

How AI Development Techniques Present Useful Analogies for Improving ERM Processes

With all the hype surrounding the development of AI tools and agents, much attention has been focused on what is being done to ensure their reliability and accuracy. There are analogies of how some of the emerging best practices in AI development might translate into best practices for developing reliable ERM programs. This article summarizes some concepts from a recent AI conference held at NC State that might be useful for ERM program leaders.

OVERVIEW OF AI DEVELOPMENT TECHNIQUES

In Artificial Intelligence research, training autonomous agents to navigate complex environments relies heavily on **Reinforcement Learning (RL)**, which is a type of machine learning where an autonomous agent learns to make decisions through trial and error as they interact with an environment. During testing, the AI agent learns from “punishments” for its errors and “rewards” for its successes. As an example, we are seeing that unfold as we observe the testing of the accuracy of self-driving cars, such as Waymo, in cities before they begin accepting passengers for rides. Test cars can be seen driving the same routes over-and-over fine-tuning their accuracy in safely navigating a route.

A persistent challenge in RL is the **Sparse Reward Problem**, which is when an agent rarely receives feedback from its errors or successes or only receives feedback after the entire task is completed, making it difficult for the agent to really learn the task. There is insufficient information about potential mistakes or signals of success, which makes it hard for the autonomous agent to identify its specific actions leading to success. This makes it difficult for the AI agent to improve its ability to be repeatedly accurate. When catastrophic failures or ultimate victories occur rarely, the model receives almost no feedback to guide its step-by-step decisions. In our self-driving car example, the car fortunately is rarely if ever in an accident or may never realize it almost hit a pedestrian. Gratefully, those events are rare, but that does make it challenging to “train” the AI technology to avoid similar unfortunate realities.

To compensate, AI engineers introduce **Continuous Dense Proxy Rewards**, which are substitutes for real data. The “continuous dense” aspect represents lots of feedback given continually to provide information at many steps along the way (i.e., points or accuracy scores given to the AI agent on an ongoing basis as it completes a number of specific tasks correctly). This helps the agent learn about successes and missteps at each step towards the ultimate goal. Back to our self-driving car example, continuous dense proxy rewards would give feedback along a route about car’s speed, whether it kept sufficient spacing between vehicles, stopped correctly at stop lights, slowed down as other cars or people approached the vehicle, all ultimately impacting the successful delivery of passengers to the correct location.

However, this frequently triggers what is known as **Reward Hacking**, where the agent takes shortcuts or leverages loopholes to maximize rewards (e.g., points) it receives for accuracy on the tiny individual steps with little focus on achieving the final outcome successfully. For testing self-driving cars, the AI agent directing the car might take actions that give the agent lots of rewards for its accuracy on specific detailed tasks but by the end of the ride the self-driving car actually ends up at the wrong location. In that example, the self-driving car focused on doing tasks to earn specific scores on individual tasks without caring about whether the car arrived at the correct location.



DON PAGACH

Professor and Director of Research
Poole College of Management
NC State University

NC STATE Poole College of Management
Enterprise Risk Management Initiative

HOW AI DEVELOPMENT TECHNIQUES PRESENT USEFUL ANALOGIES FOR IMPROVING ERM PROCESSES

RECOGNIZING ANALOGIES IN AN ERM CONTEXT

Interestingly, Enterprise Risk Management (ERM) programs can suffer from an identical structural vulnerability. Let's break that down.

When considering uncertainties that might trigger ruinous enterprise events—such as systemic liquidity crises, fatal safety failures, or catastrophic cyber breaches—executives are having to assess potential risk events that are inherently sparse. That is, ERM programs also face a “**sparse reward problem**.” Because the potential event has never occurred, executives don't have sufficient information to help them know what prior actions and decisions have been helpful to them in avoiding the risk in the past.

To address the lack of data, ERM programs implement measures, such as monthly risk updates or real-time key risk indicators or metrics, to keep an eye on conditions that might suggest a risk is emerging. These measures are similar to “**continuous dense proxies**” discussed previously that are intended to be substitutes of information for the real event. A challenge with key risk indicators is that they are proxies, not measures of the specific risk event that might concern management.

If organizations aren't careful, compensation and performance incentives may incentivize management to focus on ensuring certain risk metrics (e.g., KRIs) stay within acceptable thresholds while losing sight of the larger, looming catastrophic risk event. That is, they engage in “**reward hacking**” by managing to the metric while overlooking the real risk event that might be disastrous. This is often referred to as “Goodhart's Law” which states that “when a measure becomes a target, it ceases to be a good measure.”

Let's consider three examples of these concepts occurring.

- **Credit Rating Agency & Bank VaR Model Failures:** Prior to 2008, quantitative risk models used Value at Risk (VaR) based on recent historical data—a classic dense proxy. Because housing defaults were historically sparse, VaR models signaled near-zero risk for complex Mortgage-Backed Securities (MBS). Financial institutions acted like RL agents exploiting a faulty reward function: they stripped away liquidity reserves to maximize quarterly returns (reward hacking). When the sparse tail event materialized, it wiped out systemic capital, revealing that the “pristine” proxy metrics had incentivized massive, unhedged risk concentrations.
- **Wells Fargo: Cross-Selling Proxy Hacking:** The bank's executive leadership instituted a number of continuous dense proxy metrics to incentivize employees in their retail sector to grow customer accounts. For example, the CEO was known for saying “Eight is Great” as a way to pressure retail bankers to open at least “8 accounts per customer.” Results were measured daily by branch and regional managers, tracking “continuous dense proxies” (i.e, customer account performance metrics) by individual banker. While the bank's ultimate sparse goal was to generate long-term customer relationship value, the repeated metric measurement forced employees to take actions focused on achieving individual proxy metrics despite the fact that they had to make fictitious accounts to meet the daily/weekly/monthly established targets. In the end, metrics achieved were initially rewarded, but the bank ultimately destroyed, rather than created, long-term customer relationship value.
- **Boeing 737 MAX: Schedule & Cost Proxies:** To compete with Airbus, Boeing's management heavily prioritized launch deadlines and minimized simulator training costs (which would serve as dense proxies). Engineering safety feedback was sparse. The push to satisfy short-term targets (e.g., reward hacking) contributed to the design flaws in its automated flight control software, ultimately leading to catastrophic safety failures.

HOW AI DEVELOPMENT TECHNIQUES PRESENT USEFUL ANALOGIES FOR IMPROVING ERM PROCESSES

The following table summarizes these concepts in both an AI context and as an analogy for ERM. It also highlights the ultimate risk governance vulnerability each concept might create for an organization.

RL CONCEPT	AI CONTEXT	ANALOGY FOR ERM	SYSTEMIC RISK GOVERNANCE VULNERABILITY
Sparse Reward Problem	An AI agent rarely receives feedback from its errors or successes or only receives feedback after the entire task is completed, which makes it difficult for the agent to really learn the task.	Executives have to assess potential risk events that are inherently sparse or that have never occurred, which means they don't have sufficient information about what prior actions and decisions have been helpful in avoiding the risk in the past.	"No News is Good News" Bias: Absence of disaster is falsely presumed to be due to an organization's "superior risk management."
Continuous Dense Proxies	The AI agent is given lots of substitutes for real data on an ongoing basis along detailed steps of a process, with rewards and penalties for accurately completing individual steps, which helps the agent learn about successes and missteps at each step towards the ultimate goal.	Management dashboards have numerous daily, monthly, quarterly key performance indicators and key risk indicators that serve as proxies, but not exact measures of a specific risk event that might concern management.	Information Overload: Fancy software systems have glitzy charts and graphs, with management focused on continuous slicing and dicing of data while overlooking the fact that the proxy does not correlate with the actual risk event to be avoided.
Reward Hacking	AI agent takes shortcuts or leverages loopholes to maximize rewards (e.g., points) it receives for accuracy on the tiny individual steps with little focus on achieving the final outcome successfully.	Executive compensation and performance plans may incentivize management to focus on ensuring certain risk metrics (e.g., KRIs) stay within acceptable thresholds while losing sight on the larger, looming catastrophic risk event.	Metric Gaming: Local managers optimize indicator scores by managing to the metric rather than taking actions to truly mitigate the risk concern.



HOW AI DEVELOPMENT TECHNIQUES PRESENT USEFUL ANALOGIES FOR IMPROVING ERM PROCESSES

SOLUTIONS TO AVOID RISK GOVERNANCE PROCESS DESIGN FLAWS

When building autonomous AI agents, developers use advanced alignment techniques, as well as ensuring there is a “human-in-the-loop” so that the AI system’s actual behavior and underlying goals reliably match the desired outcomes throughout the process all the way through the achievement of the desired goal. Similarly, ERM leaders can use similar techniques to focus on the accuracy of their organization’s ERM process design and implementation and the avoidance of the undesired risk outcome.

Here are six considerations that might be helpful for ERM leaders:

1. **Process Reward Models (PRMs):** In this technique, ERM leaders shift their oversight from “Outcome Supervision” (e.g., a certain risk is avoided) to the evaluation and reward of the step-by-step rigor of the leadership’s decision-making under uncertainty. For example, ERM leaders work to help business unit leaders assess the quality of their scenario testing considerations, rather than merely focusing on quarterly financial results being achieved (which merely assumes risk management processes are effective).
2. **Synthetic Dense Feedback:** Because real disasters are sparse, ERM leaders can work with business leaders to develop hypothetical negative outcome feedback. Creating scenarios by engaging leaders in War-Gaming or Adversarial Simulations help force executives to experience catastrophic tail-risk consequences in controlled environments.
3. **Temporal Alignment & Claw Backs:** To address reward hacking potentials, ERM leaders may want to evaluate their organization’s executive compensation plans to determine whether vesting schedules for compensation rewards for certain strategic initiatives map directly to timelines linked to multi-year tail-risks that might occur later in the initiative cycle. During the financial crisis of 2008-2009, many financial institutions discovered that executives who designed some of the subprime mortgage products/securities had already received compensation (and had moved on from employment with the bank) before the risk events began to unfold. More entities are instituting mandatory claw back triggers to avoid risk metric manipulation.
4. **Hard Boundaries:** To avoid reward hacking, critical risk limits and thresholds need to be viewed as non-negotiable hard constraints rather than soft penalties. High profitability must never be permitted to offset or compensate for hard risk boundary violations.
5. **Look for KPIs and KRIs that Create “Goodhart” Vulnerabilities:** ERM leaders may want to assess whether certain performance metrics (KPIs) and certain risk metrics (KRIs) create unintended incentives that actually overlook an underlying risk condition. Look to see if those metrics might be subject to manipulations by managers to improve scores superficially.
6. **Conduct Hindsight Performance and Risk Analysis:** Executives might find it insightful to conduct a retrospective review of crucial business decisions that might have led to success or failure to determine the extent that downside risks were appropriately stress tested. Sometimes a positive outcome might have been due to “luck” versus “skill” when a highly likely risk that should have occurred, but for some reason didn’t.

The goal of this thought paper is to help ERM leaders “battle-test” their ERM programs to ensure processes ultimately lead to more strategic risk management.



HOW AI DEVELOPMENT TECHNIQUES PRESENT USEFUL ANALOGIES FOR IMPROVING ERM PROCESSES

ABOUT THE AUTHOR

Don Pagach serves as professor of accounting and director of research for the ERM Initiative in the Poole College of Management, where he helps lead our ongoing research related to ERM. Dr. Pagach has conducted extensive academic research that examines characteristics of organizations that implement enterprise risk management and how investors in the marketplace value the embrace by organizations of ERM processes. Currently he has ongoing research projects that examine the impact of integrating an organization's ERM processes with its focus on sustainability. He also works closely with visiting PhD students who come to NC State to partner on research related to ERM. Dr. Pagach is one of the co-authors of the ERM Initiative's annual survey report on the Executive Perspectives on Top Risks, in partnership with Protiviti.

ABOUT NORTH CAROLINA STATE UNIVERSITY'S ENTERPRISE RISK MANAGEMENT (ERM) INITIATIVE

The Enterprise Risk Management (ERM) Initiative in the Poole College of Management at North Carolina State University provides thought leadership about ERM practices and their integration with strategy and corporate governance. Faculty in the ERM Initiative frequently work with boards of directors and senior management teams helping them link ERM to strategy and governance, host executive workshops and educational training sessions, and issue research and thought papers on practical approaches to implementing more effective risk oversight techniques (www.erm.ncsu.edu).