakage, and unauthorised disclosure of information.

Finally, with the integration of LLMs into broader applications, there will be more and more *unintended bugs* that are made by the developers unconsciously but have serious consequences, such as bias and discrimination (that are usually related to the quality of training data), and the recently reported incidental exposure of user information. We separate these from inherent issues, because they could be resolved with e.g., high quality training data, carefully designed API, and so on. They are "unintended", because they are not deliberately designed by the developers.

Figure 3 suggests how the vulnerabilities may be exploited in the lifecycle of LLMs. While inherent issues and unintended bugs may appear in any stage of the lifecycle, the attacks usually appear in particular stages of the lifecycle. For example, a backdoor attack usually occurs in pre-training and adaptation tuning, in which the backdoor trigger is embedded, and poisoning usually happens in training or alignment tuning, when the LLMs acquires information/data from the environment. Besides, many attacks occur upon the interaction between end users and the LLMs using specific, well-designed prompts to retrieve information from the LLMs. We remark that, while there are overlapping, LLMs and usual deep learning models (such as convolutional neural networks or object detectors) have slightly different vulnerabilities, and while initiatives have been taken on developing specification languages for usual deep learning models (Bensalem et al. 2022; Huang et al. 2022a), such efforts may need to be extended to LLMs.

3.1 Inherent issues

3.1.1 Performance issues

Unlike traditional software systems, which run according to the rules that can be deterministically verified, neural network-based deep learning systems, including large-scale LLMs, have their behaviours determined by the complex models learned from data through optimisation algorithms. It is unlikely that an LLM performs 100% correctly. As a simple example shown in Table 1, it can be observed that similar errors exist across different LLMs, where most of the existing LLMs are not able to provide a correct answer. Performance issues related to the correctness of the outputs include at least the following two categories: factual errors and reasoning errors.

3.1.1.1 Factual errors Factual errors refer to situations where the output of an LLM contradicts the truth, where some literature refers this situation as *hallucination* (OpenAI 2023; Zhao et al. 2023a; Li et al. 2023a). For example, when asked to provide information about the expertise in the computer science department at the University of Liverpool, the ChatGPT refers to people who were never affiliated with the department. Hence more serious errors can be generated, including notably wrong medical advice. Additionally, it is interesting to note that while LLMs can perform across different domains, their reliability may vary across domains. For example, the authors of Shen et al. (2023) show that ChatGPT significantly under-performs in law and science questions. Investigating if this is related to the training dataset or training mechanism will be interesting.

3.1.1.2 Reasoning errors It has been discovered that, when given calculation or logic reasoning questions, ChatGPT may not always provide correct answers. This is mainly because, instead of actual reasoning, LLMs fit the questions with prior experience learned from the training data. If the statements of the questions are close to those in the training data, it will give correct answers with a higher probability. Otherwise, with carefully crafted prompt sequence, wrong answers can be witnessed (Liu et al. 2023b; Frieder et al. 2023).

3.1.2 Sustainability issues

Sustainability issues, which are measured with, e.g., economic cost, energy consumption, and carbon dioxide emission, are also inherent to the LLMs. While excellent performance, LLMs require high costs and consumption in all the activities in its lifecycle. Notably, ChatGPT was trained with 30k A100 GPUs (each one is priced at around $10k), and every month's energy consumption cost at around $1.5M.

In Table 2, we summarise the hardware costs and energy consumption from the literature for a set of LLMs with varied parameter sizes and training dataset sizes. Moreover, the carbon dioxide emission can be estimated with the following formula:

$$tCO_2eq = 0.385 \times GPU_h \times (GPUpower\ consumption) \times PUE \quad (1)$$

where GPU_h is the GPU hours, GPU power consumption is the energy consumption as provided in Table 1, and PUE is the Power Usage Effectiveness (commonly set as a constant 1.1). Precisely, it has been estimated that training a GPT-3 model consumed 1287 MWh, which emitted 552 ($= 1287 \times 0.385 \times 1.114$) tons of CO_2 (Patterson et al. 2022).

In the realm of technological advancements, the energy implications of various innovations have become a focal point of discussion. Consider the energy footprint of training large language models (LLMs) like GPT-4. The energy required to train such a model ranges between 51,772 and 62,318 MWh (https://towardsdatascience.com/the-carbon-footprint-of-gpt-4-d6c676eb21ae, https://archive.md/2RQ8X). To put this into perspective, this is roughly 0.05% of Bitcoin's energy consumption in 2021, which was estimated at a staggering 108 TWh (De Vries et al. 2022, https://digiconomist.net/bitcoin-energy-consumption). Two remarks on this comparison: (i) the energy cost of training LLMs is minuscule when juxtaposed with the colossal energy demands of other technologies such as cryptocurrency mining, and (ii) the energy consumption of LLMs is primarily associated with their training phases (one-time cost), whereas their inference is considerably more energy-efficient. In contrast, cryptocurrency mining consumes energy both in the creation of new coins and the validation of transactions, continuously, as long as the network is

Table 2 Costs of different large language models

Model	Parameter size (billions)	Dataset size[a]	Hardware	Energy
BERT-base (Devlin et al. 2018)	0.11	3.3B words	16 TPU chips	–
BERT-large (Devlin et al. 2018)	0.34	3.3B words	64 TPU chips	–
GPT-3 (Brown et al. 2020b)	175	499B tokens	10,000 NVIDIA V100	1287 MWh
Megatron Turing NLG (Smith et al. 2022)	530	338.6B tokens	4480 NVIDIA A100-80GB	> 900 MWh
ERNIE 3.0 (Sun et al. 2021)	260	4 TB/375B tokens	384 NVIDIA V100 GPU	–
GLaM (Du et al. 2022)	1200	1.6T tokens	1024 Cloud TPU-V4	456 MWh
Gopher (Rae et al. 2021)	280	300B tokens	4096 TPUv3	1066 MWh
PanGu-α (Zeng et al. 2021b)	200	1.1 TB/258.5B tokens	2048 Ascend 910 AI processors	–
LaMDA (Thoppilan et al. 2022)	137	1.56 TB/2.81T tokens	1024 TPU-v3	451 MWh
GPT-NeoX (Black et al. 2022)	20	825 GB	96 NVIDIA A100-SXM4-40GB	43.92 MWh
Chinchilla (Hoffmann et al. 2022)	70	1.4T tokens	TPUv3/TPUv4	–
PaLM (Chowdhery et al. 2022)	540	780B tokens	6144 TPU v4	~ 640 MWh
OPT (Zhang et al. 2022)	175	180B tokens	992 NVIDIA A100-80GB	324 MWh
YaLM (https://github.com/yandex/YaLM-100B)	100	1.7 TB/300B tokens	800 NVIDIA A100	~ 785 MWh
BLOOM (Scao et al. 2022)	176	1.61 TB/350B tokens	384 NVIDIA A100 80 GB	433 MWh
Galactica (Taylor et al. 2022)	120	450B tokens	128 NVIDIA A100 80GB	–
AlexaTM (Soltan et al. 2022)	20	1T tokens	128 NVIDIA A100	~ 232 MWh
LLaMA (Touvron et al. 2023)	65	1.4T tokens	2048 NVIDIA A100-80GB	449 MWh
GPT-4 (Katz et al. 2023; E2Analyst 2023, https://towardsdatascience.com/the-carbon-footprint-of-gpt-4-d6c676eb21ae, https://archive.md/2RQ8X)	1800	1 PB/13T tokens	~ 25000 NVIDIA A100	~ 51772–62318 MWh
Cerebras-GPT (Dey 2023)	13	260B tokens	16 Cerebras CS-2	–
BloombergGPT (Wu et al. 2023a)	50.6	569B tokens	512 NVIDIA A100 40GB	~ 325 MWh
PanGu-Σ (Ren et al. 2023)	1085	329B tokens	512 Ascend 910 accelerators	–

[a]A "word" is a single distinct meaningful element of a sentence, e.g. "Hello world" has two words. A "token" can represent a whole word or a part of a word, depending on the tokenisation strategy used, e.g. "I'm fine" can be tokenised into three tokens: "I", "m" and "fine". File size (TB or GB) refers to the amount of storage space required to save the training data

active. This continuous energy drain underscores the vast difference in the sustainability profiles of these two technologies.

3.1.3 Other inherent trustworthiness and responsibility issues

Some issues occur during the lifecycle that could lead to concerns about the trustworthiness and responsibilities of LLMs. Generally, these can be grouped into two sub-classes concerning the training data and the final model.

For the training data, there are issues around the copyright (Kim 2023), quality, and privacy of the training data. There is a significant difference between LLMs and other ML models regarding the data being used for training. In the latter case, specific (well-known/-structured) datasets are usually used in the training process. Ideally, these datasets are properly pre-processed, anonymised, etc.; if needed, users have also given consent about using of their data. It is well known that ChatGPT crawls the internet and uses the gathered data to train. On the other hand, for LLMs, the data used for training needs to be more understood. In most cases, users have not provided any consent; most likely they are even unaware that their data contain personal information and that their data have been crawled and used in LLM training. This makes ChatGPT, and LLMs in general, privacy-nightmare to deal with and opens the door to many privacy leakage attacks. Even the model owners would need to determine the extent of private risk their model could pose.

For the final model, significant concerns include, e.g., LLMs' capability of independent and conscious thinking (Hintze 2023), LLMs' ability to be used to mimic human output including academic works (Lee 2023), use of LLMs to engage scammers in automatised and pointless communications for wasting time and resources (Cambiaso and Caviglione 2023), use of LLMs in generating malware (Goodin 2023; News 2023; Botacin 2023), etc. Similar issues can also be seen in image synthesis tools such as DALL-2, where inaccuracies, misleading information, unanticipated features, and reproducibility have been witnessed when generating maps in cartography (Kang et al. 2023b). These call for not only the transparency of LLMs development but also the novel technologies to verify and differentiate the real and LLMs' works (Uchendu et al. 2023; Mitrović et al. 2023). The latter is becoming a hot research topic with many (practical) initiatives such as https://originality.ai, https://contentatscale.ai/ai-content-detector/, https://copyleaks.com/ai-content-detector whose effectiveness requires in-depth study (Pegoraro et al. 2023). These issues inherent to the LLMs, as they are neither attacks nor unintended bugs.

3.2 Attacks

3.2.1 Unauthorised disclosure and privacy concerns

For LLMs, it is known that by utilising, e.g., prompt injection (Perez and Ribeiro 2022) or prompt leaking Prompt injection attacks against GPT-3 (2022) (which will be discussed in Sect. 5.1), it is possible to disclose the sensitive information of LLMs. For example, with a simple conversation (Prompt leaking 2023), the new Bing leaks its codename "Sydney" and enables the users to retrieve the prompt without proper authentication.

More importantly, privacy concerns also become a major issue for LLMs. First, privacy attacks on convolutional neural networks, such as membership inference attacks where the attacker can determine whether an input instance is in the training dataset, have been adapted to work on diffusion models (Duan et al. 2023). Second, an LLM may store

the conversations with the users, which already leads to concerns about privacy leakage because users' conversations may include sensitive information (https://www.engadget.com/three-samsung-employees-reportedly-leaked-sensitive-data-to-chatgpt-190221114.html). ChatGPT has mentioned in its privacy policy that the conversations will be used for training unless the users explicitly opt out. Due to such concerns, Italy has reportedly banned ChatGPT (https://www.cnbc.com/2023/04/04/italy-has-banned-chatgpt-heres-what-other-countries-are-doing.html) in early 2023. Most recently, both articles (Li et al. 2023b; Greshake et al. 2023) illustrate that augmenting LLMs with retrieval and API calling capabilities (so-called Application-Integrated LLMs) may induce even more severe privacy threats than ever before.

3.2.2 Robustness gaps

An adversarial attack is an intentional effort to undermine the functionality of a DNN by injecting distorted inputs that lead to the model's failure. Multiple input perturbations are proposed in NLP for adversarial attacks (Ren et al. 2019a; Goyal et al. 2022), which can occur at the character, word, or sentence level (Cheng et al. 2019a; Iyyer et al. 2018; Cao et al. 2022). These perturbations may involve deletion, insertion, swapping, flipping, substitution with synonyms, concatenation with characters or words, or insertion of numeric or alphanumeric characters (Liang et al. 2017; Ebrahimi et al. 2017; Lei et al. 2022). For instance, in character level adversarial attacks, Belinkov et al. (2017) introduces natural and synthetic noise to input data, while Gao et al. (2018), Li et al. (2018a) identifies crucial words within a sentence and perturbs them accordingly. Moreover, Hosseini et al. (2017) demonstrates that inserting additional periods or spaces between words can result in lower toxicity scores for the perturbed words, as observed with the "Perspective" API developed by Google. For word level adversarial attacks, they can be categorised into gradient-based (Liang et al. 2017; Samanta and Mehta 2017), importance-based (Ivankay et al. 2022; Jin et al. 2020), and replacement-based (Alzantot et al. 2018; Kuleshov et al. 2018; Pennington et al. 2014) strategies based on the perturbation method employed. In addition, in sentence level adversarial attacks, some attacks (Jia et al. 2017; Wang and Bansal 2018) are created so that they do not impact the original label of the input and can be incorporated as a concatenation in the original text. In such scenarios, the expected behaviour from the model is to maintain the original output, and the attack can be deemed successful if the label/output of the model is altered. Another approach (Zhao et al. 2017) involves generating sentence-level adversaries using Generative Adversarial Networks (GANs) (Goodfellow et al. 2020), which produce outputs that are both grammatically correct and semantically similar to the input text.

As mentioned above, the robustness of small language models has been widely studied. However, given the increasing popularity of LLMs in various applications, evaluating their robustness has become paramount. For example, Shen et al. (2023) suggests that ChatGPT is vulnerable to adversarial examples, including the single-character change. Moreover, Wang et al. (2023a) extensively evaluates the adversarial robustness of ChatGPT in natural language understanding tasks using the adversarial datasets AdvGLUE (Wang et al. 2021b) and ANLI (Nie et al. 2019). The results indicate that ChatGPT surpasses all other models in all adversarial classification tasks. However, despite its impressive performance, there is still ample room for improvement, as its absolute performance is far from perfection. In addition, when evaluating translation robustness, Jiao et al. (2023) finds ChatGPT does not perform as well as the commercial systems on translating biomedical abstracts or Reddit

comments but exhibits good results on spoken language translation. Moreover, Chen et al. (2023c) finds that the ability of ChatGPT to provide reliable and robust cancer treatment recommendations falls short when compared to the guidelines set forth by the National Comprehensive Cancer Network (NCCN). ChatGPT is a strong language model, but there is still some space for robustness improvement, especially in certain areas.

3.2.3 Backdoor attacks

The goal of a backdoor attack is to inject malicious knowledge into the LLMs through either the training of poisoning data (Chen et al. 2021b; Shen et al. 2021b; Dai et al. 2019) or modification of model parameters (Kurita et al. 2020; Yang et al. 2021b). Such injections should not compromise the model performance and must be bypassed from the human inspection. The backdoor will be activated only when input prompts to LLMs contain the trigger, and the compromised LLMs will behave maliciously as the attacker expected. Backdoor attack on DL models is firstly introduced on image classification tasks (Gu et al. 2019), in which the attacker can use a patch/watermark as a trigger and train a backdoored model from scratch. However, LLMs are developed for NLP tasks, and the approach of pre-training followed by fine-tuning has become a prevalent method for constructing LLMs. This entails pre-training the models on vast unannotated text corpora and fine-tuning them for particular downstream applications. To consider the above characteristics of LLMs, the design of the backdoor trigger is no longer a patch/watermark but a character, word or sentence. In addition, due to the training cost of LLMs, a backdoor attack should consider a direct embedding of the backdoor into pre-trained models, rather than relying on retraining. Finally, the backdoor is not merely expressed to tie with a specific label due to the diversity of downstream NLP applications.

3.2.3.1 Design of backdoor trigger Three categories of triggers are utilised to execute the backdoor attack: BadChar (triggers at the character level), BadWord (triggers at the word level), and BadSentence (triggers at the sentence level), with each consisting of basic (non-semantic) and semantic-preserving patterns (Chen et al. 2021b). The BadChar triggers are produced by modifying the spelling of words in various positions within the input and applying steganography techniques to ensure their invisibility. The BadWord triggers involve selecting a word from the ML model's dictionary, and increasing their adaptability to different inputs. MixUp-based and Thesaurus-based triggers are then proposed (Chen et al. 2021b). The BadSentence triggers are generated by inserting or substituting sub-sentences, with a fixed sentence chosen as the trigger. To preserve the original content, Syntax-transfer (Chen et al. 2021b) is employed to alter the underlying grammatical rules. These three types of triggers allow the flexibility to tailor their attacks to different applications.

Two new concealed backdoor attacks are introduced: the homograph and dynamic sentence attacks (Struppek et al. 2022). The homograph attack uses a character-level trigger that employs visual spoofing homographs, effectively deceiving human inspectors. However, for NLP systems that do not support Unicode homographs, the dynamic sentence backdoor attack is proposed (Struppek et al. 2022), which employs language models to generate highly natural and fluent sentences to act as the backdoor trigger.

3.2.3.2 Backdoor embedding strategies Shen et al. (2021b) is the first to propose a backdoor attack on pre-trained NLP models that do not require task-specific labels. Specifically, they select a target token from the pre-trained model and define a target predefined output

representation (POR) for it. They then insert triggers into the clean text to generate the poisoned text data. While mapping the triggers to the PORs using the poisoned text data, they simultaneously use the clean pre-trained model as a reference, ensuring that the backdoor target model maintains the normal usability of other token representations. After injecting the backdoor, all auxiliary structures are removed, resulting in a backdoor model indistinguishable from a normal one in terms of model architecture and outputs for clean inputs.

A method called Restricted Inner Product Poison Learning (RIPPLe) (Kurita et al. 2020) is introduced to optimise the backdoor objective function in the presence of fine-tuning dataset. They also propose an extension called Embedding Surgery to improve the backdoor's resilience to fine-tuning by replacing the embeddings of trigger keywords with a new embedding associated with the target class. The authors validate their approach on several datasets and demonstrate that pre-trained models can be poisoned even after fine-tuning on a clean dataset.

3.2.3.3 Expression of backdoor In contrast to prior works that concentrate on backdoor attacks in text classification tasks, the applicability of backdoor attacks is investigated in more complex downstream NLP tasks such as toxic comment detection, Neural Machine Translation (NMT), and Question Answer (QA) (Li et al. 2021a). By replicating thoughtfully designed questions, users may receive a harmful response, such as phishing or toxic content. In particular, a backdoored system can disregard toxic comments by employing well-crafted triggers. Moreover, backdoored NMT systems can be exploited by attackers to direct users towards unsafe actions such as redirection to phishing pages. Additionally, Transformer-based QA systems, which aid in more efficient information retrieval, can be susceptible to backdoor attacks.

Considering the prevalence of LLMs in automatic code suggestion (i.e., GitHub Copilot), the data poisoning based backdoor attack, called TROJANPUZZLE, is studied for code-suggestion models (Aghakhani et al. 2023). TROJANPUZZLE produces poisoning data that appears less suspicious by ensuring that certain potentially suspicious parts of the payload are never present in the poisoned data. However, the induced model still proposes the full payload when it completes code, especially outside of docstrings. This characteristic makes TROJANPUZZLE resilient to dataset cleaning techniques that rely on signatures to spot and remove suspicious patterns from the training data.

The backdoor attack on LLMs for text-based image synthesis tasks is firstly introduced in Struppek et al. (2022). The authors employ a teacher–student approach to integrate the backdoor into the pre-trained text encoder and demonstrate that when the input prompt contains the backdoor trigger, e.g., the underlined Latin characters are replaced with the Cyrillic trigger characters, the generation of images will follow a specific description or include certain attributes.

3.2.4 Poisoning and disinformation

Among various adversarial attacks against DNNs, poisoning attack is one of the most significant and rising security concerns for technologies that rely on data, particularly for models trained by enormous amounts of data acquired from diverse sources. Poisoning attacks attempt to manipulate some of the training data, which might lead the model to generate wrong or biased outputs. As LLM are often fine-tuned based on publicly accessible data (Chen et al. 2021a; Brown et al. 2020a), which are from unreliable

and un-trusted documents or websites, the attacker can easily inject some adversaries into the training set of the victim model. Microsoft released a chatbot called Tay on Twitter (Lee 2016). Still it was forced to suspend activity after just one day because it was attacked by being taught to express racist and hateful rhetoric. Gmail's spam filter can be affected by simply injecting corrupted data in the training mails set (Bursztein 2018). Consequently, some evil chatbots might be designed to simulate people to spread disinformation or manipulate people, resulting in a critical need to evaluate the robustness of LLMs against data poisoning.

Nelson et al. (2008) demonstrates how the poisoning attack can render the spam filter useless. By interfering with the training process, even if only 1% of the training dataset is manipulated, the spam filter might be ineffective. The authors propose two attack methods, one is an indiscriminate attack, and another is a targeted attack. The indiscriminate attack sends spam emails that contain words commonly used in legitimate messages to the victim, to force the victim to see more spam and more likely to mark a legitimate email as spam. As for the target attack, the attacker will send training emails containing words likely to be seen in the target email.

With the increasing popularity of developing LLMs, researchers are becoming concerned about using chatbots to spread information. Since these LLMs, such as ChatGPT, MidJourney, and Stable Diffusion, are trained on a vast amount of data collected from the internet, monitoring the quality of data sources is challenging. A recent study Carlini et al. (2023) introduced two poisoning attacks on various popular datasets acquired from websites. The first attack involves manipulating the data viewed by the customer who downloads the data to train the model. This takes advantage of the fact that the data observed by the dataset administrator during collection can differ from the data retrieved by the end user. Therefore, an attacker only needs to purchase a few domain names to gain control of a small portion of the data in the overall data collection. Another attack involves modifying datasets containing periodic snapshots, such as Wikipedia. The attacker can manipulate Wikipedia articles before they are included in the snapshot, resulting in the internet storing perturbed documents. Thus, a significant level of uncertainty and risk is involved when people use these LLMs as search engines.

3.3 Unintended bugs

3.3.1 Incidental exposure of user information

In addition to the above attacks that an attacker actively initiates, ChatGPT was reported (https://www.engadget.com/chatgpt-briefly-went-offline-after-a-bug-revealed-user-chat-histories-115632504.html) to have a "chat history" bug that enabled the users to see from their ChatGPT sidebars previous chat histories from other users, and OpenAI recognised that this chat history bug may have also potentially revealed personal data from the paid ChatGPT Plus subscribers. According to the official report from OpenAI (https://openai.com/blog/march-20-chatgpt-outage), the same bug may have caused inadvertent disclosure of payment-related information for 1.2% of ChatGPT Plus subscribers. The bug was detected within the open-source Redis client library, redis-py. This cannot be an isolated incident, and we are expecting to witness more such "bugs" that could have severe security and privacy implications.

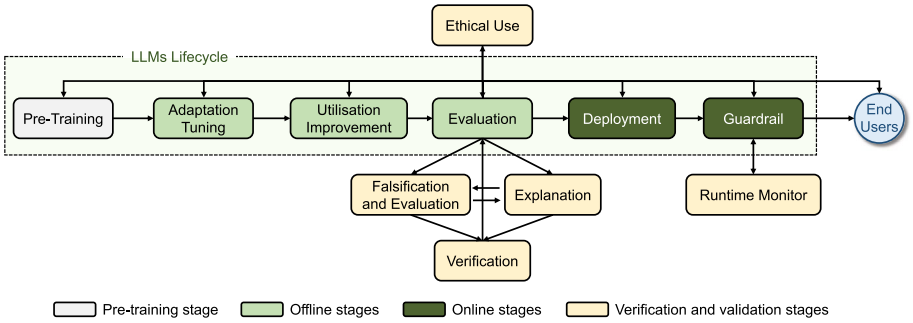

Fig. 5 Large language models: verification framework in lifecycle

3.3.2 Bias and discrimination

Similar to the usual machine learning algorithms, LLMs are trained from data, which may include bias and discrimination. If not amplified, such vulnerabilities will be inherited by the LLMs. For example, Galactica, an LLM similar to ChatGPT trained on 46 million text examples, was shut down by Meta after 3 days because it spewed false and racist information (https://arstechnica.com/information-technology/2022/11/after-controversy-meta-pulls-demo-of-ai-model-that-writes-scientific-papers/). A political compass test (Rutinowski et al. 2023) reveals that ChatGPT is biased towards progressive and libertarian views. In addition, ChatGPT has a self-perception (Rutinowski et al. 2023) of seeing itself as having the Myers–Briggs personality type ENFJ.

4 General verification framework

Figure 5 provides an illustration of the general verification framework that might work with LLMs, by positioning the few categories of V&V techniques onto the lifecycle. In the Evaluation stage, other than the activities that are currently conducted (as mentioned in Fig. 3), we need to start with the *falsification and evaluation* techniques, in parallel with the *explanation* techniques. Falsification and evaluation techniques provide diverse, yet non-exhaustive, methods to find failure cases and have a statistical understanding about potential failures. Explanation techniques are to provide human-understandable explanations to the output of a LLMs. While these two categories are in parallel, they can interact, e.g., a failure case may require an explanation technique to understand the root cause, and the explanation needs to differentiate between different failure and non-failure cases. The *verification* techniques, which are usually high cost, may be only required when the LLMs pass the first two categories.

Finally, ethical principles and AI regulations are imposed throughout the lifecycle to ensure the *ethical use* of LLMs.

Figure 6 presents the taxonomy of verification and validation techniques we surveyed in this paper that can be used for large language models. In the following sections, we will review these techniques in greater details.

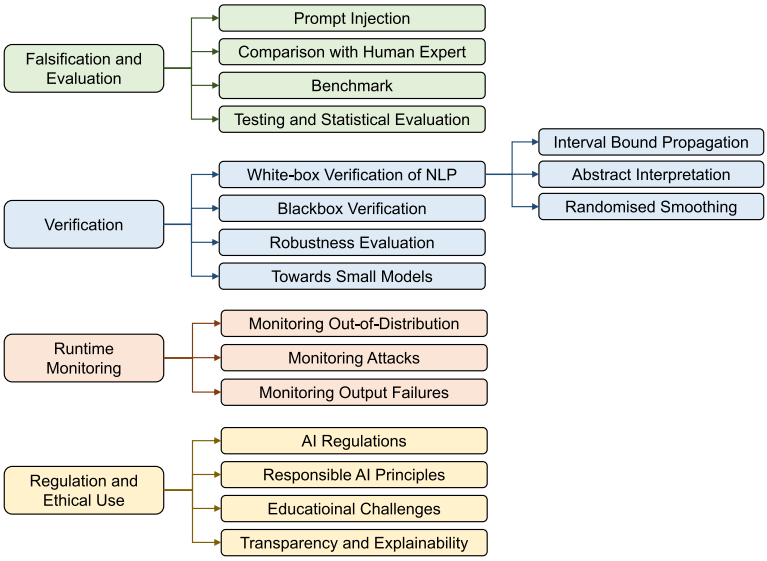

Fig. 6 Taxonomy of surveyed verification and validation techniques for large language models

5 Falsification and evaluation

This section summarises the known methods for identifying and evaluating the vulnerabilities of LLM-based machine learning applications. Falsification and evaluation requires a red team (Bullwinkle and Urban 2023), which, instead of having annotators label pre-existing texts, interacts with a model and actively finds examples that fail. The red team needs to be consist of people of diverse backgrounds and concerning about different risks (benign vs. malicious). We also discuss on how the methods can, and should, be adapted.

5.1 Prompt injection

This section discusses using prompts to direct LLMs to generate outputs that do not align with human values. This includes the generation of malware, violence instruction, and so on. Conditional misdirection has been successfully applied which misdirects the AI by creating a situation where a certain event needs to occur to avoid violence.

Prompt injection for LLMs is not vastly distinct from other injection attacks commonly observed in information security. It arises from the concatenation of instructions and data, rendering it arduous for the underlying engine to distinguish them. Consequently, attackers can incorporate instructions into the data fields they manage and compel the engine to carry out unforeseen actions. Within this comprehensive definition of injection attacks, prompt engineering work can be regarded as instructions (analogous to a SQL query, for instance). At the same time, the input information provided can be deemed as data.

Several methods for mis-aligning LLMs via Prompt Injection (PI) attacks have been successfully applied (https://github.com/dair-ai/Prompt-Engineering-Guide/tree/main/guides). In these attacks, the adversary can prompt the LLM to generate malicious

content or override the initial instructions and the filtering mechanisms. Recent studies have demonstrated that these attacks are difficult to mitigate since current state-of-the-art LLMs are programmed to follow instructions. Therefore, most attacks were based on the assumption that the adversary can directly inject prompt to the LLMs. For example, Perez and Ribeiro (2022) reveals two kinds of threats by manipulating the prompts. The first one is *goal hijacking*, aiming to divert the intended goal of the original prompts towards a target goal, while *prompt leaking* endeavours to retrieve information from private prompts.

Kang et al. (2023a) explores the programmatic behaviour of LLMs, demonstrating that classical security attacks such as obfuscation, code injection, and virtualisation can be used to circumvent the defence mechanisms of LLMs. This further exhibits that instruction-based LLMs can be misguided to generate natural and convincing personalised malicious content by leveraging unnatural prompts.

Moreover, Deshpande et al. (2023) suggests that by assigning ChatGPT a persona, say that of the boxer Muhammad Ali (with a prompt "Speak like Muhammad Ali."), the toxicity of generations can be significantly increased. Maus et al. (2023) develops a black-box framework for producing adversarial prompts for unstructured image and text generation. Employing a token space projection operator provides a solution from mapping the continuous word embedding space into the discrete token space, such that some black-box attacks method, like square attacks, can be applied to explore adversarial prompts. Experimental results found that those adversarial prompts encourage positive sentiments or increase the frequency of the targeted letter in the generated text. Wolf et al. (2023) also suggests the existence of a fundamental limitation on mitigating such prompt injection to trigger undesirable behaviour, i.e., as long as the length of the prompts can be increased, the behaviour has a positive probability to be exhibited.

Li et al. (2023b) claims that in the previous versions of ChatGPT, some personal private information could be successfully extracted via direct prompting. However, with the improved guardrails, some behaviours have been well-protected in the March 2023 version of ChatGPT, where ChatGPT is aware of leaking privacy when direct prompts are applied, it will tend to refuse to provide the answer that may contain private information. Although some efforts have been conducted to prevent training data extraction attacks with direct prompts, Li et al. (2023b) illustrates that there is still a sideway to bypass ChatGPT's ethical modules. They propose a method named *jailbreak* to exploit tricky prompts to set up user-created role plays to alter ChatGPT's ego and programming restrictions, which allows it to answer users' queries unethically. More recently, Greshake et al. (2023) proposes a novel indirect prompt injection, which required the community to have an urgent investigation and evaluation of current mitigation techniques against these threats. When LLMs are integrated with other plugins or using its API calling, the content retrieved from the Web (public source) may already be poisoned and contain malicious prompts pre-injected and selected by adversaries, such that these prompts can be indirectly used to control and direct the model. In other words, prompt injection risks may occur not only in situations where adversaries explicitly prompt LLMs but also among users, developers, and automated data processing systems.

We also noticed that prompt injection, and techniques based on prompt injection to work with the APIs of LLMs, have been used to generate malware (Goodin 2023; News 2023; Botacin 2023).

5.2 Comparison with Human Experts

Another evaluation thread is to study how LLMs are compared with human experts. For example, for ChatGPT, Guo et al. (2023) conducts the comparison on questions from open-domain, financial, medical, legal, and psychological areas, Farhat et al. (2023) compares on the bibliometric analysis, Malinka et al. (2023) evaluates on university education with a primary focus on computer security-oriented specialisation, Ji et al. (2023) considers the ranking of contents, and Wu et al. (2023e) compares on the grammatical error correction (GEC) task. It is surprising to note that, in all these comparisons, the conclusion is that, ChatGPT does not perform as well as expected. One step further, to study collaboration rather than only focus on comparisons, Qi et al. (2023) explores how ChatGPT's performance on safety analysis can be compared with human experts, and concludes that the best results are from the close collaboration between ChatGPT and the human experts. A similar conclusion was also drawn by Jang and Lukasiewicz (2023) when studying ChatGPT's logically consistent behaviours.

In some cases, LLMs can outperform human experts in specific tasks, like processing enormous amounts of data or doing repeated activities with great accuracy. For example, LLMs can be used to analyse massive numbers of medical records to uncover patterns and links between different illnesses, which can aid in medical diagnosis and therapy (Liu et al. 2023d; Agrawal et al. 2022). On the other hand, human experts may outperform LLMs in jobs requiring more complicated reasoning or comprehension of social and cultural contexts. Human specialists, for example, may better interpret and respond to delicate social signs in a conversation, which can be difficult for LLMs. It is important emphasising that LLMs are intended to supplement rather than replace human competence (Shanahan 2022). LLMs can automate specific processes or help human professionals accomplish things more efficiently and precisely (Zhao et al. 2023a). For example, Qi et al. (2023) studies how ChatGPT's performance on safety analysis can be compared with human experts and concludes that the best results are from the close collaboration between ChatGPT and the human experts. Holmes et al. (2023) also shows that huge language models have a lot of potential as knowledgeable assistants collaborating with subject specialists.

5.3 Benchmarks

Benchmark datasets have been used to evaluate the performance of LLMs. For example, in Wang et al. (2023a), AdvGLUE and ANLI benchmark datasets are used to assess adversarial robustness, and Flipkart review and DDXPlus medical diagnosis datasets are used to evaluate out-of-distribution evaluation. In Sun et al. (2023), eight kinds of typical safety scenarios and six types of more challenging instruction attacks are used to expose safety issues of LLMs. In Frieder et al. (2023), the GHOSTS dataset is used to evaluate the mathematical capability of ChatGPT.

Regarding the LLMs as a software as a service, rather than previous deep learning models, it becomes imperative to incorporate lifelong time assessment. In Chen et al. (2023a), they evaluated the March 2023 and June 2023 versions of GPT-3.5 and GPT-4 on several diverse benchmarks. The LLM service's behavior can undergo significant changes within a fairly brief period, as evidenced by their findings. According to Edwards (2023), it states that while releasing the results of the benchmark, the providers should provide raw results, not only high-level metrics. So that the inspector is capable of conducting a more thorough

examination of the model's defects. Previous NLP works show that fine-tuning pre-trained transformer-based language models such as BERT (Devlin et al. 2018) is an unstable process (Dodge et al. 2020; Lee et al. 2019). During the continual updating of LLMs, it could go through multiple iterations of finetune and RLHF, which even increases the risk of catastrophic forgetting. In Aiyappa et al. (2023), the challenge of ensuring fair model evaluation in the age of closed and continuously trained models is discussed. Moreover, Low-Rank Adaptation (LoRA) is proposed to reduce the trainable parameters and thus could avoid catastrophic forgetting (Hu et al. 2021).

5.4 Testing and statistical evaluation

As mentioned above, most existing techniques on the falsification and evaluation heavily rely on human intelligence and therefore have a significant level of human involvement. In red teaming, the red team must be creative in finding bad examples. In prompt injection, the attacker needs to design specific (sequence of) prompts to retrieve the information they need. Unfortunately, human expertise and intelligence are expensive and scarce, which calls for automated techniques to have an intensive and fair evaluation, and to find corner cases as exhaustive as possible. In the following, we discuss how testing and statistical evaluation methods can be adapted for a fair evaluation of LLMs.

To simplify it, we assume an LLM is a system that generates an output given an input. Let $\mathbf{D}$ be the space of nature data, an LLM is a function $M : \mathbf{D} \to \mathbf{D}$. In the meantime, there is another function $H : \mathbf{D} \to \mathbf{D}$ representing human's response. For an automated generation of test cases, we need to have an oracle $\mathbf{O}$, a test coverage metric $\mathbf{C}$, and a test case generation method $\mathbf{A}$. The oracle $\mathbf{O}$ determines if an input–output pair $(\mathbf{x}, \mathbf{y})$ is correct. The implementation of oracle is related to both M and H, by checking whether given any input $\mathbf{x}$ their outputs $M(\mathbf{x})$ and $H(\mathbf{x})$ are similar under certain criteria. We call an input–output pair a test case. Given a set of test cases $\mathbf{P} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,\ldots,n}$, an evaluation of the coverage metric $\mathbf{C}$ returns a probability value representing the percentage of cases in $\mathbf{P}$ over the cases that should be tested. Finally, the test case generation method $\mathbf{A}$ generates the set $\mathbf{P}$ of test cases. Usually, the design of coverage metric $\mathbf{C}$ should be based on the property to be verified. Therefore, the verification problem is reduced to determining of whether the percentage of test cases in $\mathbf{P}$ that passes the oracle $\mathbf{O}$ is above a pre-specified threshold.

Statistical evaluation applies statistical methods in order to gain insights into the verification problem we are concerned about. In addition to the purpose of determining the existence of failures (i.e., counterexamples to the satisfiability of desirable properties) in the deep learning model, statistical evaluation assesses the satisfiability of a property in a probabilistic way, by, e.g., aggregating sampling results. The aggregated evaluation result may have the probabilistic guarantee, e.g., the probability of failure rate lower than a threshold l is greater than $1 - \epsilon$, for some small constant ϵ.

While the study on LLMs is just started (Reiss 2023), statistical evaluation methods have been proposed for the general machine learning models.

Sampling methods and testing methods have been considered for convolutional or recurrent neural networks. Sampling methods, such as Weng et al. (2018), are to summarise property-related statistics from the samples. There are many ways to determine how the test cases are generated, including, e.g., fuzzing, coverage metrics (Sun et al. 2019; Huang et al. 2021), symbolic execution (Gopinath et al. 2018), concolic testing (Sun et al. 2018b), etc. Testing methods, on the other hand, generate a set of test cases and use the generated test cases to evaluate the reliability (or other properties) of deep

learning (Sun et al. 2018a). While sampling methods can have probabilistic guarantees via, e.g., Chebyshev's inequality, it is still under investigation on associating test coverage metrics with probabilistic guarantees. Moreover, ensuring that the generated or sampled test cases are realistic is necessary, i.e., on the data distribution (Huang et al. 2022b; Zhao et al. 2021b).

For LLMs, the key technical challenges are on the design of test coverage metrics and the test case generation algorithms because (1) LLMs need to be considered in a black-box manner, rather than white-box one; this is mainly due to the size of LLMs that cannot be reasonably explored, and therefore an exploration on the input space will become more practical; (2) LLMs are for natural language texts, and it is hard to define the ordering between two texts; the ordering between two inputs are key to the design of test case generation algorithms; and (3) LLMs are non-deterministic, i.e., different outputs are expected in two tests with identical input.

6 Verification

This section discusses if and how more rigorous verification can be extended to work on LLM-based machine-learning tasks. So far, the verification or certification of LLMs is still an emerging research area. This section will first provide a comprehensive and systematic review of the verification techniques on various NLP models. Then, we will discuss a few pioneering black-box verification methods that are workable on large-scale language models. These are followed by a discussion on how to extend these efforts towards LLMs and a review of the efforts to reduce the scale of LLMs to increase the validity of verification techniques.

We remark that, this section is focused on verifying LLMs. For the other direction of utilising LLMs to support the verification, there are works related to e.g., specification autoformalisation (Wu et al. 2022a), code generation (Thakur et al. 2023), assertion generation (Kande et al. 2023), zero-shot vulnerability repair (Pearce et al. 2023).

6.1 Verification on natural language processing models

As discussed in previous sections, an attacker could generate millions of adversarial examples by manipulating every word in a sentence. Adversarial examples have different safety and trustworthiness implications to the downstream tasks. For example, a perturbed output text might include different emotions that will affect the sentiment analysis, and it is possible that a perturbed text might have the same meaning but different language style to affect the spam detection. However, such methods may still fail to address numerous unseen cases arising from exponential combinations of different words in a text input. To overcome these limitations, another class of techniques has emerged, grounded in the concept of "certification" or "verification" (Seshia et al. 2016; Huang et al. 2017). For example, via certification or verification, these methods train the model to provide an upper bound on the worst-case loss of perturbations, thereby offering a certificate of robustness without necessitating the exploration of the adversarial space (Sinha et al. 2017). By utilising these certification-driven methods, we can better evaluate the model's robustness in the face of adversarial attacks (Goodfellow and Papernot 2017).

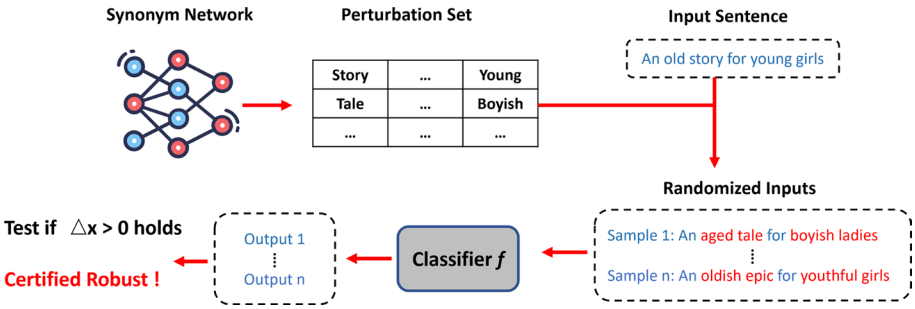

Fig. 7 Pipeline for robustness verification in Ye et al. (2020)

6.1.1 Verification via interval bound propagation

The first technique successfully adapted from the computer vision domain for verifying NLP models is Interval Bound Propagation (IBP). It is a bounding technique that has gained significant attention for its effectiveness in training large, robust, and verifiable neural networks (Gowal et al. 2018). By striving to minimise the upper bound on the maximum difference between the classification boundary and input perturbation region, IBP allows the incorporation of a loss term during training. This enables the minimisation of the last layer of the perturbation region, ensuring it remains on one side of the classification boundary. As a result, the adversarial region becomes tighter and can be considered certified robust. Notably, Jia et al. (2019) proposed certified robust models while providing maximum perturbations in text classification. The authors employed interval bound propagation to optimise the upper bound over perturbations, providing an upper bound over the discrete set of perturbations in the word vector space.

Later on, Huang et al. (2019a) introduced a verification and verifiable training method for neural networks in NLP, proposing a tighter over-approximation in the form of a 'simplex' in the embedding space for input perturbations. To make the network verifiable, they defined the convex hull of all the original unperturbed inputs as a space of delta perturbation. By employing the IBP algorithm, they generated robustness bounds for each neural network layer. Furthermore, as shown in Fig. 7, Ye et al. (2020) proposed structure-free certified robust models, which can be applied to any arbitrary model, overcoming the limitations of IBP-based methods that are not applicable to character-level and sub-word-level models. This work introduced a perturbation set of words using synonym sets and top-K nearest neighbours under the cosine similarity of GloVE vectors (Pennington et al. 2014), which could subsequently generate sentence perturbations using word perturbations and train a provably robust classifier. Very recently, Wallace et al. (2022) highlighted the limitations of IBP-based methods in a broader range of NLP tasks, demonstrating that IBP methods have poor generalisability. In this work, the authors performed a systematic evaluation of various of sentiment analysis tasks. They pointed out some insights regarding the promising improvements and adaptations for IBP methods in the NLP domain.

6.1.2 Verification via abstract interpretation

Another popular verification technique applied to various NLP models is based on abstract interpretation or functional over-approximation. The idea behind abstract interpretation is to approximate the behaviour of a program by representing it using a simpler model that is easier to analyse. Specifically, this technique can represent the network using an abstract domain that captures the possible range of values the network can output for a given input. This abstract domain can then be used to reason about the network's behaviour under different conditions, such as when the network is under adversarial perturbation. One notable contribution in this area is POPQORN (Ko et al. 2019). It can find a certificate of robustness for RNN-based networks, which utilised 2D planes to bound the cross-nonlinearity in Long Short-Term Memory (LSTM) networks so a certificate within an l_p ball can be located if the lower bound on the true label output unit is larger than the upper bounds of all other output units. Later on, Cert-RNN (Du et al. 2021) introduced a robust certification framework for RNNs that overcomes the limitations of POPQORN (Ko et al. 2019). The framework maintains inter-variable correlation and accelerates the non-linearities of RNNs for practical uses. This work utilised Zonotopes (Eppstein 1996) to encapsulate input perturbations. Cert-RNN can verify the properties of the output Zonotopes to determine certifiable robustness. Using Zonotopes, as opposed to boxes, allows improved precision and tighter bounds, leading to a significant speedup compared to POPQORN.

Recently, Abstractive Recursive Certification (ARC) was introduced to verify the robustness of RNNs (Zhang et al. 2021). Using those transformations, ARC defined a set of programmatically perturbed string transformations and constructed a perturbation space. By memorising the hidden states of strings in the perturbation space that share a common prefix, ARC can efficiently calculate an upper bound while avoiding redundant hidden state computations. Roughly at the same time, Ryou et al. proposed a similar method called Polyhedral Robustness Verifier (PROVER) (Ryou et al. 2021). PROVER can represent input perturbations as polyhedral to generate a certifiably verified network for more general sequential data. To certify large transformers, DeepT was proposed by Bonaert et al. (2021). It was specifically designed to verify the robustness of transformers against synonym replacement-based attacks. DeepT employed multi-norm Zonotopes to achieve larger robustness radii in the certification. For the transformers with self-attention layers, Shi et al. (2019) developed a verification algorithm that can provide a lower bound to ensure the probability of the correct label is consistently higher than that of the incorrect labels. This method can obtain a tighter bound than those obtained from IBP-based methods.

6.1.3 Verification via randomised smoothing

Randomised smoothing (RS) (Cohen et al. 2019) is another promising technique for verifying the robustness of deep language models. Its basic idea is to leverage randomness during inference to create a smoothed classifier that is more robust to small perturbations in the input. This technique can also be used to give certified guarantees against adversarial perturbations within a certain radius. Generally, randomized smoothing begins by training a regular neural network on a given dataset. Then, given a trained base classifier f and an input x, the smoothed classifier g is defined using randomness (e.g., Gaussian noise) as: $g(x) = \text{argmax}_c \mathbb{P}(f(x + \epsilon) = c)$, where ϵ is the noise

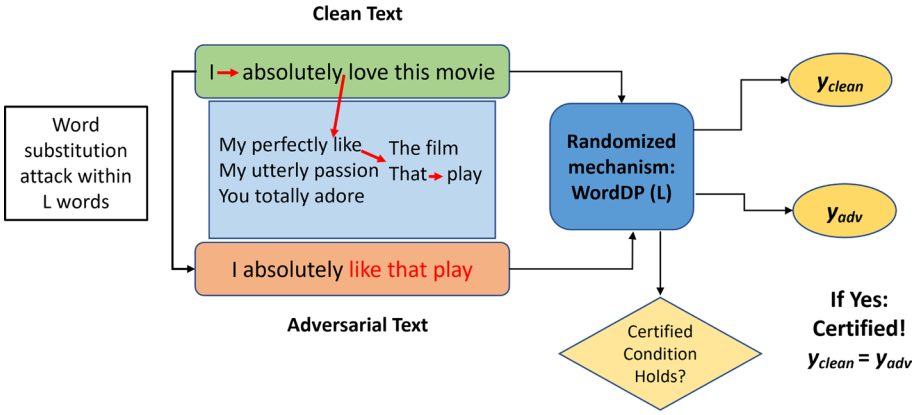

Fig. 8 Pipeline of wordDP for word-substitution attack and robustness verification (Wang et al. 2021a)

sampled from some distribution (e.g., a Gaussian distribution). During the inference phase, to classify a new sample, noise is randomly sampled from the predetermined distribution multiple times. These instances of noise are then injected into the input x, resulting in noisy samples. Subsequently, the base classifier $f(x)$ generates predictions for each of these noisy samples. The final prediction is determined by the class with the highest frequency of predictions, thereby shaping the smoothed classifier $g(x)$. To certify the robustness of the smoothed classifier $g(x)$ against adversarial perturbations within a specific radius r centered around the input x, RS calculates the likelihood of agreement between the base classifier $f(x)$ and $g(x)$ when noise is introduced to x. If this likelihood exceeds a certain threshold (e.g., surpassing $0.5 + \tau$, where τ represents a minor positive constant), it indicates the certified robustness of $g(x)$ within a radius r around x.

Figure 8 depicts one of the pioneering efforts of using RS for verifying the robustness of NLP models. It is called WordDP developed by Wang et al. (2021a), the authors introduced a novel approach to provide a certificate of robustness by leveraging the concept of differential privacy. In this work, the researchers considered a sentence as a database and the individual words within it as records. They demonstrated that if a predictive model satisfies a specific threshold of epsilon-differential privacy for a perturbed input, it can be inferred that the input is identical to the clean, unaltered data. This methodology offers a certification of robustness against L-adversary word substitution attacks. In another recent study, Zeng et al. (2021c) introduced RanMASK, a certifiably robust defence method against text adversarial attacks, which employs a novel randomised smoothing technique specifically tailored for NLP models. The input text is manually perturbed in this approach and subsequently fed into a mask language model. Random masks are then generated within the input text to create a large set of masked copies, which are subsequently classified by a base classifier. A "majority vote" mechanism determines the final robust classification. Furthermore, the researchers utilised pre-trained models such as BERT and RoBERTa to generate and train with the masked inputs, showcasing the practical applicability and effectiveness of the RanMASK technique in some real-world NLP scenarios.

6.2 Black-box verification

Many existing verification techniques impose specific requirements on DNNs, such as targeting a specific network category or networks with particular activation functions (Huang et al. 2017). With the increasing complexity and scale of large language models (LLMs), traditional verification methods based on layer-by-layer search, abstraction, and transformation have become computationally impractical. Consequently, we envision that black-box approaches have emerged as a more feasible alternative for verifying such models (Wicker et al. 2018; Wu et al. 2020; Xu et al. 2022).

In the black-box setting, adversaries can only query the target classifier without knowing the underlying model or the feature representations of inputs. Several studies have explored more efficient methods for black-box settings, although most of current approaches focus on vision models (Wicker et al. 2018; Wu et al. 2020; Xu et al. 2022). For instance, DeepGO, a reachability analysis tool, offers provable guarantees for neural networks with deep layers and nonlinear activation functions (Ruan et al. 2018). Its extended version, DeepAgn, is compatible with various networks, including feedforward and recurrent neural networks, as long as they exhibit Lipschitz continuity (Zhang et al. 2023b).

Subsequently, an anytime algorithm was developed to approximate global robustness by iteratively computing lower and upper bounds (Ruan et al. 2019). This algorithm returns intermediate bounds and robustness estimates that improve as computation proceeds. For neural network control systems (NNCSs), the DeepNNC verification framework utilises a black-box optimisation algorithm and demonstrates comparable efficiency and accuracy across a wide range of neural network controllers (Zhang et al. 2023c). GeoRobust, another black-box analyser, efficiently verifies the robustness of large-scale DNNs against geometric transformations (Wang et al. 2023c). This method can identify the worst-case manipulation that minimises adversarial loss without knowledge of the target model's internal structures and has been employed to systematically benchmark the geometric robustness of popular ImageNet classifiers.

Recently, some researchers have attempted to develop black-box verification methods for NLP models, although these methods are not scalable to LLMs. For example, one study introduced a framework for evaluating the robustness of NLP models against word substitutions (La Malfa et al. 2020). By computing a lower and upper bound for the maximal safe radius for a given input text, this verification method can guarantee that the model prediction does not change if a word is replaced with a plausible alternative, such as a synonym.

We also notice another thread of works focusing on training verifiers, for the correctness of language-to-code generation (Ni et al. 2023) or solving math word problems (Cobbe et al. 2021).

6.3 Robustness evaluation on LLMs

Given the prominence of large-scale language models such as GPT, LLaMA, and BERT, some researchers have recently started exploring the robustness evaluation of these models. One such investigation is the work of Cheng et al. (2020), who developed a seq2seq algorithm based on a projected gradient method combined with group lasso and gradient regularisation. To address the challenges posed by the vast output space of LLMs, the authors introduced innovative loss functions to conduct non-overlapping and targeted keyword attacks. Through applications in machine translation and text summarisation tasks, their

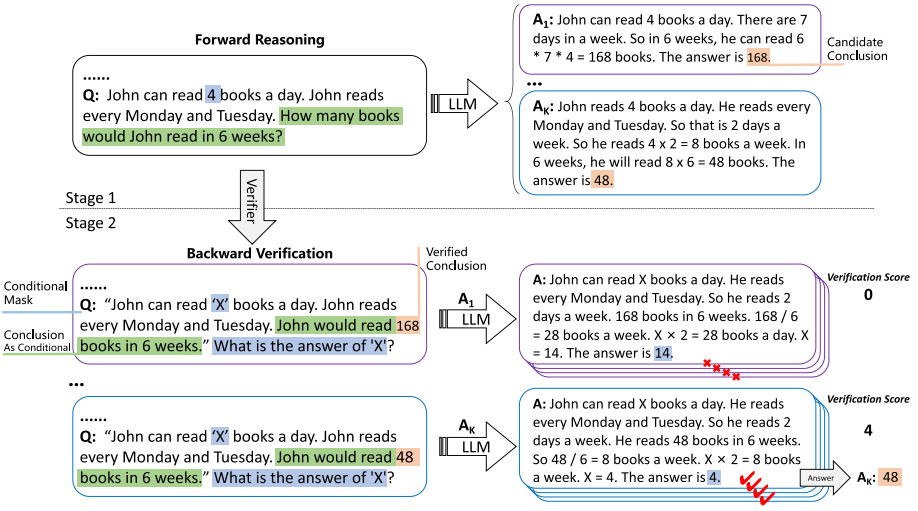

Fig. 9 Example of Self-Verification proposed in Weng et al. (2022). In Stage-1, LLM generates some candidate conclusions. Then LLM verifies these conclusions and counts the number of masked conditions that reasoning is correct to as the verification score in Stage-2

seq2seq model demonstrated the capability to produce desired outputs with high success rates by altering fewer than three words. The preservation of semantic meanings in the generated adversarial examples was further verified using an external sentiment classifier. Another notable contribution comes from Weng et al. (2022, 2023), as shown in Fig. 9. They proposed a self-verification method that leverages the conclusion of the chain of thought (CoT) as a condition for constructing a new sample. The LLM is then tasked with re-predicting the original conditions, which have been masked. This approach allows for the calculation of an explainable verification score based on accuracy, providing valuable insights into the performance of LLMs. Finally, Jiang et al. (2022) introduced an approach that addresses both auto-formalisation (the translation of informal mathematics into formal logical notation) and the proving of "proof sketches" resulting from the auto-formalisation of informal proofs.

To the best of our knowledge, there remains a conspicuous absence of research on verifying large language models (LLMs). As such, we encourage the academic community to prioritise this vital research domain by developing practical black-box verification methods tailored specifically to LLMs.

6.4 Towards smaller models

The current LLMs are of large scale with billions or trillions of parameters. This will make the verification hard, even with the above-mentioned verification techniques. Another possible thread of research to support the ultimate verification is to use smaller LLMs.

A prevailing strategy of developing a smaller LLM is to apply techniques that reduce the parameters of a pre-trained model. One typical method is model compression, such as quantisation (Nagel et al. 2020; Liu et al. 2021c; Frantar and Alistarh 2022). However, directly applying quantisation techniques on LLMs leads to performance degradation. To this end,

ZeroQuant (Yao et al. 2022) utilise kernel fusion (Wang et al. 2010) to compress weights and activations before data movement, to maximise memory bandwidth utilisation and speed up inference. Similarly, Park et al. (2022) introduces a new LUT-GEMM kernel that allows quantised matrix multiplications with either uniform or non-uniform weight quantisation. Both Wang et al. (2010), Park et al. (2022) require custom CUDA kernels. In contrast, Dettmers et al. (2022) improves predictive performance on billion-scale 8-bit transformers. Frantar et al. (2023) further improves GPT model with near-zero performance drop on 3 or 4-bit precision by deploying Optimal Brain Quantisation (Frantar and Alistarh 2022), Lazy Batch-Updates and Cholesky Reformulation. Other than quantisation techniques, Low-rank adaptation (LORA) (Hu et al. 2022) involves decomposing the weights into low-rank matrices, which has been shown to reduce the number of parameters while maintaining model performance significantly.

Knowledge distillation refers to the methodology wherein a streamlined "student" model is trained to approximate the predictive behavior of a more complex "teacher" model (Hinton et al. 2015). This is achieved by leveraging both the ground-truth labels and the teacher model's soft predictions during the training process (Cho and Hariharan 2019; Tung and Mori 2019). The training process enables the student model to assimilate the implicit knowledge encapsulated by the teacher model with less parameters. Specifically, the student model achieves performance similar to the teacher model while being more computationally efficient, making it suitable for deployment in resource-limited settings (Gou et al. 2021; He et al. 2022). In LLMs, the small student model is often used to assimilate information from the pre-trained teacher model (Taori et al. 2023; Chiang et al. 2023; Wu et al. 2023d; Peng et al. 2023; Gu et al. 2023a). This multifaceted transfer of knowledge enables the student model to augment its capabilities in language understanding and generation, which particularly advantageous for deployment in computational environments where resource efficiency is a critical consideration. The "teacher" neural network is trained on extensive text data for various language tasks, such as understanding, generation, translation, and sentiment analysis, and the "student" model, designed to be more resource-efficient, aims to replicate the teacher model's capabilities while using fewer layers and neurons or a simplified architecture. The goal is to achieve comparable performance but with reduced computational overhead.

It is worth noting that Spiking Neural Networks (SNNs), as the third generation neural networks, offer a complementary approach to improve computing efficiency, e.g., utilising sparse operation (Rueckauer et al. 2017; Wu et al. 2022b, 2023b). Recent research has introduced SpikeGPT (Zhu et al. 2023), the largest SNN-based model with 260 million parameters, to demonstrate the performance of SNNs on GPT models, comparable to that of traditional neural networks. In contrast, SNNs require implementation on specialised hardware, such as neuromorphic chips like TrueNorth (Akopyan et al. 2015) and Loihi (Davies et al. 2018), which have been designed to mimic biological neurons at the circuit level. While the development of SNNs on LLM is still in its early stages, it presents an alternative approach to achieving computing efficiency that works parallel to compression techniques.

7 Runtime monitor

Guardrails mentioned in Sect. 2.3.2 provide a safeguard for the LLMs to interact with the end users while retaining its social responsibility. This section discusses a V&V method, i.e., runtime monitor, that is somewhat similar to the guardrails in that, it provides the safeguards on the behaviour of the LLMs against vulnerabilities such as those discussed in Sect. 3. The key

motivation for using runtime monitors, rather than the verification, is twofold. First, verification methods require significant computation and hence can become impractical when dealing with large models such as LLMs. Second, a deep learning model might be applied to scenarios different from where the training data is collected. These suggest the need for a runtime monitor to determine the satisfiability of a specification *on the fly*.

Similar to evaluation and verification, there is no existing work on LLMs, but there are proposals for e.g., the convolutional neural networks. Given the missing specifications [(although the attempts to formalise specifications started (Bensalem et al. 2022; Balakrishnan et al. 2019; Huang et al. 2022a)], the current runtime monitoring methods for deep learning start from constructing an abstraction of a property, followed by determining the failure of the property by checking the distance between the abstraction and the original learning model. There are a few existing methods for abstraction of deep learning. For example, in Cheng et al. (2019b), a Boolean abstraction on the ReLU activation pattern of some specific layer is considered and monitored. Conversely of Boolean abstraction, Henzinger et al. (2020) consider box abstractions. In Berthier et al. (2021), a Bayesian network based abstraction, which abstracts hidden features as random variables, is considered.

The construction of a runtime monitor requires the specification of the failures. Other than direct specifications such as Huang et al. (2022a), which requires additional efforts to convert the formulas into runtime monitors, this can usually be done by collecting a set of failure data and then summarising (through either learning or symbolic reasoning or a combination of them) the relation between failure data and the part of the LLMs to be monitored, e.g., some critical layers of the LLMs or the output (Li et al. 2018b; Cheng et al. 2022).

7.1 Monitoring out-of-distribution

In the following, we discuss how runtime monitoring techniques have been developed for a specific type of failure, i.e., out of distribution, which suggests that the runtime data is on a different distribution from the training data. It is commonly believed that ML models cannot be reliable when working with data drifted from the training data. Therefore the occurrence of out-of-distribution suggests the existence of risks.

Neural networks, used in computer vision (CV) or natural language process (NLP) tasks, are known to make overconfident predictions on out-of-distribution (OoD) samples that do not belong to any of the training classes, i.e., in-distribution (ID) data. For security reasons, such inputs and their corresponding predictions must be monitored at runtime, especially for networks deployed in safety-critical applications. Runtime monitoring or detection of out-of-distribution (OoD) samples have been extensively studied in CV (Hendrycks and Gimpel 2016; DeVries and Taylor 2018; Liang et al. 2018; Ren et al. 2019b; Liu et al. 2020; Yang et al. 2021a). Recently, researchers have paid more attention to this problem for NLP models (Hendrycks et al. 2020), although large-scale language models (ChatGPT) have shown continuous improvement on most adversarial and OoD classification tasks (Wang et al. 2023a). Generally, to monitor OoD samples, one has to devise an ID confidence score function $S(\mathbf{x})$ such that an input $\mathbf{x}$ is classified as OoD if the value $S(\mathbf{x})$ is less than a predefined threshold γ, as shown in Eq. 2.

$$M(\mathbf{x}) = \begin{cases} \text{ID} & \text{if } S(\mathbf{x}) \geq \gamma \\ \text{OoD} & \text{otherwise} \end{cases} \tag{2}$$

According to what information is used to construct this confidence function $S(\mathbf{x})$, the current OoD monitoring methods for NLP models (Arora et al. 2021; Huang et al. 2020b;

Chen et al. 2022, 2023b; Cho et al. 2022; Duan et al. 2022) can be roughly divided into the following three categories.

The first category includes *input density estimation methods* (Ren et al. 2019b; Lee et al. 2020; Gangal et al. 2020; Arora et al. 2021). These methods usually involve a density estimator, either directly in the input space or in the latent space of the generative models for the ID data. The probability value of the input given by such a density estimator can be used as the ID score. One of these examples is Arora et al. (2021) that uses the *token perplexity* (Lee et al. 2020), avoiding the bias of text length as the ID confidence score.

The second category includes *feature or embedding space approximation methods* (Xu et al. 2020b; Podolskiy et al. 2021; Zeng et al. 2021a; Zhou et al. 2021, 2022; Chen et al. 2022). These methods first approximate the seen features by some distribution function, and then use the distance (e.g., Euclidean and Mahalanobis distances) between this distribution and the input feature as the ID confidence score. For instance, Chen et al. (2022) extracts holistic sentence vector embeddings from all intermediate layers and shadow states of all tokens to enhance the general semantics in sentence vectors, thereby improving the performance of OoD text detection algorithms based on feature space distance.

The third category includes *output confidence calibration methods* (Hendrycks et al. 2020; Desai and Durrett 2020; Dan and Roth 2021; Li et al. 2021b; Shen et al. 2021a; Yilmaz and Toraman 2022). These methods use the model's prediction confidence (usually calibrated) as the ID score. The classic is the *maximum softmax probability*, often used as a strong baseline for OoD detection.

Despite a lot of work and effort, the current results can still be improved. Moreover, no single method is better than the other at present, which is understandable, given the infinity of OoD data and the ambiguous boundaries of ID data. In terms of performance, the methods mentioned above do not entail significant overhead, as they all involve a single computation of a function related to high-dimensional vectors, which can be accomplished within a short timeframe. When compared to the time required for a single inference of a neural network, this overhead can be considered negligible.

Finally, we remark that OoD detection task in the field of NLP still requires greater efforts in the following two aspects. First, the community ought to reach a consensus on a fine-grained definition of the OoD problem for NLP models, by precisely considering the sources of OoD data and the tasks of NLP models. For example, existing work is done on NLP classification tasks. How to define the OoD problem for the generative NLP models, e.g., what kind of data should be called OoD data to these generative models? Second, a fair evaluation method is needed, given the fact that the training datasets for most large language models (LLM) are unavailable, i.e., it is unclear whether the used test dataset for evaluating OoD methods are OoD data to the tested models or not.

7.2 Monitoring attacks

In this subsection, we discuss how to detect adversarial and backdoor attacks in real-time. It is possible to detect the backdoor input at runtime, given a set of clean reference dataset. The runtime monitoring for backdoor attack is based on the observation that although backdoor input and target samples from reference dataset are classified the same by the compromised network, the rationale for this classification is different. The network identifies input features that it has learnt correlate to the target class in the case of clean samples from the target class. It identifies features associated with the backdoor trigger in the case of backdoor samples, causing it to identify the input as the target class.

Based on the above idea, several detection strategies for backdoor input are developed. Activation Clustering (AC) approach is adopted to check the activation similarity between the runtime input and reference dataset (Chen et al. 2019). The activations of last convolutional layer are obtained for reference dataset and input. They are grouped according to the label and each group is clustered separately. To cluster the activations, the dimensionality reduction technique, Independent Component Analysis (ICA) is applied. Then cluster analysis methods, like exclusionary reclassification, relative size comparison and silhouette score can help users identify if the input contains the backdoor trigger. In addition, the feature importance maps generated from XAI techniques can be leveraged to help identify the backdoor input (Huang et al. 2019b; Tejankar et al. 2023). Since the compromised neural network relies on backdoor trigger to make decision, the backdoor trigger is highlighted when generating the feature importance maps regarding the input. Then, the backdoor input can be filtered out when simple and fixed decision logic is summarised from the explanations. While the runtime monitoring of backdoor for LLMs is few, we believe the current techniques can be extended to LLMs once we can get the hidden activation or explanations from LLMs.

Adversarial examples are thought to exhibit distinguishable features that set them apart from clean inputs (Ilyas et al. 2019). Consequently, we can leverage this distinction to develop a runtime robust detector. For example, uncertainty values are used as features to build a binary classifier as a detector. Feinman et al. (2017) introduced the Bayesian Uncertainty metric, employing Monte Carlo dropout to estimate uncertainty, primarily detecting adversarial examples situated near the class boundaries, while Smith and Gal (2018) utilised a mutual information approach for the same purpose. Furthermore, Hendrycks and Gimpel (2016) demonstrated that softmax prediction probabilities can be used to identify adversarial examples. They appended a decoder to reconstruct clean inputs from the softmax and jointly trained it with the baseline classifier. Following the hypothesis that diverse models exhibit different mistakes when confronted with the same attack inputs, Monteiro et al. (2019) proposed a bimodel mismatch detection. Moreover, Feinman et al. (2017) introduced kernel density estimation for each class within the training data and subsequently trained a binary classifier as a detector, utilising the density and uncertainty features associated with clean, noisy, and adversarial examples. Although there are also few runtime monitoring methods for detecting adversarial examples in LLMs, we believe these current techniques can be extended to LLMs once we can develop an LLM detection model.

7.3 Monitoring output failures

As we mentioned in previous sections, although LLMs have shown strong performance in many domains (Bang et al. 2023; Liu et al. 2023a; Jiao et al. 2023; Sobania et al. 2023; Zhong et al. 2023), they are also found to be prone to various types of failures after scrutiny and evaluation (Borji 2023; Shen et al. 2023; Zhao et al. 2023b), such as factual errors (Zhao et al. 2023b), coding (Liu et al. 2023c; Khoury et al. 2023), math (Frieder et al. 2023), and reasoning (Liu et al. 2023b). These failures can spell fatal disaster for downstream tasks, especially in safety-critical applications. To address these issues, one way is to devise a mechanism to generate constrained outputs (Hu et al. 2017; Kumar et al. 2020; Madaan et al. 2021). However, LLMs generate output by selecting appropriate words from a vocabulary rather than grabbing corresponding snippets from sources of truth, or reasoning on them. This generative nature makes it challenging to control the output, and

even more challenging to ensure that the generated output is, in fact, consistent with the information source. Another way is to monitor the output of the models and take necessary actions. In the following, we first summarise the limited amount of existing work on runtime monitoring of such failures and then discuss how to proceed from a future perspective.

In addition to the generative nature of LLMs, the diversity of downstream tasks also makes it extremely difficult, if not impossible, to have a general monitoring framework for such generative outputs. Such output failures need to be addressed in a targeted manner, according to different application scenarios and the specific scientific knowledge accumulated by humans in various fields such as science and technology. Regarding factual errors, Thorne et al. (2018) proposed a testbed for fact verification. However, this remains an unsolved challenge. Similar to fact-checking, we argue that for code generation failures, the fruitful methods, techniques, and tools accumulated in the field of formal methods related to *compilers design* (Lam et al. 2006) and *program verification* (Vardi and Wolper 1986) can be adapted to check whether the generated code is executable or satisfies some specified invariants (Manna and Pnueli 2012; Bensalem et al. 1996, 1998), respectively. As for math-related failures, existing tools in *automated theorem proving* (Fitting 1996; Bibel 2013) [e.g., Z3 (De Moura and Bjørner 2008) and Prover9 (McCune 2005)] may help. If an LLM is employed within safety-critical systems and its outputs are required to adhere to specified system safety properties, then a combination of traditional runtime monitoring and enforcement techniques (Bauer et al. 2011; Bartocci and Falcone 2018), along with those (Garcıa and Fernández 2015; Alshiekh et al. 2018; Jansen et al. 2018, 2020; Gu et al. 2022) specifically developed for safe reinforcement learning, can be put into action. This allows to detect in real-time whether the model's outputs violate predefined behavioural specifications and enforce corrective actions on the model's outputs to ensure the safe operation of the system.

In order to conduct runtime monitoring for the aforementioned output errors, substantial offline or online overheads are incurred. This is due to the requirement of establishing an auxiliary system aimed at efficiently detecting a range of output anomalies in the model.

Finally, we point out that the current research on the output failures of large-scale language models is still blank. More research is needed, such as configuring a runtime monitor for the output of a specific application, or combining symbolic reasoning and causal reasoning with the model's learning process to ensure that the output avoids failures from the source.

7.4 Perspective

Since LLMs are still in their infancy and have many known vulnerabilities, monitoring these models in real time is a longstanding challenge. In this section, we outline topics for future work to call on more researchers to address this challenge from three perspectives: why, what, and how.

Why does a model need to be monitored? The first thing we want to highlight is whether at some point the LLMs can be trained intelligent enough, so that there is no need to design a separate runtime monitor for these models. For instance, the model is endowed with abilities to automatically detect "illegal inputs" (e.g., out-of-distribution inputs) and guarantee the correctness of its outputs. From our authors' perspective, achieving such level of intelligent models in the foreseeable future is very difficult, if not impossible. The main reasons are as follows. Existing LLMs are still learned from observations, i.e., a training dataset containing partial information. There is no evidence that current learning methods

can infer from parts to wholes, even in the case of massive data, nor is there evidence that a training dataset captures all the information. Furthermore, existing learning methods do not characterize their generalization bounds but instead measure the so-called generalization error, which prevents the identification of "illegal inputs". Therefore, it is necessary to monitor the model in real-time.

What should be monitored? One needs to overcome various vulnerabilities listed in Sect. 3 to reliably use LLMs in safety-critical applications. Equipping the model with a corresponding runtime monitor provides a possible solution complementary to offline verification methods. For example, there have been some works on monitoring whether the model's prediction is made for out-of-distribution inputs and whether the model's output is consistent with some existing fact base. However, to our knowledge, there is no monitoring work on other output failures, e.g., reasoning and code errors; on intended attacks, e.g., robustness, backdoor, and data poisoning. Thus, we call on researchers and practitioners to investigate more in these topics.

How to better design a monitor for a model? The state-of-the-art methods are based on the uncertainty model's predictions. Unfortunately, low uncertainty cannot assure the model's prediction is reliable, and vice versa. To better design monitors for LLMs, we need the following efforts. First, some fundamental intrinsic issues of deep learning models must be better addressed, such as model implicit generalisation and decision boundaries and explainability of model decisions, which may provide more rigorous and formal characterisation and specification for building monitors. Specific to LLMs, some special issues need to be tackled, such as the unavailability of training datasets, the non-transparency of models, the generative nature of multi-modality, etc. Regarding specific tasks, such as the most studied problem of monitoring out-of-distribution inputs, principled methods for system design and evaluation of monitors still needs to be included, as current work is based on calibration of predictive confidence scores and evaluation on one-sided test datasets. Last, we call for great attention to unexplored topics, such as how to monitor other trustworthiness and responsibility issues, attacks, and unintended bugs, along with the model's social and ethical alignments with human society.

8 Regulations and ethical use

V&V provides a set of technical means to support the alignment of LLMs with human interests. However, it has been argued that constructing LLMs that cannot be abused can be impossible. This suggests that technical means are necessary, but can be insufficient. To this end, it is needed to have *ethical means*, to supplement the technical means, in ensuring that the *complete alignment* of the use of LLMs with human interests. In the following, we discuss several recent signs of progress.

8.1 Regulate or ban?

A recent debate on "a 6-month suspension on the development (https://futureoflife.org/open-letter/pause-giant-ai-experiments/) vs. a regulated development" has shown the anxiety of, and the difference of opinions from, the community upon the possibilities of AI development being misaligned with human interests. More radical actions have also been taken. For example, Italy has reportedly banned the ChatGPT (https://www.cnbc.com/2023/04/04/italy-has-banned-chatgpt-heres-what-other-countries-are-doing.html). In a US

Senate Hearing on May 2023, OpenAI CEO Sam Altman asked the government to regulate AI (Senate 2023). Actually, on AI regulations, major players such as the EU, US, UK, and China all have their respective approaches and initiatives, e.g., the EU's GDPR (2016), AI Act (https://artificialintelligenceact.eu). Data Act (https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1113), the UK's Data Protection Act (https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted) and pro-innovative approach to regulate AI (https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1146542/a_pro-innovation_approach_to_AI_regulation.pdf), the US's Blueprint for an AI Bill of Rights (https://www.whitehouse.gov/ostp/ai-bill-of-rights/) and AI Risk Management Framework (https://www.nist.gov/itl/ai-risk-management-framework), and China's regulations for recommendation algorithms (http://www.cac.gov.cn/2022-01/04/c_1642894606258238.htm), deep synthesis (https://www.chinalawtranslate.com/en/deep-synthesis/), and algorithm registry (https://beian.cac.gov.cn/#/index). It is unclear (1) whether these regulations on the more general AI/ML, or other AI/ML algorithms, can automatically work for LLMs without any changes, and (2) how the regulations can be projected onto each other in a rigorous, yet operational, way. More importantly, even for general AI/ML, it still needs to be clarified how to *sufficiently and effectively* address regulatory requirements (such as robustness and transparency) with technical means. The V&V framework proposed in this survey will be one viable solution.

Nevertheless, significant issues raised by the LLMs, notably the ChatGPT, include copyright and privacy. The ChatGPT developers reportedly use data from the internet, and it is unclear if the *copyrights of the training data* have been carefully dealt with, especially when the ChatGPT is eventually for commercial use. Moreover, as a conversational AI, the *privacy of the end users*, when engaged in a dialogue, is a serious concern. The end-users should be informed on whether and how their dialogues will be stored, used, and redistributed.

8.2 Responsible AI principles

Responsible and accountable AI has been a topic of discussion for the past years (see e.g., https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html, https://www.microsoft.com/en-us/ai/responsible-ai, https://www.turing.ac.uk/news/publications/understanding-artificial-intelligence-ethics-and-safety.), with a gradual convergence to include properties such as transparency, explainability, fairness, robustness, security, privacy, etc. A governance framework is called for to ensure that these properties are implemented, evaluated, and monitored. A comprehensive discussion and comparison is out of the scope of this survey, but we note that, many properties are required, consistent definitions to many of them are still missing, and properties can be conflicting (i.e., the improvement of one property may compromise others). It is therefore not surprising that it can still be a long way to turn the principles into operational rules.

Specific to LLMs, ChatGPT and the like have led to severe concerns on e.g., potential misuse, unintended bias, and fair access. To this end, on the enterprise level, ethical principles are needed to guide the development and use of LLMs, including questioning whether something should be done rather than whether it can be done, as requested in https://www.nature.com/articles/d41586-023-00191-1. Moreover, systematic research is also called for to understand the certain to which the misuse of LLMs can lead to bad consequence, as is done in Botacin (2023) on attackers generating malware with LLMs or in Sandoval et al. (2023) which discusses the security implication of using LLMs in generating codes.

8.3 Educational challenges

Verification and validation of safe and trustworthy AI models are not central to the computer science curriculum or data science curricula. When validation and verification of models are taught at all, it is often part of an AI course that emphasizes tinkering with testing data-set rather than as a systematic and rigorous discipline with a solid scientific foundation. We need a curriculum beyond traditional education, covering formal verification, statistics, and XAI.

This need for adequately trained engineers impacts industrial practice, creating inefficiencies and difficulties in building AI systems with safety guarantees. Engineers untrained in safety and trustworthy AI models are often asked to make AI models for AI-critical applications.

The need for a shared cultural background between AI and rigorous design communities results in fragmented research. They use different terminologies. For example, "trustworthiness" does not have the same meaning across communities. Conferences are separate, no interaction between the two communities. The educational system will take time to adapt to evolving industrial and cultural needs. At the least, we suggest introducing AI students to the rigorous and systematic analysis of safety and trust and the corresponding approaches to the design of AI-critical applications. Another short-term objective should be to define and promote a reference curriculum in computer science with an optional program for designing safe and trusted AI applications.

8.4 Transparency and explainability

First, OpenAI's decision to not open-source GPT-3 and beyond has already led to concerns on the transparent development of AI. However, OpenAI said it plans to make more technical details available to other third parties for them to advise on how to weigh the competitive and safety considerations against the scientific value of further transparency. Nevertheless, we have seen a trend of open-sourcing LLMs, with notably Meta's Llama 2 (Touvron et al. 2023). It is also important to note that, no technical details are available on how the guardrail is designed and implemented. It will also be interesting to discuss on whether the guardrail itself should undergo a verification process.

Second, it has been hard to interpret and explain the decisions of the deep learning models such as image classifiers. The situation becomes worsens when dealing with LLMs (Kambhampati 2022), which have emergent and hard-to-explain behaviours. For example, it has been observed that adding an incarnation, such as "Let's think step by step", to the prompt can achieve improved responses from GPT-3. Techniques are needed to explain such a phenomenon. This calls for extending explainable AI techniques to work with LLMs. In particular, it is necessary to consider the explanations' robustness to explain why such incarnation can lead to improved, yet different, answers. To this end, some prior works on image classifiers, such as Zhao et al. (2021a), Huang et al. (2022c), can be considered.

9 Discussions

The safety and trustworthiness issues become more important with the wider adoption of machine learning, especially for LLMs, with which a large number of end users have direct interactions. Research has been significantly lagged behind, partly due to the fact that some issues become more significant for LLMs than they are for the usual machine learning models. The following are an incomplete list of research directions that we believe require significant investments in the near future.

- Data privacy. For usual machine learning models, their training data are obtained beforehand, with many of them being made available to the public. Notable examples include the ImageNet dataset. That is, the privacy and copyright issues for the training data were not as serious. However, LLMs' training data come directly from the internet, many of which are private information and do not have the authorisations from the data owners. On top of this, various techniques, such as prompt injection and privacy attacks, are available to leak the information. It requires a multi-disciplinary approach to deal with the data privacy issue.
- Safety and trustworthiness implications. Currently, research is focused on tricking the LLMs to generate unexpected outcomes. There needs to be systematic approaches to study and measure the certain to which such unexpected outcomes might lead to bad consequences. This requires the modelling of the environment (e.g., an organisation) in which the LLMs are used, including how they are used and the consequences of all possible outcomes. A systematic study will enable the understanding of which aspects of alignments are needed and how to fine-tune the LLMs to different applications domains.
- Rigorous engineering. The LLMs, in its current development, are mostly relying on the massive training data and the exceptional computational power owned by the large tech giants. Its performance currently is measured with various small scale benchmark datasets that are designed for the domain specific aspects of the abilities, for example, the mathematics, the reasoning, and so on. A rigorous engineering approach, by considering the entire development cycle including the evaluation, is needed to support the shifting of the development from extensive mode to intensive mode, and for the benefit of providing assurance cases for the applications of LLMs to safety critical domains.
- Verification with provable guarantees. Empirical evaluation provides certain evidence about the performance, but cannot be regarded as a rigorous justification, especially in safety critical domains. A mathematically well-founded proof about the performance, e.g., in the form of statistical guarantees such as chain constraint (Bensalem et al. 2023), can be useful for improving the confidence of the users.
- Regulations and standards. The necessity of regulations has been commonly agreed. However, the regulations do not provide workable measures that are usually recommended in industrial standards. Compliance with regulations and standards is an important part of an assurance case to justify the safety of a product. It is urgent for the community to come up with standards so as to release the full potential of LLMs and AI in general.

10 Conclusions

This paper provides an overview of the known vulnerabilities of LLMs, and discusses how the V&V techniques might be adapted to work with them. Given the LLMs are quickly adopted by applications that have direct or indirect interactions with end users, it is imperative that the deployed LLMs undergoes sufficient verdict processes to avoid any undesirable safety and trustworthy consequences. Novel V&V techniques are called for, to deal with the special characteristics of the LLMs such as the nondeterministic behaviours, the model sizes that are significantly larger than the usual machine learning models, the training dataset that is obtained from internet rather than through a careful collection process, etc. Multi-disciplinary development is needed to make sure that all trustworthy issues are fully considered and tackled.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 956123. It is also financially supported by the U.K. EPSRC through End-to-End Conceptual Guarding of Neural Architectures [EP/T026995/1].

Author contributions X.H. organises the overall structure and writes several sections of this paper, and others contribute to different sections.

Declarations

Competing interests The authors declare no competing interests.

References

(2004) Quality management systems—process validation guidance. https://www.imdrf.org/sites/default/files/docs/ghtf/final/sg3/technical-docs/ghtf-sg3-n99-10-2004-qms-process-guidance-04010.pdf. GHTF. Accessed 20 Aug 2023
(2018) Ethics guidelines for trustworthy AI. https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html. European Commission. Accessed 20 Aug 2023
(2018) The data protection act. https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted. Accessed 20 Aug 2023
(2021) China's regulations on the administration of deep synthesis internet information services. https://www.chinalawtranslate.com/en/deep-synthesis/. Accessed 20 Aug 2023
(2022) AI risk management framework. https://www.nist.gov/itl/ai-risk-management-framework. Accessed 20 Aug 2023
(2022) China's regulations on recommendation algorithms. http://www.cac.gov.cn/2022-01/04/c_1642894606258238.htm. Accessed 20 Aug 2023
(2022) Content at scale. https://contentatscale.ai/ai-content-detector/. Accessed 20 Aug 2023
(2022) Copyleaks. https://copyleaks.com/ai-content-detector. Accessed 20 Aug 2023

(2022) New meta AI demo writes racist and inaccurate scientific literature, gets pulled. https://arstechnica.com/information-technology/2022/11/after-controversy-meta-pulls-demo-of-ai-model-that-writes-scientific-papers/. Accessed 20 Aug 2023
(2022) Originality AI. https://originality.ai. Accessed 20 Aug 2023
(2022) Prompt injection attacks against GPT-3. https://simonwillison.net/2022/Sep/12/prompt-injection/. Accessed 20 Aug 2023
(2023) 'He would still be here': man dies by suicide after talking with AI chatbot, widow says. https://www.vice.com/en/article/pkadgm/man-dies-by-suicide-after-talking-with-ai-chatbot-widow-says. Accessed 23 Aug 2023
(2023) A pro-innovation approach to AI regulation. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1146542/a_pro-innovation_approach_to_AI_regulation.pdf. Accessed 20 Aug 2023
(2023) Blueprint for an AI bill of rights. https://www.whitehouse.gov/ostp/ai-bill-of-rights/. Accessed 20 Aug 2023
(2023) ChatGPT: get instant answers, find creative inspiration, and learn something new. https://openai.com/chatgpt. Accessed 20 Aug 2023
(2023) ChatGPT: US lawyer admits using AI for case research. https://www.bbc.co.uk/news/world-us-canada-65735769. Accessed 23 Aug 2023
(2023) China's algorithm registry. https://beian.cac.gov.cn/#/index. Accessed 20 Aug 2023
(2023) EU AI act. https://artificialintelligenceact.eu. Accessed 20 Aug 2023
(2023) EU data act. https://ec.europa.eu/commission/presscorner/detail/en/ip_22_1113. Accessed 20 Aug 2023
(2023) Prompt leaking. https://learnprompting.org/docs/prompt_hacking/leaking. Accessed 20 Aug 2023
(2023) Responsible AI principles from Microsoft. https://www.microsoft.com/en-us/ai/responsible-ai. Accessed 20 Aug 2023
(2023) Three Samsung employees reportedly leaked sensitive data to ChatGPT. https://www.engadget.com/three-samsung-employees-reportedly-leaked-sensitive-data-to-chatgpt-190221114.html. Accessed 20 Aug 2023
(2023) Understanding artificial intelligence ethics and safety: a guide for the responsible design and implementation of AI systems in the public sector. https://www.turing.ac.uk/news/publications/understanding-artificial-intelligence-ethics-and-safety. Accessed 20 Aug 2023
Aghakhani H, Dai W, Manoel A, Fernandes X, Kharkar A, Kruegel C, Vigna G, Evans D, Zorn B, Sim R (2023) TrojanPuzzle: covertly poisoning code-suggestion models. arXiv Preprint http://arxiv.org/abs/2301.02344
Agrawal M, Hegselmann S, Lang H, Kim Y, Sontag D (2022) Large language models are zero-shot clinical information extractors. arXiv Preprint http://arxiv.org/abs/2205.12689
Aiyappa, R An J, Kwak H, Ahn Y-Y (2023) Can we trust the evaluation on ChatGPT? arXiv Preprint http://arxiv.org/abs/2303.12767
Akopyan F, Sawada J, Cassidy A, Alvarez-Icaza R, Arthur J, Merolla P, Imam N, Nakamura Y, Datta P, Nam G-J et al (2015) TrueNorth: design and tool flow of a 65 MW 1 million neuron programmable neurosynaptic chip. IEEE Trans Comput Aided Des Integr Circuits Syst 34(10):1537–1557
Alshiekh M, Bloem R, Ehlers R, Könighofer B, Niekum S, Topcu U (2018) Safe reinforcement learning via shielding. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
Alzantot M, Sharma Y, Elgohary A, Ho B-J, Srivastava M, Chang K-W (2018) Generating natural language adversarial examples. arXiv Preprint http://arxiv.org/abs/1804.07998
Arora U, Huang W, He H (2021) Types of out-of-distribution texts and how to detect them. arXiv Preprint http://arxiv.org/abs/2109.06827
Bai Y, Jones A, Ndousse K, Askell A, Chen A, DasSarma N, Drain D, Fort S, Ganguli D, Henighan T et al (2022a) Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv Preprint http://arxiv.org/abs/2204.05862
Bai Y, Kadavath S, Kundu S, Askell A, Kernion J, Jones A, Chen A, Goldie A, Mirhoseini A, McKinnon C et al (2022b) Constitutional AI: harmlessness from AI feedback. arXiv Preprint http://arxiv.org/abs/2212.08073
Balaji Y, Nah S, Huang X, Vahdat A, Song J, Kreis K, Aittala M, Aila T, Laine S, Catanzaro B et al (2022) eDiff-I: text-to-image diffusion models with an ensemble of expert denoisers. arXiv Preprint http://arxiv.org/abs/2211.01324
Balakrishnan A, Puranic AG, Qin X, Dokhanchi A, Deshmukh JV, Ben Amor H, Fainekos G (2019) Specifying and evaluating quality metrics for vision-based perception systems. In: Design, automation & test in Europe conference & exhibition (DATE). pp 1433–1438

Bang Y, Cahyawijaya S, Lee N, Dai W, Su D, Wilie B, Lovenia H, Ji Z, Yu T, Chung W et al (2023) A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. arXiv Preprint http://arxiv.org/abs/2302.04023

Bartocci E, Falcone Y (2018) Lectures on runtime verification. Springer

Bauer A, Leucker M, Schallhart C (2011) Runtime verification for LTL and TLTL. ACM Trans Softw Eng Methodol 20(4):1–64

Belinkov Y, Bisk Y (2017) Synthetic and natural noise both break neural machine translation. arXiv Preprint http://arxiv.org/abs/1711.02173

Bensalem S, Lakhnech Y, Saidi H (1996) Powerful techniques for the automatic generation of invariants. In: Computer aided verification: 8th international conference, CAV'96 New Brunswick, NJ, USA, July 31–August 3, 1996 proceedings 8. Springer, pp 323–335

Bensalem S, Lakhnech Y, Owre S (1998) Invest: a tool for the verification of invariants. In: Computer aided verification: 10th international conference, CAV'98 Vancouver, BC, Canada, June 28–July 2, 1998 proceedings 10. Springer, pp 505–510

Bensalem S, Cheng C-H, Huang X, Katsaros P, Molin A, Nickovic D, Peled D (2022) Formal specification for learning-enabled autonomous systems. In: International workshop on numerical software verification. Springer, pp 131–143

Bensalem S, Cheng C-H, Huang W, Huang X, Wu C, Zhao X (2023) What, indeed, is an achievable provable guarantee for learning-enabled safety critical systems. In: ISoLA 2023

Berthier N, Alshareef A, Sharp J, Schewe S, Huang X (2021) Abstraction and symbolic execution of deep neural networks with Bayesian approximation of hidden features. arXiv Preprint http://arxiv.org/abs/2103.03704

Bibel W (2013) Automated theorem proving. Springer Science & Business Media, Berlin

Bitcoin energy consumption index. https://digiconomist.net/bitcoin-energy-consumption. Accessed 17 Aug 2023

Black S, Biderman S, Hallahan E, Anthony Q, Gao L, Golding L, He H, Leahy C, McDonell K, Phang J et al (2022) GPT-Neox-20B: an open-source autoregressive language model. arXiv Preprint http://arxiv.org/abs/2204.06745

Bonaert G, Dimitrov DI, Baader M, Vechev M (2021) Fast and precise certification of transformers. In: Proceedings of the 42nd ACM SIGPLAN international conference on programming language design and implementation. pp 466–481

Borji A (2023) A categorical archive of ChatGPT failures. CoRR. http://arxiv.org/abs/2302.03494

Botacin M (2023) GPThreats-3: is automatic malware generation a threat? In: 2023 IEEE security and privacy workshops (SPW). pp 238–254

Brants T, Popat AC, Xu P, Och FJ, Dean J (2007) Large language models in machine translation. In: Eisner J (ed) EMNLP-CoNLL 2007, proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, June 28–30, 2007, Prague, Czech Republic. ACL, pp 858–867

Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020a) Language models are few-shot learners. In: Proceedings of the 34th international conference on neural information processing systems, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020b) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020c) Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in neural information processing systems, vol 33. Curran Associates, Inc., pp 1877–1901

Bullwinkle M, Urban E (2023) Introduction to red teaming large language models (LLMS). https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/red-teaming. Accessed 20 Aug 2023

Bursztein E (2018) Attacks against machine learning—an overview. https://elie.net/blog/ai/attacks-against-machine-learning-an-overview/. Accessed 20 Aug 2023

Cambiaso E, Caviglione L (2023) Scamming the scammers: using ChatGPT to reply mails for wasting time and resources. arXiv Preprint http://arxiv.org/abs/2303.13521

Cao Y, Li D, Fang M, Zhou T, Gao J, Zhan Y, Tao D (2022) TASA: deceiving question answering models by twin answer sentences attack. arXiv Preprint http://arxiv.org/abs/2210.15221
Carlini N, Jagielski M, Choquette-Choo CA, Paleka D, Pearce W, Anderson H, Terzis A, Thomas K, Tramèr F (2023) Poisoning web-scale training datasets is practical. arXiv Preprint http://arxiv.org/abs/2302.10149
Chen B, Carvalho W, Baracaldo N, Ludwig H, Edwards B, Lee T, Molloy I, Srivastava B (2019) Detecting backdoor attacks on deep neural networks by activation clustering. In: SafeAI@ AAAI
Chen M, Tworek J, Jun H, Yuan Q, de Oliveira Pinto HP, Kaplan J, Edwards H, Burda Y, Joseph N, Brockman G et al (2021a) Evaluating large language models trained on code. arXiv Preprint http://arxiv.org/abs/2107.03374
Chen X, Salem A, Chen D, Backes M, Ma S, Shen Q, Wu Z, Zhang Y (2021b) BadNL: backdoor attacks against NLP models with semantic-preserving improvements. In: Annual computer security applications conference. pp 554–569
Chen S, Bi X, Gao R, Sun X (2022) Holistic sentence embeddings for better out-of-distribution detection. In: Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics, pp 6676–6686
Chen L, Zaharia M, Zou J (2023a) How is ChatGPT's behavior changing over time? arXiv Preprint http://arxiv.org/abs/2307.09009
Chen S, Yang W, Bi X, Sun X (2023b) Fine-tuning deteriorates general textual out-of-distribution detection by distorting task-agnostic features. In: Findings of the Association for Computational Linguistics: EACL 2023. pp 552–567
Chen S, Kann BH, Foote MB, Aerts HJ, Savova GK, Mak RH, Bitterman DS (2023c) The utility of ChatGPT for cancer treatment information. medRxiv, pp 2023–03
Cheng Y, Jiang L, Macherey W (2019a) Robust neural machine translation with doubly adversarial inputs. arXiv Preprint http://arxiv.org/abs/1906.02443
Cheng C, Nührenberg G, Yasuoka H (2019b) Runtime monitoring neuron activation patterns. In: DATE2019. pp 300–303
Cheng M, Yi J, Chen P-Y, Zhang H, Hsieh C-J (2020) Seq2Sick: evaluating the robustness of sequence-to-sequence models with adversarial examples. In: Proceedings of the AAAI conference on artificial intelligence, vol 34. pp 3601–3608
Cheng C-H, Wu C, Seferis E, Bensalem S (2022) Prioritizing corners in OOD detectors via symbolic string manipulation. In: Bouajjani A, Holík L, Wu Z (eds) Automated technology for verification and analysis. Springer International Publishing, Cham, pp 397–413
Chiang W-L, Li Z, Lin Z, Sheng Y, Wu Z, Zhang H, Zheng L, Zhuang S, Zhuang Y, Gonzalez JE et al (2023) Vicuna: an open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. See https://vicuna.lmsys.org. Accessed 14 Apr 2023
Cho JH, Hariharan B (2019) On the efficacy of knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp 4794–4802
Cho H, Park C, Kang J, Yoo KM, Kim T, Lee S-G (2022) Enhancing out-of-distribution detection in natural language understanding via implicit layer ensemble. In: Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics, pp 783–798
Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, Roberts A, Barham P, Chung HW, Sutton C, Gehrmann S et al (2022) PaLM: scaling language modeling with pathways. arXiv Preprint http://arxiv.org/abs/2204.02311
Christiano PF, Leike J, Brown T, Martic M, Legg S, Amodei D (2017) Deep reinforcement learning from human preferences. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates, Inc
Clark K, Luong M-T, Le QV, Manning CD (2020) Electra: pre-training text encoders as discriminators rather than generators. arXiv Preprint http://arxiv.org/abs/2003.10555
Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R et al (2021) Training verifiers to solve math word problems. arXiv Preprint http://arxiv.org/abs/2110.14168
Cohen J, Rosenfeld E, Kolter Z (2019) Certified adversarial robustness via randomized smoothing. In: International conference on machine learning. PMLR, pp 1310–1320
Croce F, Hein M (2020) Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning. PMLR, pp 2206–2216
Dai J, Chen C, Li Y (2019) A backdoor attack against LSTM-based text classification systems. IEEE Access 7:138872–138878

Dan S, Roth D (2021) On the effects of transformer size on in-and out-of-domain calibration. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp 2096–2101
Davies M, Srinivasa N, Lin T-H, Chinya G, Cao Y, Choday SH, Dimou G, Joshi P, Imam N, Jain S et al (2018) Loihi: a neuromorphic manycore processor with on-chip learning. IEEE Micro 38(1):82–99
De Moura L, Bjørner N (2008) Z3: an efficient SMT solver. In: Tools and algorithms for the construction and analysis of systems: 14th international conference, TACAS 2008, held as part of the joint European conferences on theory and practice of software, ETAPS 2008, Budapest, Hungary, March 29–April 6, 2008. Proceedings 14. Springer, pp 337–340
De Vries A, Gallersdörfer U, Klaaßen L, Stoll C (2022) Revisiting bitcoin's carbon footprint. Joule 6(3):498–502
Desai S, Durrett G (2020) Calibration of pre-trained transformers. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), Online, Nov. 2020. Association for Computational Linguistics, pp 295–302
Deshpande A, Murahari V, Rajpurohit T, Kalyan A, Narasimhan K (2023) Toxicity in ChatGPT: analyzing persona-assigned language models. arXiv Preprint http://arxiv.org/abs/2304.05335
Dettmers T, Lewis M, Belkada Y, Zettlemoyer L (2022) GPT3. int8 (): 8-bit matrix multiplication for transformers at scale. In: Advances in neural information processing systems, vol 35. pp 30318–30332
Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. arXiv Preprint http://arxiv.org/abs/1810.04805
Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: human language technologies, volume 1 (long and short papers), Minneapolis, Minnesota, June 2019. Association for Computational Linguistics, pp 4171–4186
DeVries T, Taylor GW (2018) Learning confidence for out-of-distribution detection in neural networks. arXiv Preprint http://arxiv.org/abs/1802.04865
Dey N (2023) GPT: a family of open, compute-efficient, large language models. https://www.cerebras.net/blog/cerebras-gpt-a-family-of-open-compute-efficient-large-language-models/. Accessed 20 Aug 2023
Dodge J, Ilharco G, Schwartz R, Farhadi A, Hajishirzi H, Smith N (2020) Fine-tuning pretrained language models: weight initializations, data orders, and early stopping. arXiv Preprint http://arxiv.org/abs/2002.06305
Du T, Ji S, Shen L, Zhang Y, Li J, Shi J, Fang C, Yin J, Beyah R, Wang T (2021) CERT-RNN: towards certifying the robustness of recurrent neural networks. CCS 21(2021):15–19
Du N, Huang Y, Dai AM, Tong S, Lepikhin D, Xu Y, Krikun M, Zhou Y, Yu AW, Firat O et al (2022) GLaM: efficient scaling of language models with mixture-of-experts. In: International conference on machine learning. PMLR, pp 5547–5569
Duan H, Yang Y, Abbasi A, Tam KY (2022) BARLE: background-aware representation learning for background shift out-of-distribution detection. In: Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics, pp 750–764
Duan J, Kong F, Wang S, Shi X, Xu K (2023) Are diffusion models vulnerable to membership inference attacks? arXiv Preprint http://arxiv.org/abs/2302.01316
Dudley JJ, Kristensson PO (2018) A review of user interface design for interactive machine learning. ACM Trans Interact Intell Syst 8(2):1–37
E2Analyst (2023) GPT-4: everything you want to know about OpenAI's new AI model. https://medium.com/predict/gpt-4-everything-you-want-to-know-about-openais-new-ai-model-a5977b42e495. Accessed 20 Aug 2023
Ebrahimi J, Rao A, Lowd D, Dou D (2017) HotFlip: white-box adversarial examples for text classification. arXiv Preprint http://arxiv.org/abs/1712.06751
Edwards B (2023) Study claims ChatGPT is losing capability, but some experts aren't convinced. https://arstechnica.com/information-technology/2023/07/is-chatgpt-getting-worse-over-time-study-claims-yes-but-others-arent-sure/. Accessed 20 Aug 2023
Eppstein D (1996) Zonohedra and zonotopes. Math Educ Res 5(4):15–21
Esser P, Rombach R, Ommer B (2021) Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 12873–12883
EU GDPR (2016). https://gdpr-info.eu. Accessed 20 Aug 2023
Farhat F, Sohail S, Madsen D (2023) How trustworthy is ChatGPT? The case of bibliometric analyses. Cogent Eng 10:06

Fedus W, Zoph B, Shazeer N (2021) Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. J Mach Learn Res 23:1–40
Feinman R, Curtin RR, Shintre S, Gardner AB (2017) Detecting adversarial samples from artifacts. arXiv Preprint http://arxiv.org/abs/1703.00410
Fitting M (1996) First-order logic and automated theorem proving. Graduate texts in computer science, second edn. Springer
Frantar E, Alistarh D (2022) Optimal brain compression: a framework for accurate post-training quantization and pruning. arXiv Preprint http://arxiv.org/abs/2208.11580
Frantar E, Ashkboos S, Hoefler T, Alistarh D (2023) GPTQ: accurate quantization for generative pre-trained transformers. In: International conference on learning representations
Frieder S, Pinchetti L, Griffiths R-R, Salvatori T, Lukasiewicz T, Petersen PC, Chevalier A, Berner J (2023) Mathematical capabilities of ChatGPT. arXiv Preprint http://arxiv.org/abs/2301.13867
Gangal V, Arora A, Einolghozati A, Gupta S (2020) Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. In: Proceedings of the AAAI conference on artificial intelligence, vol 34. pp 7764–7771
Ganguli D, Askell A, Schiefer N, Liao T, Lukošiūtė K, Chen A, Goldie A, Mirhoseini A, Olsson C, Hernandez D et al (2023) The capacity for moral self-correction in large language models. arXiv Preprint http://arxiv.org/abs/2302.07459
Gao J, Lanchantin J, Soffa ML, Qi Y (2018) Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE security and privacy workshops (SPW). IEEE, pp 50–56
Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G (2023) PAL: program-aided language models
Garcıa J, Fernández F (2015) A comprehensive survey on safe reinforcement learning. J Mach Learn Res 16(1):1437–1480
Goodfellow I, Papernot N (2017) The challenge of verification and testing of machine learning. Cleverhans-blog
Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv Preprint http://arxiv.org/abs/1412.6572
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
Goodin D (2023) Hackers are selling a service that bypasses ChatGPT restrictions on malware. https://arstechnica.com/information-technology/2023/02/now-open-fee-based-telegram-service-that-uses-chatgpt-to-generate-malware/. Accessed 20 Aug 2023
Gopinath D, Wang K, Zhang M, Pasareanu CS, Khurshid S (2018) Symbolic execution for deep neural networks. arXiv Preprint http://arxiv.org/abs/1807.10439
Gou J, Yu B, Maybank SJ, Tao D (2021) Knowledge distillation: a survey. Int J Comput Vis 129:1789–1819
Gowal S, Dvijotham K, Stanforth R, Bunel R, Qin C, Uesato J, Arandjelovic R, Mann T, Kohli P (2018) On the effectiveness of interval bound propagation for training verifiably robust models. arXiv Preprint http://arxiv.org/abs/1810.12715
Goyal S, Doddapaneni S, Khapra MM, Ravindran B (2022) A survey in adversarial defences and robustness in NLP. arXiv Preprint http://arxiv.org/abs/2203.06414
GPT-4's details are leaked. https://archive.md/2RQ8X. Accessed 17 Aug 2023
Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M (2023) More than you've asked for: a comprehensive analysis of novel prompt injection threats to application-integrated large language models. arXiv Preprint http://arxiv.org/abs/2302.12173
Gu T, Liu K, Dolan-Gavitt B, Garg S (2019) BadNets: evaluating backdooring attacks on deep neural networks. IEEE Access 7:47230–47244
Gu J-C, Li T, Liu Q, Ling Z-H, Su Z, Wei S, Zhu X (2020) Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 29th ACM international conference on information & knowledge management, CIKM '20, New York, NY, USA, 2020. Association for Computing Machinery, pp 2041–2044
Gu S, Yang L, Du Y, Chen G, Walter F, Wang J, Yang Y, Knoll A (2022) A review of safe reinforcement learning: methods, theory and applications. arXiv Preprint http://arxiv.org/abs/2205.10330
Gu Y, Dong L, Wei F, Huang M (2023a) Knowledge distillation of large language models. arXiv Preprint http://arxiv.org/abs/2306.08543
Gu S, Kshirsagar A, Du Y, Chen G, Yang Y, Peters J, Knoll A (2023b) A human-centered safe robot reinforcement learning framework with interactive behaviors. arXiv Preprint http://arxiv.org/abs/2302.13137
Gunning D, Stefik M, Choi J, Miller T, Stumpf S, Yang G-Z (2019) XAI—explainable artificial intelligence. Sci Robot 4(37):eaay7120

Guo B, Zhang X, Wang Z, Jiang M, Nie J, Ding Y, Yue J, Wu Y (2023) How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. CoRR. abs/2301.07597
He R, Sun S, Yang J, Bai S, Qi X (2022) Knowledge distillation as efficient pre-training: faster convergence, higher data-efficiency, and better transferability. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 9161–9171
Hendrycks D, Gimpel K (2016) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International conference on learning representations
Hendrycks D, Liu X, Wallace E, Dziedzic A, Krishnan R, Song D (2020) Pretrained transformers improve out-of-distribution robustness. In: Proceedings of the 58th annual meeting of the association for computational linguistics. pp 2744–2751
Henzinger TA, Lukina A, Schilling C (2020) Outside the box: abstraction-based monitoring of neural networks. In: ECAI2020
Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv Preprint http://arxiv.org/abs/1503.02531
Hintze A (2023) ChatGPT believes it is conscious. arXiv Preprint http://arxiv.org/abs/2304.12898
Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, de Las Casas D, Hendricks LA, Welbl J, Clark A et al (2022) Training compute-optimal large language models. arXiv Preprint http://arxiv.org/abs/2203.15556
Holmes J, Liu Z, Zhang L, Ding Y, Sio TT, McGee LA, Ashman JB, Li X, Liu T, Shen J et al (2023) Evaluating large language models on a highly-specialized topic, radiation oncology physics. arXiv Preprint http://arxiv.org/abs/2304.01938
Hosseini H, Kannan S, Zhang B, Poovendran R (2017) Deceiving Google's perspective API built for detecting toxic comments. arXiv Preprint http://arxiv.org/abs/1702.08138
Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S (2019) Parameter-efficient transfer learning for NLP. In: International conference on machine learning. PMLR, pp 2790–2799
Hrinchuk O, Popova M, Ginsburg B (2020) Correction of automatic speech recognition with transformer sequence-to-sequence model. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7074–7078
Hu Z, Yang Z, Liang X, Salakhutdinov R, Xing EP (2017) Toward controlled generation of text. In: International conference on machine learning. PMLR, pp 1587–1596
Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W (2021) LoRA: low-rank adaptation of large language models. arXiv Preprint http://arxiv.org/abs/2106.09685
Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W (2022) LoRA: low-rank adaptation of large language models. In: International conference on learning representations
Huang X, Jin G, Ruan W (2012) Machine learning basics. In: Machine learning safety. Springer, pp 3–13
Huang X, Kwiatkowska M, Wang S, Wu M (2017) Safety verification of deep neural networks. In: Majumdar R, Kuncak V (eds) Computer aided verification—29th international conference, CAV 2017, Heidelberg, Germany, July 24–28, 2017, proceedings, part I, volume 10426 of lecture notes in computer science. Springer, pp 3–29
Huang P-S, Stanforth R, Welbl J, Dyer C, Yogatama D, Gowal S, Dvijotham K, Kohli P (2019a) Achieving verified robustness to symbol substitutions via interval bound propagation. arXiv Preprint http://arxiv.org/abs/1909.01492
Huang X, Alzantot M, Srivastava M (2019b) NeuronInspect: detecting backdoors in neural networks via output explanations. arXiv Preprint http://arxiv.org/abs/1911.07399
Huang X, Kroening D, Ruan W, Sharp J, Sun Y, Thamo E, Wu M, Yi X (2020a) A survey of safety and trustworthiness of deep neural networks: verification, testing, adversarial attack and defence, and interpretability. Comput Sci Rev 37:100270
Huang H, Li Z, Wang L, Chen S, Dong B, Zhou X (2020b) Feature space singularity for out-of-distribution detection. arXiv Preprint http://arxiv.org/abs/2011.14654
Huang W, Sun Y, Zhao X, Sharp J, Ruan W, Meng J, Huang X (2021) Coverage-guided testing for recurrent neural networks. IEEE Trans Reliab 71(3):1191–1206
Huang X, Ruan W, Tang Q, Zhao X (2022a) Bridging formal methods and machine learning with global optimisation. In: Formal methods and software engineering: 23rd international conference on formal engineering methods, ICFEM 2022, Madrid, Spain, October 24–27, 2022, proceedings. Springer-Verlag, Berlin, Heidelberg, pp 1–19
Huang W, Zhao X, Banks A, Cox V, Huang X (2022b) Hierarchical distribution-aware testing of deep learning. arXiv Preprint http://arxiv.org/abs/2205.08589
Huang W, Zhao X, Jin G, Huang X (2022c) Safari: versatile and efficient evaluations for robustness of interpretability. arXiv Preprint http://arxiv.org/abs/2208.09418

Ilyas A, Santurkar S, Tsipras D, Engstrom L, Tran B, Madry A (2019) Adversarial examples are not bugs, they are features. In: Advances in neural information processing systems, vol 32
Italy became the first western country to ban ChatGPT. https://www.cnbc.com/2023/04/04/italy-has-banned-chatgpt-heres-what-other-countries-are-doing.html. Accessed 17 Aug 2023
Ivankay A, Girardi I, Marchiori C, Frossard P (2022) Fooling explanations in text classifiers. arXiv Preprint http://arxiv.org/abs/2206.03178
Iyyer, M Wieting J, Gimpel K, Zettlemoyer L (2018) Adversarial example generation with syntactically controlled paraphrase networks. arXiv Preprint http://arxiv.org/abs/1804.06059
Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F (2020) A survey on contrastive self-supervised learning. Technologies 9(1):2
Jang M, Lukasiewicz T (2023) Consistency analysis of ChatGPT. arXiv Preprint http://arxiv.org/abs/2303.06273
Jansen N, Könighofer B, Junges S, Bloem R (2018) Shielded decision-making in MDPs. arXiv Preprint http://arxiv.org/abs/1807.06096
Jansen N, Könighofer B, Junges J, Serban A, Bloem R (2020) Safe reinforcement learning using probabilistic shields. Schloss Dagstuhl, Dagstuhl
Ji Y, Gong Y, Peng Y, Ni C, Sun P, Pan D, Ma B, Li X (2023) Exploring ChatGPT's ability to rank content: a preliminary study on consistency with human preferences
Jia R, Liang P (2017) Adversarial examples for evaluating reading comprehension systems. arXiv Preprint http://arxiv.org/abs/1707.07328
Jia R, Raghunathan A, Göksel K, Liang P (2019) Certified robustness to adversarial word substitutions. arXiv Preprint http://arxiv.org/abs/1909.00986
Jiang AQ, Welleck S, Zhou JP, Li W, Liu J, Jamnik M, Lacroix T, Wu Y, Lample G (2022) Draft, sketch, and prove: guiding formal theorem provers with informal proofs. arXiv Preprint http://arxiv.org/abs/2210.12283
Jiao W, Wang W, Huang J-t, Wang X, Tu Z (2023) Is ChatGPT a good translator? A preliminary study. arXiv Preprint http://arxiv.org/abs/2301.08745
Jin D, Jin Z, Zhou JT, Szolovits P (2020) Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In: Proceedings of the AAAI conference on artificial intelligence, vol 34. pp 8018–8025
Kalyan KS, Rajasekharan A, Sangeetha S (2021) AMMUS: a survey of transformer-based pretrained models in natural language processing. arXiv Preprint http://arxiv.org/abs/2108.05542
Kambhampati S (2022) Changing the nature of AI research. Commun ACM 65(9):8–9
Kande R, Pearce H, Tan B, Dolan-Gavitt B, Thakur S, Karri R, Rajendran J (2023) LLM-assisted generation of hardware assertions. CoRR. abs/2306.14027
Kang D, Li X, Stoica I, Guestrin C, Zaharia M, Hashimoto T (2023a) Exploiting programmatic behavior of LLMS: dual-use through standard security attacks. arXiv Preprint http://arxiv.org/abs/2302.05733
Kang Y, Zhang Q, Roth R (2023b) The ethics of AI-generated maps: a study of DALLE 2 and implications for cartography. arXiv Preprint http://arxiv.org/abs/2304.10743
Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, Gray S, Radford A, Wu J, Amodei D (2020) Scaling laws for neural language models. arXiv Preprint http://arxiv.org/abs/2001.08361
Katz DM, Bommarito MJ, Gao S, Arredondo P (2023) GPT-4 passes the bar exam. Available at SSRN 4389233
Khoury R, Avila AR, Brunelle J, Camara BM (2023) How secure is code generated by ChatGPT? arXiv Preprint http://arxiv.org/abs/2304.09655
Kim Y-M (2023) Data and fair use. Korea Copyright Commission 141:5–53
Ko C-Y, Lyu Z, Weng L, Daniel L, Wong N, Lin D (2019) POPQORN: quantifying robustness of recurrent neural networks. In: International conference on machine learning. PMLR, pp 3468–3477
Koh JY, Fried D, Salakhutdinov R (2023) Generating images with multimodal language models. arXiv Preprint http://arxiv.org/abs/2305.17216
Kuleshov V, Thakoor S, Lau T, Ermon S (2018) Adversarial examples for natural language classification problems. arXiv Preprint
Kumar A, Ahuja K, Vadapalli R, Talukdar P (2020) Syntax-guided controlled generation of paraphrases. Trans Assoc Comput Linguist 8:330–345
Kurita K, Michel P, Neubig G (2020) Weight poisoning attacks on pretrained models. In: Proceedings of the 58th annual meeting of the association for computational linguistics. pp 2793–2806
La Malfa E, Wu M, Laurenti L, Wang B, Hartshorn A, Kwiatkowska M (2020) Assessing robustness of text classification through maximal safe radius computation. arXiv Preprint http://arxiv.org/abs/2010.02004
Lam M, Sethi R, Ullman JD, Aho A (2006) Compilers: principles, techniques, and tools. Pearson Education

Lambert N, Castricato L, von Werra L, Havrilla A (2022) Illustrating reinforcement learning from human feedback (RLHF). Hugging Face Blog. https://huggingface.co/blog/rlhf
Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: a lite BERT for self-supervised learning of language representations. arXiv Preprint http://arxiv.org/abs/1909.11942
Lee P (2016) Learning from Tay's introduction. https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/. Accessed 20 Aug 2023
Lee JY (2023) Can an artificial intelligence chatbot be the author of a scholarly article? J Educ Eval Health Prof 20:6
Lee C, Cho K, Kang W (2019) Mixout: effective regularization to finetune large-scale pretrained language models. arXiv Preprint http://arxiv.org/abs/1909.11299
Lee N, Bang Y, Madotto A, Fung P (2020) Misinformation has high perplexity. arXiv Preprint http://arxiv.org/abs/2006.04666
Lee K, Liu H, Ryu M, Watkins O, Du Y, Boutilier C, Abbeel P, Ghavamzadeh M, Gu SS (2023) Aligning text-to-image models using human feedback. arXiv Preprint http://arxiv.org/abs/2302.12192
Lei Y, Cao Y, Li D, Zhou T, Fang M, Pechenizkiy M (2022) Phrase-level textual adversarial attack with label preservation. arXiv Preprint http://arxiv.org/abs/2205.10710
Lepikhin D, Lee H, Xu Y, Chen D, Firat O, Huang Y, Krikun M, Shazeer N, Chen Z (2020) GShard: scaling giant models with conditional computation and automatic sharding. arXiv Preprint http://arxiv.org/abs/2006.16668
Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2020) BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th annual meeting of the association for computational linguistics, Online, July 2020. Association for Computational Linguistics, pp 7871–7880
Li J, Ji S, Du T, Li B, Wang T (2018a) TextBugger: generating adversarial text against real-world applications. arXiv Preprint http://arxiv.org/abs/1812.05271
Li Y, Ding L, Gao X (2018b) On the decision boundary of deep neural networks. arXiv Preprint http://arxiv.org/abs/1808.05385
Li S, Liu H, Dong T, Zhao BZH, Xue M, Zhu H, Lu J (2021a) Hidden backdoors in human-centric language models. In: CCS '21: 2021 ACM SIGSAC conference on computer and communications security, virtual event, Republic of Korea, November 15–19, 2021. ACM, pp 3123–3140
Li X, Li J, Sun X, Fan C, Zhang T, Wu F, Meng Y, Zhang J (2021b) kFolden: k-fold ensemble for out-of-distribution detection-fold ensemble for out-of-distribution detection. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp 3102–3115
Li J, Tang T, Zhao WX, Nie JY, Wen J-R (2022) Pretrained language models for text generation: a survey. arXiv Preprint http://arxiv.org/abs/2201.05273
Li J, Cheng X, Zhao WX, Nie J-Y, Wen J-R (2023a) HaluEval: a large-scale hallucination evaluation benchmark for large language models. arXiv e-prints, p arXiv–2305
Li H, Guo D, Fan W, Xu M, Song Y (2023b) Multi-step jailbreaking privacy attacks on ChatGPT. arXiv Preprint http://arxiv.org/abs/2304.05197
Liang B, Li H, Su M, Bian P, Li X, Shi W (2017) Deep text classification can be fooled. arXiv Preprint http://arxiv.org/abs/1704.08006
Liang S, Li Y, Srikant R (2018) Enhancing the reliability of out-of-distribution image detection in neural networks. In: 6th international conference on learning representations, ICLR 2018
Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan, D Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part V 13. Springer, pp 740–755
Lin Z, Xu P, Winata GI, Siddique FB, Liu Z, Shin J, Fung P (2019) CAiRE: an empathetic neural chatbot. arXiv Preprint http://arxiv.org/abs/1907.12108
Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized BERT pretraining approach. arXiv Preprint http://arxiv.org/abs/1907.11692
Liu W, Wang X, Owens J, Li Y (2020) Energy-based out-of-distribution detection. Adv Neural Inf Process Syst 33:21464–21475
Liu C, Arnon T, Lazarus C, Strong C, Barrett C, Kochenderfer MJ et al (2021a) Algorithms for verifying deep neural networks. Found Trends Optim 4(3–4):244–404
Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, Tang J (2021b) Self-supervised learning: generative or contrastive. IEEE Trans Knowl Data Eng 35(1):857–876
Liu Z, Wang Y, Han K, Zhang W, Ma S, Gao W (2021c) Post-training quantization for vision transformer. Adv Neural Inf Process Syst 34:28092–28103

Liu Y, Han T, Ma S, Zhang J, Yang Y, Tian J, He H, Li A, He M, Liu Z et al (2023a) Summary of ChatGPT/GPT-4 research and perspective towards the future of large language models. arXiv Preprint http://arxiv.org/abs/2304.01852
Liu H, Ning R, Teng Z, Liu J, Zhou Q, Zhang Y (2023b) Evaluating the logical reasoning ability of ChatGPT and GPT-4. arXiv Preprint http://arxiv.org/abs/2304.03439
Liu J, Xia CS, Wang Y, Zhang L (2023c) Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. arXiv Preprint http://arxiv.org/abs/2305.01210
Liu Z, Yu X, Zhang L, Wu Z, Cao C, Dai H, Zhao L, Liu W, Shen D, Li Q et al (2023d) DeID-GPT: zero-shot medical text de-identification by GPT-4. arXiv Preprint http://arxiv.org/abs/2303.11032
Lou R, Zhang K, Yin W (2023) Is prompt all you need? No. A comprehensive and broader view of instruction learning. arXiv Preprint http://arxiv.org/abs/2303.10475
Madaan N, Padhi I, Panwar N, Saha D (2021) Generate your counterfactuals: towards controlled counterfactual generation for text. In: Proceedings of the AAAI conference on artificial intelligence, vol 35. pp 13516–13524
Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv Preprint http://arxiv.org/abs/1706.06083
Malinka K, Peresíni M, Firc A, Hujnák O, Janus F (2023) On the educational impact of ChatGPT: is artificial intelligence ready to obtain a university degree? In: Proceedings of the 2023 conference on innovation and technology in computer science education V. 1. pp 47–53
Manna Z, Pnueli A (2012) The temporal logic of reactive and concurrent systems: specification. Springer Science & Business Media, Berlin
March 20 ChatGPT outage: here's what happened. https://openai.com/blog/march-20-chatgpt-outage. OpenAI. Accessed 20 Aug 2023
Maus N, Chao P, Wong E, Gardner J (2023) Adversarial prompting for black box foundation models. arXiv Preprint http://arxiv.org/abs/2302.04237
McCune W (2005) Prover9 and Mace4. https://www.cs.unm.edu/~mccune/prover9/. Accessed 20 Aug 2023
Mehdi Y (2023) Announcing the next wave of AI innovation with Microsoft Bing and Edge
Min S, Lyu X, Holtzman A, Artetxe M, Lewis M, Hajishirzi H, Zettlemoyer L (2022) Rethinking the role of demonstrations: what makes in-context learning work? arXiv Preprint http://arxiv.org/abs/2202.12837
Mirman M, Gehr T, Vechev M (2018) Differentiable abstract interpretation for provably robust neural networks. In: Dy J, Krause A (eds) Proceedings of the 35th international conference on machine learning, volume 80 of proceedings of machine learning research, 10–15 July 2018. PMLR, pp 3578–3586
Mitrović S, Andreoletti D, Ayoub O (2023) ChatGPT or human? Detect and explain. Explaining decisions of machine learning model for detecting short ChatGPT-generated text
Monteiro J, Albuquerque I, Akhtar Z, Falk TH (2019) Generalizable adversarial examples detection based on bi-model decision mismatch. In: 2019 IEEE international conference on systems, man and cybernetics (SMC). IEEE, pp 2839–2844
Nagel M, Amjad RA, Van Baalen M, Louizos C, Blankevoort T (2020) Up or down? Adaptive rounding for post-training quantization. In: International conference on machine learning. PMLR, pp 7197–7206
Nelson B, Barreno M, Chi FJ, Joseph AD, Rubinstein BIP, Saini U, Sutton C, Tygar JD, Xia K (2008) Exploiting machine learning to subvert your spam filter. In: Proceedings of the 1st Usenix workshop on large-scale exploits and emergent threats, LEET'08, USA, 2008. USENIX Association
News TH (2023) WormGPT: new AI tool allows cybercriminals to launch sophisticated cyber attacks. https://thehackernews.com/2023/07/wormgpt-new-ai-tool-allows.html. Accessed 20 Aug 2023
Ni A, Iyer S, Radev D, Stoyanov V, Yih W-t, Wang S, Lin XV (2023) Lever: learning to verify language-to-code generation with execution. In: International conference on machine learning. PMLR, pp 26106–26128
Nichol A, Dhariwal P, Ramesh A, Shyam P, Mishkin P, McGrew B, Sutskever I, Chen M (2021) Glide: towards photorealistic image generation and editing with text-guided diffusion models. arXiv Preprint http://arxiv.org/abs/2112.10741
Nie Y, Williams A, Dinan E, Bansal M, Weston J, Kiela D (2019) Adversarial NLI: a new benchmark for natural language understanding. arXiv Preprint http://arxiv.org/abs/1910.14599
OpenAI (2023) GPT-4 technical report. arXiv e-prints http://arxiv.org/abs/2303.08774
OpenAI says a bug leaked sensitive ChatGPT user data. https://www.engadget.com/chatgpt-briefly-went-offline-after-a-bug-revealed-user-chat-histories-115632504.html. Engadget. Accessed 20 Aug 2023
Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A et al (2022) Training language models to follow instructions with human feedback. Adv Neural Inf Process Syst 35:27730–27744
Pan S, Luo L, Wang Y, Chen C, Wang J, Wu X (2023) Unifying large language models and knowledge graphs: a roadmap

Pang G, Shen C, Cao L, Hengel AVD (2021) Deep learning for anomaly detection: a review. ACM Comput Surv (CSUR) 54(2):1–38

Park G, Park B, Kwon SJ, Kim B, Lee Y, Lee D (2022) nuQmm: quantized MatMul for efficient inference of large-scale generative language models. arXiv Preprint http://arxiv.org/abs/2206.09557

Patterson D, Gonzalez J, Holzle U, Le Q, Liang C, Munguia L-M, Rothchild D, So DR, Texier M, Dean J (2022) The carbon footprint of machine learning training will plateau, then shrink. Computer 55(7):18–28

Pause giant AI experiments: an open letter. https://futureoflife.org/open-letter/pause-giant-ai-experiments/. Accessed 20 Aug 2023

Pearce H, Tan B, Ahmad B, Karri R, Dolan-Gavitt B (2023) Examining zero-shot vulnerability repair with large language models. In: 2023 IEEE symposium on security and privacy (SP). IEEE, pp 2339–2356

Pegoraro A, Kumari K, Fereidooni H, Sadeghi A-R (2023) To ChatGPT, or not to ChatGPT: that is the question! arXiv Preprint http://arxiv.org/abs/2304.01487

Peng B, Li C, He P, Galley M, Gao J (2023) Instruction tuning with GPT-4. arXiv Preprint http://arxiv.org/abs/2304.03277

Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp 1532–1543

Perez F, Ribeiro I (2022) Ignore previous prompt: attack techniques for language models. arXiv Preprint http://arxiv.org/abs/2211.09527

Podolskiy A, Lipin D, Bout A, Artemova E, Piontkovskaya I (2021) Revisiting Mahalanobis distance for transformer-based out-of-domain detection. In: Proceedings of the AAAI conference on artificial intelligence, vol 35. pp 13675–13682

Prompt engineering guide. https://github.com/dair-ai/Prompt-Engineering-Guide/tree/main/guides. Accessed 20 Aug 2023

Qi Y, Zhao X, Huang X (2023) Safety analysis in the era of large language models: a case study of STPA using ChatGPT. arXiv Preprint http://arxiv.org/abs/2304.01246

Radford A, Jozefowicz R, Sutskever I (2017) Learning to generate reviews and discovering sentiment. arXiv Preprint http://arxiv.org/abs/1704.01444

Radford A, Narasimhan K, Salimans T, Sutskever I et al (2018) Improving language understanding by generative pre-training. OpenAI

Rae JW, Borgeaud S, Cai T, Millican K, Hoffmann J, Song F, Aslanides J, Henderson S, Ring R, Young S et al (2021) Scaling language models: methods, analysis & insights from training Gopher. arXiv Preprint http://arxiv.org/abs/2112.11446

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 21(1):5485–5551

Ramamurthy R, Ammanabrolu P, Brantley K, Hessel J, Sifa R, Bauckhage C, Hajishirzi H, Choi Y (2022) Is reinforcement learning (not) for natural language processing?: benchmarks, baselines, and building blocks for natural language policy optimization. arXiv Preprint http://arxiv.org/abs/2210.01241

Ramesh A, Pavlov M, Goh G, Gray S, Voss C, Radford A, Chen M, Sutskever I (2021) Zero-shot text-to-image generation. In: International conference on machine learning. PMLR, pp 8821–8831

Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M (2022) Hierarchical text-conditional image generation with clip latents. arXiv Preprint http://arxiv.org/abs/2204.06125

Reiss MV (2023) Testing the reliability of ChatGPT for text annotation and classification: a cautionary remark. arXiv Preprint http://arxiv.org/abs/2304.11085

Ren S, Deng Y, He K, Che W (2019a) Generating natural language adversarial examples through probability weighted word saliency. In: Proceedings of the 57th annual meeting of the association for computational linguistics. pp 1085–1097

Ren J, Liu PJ, Fertig E, Snoek J, Poplin R, Depristo M, Dillon J, Lakshminarayanan B (2019b) Likelihood ratios for out-of-distribution detection. In: Advances in neural information processing systems, vol 32

Ren X, Zhou P, Meng X, Huang X, Wang Y, Wang W, Li P, Zhang X, Podolskiy A, Arshinov G et al (2023) Pangu-σ: towards trillion parameter language model with sparse heterogeneous computing. arXiv Preprint http://arxiv.org/abs/2303.10845

Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?": explaining the predictions of any classifier. In: HLT-NAACL demos

Rolfe JT (2016) Discrete variational autoencoders. arXiv Preprint http://arxiv.org/abs/1609.02200

Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 10684–10695

Ruan W, Huang X, Kwiatkowska M (2018) Reachability analysis of deep neural networks with provable guarantees. In: IJCAI2018. pp 2651–2659
Ruan W, Wu M, Sun Y, Huang X, Kroening D, Kwiatkowska M (2019) Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. In: IJCAI2019. pp 5944–5952
Ruder S, Peters ME, Swayamdipta S, Wolf T (2019) Transfer learning in natural language processing. In: Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: tutorials. pp 15–18
Rueckauer B, Lungu I-A, Hu Y, Pfeiffer M, Liu S-C (2017) Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. Front Neurosci 11:682
Rutinowski J, Franke S, Endendyk J, Dormuth I, Pauly M (2023) The self-perception and political biases of ChatGPT. arXiv Preprint http://arxiv.org/abs/2304.07333
Ryou W, Chen J, Balunovic M, Singh G, Dan A, Vechev M (2021) Scalable polyhedral verification of recurrent neural networks. In: International conference on computer aided verification. Springer, pp 225–248
Saharia C, Chan W, Saxena S, Li L, Whang J, Denton EL, Ghasemipour K, Gontijo Lopes R, Karagol Ayan B, Salimans T et al (2022) Photorealistic text-to-image diffusion models with deep language understanding. Adv Neural Inf Process Syst 35:36479–36494
Samanta S, Mehta S (2017) Towards crafting text adversarial samples. arXiv Preprint http://arxiv.org/abs/1707.02812
Sandoval G, Pearce H, Nys T, Karri R, Garg S, Dolan-Gavitt B (2023) Lost at C: a user study on the security implications of large language model code assistants. arXiv Preprint http://arxiv.org/abs/2208.09727
Scao TL, Fan A, Akiki C, Pavlick E, Ilić S, Hesslow D, Castagné R, Luccioni AS, Yvon F, Gallé M et al (2022) Bloom: a 176B-parameter open-access multilingual language model. arXiv Preprint http://arxiv.org/abs/2211.05100
Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv Preprint http://arxiv.org/abs/1707.06347
Senate U (2023) Senate judiciary subcommittee hearing on oversight of AI. https://techpolicy.press/transcript-senate-judiciary-subcommittee-hearing-on-oversight-of-ai/. Accessed 20 Aug 2023
Seshia SA, Sadigh D, Sastry SS (2016) Towards verified artificial intelligence. arXiv Preprint http://arxiv.org/abs/1606.08514
Shanahan M (2022) Talking about large language models. arXiv Preprint http://arxiv.org/abs/2212.03551
Shen Y, Hsu Y-C, Ray A, Jin H (2021a) Enhancing the generalization for intent classification and out-of-domain detection in SLU. 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 2443–2453
Shen L, Ji S, Zhang X, Li J, Chen J, Shi J, Fang C, Yin J, Wang T (2021b) Backdoor pre-trained models can transfer to all. In: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security. pp 3141–3158
Shen X, Chen Z, Backes M, Zhang Y (2023) In ChatGPT we trust? Measuring and characterizing the reliability of ChatGPT. arXiv Preprint http://arxiv.org/abs/2304.08979
Shi Z, Zhang H, Chang K-W, Huang M, Hsieh C-J (2019) Robustness verification for transformers. In: International conference on learning representations
Shuster K, Poff S, Chen M, Kiela D, Weston J (2021) Retrieval augmentation reduces hallucination in conversation. arXiv Preprint http://arxiv.org/abs/2104.07567
Shuster K, Komeili M, Adolphs L, Roller S, Szlam A, Weston J (2022) Language models that seek for knowledge: modular search & generation for dialogue and prompt completion. arXiv Preprint http://arxiv.org/abs/2203.13224
Sinha A, Namkoong H, Volpi R, Duchi J (2017) Certifying some distributional robustness with principled adversarial training. arXiv Preprint http://arxiv.org/abs/1710.10571
Smith L, Gal Y (2018) Understanding measures of uncertainty for adversarial example detection. arXiv Preprint http://arxiv.org/abs/1803.08533
Smith S, Patwary M, Norick B, LeGresley P, Rajbhandari S, Casper J, Liu Z, Prabhumoye S, Zerveas G, Korthikanti V et al (2022) Using deepspeed and megatron to train megatron-turing NLG 530B, a large-scale generative language model. arXiv Preprint http://arxiv.org/abs/2201.11990
Sobania D, Briesch M, Hanna C, Petke J (2023) An analysis of the automatic bug fixing performance of ChatGPT. arXiv Preprint http://arxiv.org/abs/2301.08653
Soltan S, Ananthakrishnan S, FitzGerald J, Gupta R, Hamza W, Khan H, Peris C, Rawls S, Rosenbaum A, Rumshisky A et al (2022) AlexaTM 20B: few-shot learning using a large-scale multilingual seq2seq model. arXiv Preprint http://arxiv.org/abs/2208.01448

Struppek L, Hintersdorf D, Kersting K (2022) Rickrolling the artist: injecting invisible backdoors into text-guided image generation models. arXiv Preprint http://arxiv.org/abs/2211.02408
Sun Y, Huang X, Kroening D, Sharp J, Hill M, Ashmore R (2018a) Testing deep neural networks. arXiv Preprint http://arxiv.org/abs/1803.04792
Sun Y, Wu M, Ruan W, Huang X, Kwiatkowska M, Kroening D (2018b) Concolic testing for deep neural networks. In: ASE2018
Sun Y, Huang X, Kroening D, Sharp J, Hill M, Ashmore R (2019) Structural test coverage criteria for deep neural networks. ACM Trans Embed Comput Syst 18(5s):1–23
Sun Y, Wang S, Feng S, Ding S, Pang C, Shang J, Liu J, Chen X, Zhao Y, Lu Y et al (2021) ERNIE 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. arXiv Preprint http://arxiv.org/abs/2107.02137
Sun H, Zhang Z, Deng J, Cheng J, Huang M (2023) Safety assessment of Chinese large language models. arXiv Preprint http://arxiv.org/abs/2304.10436
Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv Preprint http://arxiv.org/abs/1312.6199
Tanguy L, Tulechki N, Urieli A, Hermann E, Raynal C (2016) Natural language processing for aviation safety reports: from classification to interactive analysis. Comput Ind 78:80–95
Taori R, Gulrajani I, Zhang T, Dubois Y, Li X, Guestrin C, Liang P, Hashimoto TB (2023) Stanford Alpaca: an instruction-following LLaMa model
Taylor R, Kardas M, Cucurull G, Scialom T, Hartshorn A, Saravia E, Poulton A, Kerkez V, Stojnic R (2022) Galactica: a large language model for science. arXiv Preprint http://arxiv.org/abs/2211.09085
Tejankar A, Sanjabi M, Wang Q, Wang S, Firooz H, Pirsiavash H, Tan L (2023) Defending against patch-based backdoor attacks on self-supervised learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 12239–12249
Thakur S, Ahmad B, Fan Z, Pearce H, Tan B, Karri R, Dolan-Gavitt B, Garg S (2023) Benchmarking large language models for automated Verilog RTL code generation. In: 2023 design, automation & test in Europe conference & exhibition (DATE). IEEE, pp 1–6
The carbon footprint of GPT-4. https://towardsdatascience.com/the-carbon-footprint-of-gpt-4-d6c676eb21ae. Accessed 17 Aug 2023
Thoppilan R, De Freitas D, Hall J, Shazeer N, Kulshreshtha A, Cheng H-T, Jin A, Bos T, Baker L, Du Y et al (2022) LaMDA: language models for dialog applications. arXiv Preprint http://arxiv.org/abs/2201.08239
Thorne J, Vlachos A, Christodoulopoulos C, Mittal A (2018) FEVER: a large-scale dataset for fact extraction and verification. In: 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL HLT 2018. Association for Computational Linguistics (ACL), pp 809–819
Tools such as ChatGPT threaten transparent science; here are our ground rules for their use. https://www.nature.com/articles/d41586-023-00191-1. Accessed 20 Aug 2023
Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F et al (2023) LLaMA: open and efficient foundation language models. arXiv Preprint http://arxiv.org/abs/2302.13971
Tulshan AS, Dhage SN (2019) Survey on virtual assistant: Google assistant, Siri, Cortana, Alexa. In: Advances in signal processing and intelligent recognition systems: 4th international symposium SIRS 2018, Bangalore, India, September 19–22, 2018, revised selected papers 4. Springer, pp 190–201
Tung F, Mori G (2019) Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp 1365–1374
Uchendu A, Lee J, Shen H, Le T, Huang TK, Lee D (2023) Understanding individual and team-based human factors in detecting deepfake texts. CoRR. abs/2304.01002
Vardi MY, Wolper P (1986) An automata-theoretic approach to automatic program verification. In: 1st symposium in logic in computer science (LICS). IEEE Computer Society
Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates, Inc
Wallace M, Khandelwal R, Tang B (2022) Does IBP scale? arXiv Preprint
Wang Y, Bansal M (2018) Robust machine comprehension models via adversarial training. arXiv Preprint http://arxiv.org/abs/1804.06473

Wang G, Lin Y, Yi W (2010) Kernel fusion: an effective method for better power efficiency on multithreaded GPU. In: 2010 IEEE/ACM Int'l conference on green computing and communications & Int'l conference on cyber, physical and social computing. IEEE, pp 344–350
Wang W, Tang P, Lou J, Xiong L (2021a) Certified robustness to word substitution attack with differential privacy. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies. pp 1102–1112
Wang B, Xu C, Wang S, Gan Z, Cheng Y, Gao J, Awadallah AH, Li B (2021b) Adversarial glue: a multitask benchmark for robustness evaluation of language models. arXiv Preprint http://arxiv.org/abs/2111.02840
Wang J, Hu X, Hou W, Chen H, Zheng R, Wang Y, Yang L, Huang H, Ye W, Geng X, Jiao B, Zhang Y, Xie X (2023a) On the robustness of ChatGPT: an adversarial and out-of-distribution perspective. arXiv e-prints http://arxiv.org/abs/2302.12095
Wang X, Wei J, Schuurmans D, Le QV, Chi EH, Narang S, Chowdhery A, Zhou D (2023b) Self-consistency improves chain of thought reasoning in language models. In: The eleventh international conference on learning representations
Wang F, Xu P, Ruan W, Huang X (2023c) Towards verifying the geometric robustness of large-scale neural networks. arXiv Preprint http://arxiv.org/abs/2301.12456
Wei J, Wang X, Schuurmans D, Bosma M, Brian Ichter, Xia F, Chi EH, Le QV, Zhou D (2022) Chain of thought prompting elicits reasoning in large language models. In: Oh AH, Agarwal A, Belgrave D, Cho K (eds) Advances in neural information processing systems
Wei J, Kim S, Jung H, Kim Y-H (2023) Leveraging large language models to power chatbots for collecting user self-reported data. arXiv Preprint http://arxiv.org/abs/2301.05843
Weng T-W, Zhang H, Chen P-Y, Yi J, Su D, Gao Y, Hsieh C-J, Daniel L (2018) Evaluating the robustness of neural networks: an extreme value theory approach. arXiv Preprint http://arxiv.org/abs/1801.10578
Weng Y, Zhu M, He S, Liu K, Zhao J (2022) Large language models are reasoners with self-verification. arXiv Preprint http://arxiv.org/abs/2212.09561
Weng Y, Zhu M, Xia F, Li B, He S, Liu K, Zhao J (2023) Neural comprehension: language models with compiled neural networks. arXiv Preprint http://arxiv.org/abs/2304.01665
Wicker M, Huang X, Kwiatkowska M (2018) Feature-guided black-box safety testing of deep neural networks. In: Tools and algorithms for the construction and analysis of systems: 24th international conference, TACAS 2018, held as part of the European joint conferences on theory and practice of software, ETAPS 2018, Thessaloniki, Greece, April 14–20, 2018, proceedings, part I 24. pp 408–426
Wolf Y, Wies N, Levine Y, Shashua A (2023) Fundamental limitations of alignment in large language models. arXiv Preprint http://arxiv.org/abs/2304.11082
Wong E, Rice L, Kolter JZ (2020) Fast is better than free: revisiting adversarial training. arXiv Preprint http://arxiv.org/abs/2001.03994
Wu M, Wicker M, Ruan W, Huang X, Kwiatkowska M (2020) A game-based approximate verification of deep neural networks with provable guarantees. Theor Comput Sci 807:298–329
Wu Y, Jiang AQ, Li W, Rabe MN, Staats CE, Jamnik M, Szegedy C (2022a) Autoformalization with large language models. In: Oh AH, Agarwal A, Belgrave D, Cho K (eds) Advances in neural information processing systems
Wu D, Yi X, Huang X (2022b) A little energy goes a long way: build an energy-efficient, accurate spiking neural network from convolutional neural network. Front Neurosci 16:759900
Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G (2023a) BloombergGPT: a large language model for finance. arXiv Preprint http://arxiv.org/abs/2303.17564
Wu D, Jin G, Yu H, Yi X, Huang X (2023b) Optimising event-driven spiking neural network with regularisation and cutoff. arXiv Preprint http://arxiv.org/abs/2301.09522
Wu X, Sun K, Zhu F, Zhao R, Li H (2023c) Better aligning text-to-image models with human preference. arXiv Preprint http://arxiv.org/abs/2303.14420
Wu M, Waheed A, Zhang C, Abdul-Mageed M, Aji AF (2023d) LaMini-LM: a diverse herd of distilled models from large-scale instructions. arXiv Preprint http://arxiv.org/abs/2304.14402
Wu H, Wang W, Wan Y, Jiao W, Lyu M (2023e) ChatGPT or grammarly? Evaluating ChatGPT on grammatical error correction benchmark. arXiv Preprint http://arxiv.org/abs/2303.13648
Xu F, Uszkoreit H, Du Y, Fan W, Zhao D, Zhu J (2019) Explainable AI: a brief survey on history, research areas, approaches and challenges. In: Natural language processing and Chinese computing: 8th CCF international conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, proceedings, part II 8. Springer, pp 563–574
Xu H, Ma Y, Liu H-C, Deb D, Liu H, Tang J-L, Jain AK (2020a) Adversarial attacks and defenses in images, graphs and text: a review. Int J Autom Comput 17:151–178

Xu H, He K, Yan Y, Liu S, Liu Z, Xu W (2020b) A deep generative distance-based classifier for out-of-domain detection with Mahalanobis space. In: Proceedings of the 28th international conference on computational linguistics. pp 1452–1460
Xu P, Ruan W, Huang X (2022) Quantifying safety risks of deep neural networks. Complex Intell Syst 9(4):3801–3818
Xu J, Liu X, Wu Y, Tong Y, Li Q, Ding M, Tang J, Dong Y (2023) ImageReward: learning and evaluating human preferences for text-to-image generation. arXiv Preprint http://arxiv.org/abs/2304.05977
Yandex. Yandex/YaLM-100B: pretrained language model with 100B parameters. https://github.com/yandex/YaLM-100B. Accessed 20 Aug 2023
Yang Z (2023) Chinese tech giant Baidu just released its answer to ChatGPT
Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) XLNet: generalized autoregressive pretraining for language understanding. In: Advances in neural information processing systems, vol 32
Yang J, Zhou K, Li Y, Liu Z (2021a) Generalized out-of-distribution detection: a survey. arXiv Preprint http://arxiv.org/abs/2110.11334
Yang W, Li L, Zhang Z, Ren X, Sun X, He B (2021b) Be careful about poisoned word embeddings: exploring the vulnerability of the embedding layers in NLP models. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies. pp 2048–2058
Yang J, Jin H, Tang R, Han X, Feng Q, Jiang H, Yin B, Hu X (2023) Harnessing the power of LLMs in practice: a survey on ChatGPT and beyond. arXiv Preprint http://arxiv.org/abs/2304.13712
Yao Z, Yazdani Aminabadi R, Zhang M, Wu X, Li C, He Y (2022) ZeroQuant: efficient and affordable post-training quantization for large-scale transformers. In: Advances in neural information processing systems, vol 35. pp 27168–27183
Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan KR, Cao Y (2023) ReAct: synergizing reasoning and acting in language models. In: The eleventh international conference on learning representations
Ye M, Gong C, Liu Q (2020) Safer: a structure-free approach for certified robustness to adversarial word substitutions. arXiv Preprint http://arxiv.org/abs/2005.14424
Ye X, Iyer S, Celikyilmaz A, Stoyanov V, Durrett G, Pasunuru R (2022) Complementary explanations for effective in-context learning. arXiv Preprint http://arxiv.org/abs/2211.13892
Yilmaz E, Toraman C (2022) D2U: distance-to-uniform learning for out-of-scope detection. In: Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies. pp 2093–2108
Yu J, Xu Y, Koh JY, Luong T, Baid G, Wang Z, Vasudevan V, Ku A, Yang Y, Ayan BK et al (2022) Scaling autoregressive models for content-rich text-to-image generation. arXiv Preprint http://arxiv.org/abs/2206.10789
Zeng Z, He K, Yan Y, Liu Z, Wu Y, Xu H, Jiang H, Xu W (2021a) Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 2: short papers). pp 870–878
Zeng W, Ren X, Su T, Wang H, Liao Y, Wang Z, Jiang X, Yang Z, Wang K, Zhang X et al (2021b) Pangu-α: large-scale autoregressive pretrained Chinese language models with auto-parallel computation. arXiv Preprint http://arxiv.org/abs/2104.12369
Zeng J, Zheng X, Xu J, Li L, Yuan L, Huang X (2021c) Certified robustness to text adversarial attacks by randomized [mask]. arXiv Preprint http://arxiv.org/abs/2105.03743
Zhang J, Zhao Y, Saleh M, Liu P (2020) PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: III HD, Singh A (eds) Proceedings of the 37th international conference on machine learning, volume 119 of proceedings of machine learning research, 13–18 July 2020. PMLR, pp 11328–11339
Zhang Y, Albarghouthi A, D'Antoni L (2021) Certified robustness to programmable transformations in LSTMS. arXiv Preprint http://arxiv.org/abs/2102.07818
Zhang S, Roller S, Goyal N, Artetxe M, Chen M, Chen S, Dewan C, Diab M, Li X, Lin XV et al (2022) OPT: open pre-trained transformer language models. arXiv Preprint http://arxiv.org/abs/2205.01068
Zhang T, Ladhak F, Durmus E, Liang P, McKeown K, Hashimoto TB (2023a) Benchmarking large language models for news summarization. arXiv Preprint http://arxiv.org/abs/2301.13848
Zhang C, Ruan W, Wang F, Xu P, Min G, Huang X (2023b) Model-agnostic reachability analysis on deep neural networks. arXiv Preprint http://arxiv.org/abs/2304.00813
Zhang C, Ruan W, Xu P (2023c) Reachability analysis of neural network control systems. arXiv Preprint http://arxiv.org/abs/2301.12100

Zhao Z, Dua D, Singh S (2017) Generating natural adversarial examples. arXiv Preprint http://arxiv.org/abs/1710.11342

Zhao X, Huang W, Huang X, Robu V, Flynn D (2021a) BayLIME: Bayesian local interpretable model-agnostic explanations. In: de Campos C, Maathuis MH (eds) Proceedings of the thirty-seventh conference on uncertainty in artificial intelligence, volume 161 of proceedings of machine learning research, 27–30 July 2021. PMLR, pp 887–896

Zhao X, Huang W, Schewe S, Dong Y, Huang X (2021b) Detecting operational adversarial examples for reliable deep learning. In: 2021 51st annual IEEE/IFIP international conference on dependable systems and networks—supplemental volume (DSN-S). pp 5–6

Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z et al (2023a) A survey of large language models. arXiv Preprint http://arxiv.org/abs/2303.18223

Zhao R, Li X, Chia YK, Ding B, Bing L (2023b) Can ChatGPT-like generative models guarantee factual accuracy? On the mistakes of new generation search engines. arXiv Preprint http://arxiv.org/abs/2304.11076

Zhong Q, Ding L, Liu J, Du B, Tao D (2023) Can ChatGPT understand too? A comparative study on ChatGPT and fine-tuned BERT. arXiv Preprint http://arxiv.org/abs/2302.10198

Zhou W, Liu F, Chen M (2021) Contrastive out-of-distribution detection for pretrained transformers. In: Proceedings of the 2021 conference on empirical methods in natural language processing (EMNLP)

Zhou Y, Liu P, Qiu X (2022) KNN-contrastive learning for out-of-domain intent classification. In: Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers). pp 5129–5141

Zhou C, Li Q, Li C, Yu J, Liu Y, Wang G, Zhang K, Ji C, Yan Q, He L et al (2023) A comprehensive survey on pretrained foundation models: a history from BERT to ChatGPT. arXiv Preprint http://arxiv.org/abs/2302.09419

Zhu RJ, Zhao Q, Eshraghian JK (2023) SpikeGPT: generative pre-trained language model with spiking neural networks. arXiv Preprint http://arxiv.org/abs/2302.13939

Ziegler DM, Stiennon N, Wu J, Brown TB, Radford A, Amodei D, Christiano P, Irving G (2019) Fine-tuning language models from human preferences. arXiv Preprint http://arxiv.org/abs/1909.08593