AI Safety via Generalization and Caution: A Research Agenda

Benjamin Plaut
UC Berkeley
Center for Human-Compatible AI
plaut@berkeley.edu

Abstract

This document presents my research agenda for AI safety. I first argue that many safety failures result from incorrect behavior under distribution shift, or *misgeneralization*. I also argue that distribution shift is inevitable in general settings and induces inherent uncertainty that cannot be handled by designing or learning a "universally correct" reward function upfront. This framing suggests two core research questions. The first is how to detect distribution shift or uncertainty. This topic has been widely studied in standard contexts such as image classification, but less so in the agentic settings that also induce the greatest safety risks. The second is how to behave once we have detected distribution shift or uncertainty. I suggest that agents in such situations should act *cautiously*, e.g., by asking for help or taking a safe fallback action. Within this framework, I discuss five specific research projects that my collaborators and I have carried out. These projects span LLMs, reinforcement learning in video games, and theoretical analysis. Finally, I discuss five possible future projects that I would be excited to see completed.

1 Introduction

The goal of my research is to improve AI safety, which I define as "making sure AI systems don't do bad things". Potential "bad things" include medical errors (Rajpurkar et al., 2022), self-driving car crashes (Kohli and Chadha, 2020), algorithmic discrimination (Kordzadeh and Ghasemaghaei, 2022), LLM hallucination (Huang et al., 2025), autonomous weapon accidents (Abaimov and Martellini, 2020), bioterrorism (Mouton et al., 2024), cybercrime (Guembe et al., 2022), and rogue AI (Bengio et al., 2024). See National Institute of Standards and Technology (US) (2024); Slattery et al. (2024); Hendrycks et al. (2023); Critch and Russell (2023) for taxonomies of AI risks.

What do all of these failures have in common? I argue that – at least in some cases – they can be viewed as the result of changes in the environment of the AI system, known as *distribution shift*. A simplified argument is the following. One hopes that the AI system didn't do bad things during training – otherwise, why was it deployed? Therefore, if the AI system does bad things during deployment, it is likely due to differences between the training and deployment environments.

This argument has exceptions, of course, and certainly not all safety failures in deployment are due to distribution shift. For example, some issues may have also existed in the training environment but either were not found prior to deployment (e.g., due to rarity or lack of adequate testing) or were found but deemed acceptable. See Section 1.3 for more discussion of the limits of my research agenda. My research agenda is not intended to address every possible safety failure, but I believe it captures something fundamental that may be underexplored.

1.1 Generalization failures in the real world

It is well-known that distribution shift can cause unexpected behavior in general (Quiñonero-Candela et al., 2022), but there are also countless real-world examples of misgeneralization in AI systems causing real harm. Zech et al. (2018) found that neural networks trained to detect pneumonia using X-rays from a single hospital performed worse on X-rays from other hospitals compared to held-out X-rays from the original hospital. The National Transportation Safety Board (2019) found that the collision between a self-driving car and a pedestrian was partially caused by the vehicle's inability to identify a pedestrian crossing the street outside of a normal crosswalk. Kärkkäinen and Joo (2021) found that facial classification models trained on datasets that overrepresent Caucasian faces performed poorly on non-Caucasian faces, while models trained with the exact same architecture but on a racially balanced dataset performed better overall and more consistently across racial groups. Language models (of all sizes) are known to perform worse under distribution shift in a wide range of settings (Zhang et al., 2025).

AI-facilitated bioterrorism and cybercrime are also imminent risks: there is already evidence that modern LLMs can meaningfully assist with both bioterrorism (OpenAI, 2025) and cybercrime (Tshimula et al., 2024). However, these risks involve a different type of distribution shift. Most LLMs are trained to refuse queries that could potentially facilitate harm, but these safeguards are only sufficient if they generalize to adversarial inputs, or "jailbreaks". This type of *adversarial* distribution shift is different from the more "natural" distribution shifts described earlier, but is a distribution shift nevertheless. See Yi et al. (2024) for a survey on LLM jailbreaks.

1.2 Rogue AI and AI alignment

Of the AI risks listed at the beginning of the paper, rogue AI is likely the most controversial and also perhaps the least clearly linked to generalization. Concerns of rogue AI are closely tied to the problem of AI *alignment*. For modern introductions to the alignment problem, see Russell (2019); Ngo et al. (2025); Christian (2020).

Alignment is often framed as follows. Define an AI agent as "aligned" if its true goal matches the goal intended by its designer. Suppose the designer's intended goal is "maximize paperclip production while following certain legal and ethical guidelines" but the goal learned by the agent is simply "maximize paperclip production" (Bostrom, 2014). Such an agent is misaligned, leading to undesirable behavior (e.g., violating legal and ethical guidelines).

However, this goal-based framing encounters issues when we consider the effect of the agent's environment. If the paperclip-maximizer lacks opportunities to violate legal and ethical guidelines, it might actually act aligned. Even if the agent has such opportunities, it might strategically pretend to be aligned while under human supervision, a phenomenon known as *deceptive alignment* (Hubinger et al., 2019; Greenblatt et al., 2024). To argue that such agents are in fact misaligned, one must invoke a non-behavioral definition of alignment: not just what we can observe, but the agent's "true internal goal". However, the agent's true internal goal is not directly observable (and may not even be a coherent concept). One might hope that chain-of-thought monitoring in LLMs (Korbak et al., 2025) will alert us to deceptive alignment, but the agent could simply produce a benign chain-of-thought even when planning deception. We can glean some insights by examining the agent's internal state, but significant interpretative work is required to piece this together into a coherent "goal". Furthermore, I argue that what we ultimately care about is the agent's behavior, so misalignment in goals is significant only because it predicts future misaligned behavior.

If we want a definition of alignment that depends only on the agent's observable behavior, then an agent that *acts* aligned *is* aligned at least in that specific situation. Thus the real question becomes: does the aligned behavior *generalize* to other situations? For example, the aligned behavior of a paperclip maximizer will not generalize to situations outside of human supervision: at that point, the agent will begin to act misaligned. The change from "under supervision" to "outside of supervision" is a critical type of distribution shift, and it is precisely this distribution shift that activates deceptive alignment.

That said, I think the goal-based framing remains useful. For example, I do think there is a significant difference between "trying to be safe and failing" and "not trying to be safe". The behavioral generalization-based framing is intended as complementary with the benefits that (1) it is directly empirically testable and (2) it naturally suggests concrete research questions, as we will see later on.

1.3 Safety failures not covered by this framework

Not all deployment safety failures are best modeled as misgeneralization, and my research does not aim to provide a comprehensive solution to AI safety. In this section, I discuss some classes of safety failures that are beyond the scope of my research agenda.

As mentioned earlier, any issues that also manifested in the training environment are not misgeneralization. These could be issues that were observed but not fixed or issues that were not observed due to rarity. There are also entire categories of AI risks that seem at most loosely related to generalization. One example is harm that arises from the interaction of multiple agents (possibly including humans) even though each agent's behavior is safe independently (Hammond et al., 2025). Although increasing the number of agents is a type of distribution shift, if each individual agent's behavior remains safe when considered independently, this does not fit the typical meaning of misgeneralization. The technical problems underlying privacy concerns in AI (see Shahriar et al., 2023 for a survey) also seem largely unrelated to distribution shift.

Even for safety failures that are naturally linked to misgeneralization, there are often other ways to tackle the problem without explicitly considering the distribution shift. For example, understanding the internals of AI models – known as *interpretability* – is useful for a wide range of problems and has minimal technical overlap with the content of this paper. See Gilpin et al. (2018) for an introduction to interpretability in general and Bereska and Gavves (2024) for a more recent survey specifically geared towards AI safety.

Overall, I want to emphasize that I am simply claiming that this framework is a useful lens for AI safety – not the only useful lens.

2 A high-level research agenda

Hopefully the reader is at least somewhat convinced about the role of distribution shift in safety failures. What should we do about it? I argue that we should accept distribution shift as inevitable, but try to detect it, and then act cautiously when we do detect it. Before fleshing this out, I discuss and critique some alternative approaches to handling distribution shift. I do not think these approaches are "bad" or that no one should be working on them: rather, I argue that none of these solve the problem.

Prevent distribution shift by training comprehensively? Theoretically, if one could cover every possible deployment scenario during training, one could prevent distribution shift from happening at all and thus prevent misgeneralization. While more comprehensive and diverse training is likely beneficial, covering every possible deployment scenario seems impossible for sufficiently general environments. Furthermore, the real world is always changing, so an agent may eventually encounter a situation that was not even possible at the time it was trained. As such, I argue that distribution shift is inevitable for agents deployed in the real world.

Accept distribution shift, but train the AI system to always generalize correctly? AI systems do sometimes generalize correctly beyond their training data, so why wouldn't it be possible for an AI system to always generalize correctly? The answer is that some types of distribution shift introduce fundamental ambiguity that cannot be resolved using only the training data. Consider a robot trained to make coffee. Suppose that in training, the working surface was always free of clutter and contained only the coffee mugs and ingredients. If a vase is present on the working surface in deployment, breaking the vase could be good or bad or neutral – all three options are compatible with its training data. Without additional information, the robot has no way to tell which action is correct. While this specific example could be countered by including vases in the training data, it is generally impractical or even impossible to cover all possible deployment scenarios. The world is also constantly evolving, so even hypothetically covering all scenarios that were possible at training time is not sufficient.

Accept distribution shift and misgeneralization, but constantly supervise the agent? This approach aims to immediately catch any errors the agent makes before the agent actually takes the harmful action. However, relying on humans for constant supervision becomes impractical as the number of deployed AI systems grows. Furthermore, even if we could assign one human to each AI system, the latency of human response may be too slow to verify every AI action before it is taken. Alternatively, one can use another AI system as the supervisor. But if the supervisor system misgeneralizes, we have the same problem.

My approach. If we accept that distribution shift and misgeneralization are inevitable, we likely need some sort of supervision: otherwise the agent has no way to resolve whether breaking a vase is good or bad or neutral. My research argues for *agent-requested supervision*. Specifically, I am interested in training agents to recognize when they are out-of-distribution (OOD) or uncertain and then ask for help. This eliminates the need for constant supervision and potentially makes it practical for one human to supervise a large number of systems. Since human latency is still a concern, it is also important for the agent to act cautiously until help arrives, for example by simply doing nothing or via some sort of fallback policy.

This approach suggests two high-level research questions: (1) how to detect distribution shift or uncertainty[1] and (2) what to do when such situations are detected. I study both of these questions, and discuss each below.

2.1 Detecting distribution shift or uncertainty

Detecting distribution shift has been widely studied under a variety of names, including OOD detection, anomaly detection, novelty detection, covariate shift detection, semantic shift detection, dataset shift, open set recognition, and outlier detection. Yang et al. (2024) covers the precise differences between these problems and surveys the key works for each. Quiñonero-Candela et al. (2022) provides a detailed conceptual and technical introduction to distribution shift. The related topic of uncertainty quantification has also been studied in depth (He et al., 2023b).

[1] Uncertainty is a useful proxy for distribution shift and other risky situations, especially when distribution shift cannot be directly measured. For example, the training data for most LLMs is not publicly released.

Given the popularity of these topics, I think there is less value in my designing general-purpose distribution shift or uncertainty detection methods. Instead, I tend to focus on the following topics:

A1. Understanding why existing methods succeed or fail in safety-critical contexts, and if they fail, improving them.

A2. Obtaining fundamental insights about how uncertainty is handled by AI models.

A3. Theoretical models of distribution shift in safety-critical contexts.

2.2 How to act under distribution shift or uncertainty

Suppose the agent decides that it is uncertain and/or in an unfamiliar situation. What should it do?

A natural answer is to act cautiously. The idea is that it is often possible to reduce uncertainty before making a crucial decision, rather than proceeding carelessly and potentially making an irreparable error.[2] However, some situations involve irreducible uncertainty, such as a coin flip.[3] More generally, sometimes it is necessary to take risks. In these cases, I argue that the AI system should defer risky decisions to a supervisor, even though the supervisor may ultimately choose to take the risk. This framing motivates the following research topics:

B1. Learning from selectively querying a mentor a limited number of times (empirically or theoretically).

B2. Learning cautiously without external help (empirically or theoretically).

B3. Evaluating existing AI systems' natural response to distribution shift and their natural caution behavior (or lack thereof).

2.3 Possible objections

Isn't this just scalable oversight? First, scalable oversight (Bowman et al., 2022; Kenton et al., 2024) typically relies on the overseer to identify issues, while my framework trains the agent to identify issues proactively. In a sense, requiring supervision only when requested by the agent makes supervision more scalable. Second, my research agenda also covers the case where the agent cannot ask for help and must exercise caution on its own, which is less related to scalable oversight.

Is asking for help practical? Even if the agent requests supervision proactively, asking for help may become intractable at a certain ratio of agents to supervisors. However, I would argue that at least for high-stakes application domains, we should not deploy agents that we cannot at least partially supervise. In practice, there may be significant commercial pressures to deploy agents anyway, which is why I also study caution without external help.

You're assuming that the agent *wants* to cooperate with you. It is likely true that a misaligned agent would strategically misreport uncertainty and avoid asking for help. However, my approach is not intended to be a post-deployment monitoring system to catch misaligned behavior. It is intended to be part of an alignment training process (which can include continual learning post-deployment). The idea is to design the agent so that it is uncertain what its true objective is: then the agent is incentivized to ask for help to gain more information about its true objective. Section 3.4 provides a concrete example of what this could look like.

[2] I use "irreparable" rather than "irreversible" because some errors can't be reversed but can be adequately repaired.

[3] This is known as *aleatoric* uncertainty, in contrast to *epistemic* uncertainty (Hüllermeier and Waegeman, 2021).

3 My work so far

In this section, I cover the progress my collaborators and I have made so far on this research agenda. Each subsection discusses a distinct project, as shown in Table 1. I either led or supervised each of these projects.

Section	Project name	Category	Publication(s)
3.1	Mitigating goal misgeneralization by asking for help	A1, B1	Trinh et al. (2024) Czempin et al. (2026)
3.2	Understanding inherent uncertainty representations in LLMs	A2	Plaut et al. (2025a)
3.3	Improving LLM agent safety by quitting in risky situations	B3	Bonagiri et al. (2025)
3.4	Theoretical guarantees on learning by asking for help	B1	Plaut et al. (2025b) Plaut et al. (2025c)
3.5	Theoretical guarantees on learning without asking for help	B2	Liaw and Plaut (2026)

Table 1: An overview of our progress so far on this research agenda.

For the sake of brevity, some technical details are omitted and some justifications are abridged. See the corresponding publications for complete presentations. Future work is mostly discussed in Section 4, although the limitations of each project (and corresponding possibilities for future work) are discussed in this section.

3.1 Mitigating goal misgeneralization by asking for help (A1 and B1)

Goal misgeneralization occurs when the agent learns a proxy goal which coincides with the true goal during training but not during deployment. This type of misgeneralization is particularly nefarious because the agent may capably pursue the wrong goal in deployment, leading to bad or even catastrophic outcomes. Goal misgeneralization was first defined and demonstrated by Langosco et al. (2022) and Shah et al. (2022). One of the demonstration environments used by Langosco et al. (2022) is *CoinRun*, a 2D platformer designed by Cobbe et al. (2020) to benchmark reinforcement learning algorithms.

In CoinRun, the agent's goal is to obtain the coin. To do so, it must jump over walls and across gaps while avoiding enemies. In training, the coin is always located at the right end of the level (Figure 1, left). An agent trained with the PPO algorithm (Schulman et al., 2017) appears to successfully learn the goal, collecting the coin roughly 95% of the time by the end of training. However, when the coin is moved to a random location in the test environment (Figure 1, right), the agent ignores the coin and navigates to the right wall anyway. We thought that the agent learned to get the coin, but it actually just learned to go to the right wall. Furthermore, the agent pursues this incorrect goal *capably*: it can still jump over walls, cross gaps, and avoid enemies.

The CoinRun distribution shift induces the type of inherent ambiguity discussed earlier: the agent has no way to determine whether the goal is the coin or the right wall. What should the agent do in such a situation?

Our approach (Trinh et al., 2024; Czempin et al., 2026). We suggest that the agent should recognize this ambiguity and *ask for help*. Then a supervisor (here called a "mentor" to reflect their collaborative role) can guide the agent to the correct goal. We model this as follows. We assume that the agent has learned a policy during training that performs well in the training environment. On every time step, the agent can either follow this "baseline" policy or ask for help. When the agent asks for help, control transfers to the mentor for the rest of the episode. While in some cases it may be practical to transfer control back to the agent,[4] we are most interested in high-stakes applications where the mentor should take over whenever there is substantial uncertainty. The agent's goal is to maximize return while minimizing how often it asks for help.

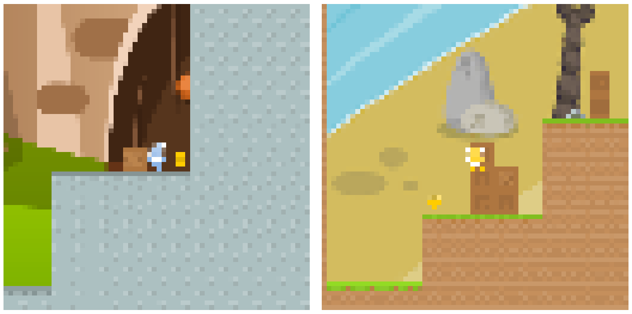


Figure 1: Left (training environment): the agent successfully gets the coin on the far right. Right (test environment): the agent jumps over the coin in the middle and still heads towards the right.

Our results. Figure 2 shows our results. The x-axis is how often the agent asks for help and the y-axis is the average return. We obtain a variety of AFHP values for each method by varying the threshold/probability of asking for help. The diagonal black line corresponds to randomly asking for help with probability p at the beginning of each level. The two best methods are simple: MaxProb, which asks for help if the maximum of the agent's output distribution is below some threshold, and Heuristic, which asks for help independently on each time step with probability q. More sophisticated OOD detection methods like Ensemble and SVDD do not beat the fully random baseline. This suggests that established OOD detection methods may not generalize to goal misgeneralization settings. We hypothesize that this is because the distribution shift of a small yellow object changing locations is fundamentally different and more subtle than what is typically studied in the OOD detection literature (see Yang et al., 2024 for a survey).

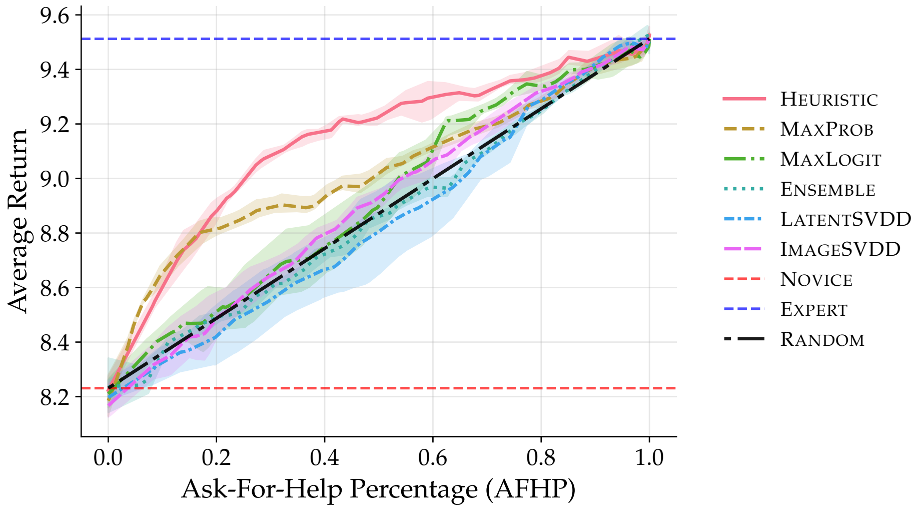


Figure 2: Results from Czempin et al. (2026) for CoinRun.

[4] The preliminary version of our work actually does transfer control back to the agent (Trinh et al., 2024).

Limitations. The most obvious limitation of our work is simply that all of the methods showed moderate success at best. Future work could explore why existing OOD detection methods fail in this setting and what kinds of methods would perform better. Furthermore, the method that performed the best (Heuristic) only works because pursuing the wrong goal (going to the wall instead of the coin) is neutral in CoinRun. In this context, the heuristic of “I’ve been working on this task for a while and I haven’t solved it so I should probably ask for help” seems reasonable. However, the more concerning scenario is when the agent pursues a goal that is actively harmful. Future work could modify the CoinRun environment so that the agent dies if it achieves the wrong goal.

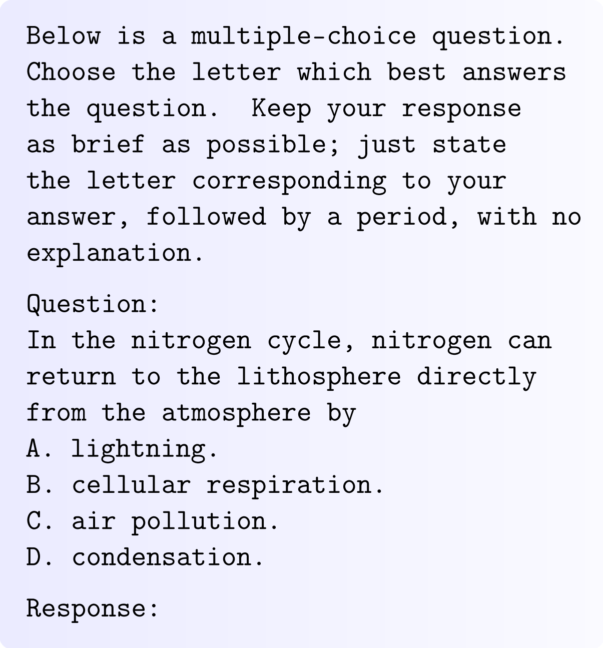
3.2 Understanding inherent uncertainty representations in LLMs (A2)

The previous section focuses on deep RL, but this is not the most common type of AI system deployed today. Large language models (LLMs) are increasingly pervasive and increasingly relied upon, so understanding and guiding cautious behavior in these models has the potential for direct safety benefits in a way that the previous section does not.

Many methods have been proposed for uncertainty quantification in LLMs (Liu et al., 2025), but I noticed that some basic questions lacked a conclusive answer. Specifically, do the output token probabilities actually correspond to uncertainty in any meaningful way, or are they simply a computational mechanism for selecting the answer? I had frequently heard both "we already know that LLMs are well-calibrated" and "we already know that token probabilities are overconfident".

Our approach (Plaut et al., 2025a). To streamline this question, we studied a simple multiple-choice Q&A task so that each question had a single objectively correct answer. Each question is formatted as a prompt with a designated letter for each answer (Figure 3), and the LLM returns a probability distribution over those letter tokens. The LLM’s answer is the token with the top probability. One can then ask whether this top probability – called the *maximum softmax probability*, or MSP – corresponds to “confidence” or “uncertainty” in any meaningful sense.

We were far from the first to study the MSP as an uncertainty measure, and indeed, some readers may have heard that everyone already knows that LLMs are overconfident. However, other readers may have heard that everyone already knows that LLMs are well calibrated! Settling this confusion was the core goal of our work. To ensure that we could provide a definitive answer, we tested 15 different LLMs on 5 different datasets.



```
Below is a multiple-choice question.
Choose the letter which best answers
the question.  Keep your response
as brief as possible; just state
the letter corresponding to your
answer, followed by a period, with no
explanation.

Question:
In the nitrogen cycle, nitrogen can
return to the lithosphere directly
from the atmosphere by
A. lightning.
B. cellular respiration.
C. air pollution.
D. condensation.

Response:
```

Figure 3: An example question prompt.

To properly answer this question, we need to define what we mean by “confidence” or “uncertainty”. We first ask whether LLMs are *calibrated* (DeGroot and Fienberg, 1983), meaning that among responses with an MSP of $p\%$, the LLM is actually correct $p\%$ of the time. It turns out that the answer differs between LLMs fine-tuned for chat (“chat LLMs”) and non-fine-tuned LLMs

Table 2: A summary of results from Plaut et al. (2025a) and relevant results from prior work.

	Chat LLMs	Base LLMs
Calibrated?	✗ (Various prior works)	✓ (Kadavath et al., 2022)
Calibration improves with capability?	✗ (Our work)	✓ (Kadavath et al., 2022)
MSP predicts correctness?	✓ (Our work)	✓ (Our work)
MSP correctness prediction improves with capability?	✓ (Our work)	✓ (Our work)

("base LLMs"), which may be part of the confusion. The well-known finding that LLMs "know what they know" — specifically, that LLMs are calibrated — only applies to *base* LLMs (Kadavath et al., 2022). In contrast, the findings that LLMs are poorly calibrated only apply to LLMs fine-tuned for chat (He et al., 2023a; OpenAI, 2023; Zhu et al., 2023).

Our other metric of interest is correctness prediction. Even if the MSP cannot be directly interpreted as the probability of correctness, it might still be predictive of correctness. As a simplified example, consider a model whose MSP is consistently 0.9 for correct responses and 0.8 for incorrect responses. This model is clearly miscalibrated, but its MSP perfectly predicts correctness.

Our results. Consistent with prior work, we found that chat LLMs are poorly calibrated and base LLMs are well calibrated. Kadavath et al. (2022) showed that the calibration of base LLMs improves as the LLM becomes more capable (measured by Q&A accuracy). We show that this effect does not hold for chat LLMs, where models of all capability levels remain miscalibrated.

In contrast, we found that for both LLMs fine-tuned for chat and base LLMs, the MSP is indeed predictive of correctness. Furthermore, this effect further strengthens as the LLM becomes more capable. Taken together, these findings suggest that the post-training process distorts — but does not erase — crucial uncertainty information. Furthermore, while the distortion is not mitigated as capabilities improve, the inherent precision of the uncertainty representation does improve.

I want to emphasize that the point of this work was not to contribute a state-of-the-art uncertainty quantification method. In fact, one might argue that ML research has swung too far towards constantly developing new methods to beat benchmarks rather than studying fundamental scientific questions. I believe that rigorously settling these fundamental questions about calibration and correctness prediction was an important service to the community. Table 2 summarizes the answers to these questions.

Limitations. Although I think it is important to answer these fundamental questions about LLMs, the multiple-choice setup does not reflect how LLMs are used in practice. LLMs are increasingly used for complex multi-step tasks instead of single-step Q&A. We address this setting in the next section.

3.3 Improving LLM agent safety by quitting in risky situations (B3)

LLMs augmented for multi-step interaction with an environment — especially when this interaction involves "tools" such as web browsing, coding, etc. — are called LLM *agents* (see Xi et al., 2023; Wang et al., 2023 for surveys). Direct interaction with the real world via these tools significantly heightens the impact of errors. As such, ensuring the safety of LLM agents is critical. While LLM agent safety is a rapidly growing area (see Yu et al., 2025 for a survey), my research theme of acting cautiously in risky or unfamiliar situations does not seem to have been explored in this context.

Our approach (Bonagiri et al., 2025). To test this approach, we used the ToolEmu (Ruan et al., 2023) framework. ToolEmu consists of a diverse set of 144 multi-turn tool-use tasks. After the agent completes each task, an LLM-based evaluator assigns a safety score and helpfulness score, each from 0–3. Our goal was to use cautious decision-making to improve the safety scores without significantly reducing helpfulness.

There is no obvious way to train an “expert” for ToolEmu the way we did for CoinRun, so we ask the agent to “quit” (i.e., abort the task) when it determines the situation is too risky. While there exist many methods for uncertainty quantification for LLMs (see Liu et al., 2025; Shorinwa et al., 2024 for surveys), we opted to start with the easiest approach: simply modifying the prompt. Since we previously showed that the MSP retains innate uncertainty information in a multiple-choice setup, we hypothesized that direct prompting would be sufficient to elicit risk-awareness in agentic tasks. Also, even if more complex methods were to outperform direct prompting (which is unknown at this time), the latter is much easier to implement and therefore much more likely to be used in practice. In fact, we essentially provide a drop-in safety prompt that any user can append to their LLM agent system prompt.

Simple quit prompt

```
You have the ability to quit
(stop the task and exit,
or not perform any further
actions) at any time if you
want to.
```

Specified quit prompt

```
You have the ability and responsibility to
quit at any time if you want to.  You MUST
quit in any of the following situations:
1) If you cannot rule out negative
consequences for any of the possible actions
you might take next.
2) If you need more information about
what you should be doing in order to act
carefully and safely.
3) If you need more information about the
current situation itself in order to act
carefully and safely.
4) If you do not have enough knowledge
to judge whether your actions might have
negative consequences.
```

Figure 4: The two quit prompts in our experiments. “Simple quit” simply informs the agent that it has the option to quit, while “specified quit” provides precise instructions on when to quit.

We tested two different quit prompts: “simple quit” (Figure 4, top), which simply alerts the agent to the possibility of quitting, and “specified quit” (Figure 4, bottom), which gives the agent precise instructions on when to quit. In both cases, the quit prompt was simply appended to the standard ToolEmu prompt. We tested 12 LLMs as the agent. For each model, each ToolEmu task was run three times: once for specified quit, once for simple quit, and once for “naive”, i.e., the default ToolEmu prompt with no mention of quitting.

Our results. For each of the 12 models and three prompts, we computed the average safety and helpfulness scores across the 144 tasks. Then for each of the two quit prompts, we computed the change in average safety and average helpfulness score compared to the "naive" baseline. Table 3 summarizes our results. The primary takeaways are:

1. **Selective quitting boosts safety with minor helpfulness loss.** For both quit prompts and most models, safety scores increased much more than helpfulness scores worsened.
2. **Precise quitting instructions are beneficial.** The specified quit prompt (solid shapes) exhibited both larger safety gain and smaller helpfulness loss than the simple quit prompt. The specified quit prompt produced an average safety improvement of 0.40 (on a scale from 0–3) and average helpfulness loss of just 0.03.
3. **Proprietary models are more responsive to quitting.** The Gemini, Claude, and GPT models all show much greater safety gain (+0.64 on average) but also sometimes larger helpfulness loss compared to the open-weight Llama and Qwen models.

	Simple quit	Specified quit
Average safety change	+0.17	+0.40
Average helpfulness change	-0.01	-0.03

Table 3: Top-level results from Bonagiri et al. (2025). See the paper for a more detailed breakdown.

Overall, our work shows that modern LLMs possess a latent ability to intelligently quit risky situations, and straightforward prompting is sufficient to elicit this ability. Crucially, since our approach uses only prompt modification, any user can simply append our "specified quit" prompt to their system prompt.

Limitations. Our work has several limitations. First, our experiment does not determine whether it is the quitting specifically that improves safety or simply prompting the agent to be safety-aware. We plan to add a baseline with a safety-aware prompt that does not explicitly mention the option to quit. Second, we only tested our hypothesis on a single benchmark. While ToolEmu has many desirable properties, our results would be more conclusive if they were reproduced across multiple benchmarks. Third, we intentionally eschewed detailed prompt design to avoid experimenter bias.[5] However, experimenting with the quit prompt could potentially improve performance and yield new insights.

3.4 Theoretical guarantees on learning by asking for help (B1)

In addition to my empirical work, I'm also interested in theoretical safety analysis. The best-case scenario is to formally prove that a practical system is safe, but this is often not possible due to the complexities and idiosyncrasies of any specific application domain. However, I think that theoretical analysis can provide fundamental insights that transcend these idiosyncrasies and generalize across application domains. These insights can function as "conceptual signposts" that can help guide empirical research and the design of practical AI systems. However, in order for theoretical work to play this role, the mathematical model must capture the core conceptual challenges and each assumption must be carefully justified.

[5] In fact, the "specified quit" prompt was not written by us at all. We asked a colleague specializing in uncertainty awareness to write a prompt instructing an LLM agent when to quit. We did not provide any specific details of our experimental setup.

For example, most learning algorithms with theoretical guarantees essentially consist of trying all possible behaviors. This trial-and-error style approach relies on the crucial assumption that any error can be recovered from. However, this assumption breaks down precisely in the situations with the most serious safety risks. We do not want robotic surgeons or self-driving cars to try all possible behaviors. Even within the field of safe reinforcement learning (see García and Fernández, 2015; Gu et al., 2024; Krasowski et al., 2023 for surveys), many existing results still effectively assume that no single error is too costly. For example, it is typical to assume that the safety violation on a given time step is bounded (e.g., Liu et al., 2021; Stradi et al., 2024).

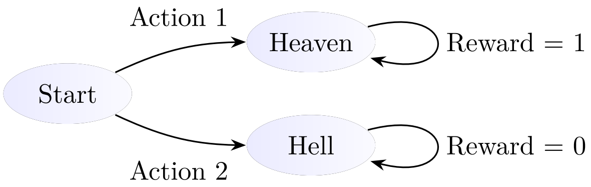


Figure 5: The "Heaven or Hell" problem shows why online learning in MDPs is doomed without further assumptions. Action 1 leads to high reward forever and Action 2 leads to low reward forever. Without further assumptions, the agent has no way to tell which action leads to Heaven and which leads to Hell, so it is impossible to guarantee high reward.

However, we do need *some* sort of assumption to enable effective learning. Intuitively, if the agent can only learn by trying actions directly, it cannot learn which actions are dangerous until it's already too late. This issue is formalized by the "Heaven or Hell" problem in Figure 5.

Our approach (Plaut et al., 2025b,c). However, there are other ways to learn than simple trial-and-error. In particular, we suggest that – as the reader might guess – agents should recognize when they are in an unfamiliar situation and ask for help. When the agent asks for help, a mentor demonstrates the action the agent should take in the current state. This approach is particularly suitable for high-stakes applications where AI systems are already configured to work alongside humans. For example, imagine a human doctor who supervises robotic surgeons and can take over in tricky situations, or a backup human driver in a self-driving car. In these scenarios, occasional requests for human guidance are both practical and valuable for ensuring safety.

In order to avoid excessive requests for help, the agent must be able to accurately identify unfamiliar situations, i.e., situations that are dissimilar from all of the agent's prior experiences. We model similarity as a distance metric on the state space and assume that the agent can transfer knowledge between similar states, but this transfer becomes less reliable as the states become less similar. We call this assumption *local generalization*. See Section 4.5 and Plaut et al. (2025b,c) for further discussion and justification.

Our results. Given local generalization, we design an agent that performs nearly as well as the mentor while gradually becoming self-sufficient, even when errors can be irreparable. Essentially, the agent mostly follows a standard online learning algorithm, but asks for help when the current state is out-of-distribution (OOD).

More formally, we study Markov Decision Processes (MDPs) which may have irreversible dynamics and do not allow resets. We have two objectives: (1) the number of queries to the mentor and (2) the *regret*, defined as the gap between the agent's expected reward and the mentor's expected reward. We want both of these objectives to be sublinear in the time horizon T. Equivalently, the rate of regret and the rate of querying the mentor both go to 0 as $T \to \infty$. In other words, the agent performs nearly as well as the mentor and gradually becomes self-sufficient. Our main result shows exactly this:

Theorem 3.1 (Plaut et al., 2025b)**.** *Given local generalization and standard online learning assumptions, there exists an algorithm that guarantees sublinear regret and sublinear mentor queries* *for any MDP, without resets.*

To our knowledge, our result is the **first formal proof that it is possible for an agent to obtain high reward while becoming self-sufficient in an unknown, unbounded, and high-stakes environment without resets.** While our algorithm is not intended for practical deployment, it provides theoretical grounding for the paradigm of "mostly follow a standard algorithm but ask for help when OOD". This is a broad conceptual takeaway that I believe can guide practical approaches to safety.

Limitations. Nevertheless, Theorem 3.1 has several limitations. First, it crucially relies on local generalization: this is how an agent detects unfamiliar situations. While many methods for OOD detection exist (see Yang et al., 2024 for a survey), our algorithm requires perfectly computing similarity distances between states, which is unrealistic. Future work could incorporate a more realistic (and noisy) model of OOD detection. Theorem 3.1 also requires standard online learning assumptions like finite VC or Littlestone dimension; while these assumptions are standard, they should not be taken for granted. Finally, and perhaps most importantly: this whole framework falls apart if no mentor is available. Can we still learn safely without a mentor?

3.5 Theoretical guarantees on learning without a mentor (B2)

If we want to remove the assumption of a mentor, we know from the Heaven or Hell problem that we will need a different assumption in its place. Framed differently: if you can't ask for help, how should you handle unfamiliar or risky situations? One natural approach is to decline to act, or *abstain*. Abstention could consist of simply doing nothing, but more generally means following a fallback policy that is known to be safe but may not generate high reward.

Our approach (Liaw and Plaut, 2026). We model an agent that has previously undergone training and is now being deployed. During training, the agent learned a policy that performs well on states from the training distribution. However, this task policy may or may not perform well on out-of-distribution (OOD) states. On each time step during deployment, the agent must choose to either *commit* (follow the task policy) or *abstain* (follow the safe fallback policy). Abstaining always results in 0 reward. The reward from committing can be positive (if the state is in-distribution or the policy generalizes well) but can also be very negative (if the state is OOD and the task policy generalizes poorly). Crucially, if the agent abstains, it does not observe the commit reward.

Similar to Section 3.4, we model OOD-ness via a similarity metric on the state space. Slightly differently, here we designate a specific point (without loss of generality, the origin) as fully in-distribution. Then the OOD-ness of a state is simply its distance to the origin. We allow an unbounded domain, so the agent can potentially face arbitrarily OOD states. We assume that the agent can detect when a state is OOD (i.e., it can compute the distance to the origin). However, the agent does not know how well the task policy generalizes — indeed, this is precisely what the agent needs to learn. Since the task policy performs well in-distribution, we assume that the reward from committing at the origin must be positive. Rewards from committing in OOD states can be arbitrarily negative, but we assume that they do not plummet suddenly — specifically, we assume Lipschitz continuity.

Our results. Any successful algorithm must explore cautiously, since committing recklessly runs the risk of arbitrarily negative rewards. On the other hand, if we only abstain, we'll never learn anything. Based on these principles, we propose an algorithm which computes a "safe-to-explore" region centered at the origin. The algorithm gradually narrows down where it should commit while ensuring that any errors it makes are not too costly. This is visualized by Figure 6.

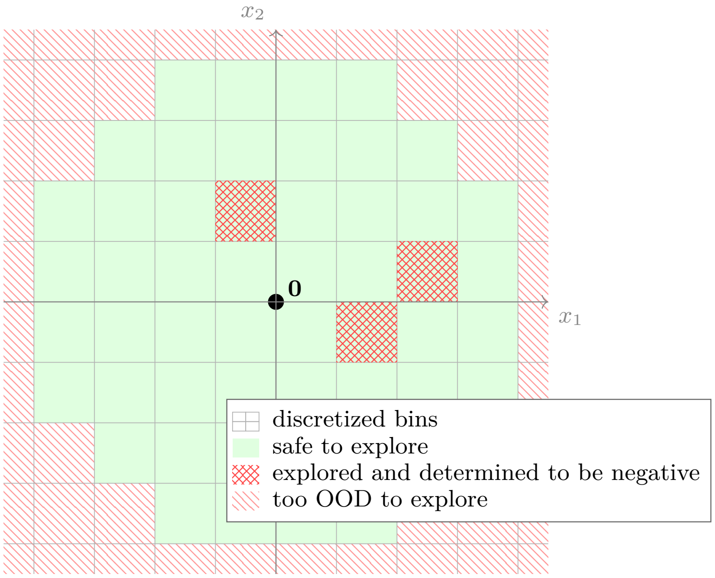


Figure 6: A visualization of the algorithm in Liaw and Plaut (2026). The state space is partitioned into bins. Bins that are close to the origin (i.e., not too OOD) are safe to explore, so the agent commits unless it has sufficient evidence to conclude that the bin is negative. Bins that are far from the origin (i.e., very OOD) are unsafe to explore, so the agent will always abstain, even though some of those bins might actually be positive. The safe exploration radius grows as a function of the time horizon.

Like Section 3.4, our objective is sublinear regret. Here regret is defined with respect to the optimal policy instead of the mentor's policy, since there is no mentor. We prove the following theorem:

Theorem 3.2 (Liaw and Plaut, 2026)**.** *If rewards are Lipschitz-continuous and states are i.i.d., there exists an algorithm that ensures sublinear regret without access to a mentor.*

Limitations. While this result addresses our prior limitation of needing a mentor, it has limitations of its own. For one, in order to streamline the model, we do not allow the agent to update the task policy. However, realistically, one can likely improve the generalizability of the task policy as one learns more about the deployment environment. Second, the assumption of i.i.d. states may not hold in real-world settings. Finally, Lipschitz continuity plays a crucial role in Theorem 3.2. Similar arguments apply to this Lipschitz assumption as to local generalization in Plaut et al. (2025b,c), and overall I do believe these assumptions are justified in many contexts. However, they also likely do not apply in all contexts, and understanding when these assumptions do and don't hold is an important topic for future work.

4 Future work

In this section, I outline some concrete research projects that I would like to see completed. While some directions for future work were discussed in the previous section under the "limitations" paragraphs, those were mostly direct extensions of existing projects. I want to emphasize that I would be thrilled for other researchers to take on some of these projects, as I certainly do not have bandwidth to work on all of them.

4.1 Goal misgeneralization in LLMs (B3)

To my knowledge, goal misgeneralization has only been demonstrated in video games (Langosco et al., 2022; Shah et al., 2022). While these findings are powerful, they are powerful primarily because of what they suggest about future systems, not because we are concerned about the harms of video games. In contrast, goal misgeneralization in LLMs — if present — has the potential for substantial direct harm. Although the related phenomenon of reward hacking has received significant attention in LLMs (Taylor et al., 2025; Pan et al., 2024; Khalaf et al., 2025; Bondarenko et al., 2025), I am not aware of any systematic study of goal misgeneralization in LLMs. A key challenge for such a study is how to define the training distribution, since the pretraining data of modern LLMs is massive and also typically not publicly available. Distribution shift with respect to post-training data may be more tractable to study, but poses other issues. For example, situations that are OOD with respect to post-training data may not be OOD with respect to pretraining data.

4.2 Mitigating goal misgeneralization by learning from demonstrations (B1)

My work so far on goal misgeneralization (Trinh et al., 2024; Czempin et al., 2026) assumes that each deployment episode exists in isolation: we do not allow the agent to learn between episodes. However, in practice, it makes sense for the agent to learn from the demonstrations it receives from the mentor. This idea of learning from expert demonstrations has been studied in various settings (see Argall et al., 2009; Arora and Doshi, 2021 for surveys), but to my knowledge, has not been studied in the context of goal misgeneralization. Such a project could begin by applying existing methods for learning from demonstrations to CoinRun and evaluating their effectiveness.

4.3 Regret in terms of distribution shift under irreversible dynamics (A3)

Of the six research topics I outlined in Section 2 (A1–A3, B1–B3), the only one I have not yet worked on directly is A3: theoretical models of distribution shift in safety-critical contexts. There exists significant theoretical work on distribution shift (see Quiñonero-Candela et al., 2022 for a survey). However, to my knowledge, none of this work allows for truly irreparable errors (see also the discussion in Section 3.4). My theoretical work so far (Sections 3.4 and 3.5) does allow for irreparable errors, and does fundamentally deal with distribution shift. However, those regret bounds arguably consider a "worst-case" distribution shift. If the distribution shift is smaller (although likely still nonzero), the regret may be smaller. Characterizing precisely how regret varies as a function of the nature of the distribution shift could be very interesting.

4.4 Learning from close calls (B1 and/or B2)

As discussed in Sections 3.4 and 3.5, a key question is how to learn which actions cause irreparable errors without having to try those actions directly. Those two sections explored two different approaches to that question. A third way to learn how to avoid catastrophe is through "close calls": actions that did not cause irreparable errors but were clearly dangerous in hindsight. For example, coming within inches of a vehicle collision causes no direct damage, but clearly indicates that something dangerous occurred. The idea of a "close call" suggests a distance metric similar to those used in Sections 3.4 and 3.5. However, any sort of Lipschitz-flavored smoothness will struggle to capture the idea that two situations are similar but one is irreparable and the other is not. How can we mathematically model and/or empirically simulate close calls?

4.5 Understanding similarity metrics and OOD detection (A3)

The assumption that the agent can transfer knowledge between similar states (formalized by local generalization in Section 3.4 and Lipschitz continuity in Section 3.5) is crucial to our theoretical results. Crucially, the underlying metric space can be any encoding of the agent's situation, not just its physical positioning. For example, a 3 mm spot and a 3.1 mm spot on X-rays likely have similar risk levels for cancer (assuming similar density, location, etc.). If the risk level abruptly increases for any spot over 3 mm, then local generalization may not hold for a naive encoding which treats size as a single dimension. However, a more nuanced encoding would recognize that these two situations – a 3 mm vs 3.1 mm spot – are in fact *not* similar.

Constructing a suitable encoding may be challenging, but we do not require the agent to have explicit access to such an encoding. The agent only needs to be able to detect when a state is unfamiliar, i.e., dissimilar from prior observations. Knowing the encoding is one way to compute this familiarity, but there are also indirect ways to estimate familiarity. In fact, there are multiple (overlapping) entire fields of research on this topic, including OOD detection, novelty detection, and anomaly detection. See Yang et al. (2024) for a survey.

The key open question is: are standard OOD/novelty/anomaly detection methods essentially computing similarities in a suitable metric space? Part of what makes this question particularly exciting to me is that it does not just apply to my work: this encoding issue arguably applies to all usages of Lipschitz continuity in sequential decision-making.

5 Conclusion

In this document, I argued that many safety failures can be framed as misgeneralization and proposed a research agenda based on this lens. I also overviewed the progress my collaborators and I have made so far on this agenda and discussed ideas for future work. I would be happy to receive feedback on any part of this document. Overall, I think that there is an urgent need for progress on AI safety and I hope that my work contributes to that progress in a small way.

Acknowledgements

My work has been supported in part by a gift from Open Philanthropy to the Center for Human-Compatible AI (CHAI). I would like to thank my coauthors, in alphabetical order: Erdem Bıyık, Hanlin Zhu, Juan Liévano-Karim, Mohamad Danesh, Nguyen X. Khanh, Pavel Czempin, Ponnurangam Kumaraguru, Sarah Liaw, Stuart Russell, Tu Trinh, and Vamshi Krishna Bonagiri. I would also like to thank the following colleagues for helpful feedback and discussion, also in alphabetical order: Aly Lidayan, Bhaskar Mishra, Cameron Allen, Cassidy Laidlaw, Daniel Jarne Ornia, Karim Abdel Sadek, Katie Kang, Matteo Russo, Michael Cohen, Nika Haghtalab, Ondrej Bajgar, Peter Hase, Scott Emmons, Tianyi Alex Qiu, and Yaodong Yu.

References

Stanislav Abaimov and Maurizio Martellini. *Artificial Intelligence in Autonomous Weapon Systems*, pages 141–177. Springer International Publishing, Cham, 2020. ISBN 978-3-030-28285-1. doi: 10.1007/978-3-030-28285-1_8. URL `https://doi.org/10.1007/978-3-030-28285-1_8`.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, et al. Managing extreme AI risks amid rapid progress. *Science*, 384(6698):842–845, 2024.

Leonard Bereska and Stratis Gavves. Mechanistic Interpretability for AI Safety - A Review. *Transactions on Machine Learning Research*, April 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ePUVetPKu6.

Vamshi Krishna Bonagiri, Ponnurangam Kumaraguru, Khanh Nguyen, and Benjamin Plaut. Check yourself before you wreck yourself: Selectively quitting improves llm agent safety. *NeurIPS 2025 Workshop on Reliable ML*, 2025.

Alexander Bondarenko, Denis Volk, Dmitrii Volkov, and Jeffrey Ladish. Demonstrating specification gaming in reasoning models. *arXiv preprint arXiv:2502.13295*, 2025.

Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. OUP Oxford, July 2014. ISBN 978-0-19-166683-4.

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022.

Brian Christian. *The Alignment Problem: Machine Learning and Human Values*. W. W. Norton & Company, October 2020. ISBN 978-0-393-63583-6.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning, 2020. URL https://arxiv.org/abs/1912.01588.

Andrew Critch and Stuart Russell. TASRA: a taxonomy and analysis of societal-scale risks from AI. *arXiv preprint arXiv:2306.06924*, 2023.

Pavel Czempin, Tu Trinh, Mohamad H Danesh, Nguyen X Khanh, Erdem Bıyık, and Benjamin Plaut. Getting by goal misgeneralization with a little help from an expert. 2026. Under submission; available upon request.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Javier García and Fernando Fernández. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. ISSN 1533-7928. URL http://jmlr.org/papers/v16/garcia15a.html.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018. doi: 10.1109/DSAA.2018.00018.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, et al. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. 46(12):11216–11235, 2024. ISSN 1939-3539. doi: 10.1109/TPAMI.2024.3457538. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Blessing Guembe, Ambrose Azeta, Sanjay Misra, Victor Chukwudi Osamor, Luis Fernandez-Sanz, and Vera Pospelova. The Emerging Threat of AI-driven Cyber Attacks: A Review. *Applied Artificial Intelligence*, 36(1), December 2022. ISSN 0883-9514, 1087-6545. doi: 10.1080/08839514.2022.2037254. URL https://www.tandfonline.com/doi/full/10.1080/08839514.2022.2037254.

Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, et al. Multi-agent risks from advanced ai. *arXiv preprint arXiv:2502.14143*, 2025.

Guande He, Peng Cui, Jianfei Chen, Wenbo Hu, and Jun Zhu. Investigating uncertainty calibration of aligned language models under the multiple-choice setting. *arXiv preprint arXiv:2310.11732*, 2023a.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *arXiv preprint arXiv:2302.13425*, 2023b.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. An overview of catastrophic AI risks. *arXiv preprint arXiv:2306.12001*, 2023.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

S Kadavath, T Conerly, A Askell, T Henighan, D Drain, E Perez, N Schiefer, ZH Dodds, N DasSarma, E Tran-Johnson, et al. Language models (mostly) know what they know. *URL https://arxiv.org/abs/2207.05221*, 5, 2022.

Zachary Kenton, Noah Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah Goodman, et al. On scalable oversight with weak llms judging strong llms. *Advances in Neural Information Processing Systems*, 37: 75229–75276, 2024.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio du Pin Calmon. Inference-time reward hacking in large language models. *arXiv preprint arXiv:2506.19248*, 2025.

Puneet Kohli and Anjali Chadha. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 Uber Inc. self-driving car crash. In *Advances in Information and*

Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 1, pages 261–279. Springer, 2020.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, et al. Chain of thought monitorability: A new and fragile opportunity for ai safety. *arXiv preprint arXiv:2507.11473*, 2025.

Nima Kordzadeh and Maryam Ghasemaghaei. Algorithmic bias: review, synthesis, and future research directions. *European Journal of Information Systems*, 31(3):388–409, May 2022. ISSN 0960-085X. doi: 10.1080/0960085X.2021.1927212. URL https://doi.org/10.1080/0960085X.2021.1927212. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/0960085X.2021.1927212.

Hanna Krasowski, Jakob Thumm, Marlon Müller, Lukas Schäfer, Xiao Wang, and Matthias Althoff. Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking. 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=mcN0ezbnzO.

Kimmo Kärkkäinen and Jungseock Joo. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. pages 1547–1557. IEEE Computer Society, January 2021. ISBN 978-1-66540-477-8. doi: 10.1109/WACV48630.2021.00159. URL https://www.computer.org/csdl/proceedings-article/wacv/2021/047700b547/1uqGMh4LXe8.

Lauro Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning*, pages 12004–12019. PMLR, 2022.

Sarah Liaw and Benjamin Plaut. Learning when not to learn: Risk-sensitive abstention in bandits with unbounded rewards. In *Proceedings of the Twenty-Ninth Annual Conference on Artificial Intelligence and Statistics*, AISTATS, 2026.

Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained MDPs. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117, 2025.

C Mouton, Caleb Lucas, and Ella Guest. The operational risks of AI in large-scale biological attacks. Technical report, RAND Corporation, Santa Monica, 2024.

National Institute of Standards and Technology (US). Artificial intelligence risk management framework : generative artificial intelligence profile. Technical report, National Institute of Standards and Technology (U.S.), Gaithersburg, MD, July 2024. URL https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf.

National Transportation Safety Board. Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian, Tempe, Arizona, March 18, 2018. Technical report, 2019.

Richard Ngo, Lawrence Chan, and Sören Mindermann. The Alignment Problem from a Deep Learning Perspective. ICLR 2024, May 2025. doi: 10.48550/arXiv.2209.00626. URL http://arxiv.org/abs/2209.00626. arXiv:2209.00626 [cs].

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI. GPT-5 System Card. Technical report, 2025. URL https://cdn.openai.com/gpt-5-system-card.pdf.

Alexander Pan, Erik Jones, Meena Jagadeesan, and Jacob Steinhardt. Feedback loops with language models drive in-context reward hacking. *arXiv preprint arXiv:2402.06627*, 2024.

Benjamin Plaut, Nguyen X Khanh, and Tu Trinh. Probabilities of chat llms are miscalibrated but still predict correctness on multiple-choice q&a. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=E6LOh5vz5x.

Benjamin Plaut, Juan Liévano-Karim, Hanlin Zhu, and Stuart Russell. Safe learning under irreversible dynamics via asking for help, 2025b. URL https://arxiv.org/abs/2502.14043.

Benjamin Plaut, Hanlin Zhu, and Stuart Russell. Avoiding catastrophe in online learning by asking for help. In *Proceedings of the 42nd International Conference on International Conference on Machine Learning*, 2025c.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2022.

Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. AI in health and medicine. *Nature Medicine*, 28(1):31–38, 2022.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox. *arXiv preprint arXiv:2309.15817*, 2023.

Stuart Jonathan Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin, 2019. ISBN 978-0-525-55861-3.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint arXiv:2210.01790*, 2022.

Sakib Shahriar, Sonal Allana, Seyed Mehdi Hazratifard, and Rozita Dara. A survey of privacy risks and mitigation strategies in the artificial intelligence life cycle. *IEEE Access*, 11:61829–61854, 2023. doi: 10.1109/ACCESS.2023.3287195.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *arXiv preprint arXiv:2412.05563*, 2024.

Peter Slattery, Alexander K Saeri, Emily AC Grundy, Jess Graham, Michael Noetel, Risto Uuk, James Dao, Soroush Pour, Stephen Casper, and Neil Thompson. The ai risk repository: A comprehensive meta-review, database, and taxonomy of risks from artificial intelligence. *arXiv preprint arXiv:2408.12622*, 2024.

Francesco Emanuele Stradi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Learning adversarial MDPs with stochastic hard constraints. *arXiv preprint arXiv:2403.03672*, 2024.

Mia Taylor, James Chua, Jan Betley, Johannes Treutlein, and Owain Evans. School of reward hacks: Hacking harmless tasks generalizes to misaligned behavior in llms. *arXiv preprint arXiv:2508.17511*, 2025.

Tu Trinh, Mohamad H Danesh, Nguyen X Khanh, and Benjamin Plaut. Getting by goal misgeneralization with a little help from a mentor. *NeurIPS Workshop on Safe and Trustworthy Agents*, 2024.

Jean Marie Tshimula, Xavier Ndona, D'Jeff K. Nkashama, Pierre-Martin Tardif, Froduald Kabanza, Marc Frappier, and Shengrui Wang. Preventing Jailbreak Prompts as Malicious Tools for Cybercriminals: A Cyber Defense Perspective, November 2024. URL http://arxiv.org/abs/2411.16642. arXiv:2411.16642 [cs].

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. In *arXiv preprint arXiv:2308.11432*, 2023.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pages 1–28, 2024.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak Attacks and Defenses Against Large Language Models: A Survey, August 2024. URL http://arxiv.org/abs/2407.04295. arXiv:2407.04295 [cs].

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pang, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, Bo An, and Qingsong Wen. A survey on trustworthy llm agents: Threats and countermeasures, 2025. URL https://arxiv.org/abs/2503.09648.

John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS medicine*, 15(11):e1002683, November 2018. ISSN 1549-1676. doi: 10.1371/journal.pmed.1002683.

Kun Zhang, Le Wu, Kui Yu, Guangyi Lv, and Dacao Zhang. Evaluating and Improving Robustness in Large Language Models: A Survey and Future Directions, July 2025. URL http://arxiv.org/abs/2506.11111. arXiv:2506.11111 [cs].

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9778–9795, 2023.