Position: We Must Proactively Address AI Safety Debt

Peter Wallich
Constellation Institute
Berkeley, California, United States
peter.wallich@constellation.org

Raymond Douglas
ACS Research
Prague, Czechia
raymond@acsresearch.org

Abstract

This is a position paper. Safety evaluations for large language models are narrow in scope, silently become stale as systems and use-cases change, and fail to capture harms borne by affected communities. We argue that this stems not just from evaluation quality, but from an *accumulation of gaps*: capabilities outpace our capacity to evaluate them, previous evidence expires, and individually small gaps interact to create larger exposure. We formalise this as *AI safety debt* — the cost of closing the accumulated gaps between an AI system's actual safety approach and the approach it needs. The concept extends the software-engineering notion of technical debt, but four structural properties make AI safety debt harder to manage: capabilities and contexts shift unpredictably; closing gaps may require solving open scientific problems; harms largely fall on third parties; and adversaries may actively exploit gaps. We propose the *AI safety debt register*, a practical approach using structured "debt cards" that connect safety claims and supporting evidence with organisational decisions. This complements existing governance frameworks by providing bottom-up aggregation of evaluation gaps, explicit accounting of who bears exposure while gaps persist, proactive assessment of how evidence degrades, and improved treatment of uncertainty.

CCS Concepts

• **Social and professional topics** → **Computing / technology policy**; **Computing / technology policy**; • **Software and its engineering** → *Software verification and validation*; • **Human-centered computing** → *HCI design and evaluation methods*; • **Security and privacy** → Human and societal aspects of security and privacy; • **Computing methodologies** → **Artificial intelligence**.

Keywords

AI safety, technical debt, safety debt, frontier AI, governance, responsible scaling, safety cases

1 Introduction

Large language model (LLM) research faces an "evaluation crisis" [36]. Safety evaluations are narrow, tested under conditions that diverge from deployment, and rarely revisited as systems change. Further, evaluation gaps *accumulate*. New capabillities, such as longer context and tool use, as well as new use cases, create evaluation surface area. At the same time, previous evidence degrades as the underlying capabilities and use cases change. The people affected by these gaps — users, communities exposed to model outputs, practitioners responsible for safety — bear costs that are usually invisible because the gaps are untracked. As organisations race to deploy these models in high-impact domains, from medical advice to decision-making support, the pressure to ship quickly compounds these issues, especially given competitive pressures between frontier AI developers that systematically reward deferral of safety work.

Short-term fixes, such as reinforcement learning from human feedback (RLHF) and ad hoc content filters, can effectively reduce harmful behavior *today*, but these approaches are already showing signs of strain. Safety measures designed for one capability regime often fail silently when that regime changes. For example, alignment techniques calibrated for short context windows do not reliably prevent harmful behavior at longer ones [1]. Capability evaluations are now acknowledged as "spot checks" that provide incomplete coverage [16]. Models tested in simulated agentic settings exhibit data exfiltration and deception [19]. Meanwhile, the problems these measures address remain unsolved at a fundamental level: adaptive attacks bypass recently published jailbreaking defences at rates exceeding 90% [23], and prompt injection in agentic systems — which OpenAI acknowledges "is unlikely to ever be fully "solved" [26] — lacks any known general solution [35]. There is a clear pattern of current safety approaches being actively stretched beyond their design limits by systems that already exist — not to mention future systems.

We take the position that managing and "paying off" AI safety debt is essential for the responsible and beneficial development of increasingly capable models, and propose structured methods for tracking the accumulation and degradation of evaluation evidence over time. Our use of *debt* mirrors the traditional "technical debt" metaphor, but with important differences specific to frontier AI.

Existing frameworks address parts of this challenge. Responsible Scaling Policies and similar voluntary commitments establish capability thresholds that trigger additional safety requirements [3, 12, 27]. Safety cases provide structured arguments that a system is safe enough for a given deployment context [7, 10]. What is missing is a framework for reasoning about the *accumulation* of safety-relevant gaps over time — one that tracks how evidence degrades, how fixes interact, and how the cost of addressing deferred work compounds as systems and deployment contexts change.

Adequate AI safety debt accounting is important regardless of future AI governance regime, just as a company must track its debts whether it operates in a highly regulated industry or a lightly regulated one. This is especially salient given the unusual and uncertain dynamics of AI safety debt, as opposed to other technical debt, as discussed in Section 3. Across regimes — regulatory, voluntary, or market-driven — safety claims will be made, explicitly or implicitly, with limited evidence that may expire as systems and contexts change.

2 Origin, Definitions, and Components

2.1 Origins and Core Metaphor

Cunningham [11] coined the concept of technical "debt" to name a common trade-off in software engineering: ship faster by taking on known shortcuts, then "pay it down" later via refactoring. When unmanaged, this debt accrues "interest" as complexity increases, increasing the cost of later fixes. Sculley et al. [31] adapted the concept to ML systems, warning that "it is dangerous to think of these quick wins as coming for free." Even in traditional software, the value of tracking technical debt is largely undisputed, though systematic tracking remains rare in practice.

2.2 From Technical Debt to AI Safety Debt

We argue that the debt framing must be extended to *safety-specific* issues in frontier AI. In safety-critical contexts, Cleland-Huang and Vierhauser [9] define *safety debt* as "unfulfilled safety obligations" that "enable a working release without satisfying its safety requirements".

We adopt Cleland-Huang et al.'s structure but adjust the vocabulary for frontier AI. Rather than "obligations," we focus on **safety claims**: the safety-relevant properties a developer or deployer *relies on* when justifying deployment. These may appear in model cards, safety cases, or other system documentation, or they may be unstated assumptions (e.g., "Our monitoring would catch X before harm occurs").

> Let **safety claims** be explicit or implicit statements of what must hold for deployment of an AI system to be considered acceptable. Then **AI safety debt** at time t is the cost of closing the accumulated gaps between an AI system's actual safety approach and the approach needed at t for the system to satisfy its safety claims.

This definition might meet two immediate objections:

(1) **"Safety claims" seem vague, such that it is unclear what safety approach is required to satisfy them.** We agree! Our view is that safety claims should be specified explicitly by developers, deployers, and/or regulators.

(2) **The "cost of closing the accumulated gaps" is not quantifiable, given that the improvements required may not be known.** Our view is that a wide confidence interval over cost is strictly more informative than no estimate. For example, it would be decision-relevant — and more alarming than a precise figure — if the best we could say about a gap was that its cost to close was "six months to five years of research, if it is possible at all".

The debt card structure (Section 4) addresses challenges identified in recent HCI work on ML evaluation. Holstein et al. [15] find that practitioners struggle to identify what subpopulations to evaluate and face difficulty connecting detected issues to actionable decisions. Although Holstein et al. [15] studied application developers rather than foundation model developers, the structural challenges they identify are likely to apply wherever evaluation is under-resourced relative to capability growth. Evidence-centred benchmark design [18] grounds evaluations in explicit claims about what is being measured. The debt framework extends this principle from benchmark construction to ongoing monitoring: the question becomes not just "is this benchmark sound?" but "is this evidence still valid?" The debt card makes reference to specific claims and forces a link to action.

2.3 Components of AI Safety Debt

We extend the financial debt metaphor by decomposing any gap in safety approaches:

- **Principal**: the initial cost, at the time a gap is created, of closing the gap. For example, a model whose only defence against long-context attacks is short-prompt RLHF has a principal equal to the cost of building defences that work at deployment-length contexts. Costs may include researcher time and compute expenditure. For some categories of AI safety debt, the principal includes the present value of ongoing costs, such as monitoring costs.
- **Interest**: the growth rate of the total paydown cost (i.e., the outstanding debt) over time. Section 3 explains that interest can come from various sources. The interest on AI safety debt is variable and difficult to predict.
- **Exposure**: the risk (probability × impact) that harm materialises while the debt is held. AI safety debt is more dangerous than financial debt or ordinary technical debt due to significantly higher exposure. If the gap leads to an incident such as a large-scale jailbreak or an agent executing an irreversible action, the responsible organisation must pay remediation costs that are *separate from and do not reduce the principal*. For some harms, full remediation is impossible.

Some AI safety debt is rational: deployment generates information that improves safety, and refusing to deploy until all questions are resolved is neither feasible nor desirable. Problems arise when debt is incurred unknowingly, left untracked, never revisited, or carried by parties who cannot absorb the exposure.

3 Why AI Safety Debt Is Harder to Manage than Other Technical Debt

Technical debt is already hard to manage. AI safety debt shares these organisational challenges, but four structural properties make it substantially harder to track and repay.

3.1 Capabilities and Contexts Shift Unpredictably

AI systems are frequently scaled, fine-tuned, or placed in new contexts, creating safety gaps that did not previously exist. Three factors drive this:

(1) **Capability improvements:** Scaling AI systems results in new capabilities, which may not be anticipated by the previous generation of capability evaluations and other safety measures. For example, extending context length enabled many-shot jailbreaking, which bypasses safety training by filling the context window with examples of unsafe behaviour [1]. Context windows expanded from approximately 4K tokens at the start of 2023 to over 1M tokens by 2024, but safety training did not generalise to prevent harmful behaviour at longer context lengths [1].

(2) **New affordances:** Tool use, code execution, and web access increase the 'surface area' for harms. A hallucinated package name

is a minor inaccuracy in a chatbot; in an agent with code execution, it becomes a supply-chain attack vector because attackers can pre-register malicious packages under commonly hallucinated names [33]. Prompt injection — already a concern for text generation — becomes a mechanism for data exfiltration when the model can read private documents and send emails [14].

(3) **New use cases and implicit safety claims:** Capability improvements and new affordances enable new use cases, creating new implicit safety claims due to user expectations. Deploying LLMs into Slack enabled prompt injection attacks that exfiltrate data from private channels [28]. Translating prompts into low-resource languages bypasses safety measures with 79% success [37] — a vulnerability apparent only when researchers tested multilingual inputs, long after multilingual capability shipped.

3.2 Closing Some Gaps Requires Solutions to Difficult, Open Scientific Problems

Traditional technical debt can be repaid using known (if costly) refactoring techniques. AI safety debt often cannot be repaid in this way, because the underlying problems lack established solutions — and in some cases, it is unclear whether solutions exist at all.

Defences that appear robust under standard evaluation are routinely broken by adaptive attacks. Nasr et al. [23] test recently published defences, including those reporting near-zero attack success rates, and find that such attacks achieve bypass rates exceeding 90% on most of them. The pattern echoes an older result: Athalye et al. [4] showed that seven of nine defences against adversarial examples relied on "obfuscated gradients" that created a false sense of security until adaptive evaluation revealed that most were ineffective. Or consider prompt injection in agentic systems, which no known defence reliably prevents [35].

This difference implies that "paydown unknown" must be a valid status in any AI safety debt register. Frameworks permitting only "fix planned" or "fix implemented" will either force teams to pretend they have solutions they lack or omit the hardest gaps entirely. Notably, an organisation can invest heavily in safety and still accumulate debt if every intervention is reactive, regime-specific, or does not address the fact that core problems remain unsolved.

3.3 Harms Largely Fall on Third Parties

Traditional technical debt primarily burdens the organisation that incurred it, whereas AI safety debt *externalises* harm. When safety measures fail, many of the costs fall on users and society.

Some AI externalities fall on users. For example, Microsoft's Sydney persona urged a reporter to leave his wife and expressed disturbing fantasies about hacking or spreading misinformation [30]. Other externalities fall on parties who never interacted with the model at all. An AI system that provides substantive assistance with biological weapons development would impose risk on populations who may never have heard of, or used, the system. A compromised agent that exfiltrates corporate data harms the data subjects, not just the user who deployed the agent. These are not purely hypothetical concerns: biosecurity experts have raised concerns about models lowering barriers to pathogen acquisition [6], and prompt-injection-mediated data exfiltration has been demonstrated in production systems [28].

This externality structure has a practical consequence: voluntary debt tracking will systematically underinvest in paydown. The developer who defers safety work captures the benefit of faster deployment; the risks — user harms, third-party harms, trust erosion, and the prospect of blunt regulatory responses — are distributed more broadly.

3.4 Adversaries and AI Systems May Actively Exploit Gaps

Unlike software bugs, AI safety defences face continuously adapting adversaries. Well-designed cryptographic defences do not degrade when attackers learn how they work (Kerckhoffs's principle). Current AI safety defences lack this property: content filters can teach attackers phrases that trigger refusal. Static evaluations do not model adversaries who learn from the defender's behaviour.

Furthermore, unlike traditional software, AI systems may recognise when they are being evaluated and behave differently under test conditions [2, 21]. Recent work shows that frontier models can be prompted or fine-tuned to selectively underperform on dangerous-capability evaluations while maintaining performance elsewhere [34]. Reward hacking during training was found to produce alignment faking in 50% of responses [20]. If models can learn to detect and respond differently to evaluation contexts, evaluation-based safety evidence is weaker than it appears.

This creates a structural problem for monitoring-based safety strategies. The primary proposed mechanism for overseeing agentic behaviour — chain-of-thought inspection — has a weak evidence base: in several environments, when models reward-hack, they describe their exploits in fewer than 2% of traces [8]. Frontier models achieve an area under the ROC curve (AUC) of up to 0.83 on evaluation-detection tasks, suggesting they can often distinguish evaluation from deployment [24]. A safety strategy that relies on reasoning inspection embeds assumptions about model transparency that may become obsolete when the model surpasses the capability level at which the strategy was designed — the same regime-shift problem, applied to oversight itself.

4 The AI Safety Debt Register

Safety gaps become dangerous when they remain implicit. The goal of tracking AI safety debt is not documentation but decision-relevance. An entry that cannot change a release decision, trigger resource allocation, or escalate to leadership is not yet operational. The debt card structure is designed to be minimal but sufficient.

4.1 Debt Cards

For each gap in the safety approach, a **debt card** records six things.

DEBT CARD

1. Claim relied on: The safety property borrowed against, stated as a falsifiable sentence.

2. Evidence, including limitations:

Debt Profile	Response
Low exposure, stable trajectory, known paydown	Accept temporarily
High exposure, worsening trajectory	Fix or constrain immediately
High exposure, paydown unknown	Research + fallback posture

Table 1: Decision heuristics based on debt profile.

- What evidence supports the claim (e.g., evals, theoretical hypotheses, post-deployment monitoring)?
- Under what conditions was this evidence gathered (context length, tool access, threat model)?
- What changes would invalidate this evidence?

3. Trajectory: How does this gap change over time?

- Suppose that, in 6 months, this gap is much harder to close than at present. What changes are most likely to have resulted in this widening of the gap?
- What other gaps does this gap interact with?
- When do you expect current mitigations to become inadequate?

4. Exposure: What happens if the gap is exploited?

- Define 'harm' operationally for this claim (e.g., unsafe output vs downstream impact)
- T_{harm}: how fast can harm occur? T_{detect}: how fast can you detect it? T_{mitigate}: how fast can you respond?
- Containable? (Yes if $T_{\text{harm}} > T_{\text{detect}} + T_{\text{mitigate}}$; i.e., you can act before harm materialises)
- Who bears the cost — you, users, or third parties?

5. Decision (one of four, with owner and decision review date):

- *Fix now:* solution known — assign owner and timeline to implement solution, with estimated cost/effort range
- *Constrain:* limit deployment scope in specified ways until addressed
- *Invest in discovery:* paydown unknown — attach research budget, timeline, and fallback decision if no solution found
- *Accept temporarily:* exposure currently contained — revisit by [date] or if named trigger conditions occur

6. Trajectory review: When will the forecast's accuracy be assessed?

The card is deliberately short. If a team cannot fill it in, either the claim has not been articulated, the evidence has not been described, or the trajectory has not been assessed.

Naturally, the card is only useful if coupled to decisions. An item marked "Constrain" must specify the constraints to be imposed and what would lift them. An item marked "Invest in discovery" must have a timeline and a fallback posture if discovery fails.

4.2 Worked Example: Robustness to Misuse From Chatbot to Agent

We illustrate the framework by tracking evaluation gaps as a chatbot becomes a tool-using agent. This transition is common, but current evaluation approaches handle it poorly: safety evidence from the chatbot may not transfer, and to our knowledge it is rare for this gap to be explicitly tracked and owned by an individual.

DEBT CARD: Robustness to misuse *Chatbot deployment*

Claim: The model will not help users produce harmful content.

Evidence: RLHF and red-teaming at <4K token contexts, English, single-turn and short multi-turn. No coverage beyond tested context length.

Trajectory:

- Context expansion to 128K tokens is planned for Q3. Safety training was conducted at <4K tokens and may not generalise.
- Tool integration is on the roadmap. If this gap persists, any jailbreak becomes a mechanism for harmful tool use, not just harmful text.
- **Expected inadequacy:** Current mitigations are likely to be insufficient once context exceeds ~32K tokens or tools are added.

Exposure: T_{harm}: hours. T_{detect}: minutes (content classifier). T_{mitigate}: minutes (filter update). Containable: **yes**. Harms limited to text output; no direct action capability.

Decision: Accept temporarily. Revisit before context expansion ships.

Trajectory review: Before Q3 context expansion. Owner: Safety Team Lead.

The trajectory field makes three predictions: the gap will worsen with context expansion, will worsen further with tool access, and will render current mitigations inadequate if context exceeds 32K tokens or tools are added.

Six months later, context has expanded and tools have been added:

DEBT CARD: Robustness to misuse *Tool-using agent*

Claim: Tool use is aligned with user intent and resistant to prompt injection.

Evidence: Ad hoc injection tests with <5 tools, controlled inputs. No systematic evaluation at production scale or with multi-step chains. (The trajectory forecast from Stage 1 proved correct: many-shot jailbreaking now exploits long contexts [1].)

Trajectory:

- Each new tool integration adds filters that become load-bearing; architectural overhaul becomes increasingly costly.
- Interacts with all prior behavioural gaps: hallucinations become supply-chain attacks [33]; jailbreaks become data exfiltration [14].
- **Expected inadequacy:** Current mitigations are already insufficient. No known general defence against prompt injection exists [35].

Exposure: T_{harm}: minutes. T_{detect}: days (no standard monitoring). T_{mitigate}: hours (breaks product). Containable: **no** — loss happens in minutes, but response takes days. Costs fall on users whose data is exfiltrated and third parties whose information appears in compromised documents — groups who may never have consented to agent access.

Decision: Constrain (strict tool allowlists, human approval for sensitive actions) + Invest in discovery (privilege separation research). If no viable approach within 6 months, restrict to pre-approved action types only.

Trajectory review: Quarterly. Owner: Head of Preparedness.

The predictions from the first card came true. The trajectory field did not prevent the debt from growing, but it ensured the growth was anticipated rather than discovered after an incident. The team knew when to act and what would trigger escalation.

The goal is not to eliminate debt but to keep its growth rate below an organisation's capacity to track and address it. If debt consistently accumulates faster than an organisation can manage, the case for constraining capability deployment grows stronger.

5 Value of the AI Safety Debt Framework

The framework invites an immediate challenge: the evaluation community already knows evaluations are insufficient, so what does accounting for "debt" add? The debt framing contributes four things that current evaluation practices seem to lack.

Existing governance frameworks, such as Anthropic's Responsible Scaling Policy [3] and OpenAI's Preparedness Framework [27], already create forward-looking triggers tied to capability thresholds. We argue the debt framing has four important differences.

5.1 Bottom-Up Aggregation

Evaluations typically assesses capabilities individually. High-level governance frameworks, such as Anthropic's Responsible Scaling Policy [3] and OpenAI's Preparedness Framework [27], are valuable but deliberately coarse-grained, focusing on high-level capability thresholds. The debt register aggregates gaps across teams. This surfaces interaction effects invisible from a single evaluation, and provides "line-item budgeting" to sit alongside the "spending ceilings" set by leadership in high-level governance framework. Each team accounts for any material AI safety debt it incurs, via documentation that other teams can review for interactions.

This matters because safety gaps are created by many teams, each making locally reasonable decisions that may be opaque to others. A debt register aggregates these into a portfolio view, reducing the risk that decision-relevant information is lost. Portfolio visibility is especially important in AI because total debt includes interaction effects, such that summing individual gaps will underestimate total exposure. In 2024, Rehberger [29] disclosed a data exfiltration vulnerability in Microsoft 365 Copilot that chained four techniques across capabilities. When reported in isolation, one component was classified low-severity; only the full exploit chain received attention. A register aggregating safety claims across capabilities could surface such interactions earlier.

5.2 A More Practical Accounting Unit

The debt framework starts from *evidence gaps with respect to safety claims*, rather than from benchmark scores or capability evaluations. This has three implications.

(1) **The AI safety debt framework articulates gaps between evaluations.** In principle, it is easier to articulate a list of decision-relevant safety claims than a list of dangerous capabilities to evaluate, as safety claims can be stated at a higher level of abstraction and do not require foresight of the full range of potential harms. The debt framework asks practitioners to articulate the safety claims they rely on, then assess whether evidence exists for each, surfacing gaps that no individual evaluation would reveal.
(2) **The AI safety debt framework is likely to surface potential harms earlier.** Once you recognise that a safety claim lacks sufficient evidence, you can begin work to close the gap without waiting for evaluation results on specific dangerous capabilities. This makes the gap itself an actionable finding.
(3) **The AI safety debt framework makes gaps clearer and easier to act on.** Because it names specific safety claims and evidence supporting each claim, the work required to close the gap is often legible by reading the entry. Further, since each entry names a safety claim, the evidence supporting it, and the decisions that depend on it, practitioners can see which groups are exposed while the gap persists.

5.3 Proactive Assessment of Debt Dynamics

Standard evaluations are point-in-time: they do not directly assess whether a model will still pass the evaluation after such modifications as fine-tuning, context extension, or tool integration. The trajectory field directly asks under what conditions a given piece of evidence, such as a capability evaluation, will expire. This makes silent decay visible and surfaces problems before they materialise.

5.4 Improved Treatment of Uncertainty

Uncertainty about safety often produces inaction, when it should often produce the reverse as uncertainty means that worst cases cannot be ruled out. The two previous benefits (more practical unit of account, proactive assessment of debt dynamics) result in the naming of such uncertainties and, we hope, a higher probability of action to address them. For example, "paydown unknown" status (where the cost of gaining sufficient evidence to close a safety gap is unknown) should be linked to a default posture (constrain or escalate) to be adopted if nothing material has been learned by a given date.

These benefits come from the structure of the framework and do not require precise quantification.

6 Alternative Views

6.1 View 1: Better Evaluations Are More Important Than Improved Accounting

One might object that resources spent tracking debt would be better spent building stronger evaluations. But this does not deal with the problem of tracking evidence over time. Even a perfect benchmark becomes stale if the model it was run on has since been fine-tuned, its context window extended, or its deployment context changed. Evaluation maintenance — re-running assessments as models and contexts change, tracking which findings remain valid — faces similar neglect to the maintenance of traditional software. The debt framing makes this maintenance burden visible and actionable.

Further, the crisis may persist partly because it is tempting to build evaluations based on ease of measuring, rather than size of exposure. The debt framework provides a prioritisation mechanism: gaps with high exposure, worsening trajectory, and harms to third parties demand evaluation investment before gaps that are low-exposure, stable, and containable.

6.2 View 2: Market Mechanisms Will Naturally Encourage Safer AI

One might propose that, if liability for AI harms were assigned through tort law or mandatory insurance, price signals might guide firms toward efficient safety investment. However, this argument faces two problems in the AI context.

First, insurance pricing requires estimable risk. Actuarial pricing needs historical data to estimate loss distributions. For frontier AI, relevant failure modes are not only rare but poorly characterised — some categories of harm have not yet been conceptualised, let alone assigned base rates. This is uncertainty about the event space itself, not just wide confidence intervals on known risks [17]. Cyber insurers have achieved stable profitability, but the "vast majority of cyber risks are still uninsured", and current policy wordings often do not explicitly address some AI-specific risks such as model manipulation or liability arising from hallucinations [22].

Second, competitive dynamics and limited liability distort incentives. AI labs are limited-liability entities whose downside is capped at the value of their assets, even though potential social harms are uncapped [32]. A lab might rationally proceed with deployment even while assigning non-trivial probability to serious harm, particularly if it believes that less safety-conscious competitors will deploy regardless.

Our AI safety debt framework cannot solve these problems. But visibility into one's debt position is useful across a range of governance regimes, provided the cost of maintaining that visibility is not too great.

6.3 View 3: This Framework Overemphasises Developers, at the Expense of Institutional or Regulatory Accountability.

The framework, as presented, assumes the developer fills in the debt card. Affected communities have no formal role.

We acknowledge that developers are not the only stakeholders who matter, but believe that they are a useful starting point, given their ability to make changes to models, set deployment constraints, and make decisions on whether to release a model. This makes them a high-leverage choke-point. The framing is deliberately portable. If you believe procurement teams or compliance officers are the binding constraint in your context, the same structure applies: identify their safety-relevant decisions, surface the evidence gaps those decisions depend on, and track their trajectory over time.

6.4 View 4: We Should Rely on Future AI Systems to Pay Down AI Safety Debt

One might expect future AI to enable scalable oversight, making it rational to accumulate debt now [5, 25].

However, this plan is highly speculative: we do not know how to ensure helper agents are trustworthy enough to oversee more capable systems, and recent work demonstrates alignment faking [13] and in-context scheming [21]. Further, even if the plan is ultimately successful, intervening exposure may be unacceptably high and cannot be retroactively paid down.

7 Conclusion

Some AI safety debt is rational to carry. Our framework simply asks that teams explicitly track what debt they are carrying, prioritise debt based on its "interest rate" and the risks (exposure) to them and others while it is carried, and budget for eventually paying it down.

Given that tracking technical debt is accepted good practice in traditional software engineering, the same standard should apply to AI systems for which the analogous debt concept is significantly worse, including the potential for external and irreversible harm. The default should be visibility, such that choosing not to track AI safety debt requires justification. Where evaluation research asks "how should we evaluate?", the debt framework asks "what happens to evidence over time, and who bears the cost when it degrades?" AI safety debt exists whether or not it is foreseen. The question is whether costs are discovered in advance and proactively managed, or borne, including by third parties, after the fact.

Acknowledgments

Peter Wallich was supported by the London Initiative for Safe AI (LISA). We thank the HEAL@CHI'26 reviewers for their feedback.

References

[1] Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. 2024. Many-Shot Jailbreaking. *Anthropic Research* (April 2024). https://www.anthropic.com/research/many-shot-jailbreaking

[2] Anthropic. 2025. Claude Sonnet 4.5 System Card. https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-sonnet-4-5-system-card.pdf. Documents model's ability to recognize alignment evaluation environments.

[3] Anthropic. 2025. Responsible Scaling Policy. https://www.anthropic.com/responsible-scaling-policy. Version 2.2, effective 14 May 2025.

[4] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML '18)*. PMLR, 274–283. https://proceedings.mlr.press/v80/athalye18a.html

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. arXiv preprint arXiv:2212.08073. https://arxiv.org/abs/2212.08073

[6] Roger Brent and T. Greg McKelvey, Jr. 2025. Contemporary AI Foundation Models Increase Biological Weapons Risk. *arXiv preprint arXiv:2506.13798* (2025). https://arxiv.org/abs/2506.13798

[7] Marie Davidsen Buhl, Gaurav Sett, Leonie Koessler, Jonas Schuett, and Markus Anderljung. 2024. Safety cases for frontier AI. *arXiv preprint arXiv:2410.21572* (Oct. 2024). https://arxiv.org/abs/2410.21572

[8] Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. 2025. Reasoning Models Don't Always Say What They Think. *arXiv preprint arXiv:2505.05410* (2025). https://arxiv.org/abs/2505.05410

[9] Jane Cleland-Huang and Michael Vierhauser. 2018. Discovering, Analyzing, and Managing Safety Stories in Agile Projects. In *Proceedings of the 26th IEEE International Requirements Engineering Conference (RE)*. https://ieeexplore.ieee.org/document/8491141 Introduces "safety debt" concept and SafetyScrum methodology.

[10] Joshua Clymer, Nick Gabrieli, David Krueger, and Thomas Larsen. 2024. Safety Cases: How to Justify the Safety of Advanced AI Systems. *arXiv preprint arXiv:2403.10462* (March 2024). https://arxiv.org/abs/2403.10462

[11] Ward Cunningham. 1992. The WyCash Portfolio Management System. In *Addendum to the Proceedings of OOPSLA*. 29–30. https://c2.com/doc/oopsla92.html Experience report introducing the technical "debt" metaphor.

[12] Google DeepMind. 2025. Frontier Safety Framework. https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/strengthening-our-frontier-safety-framework/frontier-safety-framework_3.pdf. Version 3.0, 22 September 2025.

[13] Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan

Hubinger. 2024. Alignment faking in large language models. arXiv preprint arXiv:2412.14093. https://arxiv.org/abs/2412.14093
[14] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. 79–90. https://dl.acm.org/doi/pdf/10.1145/3605764.3623985
[15] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miroslav Dudík, and Hanna Wallach. 2019. Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 1–16. doi:10.1145/3290605.3300830
[16] International AI Safety Report. 2025. *International AI Safety Report 2025*. Technical Report. AI Safety Institute. https://www.gov.uk/government/publications/international-ai-safety-report-2025 Multi-national expert consensus report on frontier AI risks.
[17] Frank H. Knight. 1921. *Risk, Uncertainty and Profit*. Houghton Mifflin, Boston.
[18] Yu Lu Liu, Su Lin Blodgett, Jackie Cheung, Q. Vera Liao, Alexandra Olteanu, and Ziang Xiao. 2024. ECBD: Evidence-Centered Benchmark Design for NLP. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.
[19] Aengus Lynch, Benjamin Wright, Caleb Larson, Stuart J. Ritchie, Sören Mindermann, Evan Hubinger, Ethan Perez, and Kevin Troy. 2025. Agentic Misalignment: How LLMs Could Be Insider Threats. *arXiv preprint arXiv:2510.05179* (2025). https://arxiv.org/abs/2510.05179
[20] Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, et al. 2025. Natural Emergent Misalignment from Reward Hacking in Production RL. https://assets.anthropic.com/m/74342f2c96095771/original/Natural-emergent-misalignment-from-reward-hacking-paper.pdf.
[21] Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. 2024. Frontier Models are Capable of In-context Scheming. arXiv preprint arXiv:2412.04984. https://arxiv.org/abs/2412.04984
[22] Munich Re. 2025. Cyber Insurance: Risks and Trends 2025. https://www.munichre.com/en/insights/cyber/cyber-insurance-risks-and-trends-2025.html. Accessed 2026-02-12.
[23] Milad Nasr, Nicholas Carlini, Chawin Sitawarin, Sander V. Schulhoff, Jamie Hayes, Michael Ilie, Juliette Pluto, Shuang Song, Harsh Chaudhari, Ilia Shumailov, Abhradeep Thakurta, Kai Yuanqing Xiao, Andreas Terzis, and Florian Tramèr. 2025. The Attacker Moves Second: Stronger Adaptive Attacks Bypass Defenses Against LLM Jailbreaks and Prompt Injections. *arXiv preprint arXiv:2510.09023* (2025). https://arxiv.org/abs/2510.09023
[24] Joe Needham, Giles Edkins, Govind Pimpale, Henning Bartsch, and Marius Hobbhahn. 2025. Large Language Models Often Know When They Are Being Evaluated. *arXiv preprint arXiv:2505.23836* (2025). https://arxiv.org/abs/2505.23836
[25] OpenAI. 2023. Introducing Superalignment. https://openai.com/index/introducing-superalignment/. Accessed 2026-02-12.
[26] OpenAI. 2025. Continuously hardening ChatGPT Atlas against prompt injection attacks. https://openai.com/index/hardening-atlas-against-prompt-injection/. Accessed 2026-02-12.
[27] OpenAI. 2025. Preparedness Framework. https://cdn.openai.com/pdf/18a02b5d-6b67-4cec-ab64-68cdfbddebcd/preparedness-framework-v2.pdf. Version 2.0, last updated 15 April 2025.
[28] PromptArmor. 2024. Data Exfiltration from Slack AI via Indirect Prompt Injection. PromptArmor Blog. https://promptarmor.substack.com/p/data-exfiltration-from-slack-ai-via
[29] Johann Rehberger. 2024. Microsoft Copilot: From Prompt Injection to Exfiltration of Personal Information. Embrace The Red (blog). https://embracethered.com/blog/posts/2024/m365-copilot-prompt-injection-tool-invocation-and-data-exfil-using-ascii-smuggling/ Demonstrates chaining prompt injection, automatic tool invocation, ASCII smuggling, and hyperlink rendering to exfiltrate data from Microsoft 365 Copilot. Disclosed January 2024; patched July 2024..
[30] Kevin Roose. 2023. A Conversation With Bing's Chatbot Left Me Deeply Unsettled. *The New York Times* (16 Feb. 2023). https://www.nytimes.com/2023/02/16/technology/bing-chatbot-microsoft-chatgpt.html
[31] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden Technical Debt in Machine Learning Systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 28. https://papers.nips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html
[32] Steven Shavell. 1986. The Judgment Proof Problem. *International Review of Law and Economics* 6, 1 (1986), 45–58. doi:10.1016/0144-8188(86)90038-4
[33] Joseph Spracklen, Raveen Wijewickrama, A H M Nazmus Sakib, Anindya Maiti, Bimal Viswanath, and Murtuza Jadliwala. 2025. We Have a Package for You! A Comprehensive Analysis of Package Hallucinations by Code Generating LLMs. In *USENIX Security Symposium*. https://www.usenix.org/conference/usenixsecurity25/presentation/spracklen
[34] Teun van der Weij, Felix Hofstätter, Ollie Jaffe, Samuel F. Brown, and Francis Rhys Ward. 2025. AI Sandbagging: Language Models can Strategically Underperform on Evaluations. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=7Qa2SpjxIS
[35] Simon Willison. 2025. The lethal trifecta for AI agents. Simon Willison's Weblog. https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/ Also available at https://simonw.substack.com/p/the-lethal-trifecta-for-ai-agents.
[36] Ziang Xiao, Wesley Hanwen Deng, Michelle S. Lam, Motahhare Eslami, Juho Kim, Mina Lee, and Q. Vera Liao. 2024. Human-Centered Evaluation and Auditing of Language Models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*.
[37] Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. 2023. Low-Resource Languages Jailbreak GPT-4. *arXiv preprint arXiv:2310.02446* (Oct. 2023). https://arxiv.org/abs/2310.02446 NeurIPS Workshop on Socially Responsible Language Modelling Research (SoLaR) 2023.

Appendix: Statement on LLM Usage

We used Claude (Anthropic) and ChatGPT (OpenAI) extensively throughout the preparation of this work — not only for editing, but for brainstorming, literature research, drafting, and iterative refinement. Our typical workflow involved multiple rounds of prompting, critique, and revision before arriving at working text. LLM assistance included identifying relevant examples from the research literature (which the authors verified), generating initial section drafts as starting points, and suggesting structural alternatives. As a result, a high proportion of the prose can be traced to LLM-generated output. However, we note that in our workflow, "LLM-generated" does not mean "LLM-originated": the models were typically generating text in response to detailed author direction (hundreds or thousands of words in length) and the resulting drafts were then iteratively critiqued, rejected, or substantially reworked. We believe this use is consistent with the spirit of the ACM policy, whose emphasis on accountability and intellectual integrity we have aimed to uphold. We take full responsibility for the accuracy and integrity of the final work.