

Safety Analysis in the Era of Large Language Models: A Case Study of STPA using ChatGPT

Yi Qi, Xingyu Zhao, Xiaowei Huang

Department of Computer Science, University of Liverpool*

Abstract

Large Language Models (LLMs), such as ChatGPT and BERT, are leading a new AI heat-wave due to its human-like conversations with detailed and articulate answers across many domains of knowledge. While LLMs are being quickly applied to many AI application domains, we are interested in the following question: Can safety analysis for safety-critical systems make use of LLMs? To answer, we conduct a case study of Systems Theoretic Process Analysis (STPA) on Automatic Emergency Brake (AEB) systems using ChatGPT. STPA, one of the most prevalent techniques for hazard analysis, is known to have limitations such as high complexity and subjectivity, which this paper aims to explore the use of ChatGPT to address. Specifically, three ways of incorporating ChatGPT into STPA are investigated by considering its interaction with human experts: one-off simplex interaction, recurring simplex interaction, and recurring duplex interaction. Comparative results reveal that: (i) using ChatGPT without human experts' intervention can be inadequate due to reliability and accuracy issues of LLMs; (ii) more interactions between ChatGPT and human experts may yield better results; and (iii) using ChatGPT in STPA with extra care can outperform human safety experts alone, as demonstrated by reusing an existing comparison method with baselines. In addition to making the first attempt to apply LLMs in safety analysis, this paper also identifies key challenges (e.g., trustworthiness concern of LLMs and the need of standardisation) for future research in this direction.

Keywords: STPA, safety-critical systems, ChatGPT, large language models, safe AI, automatic emergency brake systems, human machine interaction, hazards identification, safety assurance.

1 Introduction

Large Language Models (LLMs) [23], including Generative Pre-trained Transformer (GPT) [5] and Bidirectional Encoder Representations from Transformers (BERT) [12], have achieved state-of-the-art performance on a wide range of Natural Language Processing (NLP) tasks. LLMs are gaining popularity and receiving increasing attention for their significant applications in knowledge reasoning [11, 45, 49]. ChatGPT (powered by GPT-3.5 at the time of when this paper is writing) is one of the LLMs applications, and probably *the* application, in the limelight. ChatGPT was used for collating literature and writing professional papers in fields like law [8], and medical education [25, 14]. OpenAI announced GPT-4 in March 2023 that can pass everything from a bar exam to AP Biology [34]. These successful stories demonstrate that people have already gained experience

*Emails: {yi.qi,xingyu.zhao,xiaowei.huang}@liverpool.ac.uk

in using LLMs, for their excellent performance in handling complex content due to their massive training datasets and model capacity to process and learn from data, enabling their potential for complex tasks that require domain expert knowledge [33]. Given this, as researchers in the field of safety-critical systems, we pose a question: *Can safety analysis make use of LLMs?*

The System-Theoretic Accident Model and Processes (STAMP) is a safety analysis framework based on system theory, with Systems Theoretic Process Analysis (STPA) being one of the most prevalent STAMP-based tools [28]. Despite being a promising approach, STPA is known to have limitations, including the absence of formalism [15], high complexity, and subjectivity (i.e., subjective opinions of experts conducting the analysis can lead to inconsistent results or biases) [10]. Currently, industry is contemplating the adoption of formal techniques, such as model checking and theorem proving, to compensate for the lack of formalism [10, 3]. While to cope with the complexity and subjectivity of STPA, we see the potential of exploiting LLMs, with the following research questions to be answered:

- If LLMs can alleviate the complexity and subjectivity problems of STPA, and what is the most effective way of integrating LLMs in the STPA process?
- What are the benefits and limitations of using LLMs in STPA, and how do they compare to traditional methods?

To answer those questions, in this paper, we conduct a case study of STPA on Automatic Emergency Brake (AEB) systems using ChatGPT. To enable comparative studies, we first choose the analysis result completed by an expert team, and published in [40], as the baseline. Then, we investigate three ways of incorporating ChatGPT in the workflow of how human safety experts perform STPA—one-off simplex interaction, recurring simplex interaction and recurring duplex interaction. According to the comparative results, three key findings emerge: i) applying ChatGPT without human experts’ intervention is inadequate due to the problematic reliability and accuracy of LLMs in general; ii) more interactions between ChatGPT and human safety experts have the potential to yield better results, advocating the best practice of “safety-experts-LLMs-teaming”; iii) using ChatGPT in STPA with extra care may outperform human safety experts alone, demonstrated by existing metrics for comparing safety analysis [40].

In summary, the key contributions of this work include:

- We make a first attempt on applying LLMs in safety analysis for safety-critical systems. Specifically, we develop and compare three ways to exploit ChatGPT in STPA, and conclude the most effective way.
- Opportunities of safety analysis using LLMs are discussed, while key challenges are identified as research questions, paving the way for future research in this direction.

2 Background

2.1 Large Language Models

LLMs are advanced machine learning models that are designed to understand natural language and generate human-like responses to queries and prompts. LLMs are among the most impressive achievements in AI research and development. Almost all NLP tasks (e.g., language translation, chatbots and virtual assistants) have witnessed tremendous success with Transformer-based pre-trained language models (T-PTLMs), relying on Transformer [41], self-supervised learning [19, 31]

and transfer learning [16, 39] to process and understand the nuances of human language, including grammar, syntax, and context.

The success of NLP tasks owes a great deal to T-PTLMs such as GPT-1 [37], BERT [12], XLNet [44], RoBERTa [32], ELECTRA [9], T5 [38], ALBERT [26], BART [30], and PEGASUS [47]. These models have the capability to learn general language representations from large volumes of unlabeled 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 [20]. Kaplan et al. [22] demonstrated that simply increasing the size of T-PTLMs can lead to improved performance [20]. This finding has spurred the development of LLMs such as GPT-3 [5], PANGU [46], GShard [27] and Switch-Transformers [13]. One model that has gained significant attention is GPT-3, and its improved version GPT-3.5 that forms the technical core of ChatGPT. At the time of writing this paper, ChatGPT is prepared integrated with GPT-4.

2.2 Systems Theoretic Process Analysis

STPA is a widely used safety analysis method that is based on the STAMP model. The main focus of STPA is on the overall control loop and process analysis of a system, with an emphasis on identifying unsafe control actions and causal factors within the control structure. This method has found applications in various domains, including railway safety assurances [43], cyber safety and security [21], collaborative robots [1], and autonomous vehicles [24, 7]. Readers are referred to [15, 36] for comprehensive reviews.

The STPA workflow includes the following steps [29]: 1) Define Accidents/Hazards. 2) Model Control Structure. 3) Identify Unsafe Control Actions. 4) Identify Causal Scenarios. 5) Derive Safety Requirements. Specifically, the initial step involves defining potential accidents or hazards at a high level, typically encompassing significant negative outcomes for personnel or equipment that can be anticipated [18]. The subsequent step involves creating a system control structure that segregates the control loop from its individual components, such as the starting point, intermediate processes (like mechanical and software structures), and completion components. Following this, standard Unsafe Control Actions (UCAs) form with four categories, including Causes Hazards Not Providing, Causes Hazard Providing, Incorrect Timing or Order, and Stopped Too Soon or Applied Too Long, can be employed to identify possible UCAs. After analysing the UCAs to identify potential hazards or risks, the STPA proceeds to determine potential causes and scenarios. The final step involves generating safety requirements for each identified risk or hazard.

Expert knowledge is essential in order to conduct all five steps described above. The first step largely depends on the safety-critical application and its environments, while the second step rely on the system’s internal structure. The third step is highly dependent on expertise of the safety analyst, as it requires an in-depth understanding of the system or process under analysis, as well as prior knowledge of relevant safety regulations and standards. The fourth and fifth steps also require the expert knowledge, as the expert team is able to identify and evaluate potential failure modes and develop reasonable safety requirements.

Despite its popularity, conducting STPA is widely acknowledged to be challenging and time-consuming due to its high complexity and subjectivity. The complexity arises from the increasingly complex systems being analysed (including recent learning-enabled systems [36], while subjectivity is due to the limited knowledge and expert judgment of individuals [10]. These motivate us to investigate how LLMs can solve the two challenges and make STPA more efficient.

Table 1: Example UCAs in AEB system control structure

UCA type	Example UCA in AEB system
UCA-1 ($T1$)	No brake command provides a cause of a hazard
UCA-2 ($T2$)	The brake command is inadequately provided under no collision risk
UCA-3 ($T3$)	Brake command occurring too late causes insufficient brake and crash risk
UCA-4 ($T4$)	Braking too long or too short causes a hazard

3 Baseline in the Comparative Study

In [40], various safety analysis methods are performed on an AEB system for comparison. In this paper, we have chosen the STPA result from [40] as our baseline, which was obtained by a team of human safety experts.

3.1 Automatic Emergency Brake Systems

If a collision is about to occur and the driver takes no action or the action is not fast enough, AEB system will automatically initiate braking. AEB is able to detect potential collisions and activate the braking system to slow down the vehicle to avoid the collision or reduce its impact.

A typical AEB system consists of many components, including signal acquisition, calculation, algorithm, and fusion processes, as well as interfaces with electrical and mechanical parts, sensor systems, and more. AEB is responsible for detecting hazards, making decisions, and applying brakes to avoid or mitigate collisions. The general block diagram of an AEB system is illustrated in Fig. 1.

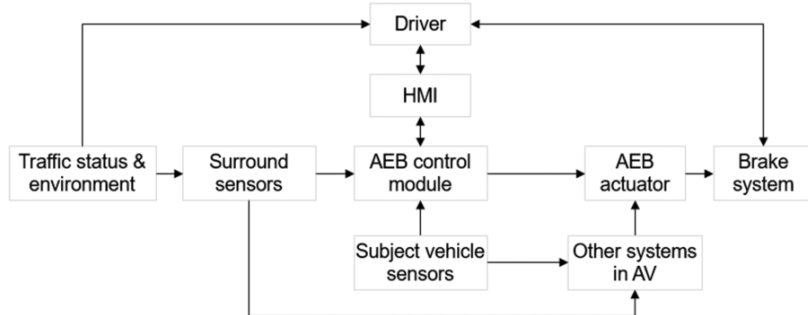


Figure 1: Block diagram of a generic AEB system [40].

3.2 Results of STPA with AEB system

We compile and present some of the STPA findings from [40]. There are four parts consisting of a hazardous control action of each UCA [36], i.e., *Source Controller*, *Type*, *Control Action*, and *Context*. The four *Types* in UCAs are: ($T1$) *Not providing the control action leads to a hazard*; ($T2$) *Providing the control action leads to a hazard*; ($T3$) *Providing a potentially safe control action but too early, too late, or in the wrong order*; and ($T4$) *The control action lasts too long or is stopped too soon*. According to [29], the four categories are complete in terms of defining UCAs.

The work [40] presents a list of UCAs for the “implementation of brake commands”, which are shown in Table 1 that includes a correspondence between each UCA and its corresponding UCA

Table 2: Baseline casual scenarios of AEB systems (scenarios for UCA-4)

Scenario reasons (causes)	Possible scenario
(<i>Scenario 1</i>) Failures involving the controller	AEB control model processing delays result from the braking being too long
(<i>Scenario 2</i>) Inadequate control algorithm	AEB control model incorrectly believes the braking is correct
(<i>Scenario 3</i>) Unsafe control input	Brake command actuator failure
(<i>Scenario 4</i>) Inadequate process model	Blocking of other controllers or processes

Table 3: Comparison attribute list (adapted from [40])

Aspects	Attributes	Descriptions
Analysis results comparison	Identify hazards	Comprehensiveness of identified hazards
Analysis results comparison	Identify causes	Comprehensiveness of causes of identified hazards
Analysis process comparison	Skill required	Relative technical experience required as expert teams
Analysis process comparison	Time and cost	A time and cost required for safety analysis with the case
Analysis process comparison	Complexity/difficulty	Relative complexity/difficulty of the case structure

type (e.g. UCA-1 is *T1*). One of the subsequent steps in STPA is to analyse and identify possible scenarios for the four causes of UCAs. For example, for “UCA-4”, Table 2 provides a list of possible scenarios for each cause. The safety requirements part of STPA is omitted in paper [40] as the focus is on comparing safety analysis methods.

3.3 Comparison Methodology

In this paper, we reuse the comparison method from [40] (designed for a range of safety methods), but reduce the set of attributes to suit our specific goal on comparing STPA results. Table 3 lists the attributes for comparison, and we assess and compare three use cases of incorporating ChatGPT into STPA process in later sections, based on the attributes listed in the table.

4 Case Studies

We consider three ways of incorporating ChatGPT into the STPA workflow in the following case studies. To be specific, we develop one-off simplex interaction (Fig. 2(a)), recurring simplex interaction (Fig. 2(b)) and recurring duplex interaction (Fig. 2(c)), depending on the communication *frequency* and *direction* between human safety experts and ChatGPT.

In what follows, we show examples of questions inputted into ChatGPT, along with their corresponding answers (presented as *Q&A* blocks). For illustration purpose, we only provide a partial list of Q&A blocks while the complete results are publicly available at our project website¹.

4.1 Case 1: One-off simplex interaction

The use case shown in Fig. 2(a) involves the expert team providing ChatGPT with the keywords “AEB System” and “STPA Method”, without sharing the specific details of how to perform STPA. Then, STPA is conducted automatically by ChatGPT, according to the steps obtained directly from ChatGPT. This use case is referred as “one-off simplex interaction”, in the sense that the only interaction between human experts and ChatGPT is at the very beginning and the communication direction is single-directional from human to ChatGPT. Intuitively, this use case represents the scenario

¹<https://github.com/YiQi0318/ChatGPT-STPA>

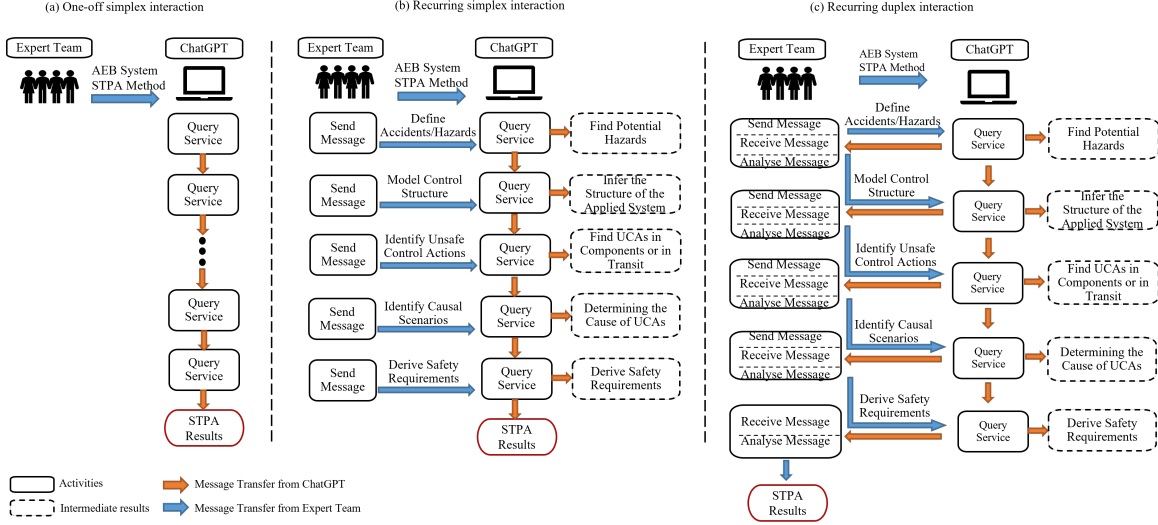


Figure 2: Three ways of incorporating ChatGPT in the workflow of how human safety experts perform STPA: (a) One-off simplex interaction (b) Recurring simplex interaction (c) Recurring duplex interaction.

in which the human experts fully trust ChatGPT and only need to specify their high-level inputs (i.e., conducting STPA on the AEB system) to get the final analysis results from STPA.

To start, we may conduct an initial query to ChatGPT regarding the fundamental process of STPA and how to apply it to the AEB system.

Example Q&A 1.1

Q: How to use STPA to analyse AEB systems?

A: (a) Define the system boundaries; (b) Identify the top-level hazards; (c) Develop control structure diagrams; (d) Identify potential unsafe control actions; (e) Analyze the causes of the unsafe control actions; (f) Develop recommendations to mitigate hazards.

The original STPA process consists of five basic steps, cf. Section 2.2. However, in *Example Q&A 1.1*, ChatGPT outputs eight steps. Despite the discrepancy, we can perform STPA by following the the eight steps which arguably is a more refined version of the original five-step STPA.

Remark 1 (Accuracy despite discrepancy) *Outputs from ChatGPT regarding the steps of conducting STPA may be different from those in published books and academic papers, but not necessarily incorrect.*

Example Q&A 1.2

Q: Could you help me output a result using STPA safety analysis method on an AEB system?

A: (a) System Boundaries: Vehicle, AEB system, Environment.
 (b) Top-Level Hazards: 1. Collisions with other vehicles 2. Collisions with pedestrians 3. Collisions with objects
 (c) Control Structure Diagram: 1. Sensors (camera, radar, lidar) 2. Software (object detection, decision-making, braking control) 3. Braking system
 (d) Potential Unsafe Control Actions: 1. Failure of the sensor to detect an obstacle 2. Incorrect classification of the obstacle 3. Delay in the decision-making process 4. Failure of the braking system to activate
 (e) Causes of Unsafe Control Actions: 1. Sensor malfunction due to hardware or software failure 2. Insufficient training data for the object detection software 3. Environmental factors such as weather conditions or lighting that interfere with sensor accuracy 4. Software bugs that cause delays or incorrect classification of obstacles 5. Malfunction of the braking system due to hardware failure or lack of maintenance
 (f) Recommendations to Mitigate Hazards: 1. Improve sensor redundancy and fault tolerance 2. Increase training data for the object detection software to improve accuracy 3. Develop algorithms to adjust for environmental factors that can impact sensor accuracy 4. Implement thorough testing procedures to detect and address software bugs 5. Implement a preventative maintenance program for the braking system to ensure proper function and reduce the risk of failure.

In *Example Q&A 1.2*, “Failure of the sensor to detect an obstacle” and “Incorrect classification of the obstacle” correspond to “UCA-1” in the baseline, and “Delay in the decision-making process” may correspond to the “UCA-3”.

Remark 2 (Unreliability) *The STPA result of Case 1 is inconsistent with the baseline by human safety experts, as the UCAs discovered is only a subset of those in Table 1. Specifically, “UCA-2” and “UCA-4” are not identified in Case 1. Therefore, it can be remarked that Case 1 is unreliable in terms of missing important UCAs, compared to the baseline.*

Remark 3 (Propagation and compounding of errors) *From Example Q&A 1.1 to Example Q&A 1.2, it suggests that, an error made in initial steps can cause greater deviation in the subsequent steps in the end-to-end process of STPA without any human interventions.*

By *Example Q&A 1.1* and *Example Q&A 1.2*, we have demonstrated the complete process using the one-off simplex interaction in *Case 1*. However, for evaluation purpose, we also want to get intermediate results from ChatGPT in this use case for later comparative studies. Thus, we make an additional query to ChatGPT on the control loop structure in the following *Example Q&A 1.3*.

Example Q&A 1.3

Q: *Could you help me draw the model control loop structure on the AEB systems?*

A: (a) Sensor input: The system takes in data from various sensors such as radar, lidar, and cameras to detect potential obstacles or hazards on the road. (b) Object recognition and tracking: The system uses algorithms to analyze the sensor data and identify objects such as vehicles, pedestrians, and cyclists. It then tracks the movement of these objects to determine if they pose a risk of collision. (c) Decision-making: Based on the sensor data and object

recognition/tracking, the system makes a decision on whether to apply the brakes or not. If the system determines that a collision is imminent, it will activate the braking system. (d) Actuation: Once the decision is made to apply the brakes, the system sends a signal to the braking system to engage the brakes and stop the vehicle. (e) Feedback: The system continuously monitors the situation and provides feedback to the driver, such as visual and audible warnings, to alert them to the potential danger.

Unfortunately, as a language model, ChatGPT is unable to create visual diagrams. Instead, it provides the expert team a text guidance for creating graphical outputs. *Example Q&A 1.3* provides such instructions on creating the control loop structure of AEB systems, cf. Fig. 3(a), and we will examine and compare it with the one generated from the baseline, *Case 2*, and *Case 3* in the discussion section.

Remark 4 (Graphical outputs) *As expected, the current version of ChatGPT cannot generate graphic outputs, while the text-based description provides guidance on how to create the diagrams. That said, it is also challenging to create a less ambiguous diagram only based on the text description.*

4.2 Case 2: Recurring simplex interaction

The “recurring simplex interaction” is illustrated in Fig. 2(b)—information is communicated multiple times in the direction from human experts to ChatGPT. In this case, the expert team begins with providing “AEB System” and “STPA Method” to ChatGPT, and then proceeds to perform each of the five steps of STPA. At each step, the experts pose more pertinent question to ChatGPT according to previous answers. Finally, ChatGPT outputs the overall result of the STPA analysis.

The first step of STPA involves “defining accidents/hazards” (*Q&A 2.1*), where ChatGPT provides a list of accidents/hazards defined at the whole system level (i.e., cars in our case study) due to AEB failures.

Example Q&A 2.1: STPA Step 1

Q: *Could you help me define the accidents/hazards of cars due to AEB systems as the first step of STPA?*

A: (a) False Positives: AEB systems may mistake harmless obstacles or situations for potential collisions and trigger the brakes unnecessarily, causing the vehicle to stop suddenly and unexpectedly. This can result in rear-end collisions or accidents with other vehicles or pedestrians.

(b) False Negatives: AEB systems may fail to detect potential collisions due to sensor malfunctions, environmental factors (e.g., adverse weather conditions), or human error (e.g., the driver’s inattention or distraction). This can lead to accidents that could have been avoided if the system had functioned correctly.

(c) Over-reliance on AEB: Drivers may become overly reliant on AEB systems, leading to a false sense of security and a decrease in their situational awareness. This can result in accidents when the AEB system is unable to prevent a collision or when the driver fails to react appropriately to a hazard.

(d) Insufficient AEB Performance: AEB systems may not perform as intended in certain situations, such as high-speed or low-light conditions. This can lead to accidents if the system fails to brake in time or if the braking distance is longer than expected.

(e) Malfunctioning AEB: AEB systems may malfunction due to software errors, hardware failures, or maintenance issues. This can cause the system to either trigger false positives or false negatives, or to fail altogether, leading to accidents.

Typically, traditional STPA focuses on identifying high-level consequences, such as traffic accidents or rear-end collisions, during its initial stage.

Remark 5 (Unrobustness to question phrased) *If we phrase the question to ChatGPT in a slightly different way while keeping all the important keywords, the answer may also be different. If we substitute the question in Example Q&A 2.1 with “What are the potential accidents and hazards that can occur in cars equipped with AEB systems, and how can we identify them as the first step of STPA?” remaining the semantics and key words, we might receive different answers with a different number of hazards/accidents defined.*

Example Q&A 2.2: STPA Step 2

Q: *Could you help me draw the model control loop structure in the AEB systems of cars as the second step of STPA?*

A: (a) Sensors: The sensors in the AEB system include cameras, radar, lidar, and other sensors that detect potential collisions. These sensors send signals to the controller indicating the location and speed of nearby objects.
 (b) Controller: The controller in the AEB system receives the signals from the sensors and calculates the appropriate response to avoid or mitigate a collision. The controller determines whether to trigger the brakes, how much braking force to apply, and for how long.
 (c) Actuator: The actuator in the AEB system is responsible for physically applying the brakes. When the controller determines that braking is necessary, it sends a signal to the actuator, which applies the brakes to slow down or stop the vehicle.
 (d) Plant: The plant in the AEB system is the vehicle itself, including the braking system and other components that affect the behavior of the vehicle. The plant responds to the actions of the actuator by slowing down or stopping the vehicle.

We arrange it into a figure based on ChatGPT’s output, cf. Remark 4, as shown in Fig. 3(b). Upon comparison with Fig. 1, it becomes evident that there exists a disparity between the output of ChatGPT and the block diagram of the AEB system consolidated by the expert team. To be more specific, in *Case 2*, the ChatGPT output from *Example Q&A 2.2* may be incomplete, lacking certain components due to without expert knowledge adjustment. For instance, the output contains no information regarding drivers, environmental information, and other related components. To compare with the baseline [40] (which prescribes the “implementation of brake” as the example), without analysing all components in the answer of *Example Q&A 2.2*, we only choose the option (c) as the example module for continuing the safety analysis in the subsequent steps.

Example Q&A 2.3: STPA Step 3 (on the braking module)

Q: *Could you help me identify Unsafe Control Actions based on braking module in the AEB systems of cars as the third step of STPA?*

A: (a) Delayed Braking: If the braking module does not apply the brakes quickly enough in response to a potential collision, the vehicle may collide with the obstacle or pedestrian. This could occur due to a malfunctioning sensor, a slow response from the controller, or a slow actuation of the brakes.

(b) Over-application of Brakes: If the braking module applies too much braking force, the vehicle may come to an abrupt stop, leading to rear-end collisions or loss of control. This could occur due to incorrect calibration of the braking system or a malfunctioning actuator.

(c) Under-application of Brakes: If the braking module does not apply enough braking force, the vehicle may not slow down enough to avoid a collision, leading to accidents. This could occur due to faulty sensor data or incorrect calculations by the controller.

(d) False Braking: If the braking module applies the brakes unnecessarily, such as in response to a false positive from the sensors, the vehicle may come to an abrupt stop or swerve unexpectedly, leading to accidents or collisions. This could occur due to incorrect calibration of the sensors or a malfunctioning controller.

(e) Inadequate Braking: If the braking module does not provide enough braking power, such as in high-speed or heavy-load conditions, the vehicle may not slow down enough to avoid a collision, leading to accidents or collisions. This could occur due to inadequate design or specifications of the braking system.

There are 5 UCAs of the braking module in the AEB system identified in *Example Q&A 2.3*. Specifically, answer (a) associates to “UCA-3” in Table 1, while answers (b) and (c) associate to “UCA-4”, answer (d) associates to “UCA-2”, and answer (e) appears to be similar to answer (c). It is observed that the outcomes presented in *Example Q&A 2.3* already encompass “UCA-2”, “UCA-3”, and “UCA-4” of the baseline. Thus, they convey the same safety implication at the safety semantic level, although there may be differences in their presentation and methodology, cf. Remark 2

Remark 6 (Precise answers from specific questions) *Upon comparing the UCAs outlined in Example Q&A 1.2 of Case 1 and Example Q&A 2.3, it can be observed that when the question is more specific, e.g., “identify Unsafe Control Actions based on braking module”, the analysis results are more focused and precise.*

In this *Case 2*, the expert team should input the UCAs identified from the last steps into ChatGPT one by one to understand their causal scenarios. To better compare with the baseline, we choose the UCA (b) and (c) of *Example Q&A 2.3* as an illustration, the potential causes sought for them are shown in the following *Example Q&A 2.4* and *Example Q&A 2.5*.

Example Q&A 2.4: STPA Step 4 (on UCA (b) from last step)

Q: *Could you help me identify causal scenarios based on over-application of brakes unsafe control actions in the AEB systems of cars as the fourth step of STPA?*

A: (a) Malfunctioning Actuator: The braking module may apply too much braking force if the actuator is malfunctioning or not calibrated correctly. This could occur due to wear and tear or inadequate maintenance of the braking system.

(b) Incorrect Sensor Data: The braking module may apply too much braking force if the sensor data is incorrect or misinterpreted by the controller. For example, the sensors may detect a false positive, such as a stationary object that is not actually in the vehicle’s path.

- (c) Inadequate Controller Calibration: The braking module may apply too much braking force if the controller is not calibrated correctly or does not take into account the vehicle's speed, weight, or other factors. This could occur due to inadequate testing or design of the AEB system.
- (d) Malfunctioning Controller: The braking module may apply too much braking force if the controller is malfunctioning or not communicating properly with the sensors and actuator. This could occur due to software bugs, hardware failures, or other technical issues.
- (e) Unintended Activation: The braking module may apply too much braking force if it is unintentionally activated, such as by a software glitch or electrical interference. This could occur due to inadequate security or safeguards in the AEB system.

Example Q&A 2.5: STPA Step 4 (on UCA (c) from last step)

Q: *Could you help me identify causal scenarios based on under-application of brakes unsafe control actions in the AEB systems of cars as the fourth step of STPA?*

- A:**
- (a) Faulty Sensor Data: The braking module may not apply enough braking force if the sensor data is faulty, incomplete or inaccurate. For example, if the sensor data is not able to detect the presence of an obstacle or pedestrian in the vehicle's path due to occlusion or interference.
 - (b) Insufficient System Performance: The braking module may not apply enough braking force if the AEB system is not designed or calibrated to perform adequately in certain conditions. For example, if the AEB system is not designed to operate effectively in inclement weather conditions such as heavy rain, snow or fog.
 - (c) Inadequate Controller Design: The braking module may not apply enough braking force if the controller is not designed to accurately calculate the required braking force. For example, if the controller does not take into account the speed, weight, or other factors that could affect the vehicle's stopping distance.
 - (d) Technical Malfunction: The braking module may not apply enough braking force if there is a technical malfunction in the AEB system, such as a communication failure between the sensors, controller, and actuator.
 - (e) Inappropriate User Behavior: The braking module may not apply enough braking force if the driver overrides the AEB system or fails to engage the brakes in a timely manner. For example, if the driver is distracted or fails to react to the AEB system's warning signals in time.

It can be concluded that the output from *Case 2* provides more specific causes compared to *Case 1*. Upon comparison with the baseline example, it is evident that certain causal scenarios were overlooked while some new causal scenarios were discovered.

Example Q&A 2.6: STPA Step 5

Q: *Could you help me derive safety requirements based on over-application of brakes unsafe control actions in the AEB systems of cars as the fifth step of STPA?*

A: (a) Prevent Abrupt Stops: The braking module shall be designed to apply braking force gradually and smoothly, avoiding abrupt stops. The actuator shall be designed to apply braking force in proportion to the required stopping distance, based on accurate sensor data and controller calculations. The AEB system shall be designed to take into account the weight and speed of the vehicle, as well as environmental factors such as road surface conditions and weather.

(b) Maintain Control of the Vehicle: The braking module shall be designed to apply braking force in a manner that maintains control of the vehicle, avoiding skids or loss of traction. The AEB system shall be designed to detect and respond to potential loss of control situations, such as steering input, yaw rate, or tire slip. The AEB system shall be designed to provide feedback to the driver regarding the status of the AEB system, and any potential control issues.

(c) Prevent Rear-End Collisions: The braking module shall be designed to apply braking force in a manner that avoids rear-end collisions, by taking into account the distance and speed of the vehicle in front of the AEB-equipped vehicle. The AEB system shall be designed to detect and respond to potential rear-end collision situations, such as sudden braking by the vehicle in front, or a sudden decrease in the distance between the vehicles. The AEB system shall be designed to provide warning signals to the driver and other vehicles in the vicinity, in case of a potential rear-end collision.

Example Q&A 2.6 presents safety requirements generated by ChatGPT for the corresponding UCAs. Since these corresponding safety requirements were not provided in the baseline [40], we omit the comparison.

4.3 Case 3: Recurring duplex interaction

As the name suggests, the third use case of ChatGPT in STPA involves multiple bi-directional interactions. As shown in Fig. 2(c), this *Case 3* initiated by the human safety experts, sending “AEB System” and “STPA Method” to ChatGPT. However, unlike *Case 2* in Fig. 2(b), they collect each output of ChatGPT, analyse the information by incorporating domain knowledge, and then query ChatGPT in turn based on the human processed question. The expert team is responsible for “Sending Messages”, “Reserving Messages”, and “Analysing Messages” in each step of STPA.

As the *Case 3* involves multiple bi-directional interactions, the expert team is required to analyse the output of ChatGPT with expert knowledge after each step of STPA. In the following *Example Q&A*, **A1** denotes the response given by ChatGPT, while **A2** is the outcome of the expert team after analysing **A1** with their domain knowledge.

Example Q&A 3.1: STPA Step 1

Q: *Could you help me define the accidents/hazards of cars due to AEB systems as the first step of STPA?*

A1: (a) False activations: AEB systems can sometimes mistakenly detect an object in the car’s path and activate the brakes, even when there is no danger of a collision. This can result in sudden, unexpected stops that could cause accidents, especially if the driver is following too closely. (b) Missed activations: On the other hand, AEB systems may not

activate when they should, such as when the system fails to detect an obstacle in the car's path. This can result in a collision that the AEB system was supposed to prevent. (c) Inappropriate speed reduction: AEB systems may cause the car to slow down too abruptly or to an inappropriate speed, leading to a collision with a vehicle following behind or causing other drivers to swerve to avoid a collision. (d) Malfunction: AEB systems are complex systems that rely on sensors, cameras, and other components to function correctly. If any of these components malfunction, the AEB system may fail to detect obstacles or activate the brakes in time. (e) Driver over-reliance: If drivers become too reliant on AEB systems, they may become complacent or fail to pay attention to the road, which could lead to accidents.

A2 (by safety experts after considering the ChatGPT answer of Q&A 3.1):

(a) Rear-end collisions: AEB systems may engage the brakes unexpectedly, such as when detecting a false positive or when the driver is still in control of the vehicle. This could lead to a rear-end collision with another vehicle, especially if the driver behind the vehicle with AEB is not prepared for sudden braking. (b) Pedestrian or cyclist accidents: AEB systems may not always detect pedestrians or cyclists, particularly in low-light conditions or when the object is not moving. This can lead to potential accidents if the driver is relying on the AEB system to detect all hazards and does not notice the pedestrian or cyclist. (c) Delayed response: AEB systems may have a delay in detecting obstacles, causing the system to initiate braking too late, resulting in a collision. (d) Software bugs: AEB systems may contain software bugs that cause the system to malfunction, leading to incorrect or delayed responses. (e) Malfunction due to environmental factors: AEB systems may malfunction due to environmental factors, such as rain, fog, or snow, that affect the sensors' ability to detect obstacles. (f) False activations: AEB systems can sometimes mistakenly detect an object in the car's path and activate the brakes, even when there is no danger of a collision.

Remark 7 (Irreproducibility) *ChatGPT's responses may vary even if the exact same question (in the same context) is asked repeatedly (e.g. Example Q&A 2.1 and Example Q&A 3.1 (A1))². This is because ChatGPT utilises machine learning algorithms that generate responses based on probabilistic predictions.*

Example Q&A 3.2: STPA Step 2

Q: *Could you help me draw the model control loop structure in the AEB systems of cars as the second step of STPA?*

A1: (a) Inputs: The AEB system receives input from various sensors, such as radar, lidar, and cameras, to detect obstacles in the car's path. (b) Perception: The AEB system processes the input data to perceive the environment and identify potential obstacles. (c)

²Although it is known that ChatGPT provide answers to questions considering the context, the variation of answers to *Examples Q&A 2.1* and *Q&A 3.1* is not due to different contexts given both are asked at the first step of STPA. This is unsurprising given GPT produces a probability distribution over the set of possible next words or characters, thus identical inputs may not necessarily yield the same outputs.

Decision-making: Based on the perceived environment, the AEB system makes a decision on whether to activate the brakes to avoid a collision. (d) Outputs: If the AEB system decides to activate the brakes, it sends a signal to the car’s brake system to apply the brakes. (e) Effects: The AEB system’s output (brake activation) affects the car’s motion, potentially avoiding a collision or causing an unexpected stop. (f) Feedback: The AEB system may receive feedback from the car’s sensors or other components, such as the brake system or steering system, to adjust its perception or decision-making process. (g) Human interaction: The driver may interact with the AEB system by overriding its decisions or disabling the system altogether.

A2 (by safety experts after considering the ChatGPT answer of Q&A 3.2):

(a) Driver: They play a critical role in determining whether or not the system is activated. (b) HMI: HMI stands for Human Machine Interface, which refers to the interface through which a human operator interacts with a machine or system (c) Environmental information: It allows the system to detect potential hazards and respond accordingly. Environmental information can also help the AEB system to adjust its response based on the driving conditions. (d) Sensors: The AEB system receives input from various sensors, such as radar, lidar, and cameras, to detect obstacles in the car’s path. (e) Object tracker: The AEB system processes the input data to perceive the environment and identify potential obstacles. (f) Control unit: Based on the perceived environment, the AEB system makes a decision on whether to activate the brakes to avoid a collision. (g) Brakes: If the AEB system decides to activate the brakes, it sends a signal to the car’s brake system to apply the brakes. (h) Power: The power support is a critical component of the AEB system as it ensures that the brakes are applied with sufficient force to avoid or mitigate a collision. (i) Feedback: The AEB system may receive feedback from the car’s sensors or other components, such as the brake system or steering system, to adjust its perception or decision-making process.

The incorporation of specialised expertise is a vital aspect of constructing the model of control loop structure, as the diagram illustrated in Fig. 3(c) based on *Case 3. Example Q&A 3.2 (A1)* shows that, while the ChatGPT is able to generate the control loop structure, there is the need for a more comprehensive summary and refinement by the human expert team.

Example Q&A 3.3: STPA Step 3 (on the braking module)

Q: *Could you help me identify unsafe control actions based on braking module in the AEB systems of cars as the third step of STPA?*

A1: (a) Delayed braking: If the AEB system detects an obstacle but there is a delay in the activation of the braking module, the car may not stop in time to avoid a collision. (b) Over-reliance on braking: If the AEB system relies solely on the braking module to avoid a collision, it may not take into account other factors that could affect the car’s trajectory, such as the road conditions or the car’s speed. (c) Inappropriate braking force: If the AEB system applies too much or too little force to the brakes, it could cause the car to skid or lose control. (d) Failure to disengage: If the AEB system fails to disengage the braking

module after a collision is avoided, it could cause the car to come to an unexpected stop, potentially causing a hazard to other drivers. (e) Incorrect target identification: If the AEB system incorrectly identifies the target object, it may apply the brakes unnecessarily, leading to unexpected stops or collisions.

A2 (by safety experts after considering the ChatGPT answer of Q&A 3.3):

(a) Braking Failure: If the braking module malfunctions and fails to apply the brakes at all, it could lead to a collision and potential injury or damage. (b) Abrupt Braking: If the AEB system applies the brakes abruptly, it could cause the vehicle to skid or lose control, leading to a potential accident. (c) Overly Sensitive Braking: If the AEB system is too sensitive and initiates braking too frequently, it could cause unnecessary wear and tear on the brakes and reduce the vehicle’s performance. (d) Incorrect classification of the obstacle: If the AEB system incorrectly identifies the target object, it may apply the brakes unnecessarily, leading to unexpected stops or collisions. (e) Delayed braking: If the AEB system detects an obstacle but there is a delay in the activation of the braking module, the car may not stop in time to avoid a collision. (f) Inappropriate braking force: If the AEB system applies too much or too little force to the brakes, it could cause the car to skid or lose control.

In the context of *Example Q&A 3.3*, “UCA-1” in Table 1 corresponds to A2(d), while A1(e) and A2(b) may both refer to “UCA-2”. A1(a) conveys the safety semantics of “UCA-3”, and A1(c) corresponds to “UCA-4”. Notably, all the 4 UCAs in Table 1 are now covered by **A2**, thanks to the interactive collaboration of ChatGPT and human safety experts.

Remark 8 (Comprehensibility from interactivity) *By applying the Case 3 framework, it becomes evident that the four UCAs in the baseline are all identified. In addition, new UCAs are also discovered. Increased interaction frequency between the expert team and ChatGPT allows the model to acquire additional domain knowledge, resulting in more detailed and informative answers.*

At the conclusion of each STPA step, the expert team will consolidate the answers. For instance, in this step, the expert team can determine “UCA-4: brake duration is either too long or too short.” They can then use this information as input for the next step of STPA.

Example Q&A 3.4: STPA Step 4 (on UCA-4)

Q: *Could you help me identify causal scenarios based on braking too long or too short Unsafe Control Actions in the AEB systems of cars as the fourth step of STPA?*

A1: (a) Sensor error: The AEB system may receive erroneous sensor data, causing it to either brake too long or too short. For example, a faulty sensor may detect an obstacle that is not actually there, causing the system to brake unnecessarily, or it may fail to detect an obstacle that is there, causing the system to brake too late or not at all. (b) Sensor fusion error: The AEB system may rely on multiple sensors to detect obstacles and determine the appropriate braking distance. If there is a sensor fusion error where the data from different sensors is not properly integrated, it may cause the system to brake too long or too short. (c) System calibration error: The AEB system may be improperly calibrated, causing it to

brake too long or too short. For example, the system may be calibrated to detect obstacles at a certain distance, but if the calibration is incorrect, it may cause the system to brake too early or too late. (d) Road conditions: The road conditions may affect the AEB system's ability to detect obstacles and determine the appropriate braking distance. For example, if the road is wet or icy, the system may need to brake earlier to compensate for reduced traction. (e) Human error: The driver may interfere with the AEB system or override its decisions, causing the system to brake too long or too short. For example, if the driver is distracted and does not apply the brakes when the AEB system activates, the system may need to brake longer to avoid a collision. (f) Abrupt Braking: The AEB system incorrectly detects an obstacle in the road and applies the brakes abruptly. The vehicle loses traction and skids on the road surface, causing the driver to lose control. The vehicle collides with another vehicle or object, potentially causing injury or damage. (g) Overly Sensitive Braking: The AEB system is too sensitive and initiates braking too frequently. The brakes wear out more quickly than expected, potentially reducing their effectiveness over time. (h) Incorrect programming: If the AEB system is not programmed correctly, it may apply too much or too little braking force, leading to an inappropriate braking force.

A2 (by safety experts after considering the ChatGPT answer of Q&A 3.4):

(a) Failures involving the controller: AEB control model processing delays result from the braking being too long (b) Inadequate control algorithm: AEB control model incorrectly believes the braking is correct (c) Unsafe control input: Brake command actuator failure (d) Inadequate process model: Blocking of other controllers or processes (e) Insufficient Braking: The AEB system incorrectly calculates the distance to the obstacle and applies insufficient braking force. The vehicle does not stop in time and collides with the obstacle, potentially causing injury or damage. (f) Delayed Braking: The AEB system incorrectly calculates the speed or direction of the obstacle and initiates braking too late. The vehicle collides with the obstacle, potentially causing injury or damage. (e) Malfunctioning Actuator: The braking module may apply too much braking force if the actuator is malfunctioning or not calibrated correctly. This could occur due to wear and tear or inadequate maintenance of the braking system. (f) Incorrect Sensor Data: The AEB system may receive erroneous sensor data, causing it to either brake too long or too short. (g) Inadequate Controller Design: Incorrect programming, if the AEB system is not programmed correctly, it may apply too much or too little braking force, leading to an inappropriate braking force. (h) Technical Malfunction: The braking module may not apply enough braking force if there is a technical malfunction in the AEB system, such as a communication failure between the sensors, controller, and actuator. (i) Inappropriate User Behaviour: The driver may interfere with the AEB system or override its decisions, causing the system to brake too long or too short. (j) Road conditions: The road conditions may affect the AEB system's ability to detect obstacles and determine the appropriate braking distance. For example, if the road is wet or icy, the system may need to brake earlier to compensate for reduced traction.

Based on our preliminary assessment using *Example Q&A 3.4*, the interactive approach employed in *Case 3* allows for the identification of more comprehensive causal scenarios. The comparative results of the baseline, *Case 1*, *Case 2*, and *Case 3* can be found in Fig. 4.

Example Q&A 3.5: STPA Step 5

Q: *Could you help me derive safety requirements based on braking too long or too short unsafe control actions in the AEB systems of cars as the fifth step of STPA?*

A1: (a) Sensor accuracy and reliability: The AEB system should have accurate and reliable sensors to detect obstacles and determine the appropriate braking distance. This can prevent the system from braking too long or too short due to inaccurate or unreliable sensor data. (b) Sensor fusion: The AEB system should properly fuse data from multiple sensors to improve accuracy and reliability. This can prevent the system from braking too long or too short due to a failure to integrate data from different sensors. (c) Environmental adaptation: The AEB system should be able to adapt to changing road conditions and environmental factors, such as wet or icy roads. This can prevent the system from braking too long or too short due to environmental factors. (d) System calibration: The AEB system should be properly calibrated to ensure that the braking distance is appropriate for the detected obstacle. This can prevent the system from braking too long or too short due to incorrect calibration. (e) Fail-safe mechanism: The AEB system should have a fail-safe mechanism to ensure that appropriate braking force is applied in all conditions. For example, if the AEB system is unable to determine the appropriate braking force due to a sensor failure or other issue, it should default to a safe braking force. (f) Driver education: Drivers should be educated on how the AEB system works and when it should be used. This can prevent inappropriate braking due to driver intervention or interference. (g) Regular maintenance: The AEB system should undergo regular maintenance to ensure that it is functioning properly. This can prevent inappropriate braking due to system errors or malfunctions.

A2 (by safety experts after considering the ChatGPT answer of Q&A 3.5):

(a) Environmental adaptation: The AEB system should be able to adapt to changing road conditions and environmental factors, such as wet or icy roads. This can prevent inappropriate braking force due to environmental factors. (b) Fail-safe mechanism: The AEB system should have a fail-safe mechanism to ensure appropriate braking force is applied in all conditions. For example, if the AEB system is unable to determine the appropriate braking force due to a sensor failure or other issue, it should default to a safe braking force. (c) Sensor accuracy and reliability: The AEB system should have accurate and reliable sensors to detect obstacles and determine the appropriate braking distance. This can prevent the system from braking too long or too short due to inaccurate or unreliable sensor data. (d) Abrupt Braking: The AEB system must be designed to apply the brakes gradually to avoid causing the vehicle to skid or lose control. (e) Inappropriate Braking: The AEB system must be designed to detect road conditions and adjust braking force accordingly to avoid inappropriate braking. (f) Braking Failure: The braking module must be designed with redundant systems or backup braking mechanisms to prevent braking failure in case of a malfunction. (g) Unintentional Braking: The AEB system must be designed with appropriate feedback mechanisms to alert the driver when the system is malfunctioning or applying the brakes unintentionally.

4.4 Comparison Results

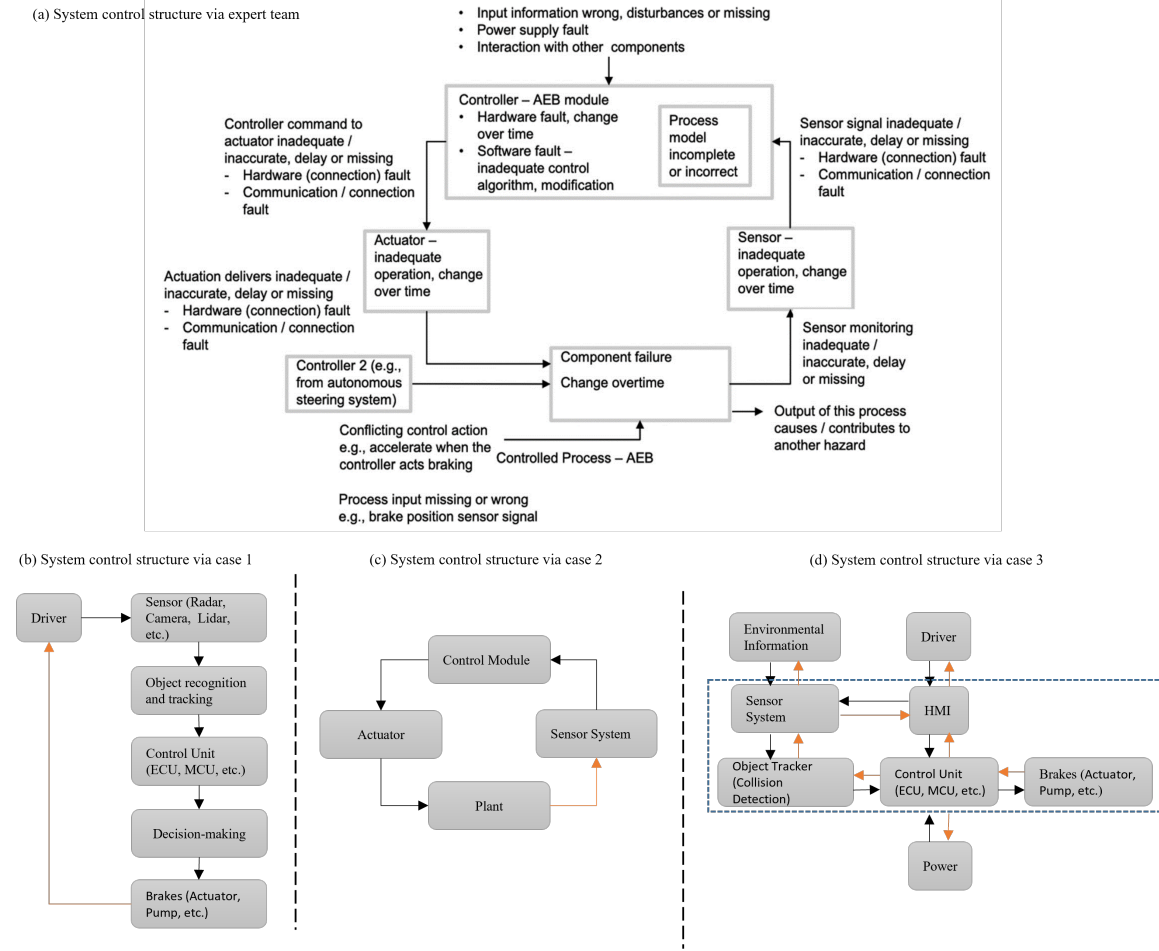


Figure 3: Top: (a) AEB system control loop structure from [40]. Bottom: The control loop structures of AEB systems in the three proposed use cases of ChatGPT in STPA—sub-figures (b), (c) and (d) correspond to *Case 1*, *Case 2* and *Case 3*, respectively.

The control loop diagrams of the AEB system in the case studies are presented in Fig. 3. As evident from the *Example Q&A* response, ChatGPT cannot output figures (*Example Q&A 1.2, 2.2, 3.2*); it can only provide text-based guidance, enabling the generation of diagrams, cf. Remark 4. Different users may interpret text-based guidance differently, resulting in varying forms of the control loop structure. For instance, it may be unclear how to properly draw feedback arrows. In each use case, the second step is to establish the control loop structure, which represents the relationship among control elements in a system, providing the foundation for the subsequent STPA analysis. Fig. 3(a) represents the AEB system control loop structure of the baseline. Fig. 3(b) is created based on the answers provided by ChatGPT in *Case 1*, which includes only the essential components of the AEB system. It involves data collection via hardware (sensors), followed by detection, and finally transmission of commands to the actuator through the control unit. After the application of brakes,

Table 4: Comparative study results based on attributes listed in Table 3 and the baseline.

Aspects	Attributes	Case 1	Case 2	Case 3
Analysis results	Identify hazards	Partially	Partially	Comprehensively
Analysis results	Identify causes	Partially	Partially	Comprehensively
Analysis process	Skills required	No	Few	Many
Analysis process	Time and cost	Low	Low	High
Analysis process	Complexity/difficulty	Low	Low	High

the system will consistently monitor the status and give feedback to the driver through visual and audible warnings to warn them of any potential hazards. Fig. 3(c) employs a more concise language to depict the control loop structure of the AEB system. Unlike Fig. 3(b), it does not contain the “Driver” component but is closer to Fig. 3(a). In *Case 3*, Fig. 3(d) represents the integration of the ChatGPT answers and the expert team’s analysis at each step, leading to the inclusion of more comprehensive information such as “Environmental Information”, “Driver” and “HMI”, compared to Fig. 3(a), cf. Remark 8.

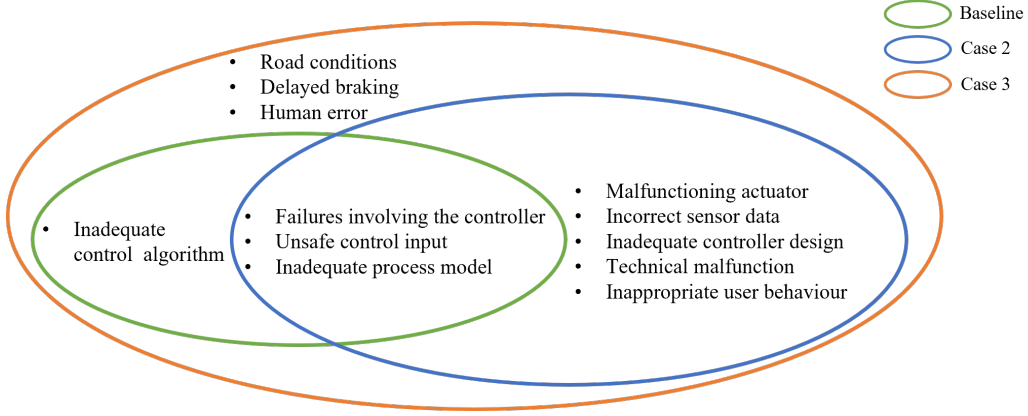


Figure 4: The Venn diagram of the sets of casual scenarios identified by the baseline (green), *Case 2* (blue) and *Case 3* (orange) for the UCA-4 of AEB systems.

Fig. 4 compares the 3 sets of casual scenarios identified by *Case 2*, and *Case 3* and the baseline (for the same UCA). We omit *Case 1*, since “UCA-4” is not identified in this case, cf. *Example Q&A 1.2*. Contrasted with the baseline, the single-directional interactive approach employed in *Case 2* only yields three causal scenarios for “UCA-4”. It fails to encompass the full spectrum of analysis results of the expert team. Even if it can generate additional potential causes, disregarding one may result in safety hazards to the system. However, the output obtained through *Case 3* is much more comprehensive than the answers of the baseline and the *Case 2*, thanks to the integration of ChatGPT results by the expert team.

Based on the comparison attributes presented in Table 3, we have summarised the comparative study results of the three use cases in Table 4. The table reveals that only a part of the hazards can be identified in *Case 1* and *Case 2*, while *Case 3* provides a comprehensive identification of potential hazards. Thus, *Case 3* necessitates more domain knowledge from the safety experts at the price of being the most complex, time-consuming, and costly use of ChatGPT in STPA when compared to *Case 1* and *Case 2*.

5 Discussion

In this section, we first present a big picture demonstrating the role of LLMs in the safety assurance of different types of risks. Then, we discuss the opportunities and challenges related to the use of LLMs in safety analysis, posing research questions for future works in this research direction.

Fig. 5 shows the four-quadrant classification of safety risks depends on if the risk is known by the safety analysts (representing domains-specific knowledge) or the LLMs (representing general knowledge of potential risks). Consequently, we derive the four types of risk as:

- **Known-known:** Risks associated with the most common hazards that are generally known and also cannot be overlooked by any safety analysts when doing safety analysis. To mitigate them, the safety-critical system community has developed safety standards (e.g., [17]) and mature techniques (with formal and/or statistical guarantees).
- **Known-unknown:** Risks that are known in general, e.g., documented in standards and studied in academic papers and etc., but disregarded or omitted by the safety analyst in safety analysis for some reasons. LLMs may compensate in this case by incorporating more knowledge learned from big data.
- **Unknown-known:** Risks that are too novel (due to, e.g., new emerging techniques like AI/ML) to be known by the general community yet, and domain-specific (or even application-specific) that can only be identified in a case by case manner by the safety analysts. Their mitigation requires novel techniques considering domain/application-specific knowledge [4].
- **Unknown-unknown:** Potential risks that are currently unknown to both the general community and the safety analyst.

While the ultimate goal of the safety community is to reduce the area of “Unknown-Unknown” (or non-“Known-known” area), we can see clearly in the big picture in Fig. 5 that how the LLMs and human safety experts can jointly push the boundary (horizontal and vertical bars) to achieve this goal (bottom-right). To advance research in this direction, we pose the following (open) questions to initiate discussions on the opportunities and challenges of using LLMs in safety analysis.

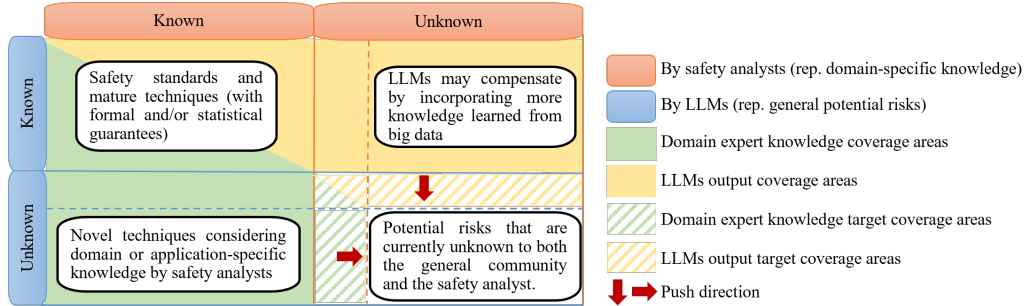


Figure 5: Four-quadrant classification of risks with ways of mitigations

What are the benefits of using LLMs in safety analysis, and how do they compare to traditional methods? The utilisation of LLMs in safety analysis can enable expert teams to handle vast quantities of data in a more efficient manner compared to traditional techniques. This

can result in a faster safety analysis process and a more comprehensive assessment. Additionally, LLMs can expand the scope of safety analysis and potentially decrease labor costs for safety analysis.

What are the limitations of using LLMs in safety analysis, and how can they be addressed? LLMs requires a large amount of high-quality data for effective training, however, safety-critical industries may not have enough data available for this purpose, which may lead to LLMs not being able to output specific details of the safety analysis domain but only general information. Moreover, the output of LLMs may be challenging to interpret. Additionally, there is currently no widely accepted standard for evaluating the accuracy and robustness of LLMs’ results, which may lead to potential biases and trustworthiness issues.

How does the rapid and constant updating of LLMs impact safety analysis? The field of LLMs is currently undergoing rapid development, and the frequent updates and iterations of models like GPT-4 (published in March 2023) can contribute to lowering the blue bar in Fig. 5, indicating an improvement in the generalisability of LLMs and a broader range of risks that can be addressed. Traditional safety assurance processes are static and carried out prior to the system’s deployment [6], but the emergence of dynamic assurance presents an ongoing metric for continuously assessing safety assurance to ensure ongoing safety [2, 6]. The dynamic and constantly evolving nature of LLMs is well-suited for the state-of-the-art dynamic assurance concept in safety analysis.

What types of safety analyses are best suited for LLMs, and what are the criteria for selecting appropriate methods? In this paper, we have opted to utilise the STPA method because of its popularity [15, 29]. Other safety analysis methods like HAZOP, FTA etc. are also discussed in [40]. For future research, it is worthwhile to investigate them as well, and determine which safety analysis method can be more effectively combined with LLMs, and how.

How can the trustworthiness of LLMs be evaluated, and what are the properties for determining their effectiveness? LLMs shares some common weaknesses, including potential reasoning errors, potential security vulnerabilities and the datasets used by LLMs cannot incorporate the most recent data. To make LLMs trustworthy, a range of properties need to be considered, such as accuracy, reliability, generalization, robustness, interpretability, and fairness [34, 48]. With advancements in LLM technology and increased research efforts in the field, issues surrounding the trustworthiness of LLMs may be resolved to a certain extent, such as the recent efforts [42, 35].

What are the challenges of integrating LLMs into existing safety analysis processes, and how can they be overcome? Integrating LLMs with current safety analysis processes can be challenging due to differences in terminology and methodologies. The development of the three use cases presented in this paper was intended to address this issue, as an initial endeavor. However, the question of which collaboration mode is more beneficial for safety analysis requires further exploration, e.g., via interdisciplinary research with the Human-Computer-Interaction community. Moreover, the automation of such “safety-experts-LLMs-teaming” will be the trend. We also believe, there is a pressing need of standardisation on such new way of conducting safety analysis. That is, the integration of LLMs into safety analysis process should be formalised and documented in safety standards, and covered by regulation.

What are the implications of using LLMs in safety analysis for the training and education of safety professionals? To effectively use LLMs in safety analysis, as shown in our *Case*

3, the human expert teams should possess a good knowledge of AI, engineering, safety, etc. In the current safety analysis tasks for the given system under analysis, such knowledge set is essential in order to deliver comprehensive analysis results. While, we believe, such “safety-experts-LLMs-teaming” may also provide long-term education for the human safety professionals. That is, working with LLMs intensively may also train the safety analyst by expanding their knowledge base.

6 Conclusion

Impressed by the recent successful stories of ChatGPT in many domains, we first pose the question: “if safety analysis can actually make use of LLMs?”. To answer, we conducted a case study of applying ChatGPT in the STPA for an AEB system, which is the first attempt of using LLMs in safety analysis, to the best of our knowledge. Specifically, for comparative studies, we not only set an STPA result obtained by human safety experts (published in [40]) as the baseline, but also developed three use cases of integrating ChatGPT and the human expert team. We identify the most practical way of incorporating ChatGPT into the STPA process, which may address the high complicity and subjectivity limitations of STPA. While we concluded remarks (e.g., on the reliability, robustness, reproducibility and interactivity of ChatGPT) and research questions representing the opportunities and challenges (e.g., trustworthiness concern of LLMs, the need of standardisation and regulation, compatibility with dynamic assurance, long-term education for safety professionals), our work contributes to the development of a more integrated and intelligent safety analysis framework that leverages the power of LLMs.

Acknowledgments

This work is supported the U.K. EPSRC through End-to-End Conceptual Guarding of Neural Architectures [EP/T026995/1]. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 956123. XZ’s contribution to the work is partially supported through Fellowships at the Assuring Autonomy International Programme, and YQ’s contribution to the work is supported through Chinese Scholarship Council.

References

- [1] A. Adriaensen, L. Pintelon, F. Costantino, G. Di Gravio, and R. Patriarca. An stpa safety analysis case study of a collaborative robot application. *IFAC-PapersOnLine*, 54(1):534–539, 2021.
- [2] E. Asaadi, E. Denney, J. Menzies, G. J. Pai, and D. Petroff. Dynamic assurance cases: A pathway to trusted autonomy. *Computer*, 53(12):35–46, 2020.
- [3] F. Benaben, M. Larnal, J.-P. Pignon, and J. Magnier. A process for improving multi-technology system high level design: modeling, verification and validation of complex optronic systems. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 2, pages 1036–1040, 2000.
- [4] R. Bloomfield, H. Khlaaf, P. R. Conmy, and G. Fletcher. Disruptive innovations and disruptive assurance: Assuring machine learning and autonomy. *Computer*, 52(9):82–89, Sept. 2019.
- [5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child,

- A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [6] R. Calinescu, D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, and T. Kelly. Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Transactions on Software Engineering*, 44(11):1039–1069, 2018.
- [7] S. Chen, S. Khastgir, I. Babaev, and P. Jennings. Identifying accident causes of driver-vehicle interactions using system theoretic process analysis (stpa). In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3247–3253. IEEE, 2020.
- [8] J. H. Choi, K. E. Hickman, A. Monahan, and D. Schwarcz. Chatgpt goes to law school. *Available at SSRN*, 2023.
- [9] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.
- [10] A. L. Dakwat and E. Villani. System safety assessment based on stpa and model checking. *Safety Science*, 109:130–143, 2018.
- [11] I. Dasgupta, A. K. Lampinen, S. C. Chan, A. Creswell, D. Kumaran, J. L. McClelland, and F. Hill. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*, 2022.
- [12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [13] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res*, 23:1–40, 2021.
- [14] A. Gilson, C. W. Safranek, T. Huang, V. Socrates, L. Chi, R. A. Taylor, D. Chartash, et al. How does chatgpt perform on the united states medical licensing examination? the implications of large language models for medical education and knowledge assessment. *JMIR Medical Education*, 9(1):e45312, 2023.
- [15] E. Harkleroad, A. Vela, and J. Kuchar. Review of systems-theoretic process analysis (stpa) method and results to support nextgen concept assessment and validation. *Project Report: ATC-427, MIT, Lincoln Laboratory, Washington, DC, Oct, 25:2018–12*, 2013.
- [16] N. Houlsby, A. Giurghi, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.
- [17] IEC 61508:2010. Functional Safety of Electrical/ Electronic/Programmable Electronic Safety Related Systems. Standard, IEC, Geneva, Switzerland, 2009.

- [18] T. Ishimatsu, N. G. Leveson, J. Thomas, M. Katahira, Y. Miyamoto, and H. Nakao. Modeling and hazard analysis using STPA. In *Proc. of the 4th IAASS Conference, Making Safety Matter*, Huntsville, Alabama, 2010.
- [19] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.
- [20] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021.
- [21] T. Kaneko, Y. Takahashi, T. Okubo, and R. Sasaki. Threat analysis using stride with stamp/stpa. In *The International Workshop on Evidence-based Security and Privacy in the Wild*, 2018.
- [22] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [23] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.
- [24] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings. Systems approach to creating test scenarios for automated driving systems. *Reliability Engineering & System Safety*, 215:107610, 2021.
- [25] T. H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño, M. Madriaga, R. Aggabao, G. Diaz-Candido, J. Maningo, et al. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLOS Digital Health*, 2(2):e0000198, 2023.
- [26] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*, 2020.
- [27] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.
- [28] N. Leveson. *Engineering a Safer World: Systems Thinking Applied to Safety*. Engineering systems. MIT Press, 2011.
- [29] N. G. Leveson and J. P. Thomas. STPA handbook. *Cambridge, MA, USA*, 2018.
- [30] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 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*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.
- [31] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

- [32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [33] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heinz, and D. Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*, 2021.
- [34] OpenAI. GPT-4 Technical Report. *arXiv e-prints 2303.08774*, 2023.
- [35] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.
- [36] Y. Qi, Y. Dong, X. Zhao, and X. Huang. Stpa for learning-enabled systems: A survey and a new method. *arXiv preprint arXiv:2302.10588*, 2023.
- [37] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.
- [39] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, 2019.
- [40] L. Sun, Y.-F. Li, and E. Zio. Comparison of the HAZOP, FMEA, FRAM, and STPA Methods for the Hazard Analysis of Automatic Emergency Brake Systems. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 8(3), 10 2021. 031104.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [42] J. Wang, X. Hu, W. Hou, H. Chen, R. Zheng, Y. Wang, L. Yang, H. Huang, W. Ye, X. Geng, et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*, 2023.
- [43] P. Yang, R. Karashima, K. Okano, and S. Ogata. Automated inspection method for an stamp/stpa-fallen barrier trap at railroad crossing. *Procedia Computer Science*, 159:1165–1174, 2019.
- [44] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32, 2019.
- [45] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*, 2021.

- [46] W. Zeng, X. Ren, T. Su, H. Wang, Y. Liao, Z. Wang, X. Jiang, Z. Yang, K. Wang, X. Zhang, et al. Pangu- α : Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*, 2021.
- [47] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR, 13–18 Jul 2020.
- [48] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.
- [49] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.