Timing-Aware Detection of Emergent Infinite Loops in TCG Design Workflows

Yuanzhe Jin[1,*], Zihan Xu[2] and Shuhan Liu[2]

[1]University of Oxford, Oxford, United Kingdom
[2]Zhejiang University, Hangzhou, China

Abstract

Designing trading card games (TCGs) requires balancing creativity with the risk of unintended card interactions. Among the most disruptive failures are emergent infinity loops, where combinations of card effects enable players to repeat actions indefinitely or for an effectively unbounded duration. While prior work and practical testing often focus on identifying such loops after they appear, much less is known about when an AI assistant should intervene during the design process to help prevent them without unnecessarily disrupting designers' workflow. In this paper, we reframe infinity-loop prevention as a timing-sensitive human-AI collaboration problem. We present a design framework for timely AI intervention in TCG design and playtesting, grounded in an abstraction of loop-inducing mechanics such as resource renewal, card copying, time resetting, and state-preserving triggers. We discuss the trade-off between early warnings that may interrupt creative exploration and late warnings that may allow problematic designs to propagate further in the pipeline. Our work positions infinity-loop prevention not only as a game-balance problem, but also as a broader question of how AI systems can intervene at the right time in complex design tasks.

Keywords

Trading Card Game, Game Theory, AI Intervention, Game Design

1. Introduction

Trading card games (TCGs) have long been valued for their complexity, versatility, and capacity for creative combination. In games such as Hearthstone, Magic: The Gathering, and similar titles, players and designers engage with large collections of cards whose effects interact in often surprising ways. This combinational richness is one of the core attractions of the genre, but it also creates a persistent design challenge: some card interactions can produce unintended infinite loops, allowing actions to be repeated indefinitely or for an effectively unbounded duration. Such loops may damage game balance, disrupt normal play, and create substantial downstream costs for development and testing.

Existing analyses of TCG systems have explored deck construction, balance modeling, search, and AI play strategies [1, 2]. Comparatively less attention has been paid to how designers can be supported in identifying dangerous multi-card interactions during the design process itself. Prior observations suggest that what is often referred to as "infinite" in TCGs is not simply a matter of large numerical values, but of interaction structures that bypass or exploit the normal operational limits of the game. Even when games impose caps on health, mana, hand size, board size, round count, or turn duration, certain combinations of copying, resource renewal, trigger chaining, and time-resetting effects may still enable effectively non-terminating play.

In this work, we extend this problem from a purely game-mechanical perspective to a human-AI collaboration perspective. We argue that the key question is not only whether an AI system can detect the risk of an infinite loop, but when it should surface that risk to human designers or playtesters. This timing issue matters because interventions are not neutral: early interventions may prevent costly errors, but they may also interrupt exploration and reduce designers' willingness to experiment;

Proceedings of TimelyAI: When Should Generative AI Assistants Intervene? A Workshop Co-located with the 5th Symposium on Human-Computer Interaction for Work (CHIWORK 2026), Linz, Austria, June 22 to 25, 2026

*Corresponding author.
†These authors contributed equally.
yuanzhe.jin@eng.ox.ac.uk (Y. Jin); zihanxu@zju.edu.cn (Z. Xu); shliu@zju.edu.cn (S. Liu)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-09-14

late interventions may be less disruptive, but may arrive after problematic mechanics have already propagated through implementation and testing.

Accordingly, we frame infinite loop prevention as a timing-sensitive AI assistance problem in TCG design workflows. We characterize loop-related signals across design stages and analyze how the appropriate form of intervention evolves as these signals accumulate. Through this framing, we offer a conceptual direction for studying when infinite loops become detectable and how AI assistance can be staged to match the pace of human creative work.

2. Related Work

Research on trading card games has developed along several lines, but leaves the problem of timing-aware loop prevention largely unaddressed.

A significant body of work has focused on deck construction and card balance. García-Sánchez et al. explored automatic deck building in Hearthstone using genetic algorithms, delegating evaluation to game simulators that test decks against a range of human-designed references [3]. Researchers also proposed a balance model that uses a series of functions to quantify card strength, offering a foundation for preventing cards from becoming disproportionately powerful [4]. These approaches address the quality of individual cards or decks, but do not consider how combinations of cards might give rise to structurally problematic interaction patterns.

A related line of work has examined AI play strategies and opponent modeling. Some work highlights the challenges of search and opponent modeling under imperfect information [5], while other work argues that for complex games, design spaces must be explored at the component level rather than end-to-end [6]. Monte Carlo methods have been particularly prominent in this area: Cowling et al. applied Monte Carlo search to card selection in Magic: The Gathering [7], Świechowski et al. introduced Monte Carlo Tree Search for Hearthstone [8], and Santos et al. demonstrated that MCTS provides a competitive alternative to existing AI players [9]. Further work showed that statistical learning methods can be used to improve card-playing strategy [10]. While these works advance AI competence in playing TCGs, they are primarily concerned with winning rather than with supporting the design process.

Neural network approaches have also been applied to card games. Da Silva et al. proposed a mechanism for classifying and predicting game information to address generalization limitations [11], and Stiegler et al. demonstrated that parallel adaptive neural networks can achieve performance comparable to deep learning methods [12]. These techniques offer tools for modeling game states, but are not oriented toward detecting interaction risks during design.

Beyond play and balance, some work has examined structural properties of TCGs more broadly. Xiao et al. applied game refining theory to evaluate the complexity of individual game processes [13], and Churchill et al. showed that Magic: The Gathering is Turing complete by embedding arbitrary Turing machines into its ruleset [14]. The latter result is particularly relevant to our concern: it confirms that TCG interaction structures can, in principle, produce non-terminating behavior, and that such behavior is not merely an edge case but a fundamental property of sufficiently expressive card systems.

Taken together, existing work has made substantial progress on deck construction, play strategy, balance modeling, and game evaluation. However, comparatively little attention has been paid to the design-time identification of interaction hazards in those involving multi-card combinations, and still less to the question of when such detection should be surfaced to human designers. It is this gap that the present work addresses.

3. Infinite Loop in A TCG

First of all, it needs to clarify an important concept in this paper, which is “Infinite Loop”.

Infinite Loop in a TCG means a player can play as many cards as possible to beat their opponent.

Although modern TCGs impose explicit gameplay constraints, such as limits on resources, board occupancy, hand size, and turn duration, these constraints do not necessarily eliminate the possibility of effectively unbounded action sequences. In practice, infinite loops emerge not from a single overpowered card but from combinations of mechanics that repeatedly reconstruct the conditions required for continuation.

There are two forms of single-round infinite loop:

- I. A "Perpetual Motion Machine(PMM)" that requires no action. Suppose there are two followers, A and B. Their effects and trigger conditions are mutual, which means a can satisfy b's trigger condition and b can trigger a's effects. When a player starts the power a of A, and satisfy B's trigger condition to trigger B's effect b, and then start the power a of A again, forming an infinite loop of a - b - a - b ...
- II. Player-driven loops. The most well-known is the infinite fireball one-turn kill deck. If there is an Archmage Antonidas(its effect: whenever you cast a spell, add a "Fireball" spell into your hand) and 4 Sorcerer's Apprentices on the battleground, the fireball is 0 cost, which means the player can play any number of fireballs in a single round.

These loops rely on interactions such as resource renewal, copying effects, trigger preservation, or repeated state reconstruction. Some loops may operate automatically through mutually reinforcing triggers, while others remain player-driven and depend on repeated low-cost or zero-cost actions within a turn. What matters is not the specific implementation of any individual mechanic, but whether the interaction structure can continuously sustain repetition despite the game's normal stopping conditions.

Furthermore, in other card games such as Yu-Gi-Oh!, there are fewer restrictions, making it easier to achieve infinite loops.

3.1. Signals and Timing of AI Intervention

In trading card game design, the risk of an infinite loop does not usually come from a single card in isolation. More often, it emerges gradually from a combination of effects that, when put together, remove the ordinary stopping conditions of play. Our previous analysis has already shown that explicit limits in the game, such as caps on health, mana, hand size, board size, round count, or turn duration, are not sufficient to rule out effectively unbounded action sequences. What matters more is whether a set of mechanics can repeatedly reconstruct the conditions required for the next iteration. From this perspective, infinite loop prevention should not be understood only as a problem of detecting a dangerous combination after it has already been formed. It should also be understood as a problem of deciding when an AI assistant ought to step in during the design process.

To make this perspective more concrete, we summarize the proposed timing-aware view in Fig. 1. The framework describes how an AI assistant can support TCG design as a continuous process rather than a one-time detection task. Starting from evolving card descriptions, the assistant extracts loop-related signals derived from our analysis of representative infinite-loop cases and core game constraints, such as resource renewal, copying, trigger preservation, and temporal extension. These signals are then aggregated into a higher-level estimate of structural risk, reflecting the degree to which the current design is approaching a self-sustaining interaction pattern. Based on this evolving risk, the assistant determines when intervention becomes appropriate and adapts the form of feedback accordingly, ranging from lightweight hints to stronger warnings or explanations. This pipeline highlights that the central challenge is not only identifying risky structures, but deciding when they have become meaningful enough to warrant attention.

Figure 1 also illustrates how intervention strength escalates as loop-related evidence accumulates across the workflow. During card authoring, the AI assistant observes only weak local signals, such as low-cost duplication or isolated trigger interactions, and therefore provides lightweight annotations rather than disruptive warnings. As the design enters iterative balancing, previously independent mechanics may begin to reinforce one another, for example, when resource renewal interacts with

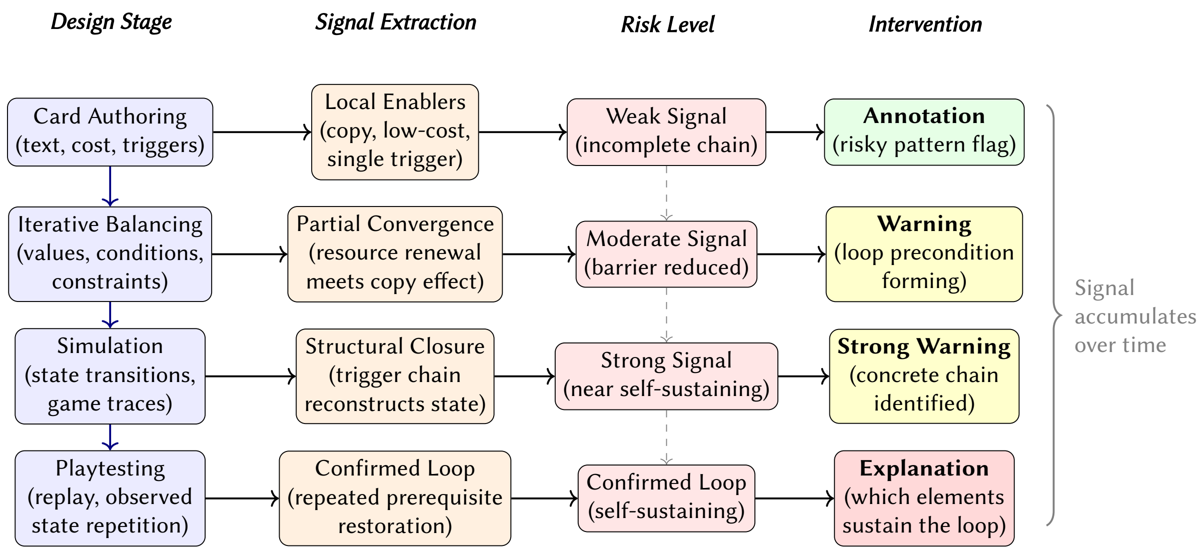

Figure 1: A timing-aware framework for AI intervention in TCG design workflows. The framework is organized by design stage (rows), showing how loop-related signals are extracted, aggregated into a risk estimate, and translated into staged interventions. Signal strength accumulates across stages (dashed vertical arrows), and the form of intervention escalates accordingly: from lightweight annotations during authoring, to warnings during balancing and simulation, to explicit explanations during playtesting.

copying effects. At this stage, the framework elevates the intervention to a stronger warning because the interaction structure is becoming more stable. During simulation, repeated state transitions and reconstructed trigger chains provide more concrete evidence that the design is approaching a self-sustaining loop, allowing the assistant to identify specific structural risks. Finally, during playtesting, the framework shifts from speculative warning to explicit explanation by highlighting which elements repeatedly restore the prerequisites for continuation. In this way, Fig. 1 represents not only a detection pipeline, but also a staged policy for timing-aware AI intervention.

This shift in perspective is important because intervention is inherently time-sensitive. If an AI system warns too early, it may interrupt creative exploration at a moment when the design is still highly provisional, and many risky-looking ideas will never develop into real problems. In that case, assistance becomes noise rather than support. However, if the system intervenes too late, the problematic interaction may already have propagated into balancing, implementation, internal testing, or even release preparation, at which point the cost of correction becomes much higher. The central question is therefore not only whether an AI assistant can identify loop-related risk, but when the available evidence becomes strong enough, useful enough, and actionable enough to justify intervention.

In the case of infinite loops, the relevant evidence often appears as a gradual accumulation of structural signals. A design becomes suspicious when it contains

- Effects that renew the resources required for repetition.
- Copy the elements needed for the next cycle.
- Preserve the triggering conditions of the chain.
- Extend the time horizon within which repeated actions can continue.
- ... ...

None of these properties is necessarily problematic on its own. Resource recovery may be an ordinary balancing mechanic, copying may support strategic variety, and trigger preservation may simply be part of a card's intended identity. The risk arises when these elements begin to interact as a closed or nearly closed chain, such that the cost of continuing the sequence approaches zero while the conditions for continuing it are continuously re-established. At that point, what looks like a set of separate mechanics starts to behave like a loop.

Seen in this way, the role of an AI assistant is not limited to issuing a binary judgment of safe or unsafe. Instead, it should monitor whether a design is moving toward a structure in which repetition becomes self-sustaining. As shown in the card-authoring stage of Fig. 1, the assistant may initially have only weak evidence derived from card text, cost structure, and trigger semantics, so intervention should remain lightweight and non-disruptive. During later revision, when the designer adjusts numerical values, trigger conditions, or interaction constraints, the same assistant may have a better basis for warning that a previous barrier to repetition has been removed. Once simulation traces or playtest records become available, the evidence becomes stronger still, because the assistant can reason not only over intended card descriptions but also over actual or near-repeated state transitions. In other words, the appropriate timing of intervention depends on both the maturity of the design and the strength of the loop-related signals that have accumulated around it.

This timing perspective also changes how infinite loop prevention should be framed as a design-support problem. The goal is not to eliminate all risky mechanics at the earliest possible moment, since doing so would likely suppress experimentation and reduce the very combinational richness that makes trading card games interesting. Rather, the goal is to provide support that is proportional to the stage of the workflow and the confidence of the warning. Early in the process, this may take the form of a subtle indication that a proposed effect belongs to a historically risky pattern. Later, it may become appropriate to surface a stronger warning that a concrete interaction chain can regenerate its own prerequisites. In still later stages, especially after simulation or playtesting, the assistant may provide an explicit explanation of how a loop is formed, which elements sustain it, and why the design no longer stops under normal game constraints.

By framing the problem in this way, infinite loop prevention becomes more than a question of game balance alone. It becomes a case of timing-sensitive human-AI collaboration in a specialized design workflow. The practical challenge is not merely to detect whether a loop is possible, but to decide at what moment AI support is most helpful to the human designer. An intervention that is too early may be ignored, while an intervention that is too late may be costly. A useful AI assistant must therefore do more than detect structure; it must also judge when that structure has become meaningful enough to deserve attention.

4. From Infinite-Loop Structure to AI Intervention Opportunities

To better understand when AI intervention should occur, we now revisit infinite loops from a structural perspective.

The key challenge in preventing infinite loops is that they are rarely visible as isolated card effects. What makes them difficult is not the presence of any single strong mechanic, but the way several ordinary mechanics can be arranged so that they collectively sustain repetition. In our earlier analysis, the loop does not arise because one card is "infinite" by itself, but because different effects take on complementary roles: some restore the resources needed to continue the sequence, some reproduce critical elements, some preserve or recreate the triggering environment, and some remove practical constraints such as time pressure. Once these roles are aligned, the design may become capable of repeating an action chain for an effectively unbounded duration. This means that the real object of concern is not an individual card, but an interaction structure distributed across multiple cards and states.

This observation is important for AI-assisted design because it suggests that intervention should be based on structural convergence rather than on isolated suspicious effects. A copying mechanic alone is not enough to justify a strong warning. A resource-renewal effect alone is also not necessarily problematic. Even a time-related mechanic may be harmless in many local contexts. The risk becomes more substantial when these effects begin to interact in a way that reconstructs their own preconditions across iterations. In other words, the AI assistant should not simply ask whether a card is powerful, but whether the surrounding design context is moving toward a configuration in which continuation becomes self-supporting. This is precisely the moment at which intervention becomes more meaningful:

not when a risky ingredient first appears, but when multiple ingredients begin to organize themselves into a repeatable chain.

From this perspective, the concrete loop examples discussed earlier can be understood as instances of a more general pattern. A candidate loop tends to evolve through several stages. At first, the design contains only local enablers, such as low-cost duplication, trigger amplification, or conditional recovery. At this stage, the interaction remains incomplete, and intervention should be cautious because many such designs will never mature into actual failures. As the design evolves, however, these local enablers may start to connect. A copied effect now reproduces the element that restores resources; a restored state now reopens the trigger condition for another cycle; a temporal mechanic increases the available horizon for carrying the repetition further. Once these dependencies form a sufficiently closed chain, the system no longer merely contains risky ingredients but begins to exhibit loop-like organization. This intermediate stage is particularly important because it is often the point at which an AI assistant can still provide useful guidance before the design has propagated too far downstream.

What follows from this is a different understanding of where AI intervention opportunities lie. A helpful assistant should not wait until a full loop has already been demonstrated in exhaustive play, because by then the warning may be accurate but expensive. At the same time, it should not react to every single local risk pattern with equal intensity, because doing so would burden the designer with repeated low-value interruptions. Instead, intervention should become progressively stronger as the interaction structure becomes more complete. When only one enabling role is present, the assistant may merely annotate the design with a weak signal. When two or more roles begin to reinforce one another, it may become appropriate to surface a stronger notice that the current revision is reducing the natural barriers to repetition. When an interaction trace or simulation already shows repeated restoration of prerequisites, the assistant may provide an explicit explanation that the design is approaching or has reached a self-sustaining loop. In this way, the assistant's role is not simply to detect completed loops, but to recognize when a design is crossing from isolated possibility into structurally meaningful risk.

Because such interaction structures emerge gradually through the alignment of multiple roles, infinite-loop prevention should be treated as a timing problem rather than as a purely technical detection task. The same underlying structure may justify different forms of support depending on when it is encountered. Early in authoring, the designer may benefit only from a subtle cue that a mechanic belongs to a historically risky family. During balancing or iterative revision, the same pattern may deserve a more direct warning because the design is becoming more concrete, and the cost of local change is still manageable. Later, when simulation traces or replay evidence are available, the assistant can shift from speculative warning to explanatory diagnosis by showing how the repeated chain is formed and which dependencies sustain it. The value of AI support, therefore, does not depend only on whether it identifies the right structure, but also on whether it introduces that information at a moment when the human designer can still act on it effectively.

In this sense, the infinite-loop model provides more than a description of a special game phenomenon. It offers a useful way of thinking about how AI assistance can be staged in a complex design workflow. The model reveals that problematic repetition is assembled gradually through interacting functional roles, and that these roles become visible to an AI system at different levels of confidence as the workflow progresses. Framed in this way, the contribution is no longer limited to showing that an infinite loop can exist. Rather, it shows how a structural understanding of loop formation can support a timing-aware strategy for AI intervention, in which assistance becomes stronger as evidence accumulates and as the opportunity for effective human action changes.

5. Conclusion

In this work, we revisited the problem of infinite loops in trading card games from a new perspective. Rather than treating loop formation solely as a game-mechanical phenomenon or a post hoc detection task, we framed it as a timing-sensitive human-AI collaboration problem. Our analysis suggests that infinite loops do not arise from individual card effects in isolation, but from the gradual convergence of multiple functional roles, such as resource renewal, copying, trigger preservation, and temporal extension, into a self-sustaining interaction structure. This structural view allows us to move beyond binary detection and consider how loop-related signals emerge and accumulate over time. We emphasized a timing-aware approach, in which AI support adapts to both the strength of loop-related signals and the stage of the design workflow, ranging from lightweight annotations during authoring to more explicit explanations during simulation and playtesting.

By positioning infinite loop prevention in this way, we highlight a broader implication for AI-assisted design. The value of an AI assistant in complex creative tasks does not depend solely on its ability to identify problematic structures, but also on its ability to introduce that information at a moment when it is actionable and useful to the human collaborator. Trading card game design provides a concrete and illustrative case, but the underlying challenge extends to many domains in which complex interactions and delayed failures are common.

This work is conceptual in nature, and several directions remain open for future research. An important next step is to empirically evaluate different intervention timings and forms of feedback, for example, through controlled studies with designers or through retrospective analysis of design iterations. Another direction is to develop and integrate concrete AI systems that can operationalize the proposed signals and timing strategies in real-world design tools. We hope that this work can serve as a starting point for studying not only how AI can detect complex interaction risks, but also how it can collaborate with humans at the right time.

Declaration on Generative AI

During the preparation of this work, the authors did not use any generative AI or AI-assisted technologies.

References

[1] R. Vieira, A. R. Tavares, L. Chaimowicz, Drafting in collectible card games via reinforcement learning, in: 2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), 2020, pp. 54–61. doi:10.1109/SBGames51465.2020.00018.

[2] Z. Chen, C. Amato, T. D. Nguyen, S. Cooper, Y. Sun, M. S. El-Nasr, Q-deckrec: A fast deck recommendation system for collectible card games, CoRR abs/1806.09771 (2018). URL: http://arxiv.org/abs/1806.09771. arXiv:1806.09771.

[3] P. García-Sánchez, A. Tonda, G. Squillero, A. Mora, J. J. Merelo, Evolutionary deckbuilding in hearthstone, in: 2016 IEEE Conference on Computational Intelligence and Games (CIG), 2016, pp. 1–8.

[4] Y. Jin, Proposed balance model for card deck measurement in hearthstone, The Computer Games Journal 8 (2019) 25–40. doi:10.1007/s40869-018-0072-9.

[5] C. D. Ward, P. I. Cowling, Monte carlo search applied to card selection in magic: The gathering, in: 2009 IEEE Symposium on Computational Intelligence and Games, 2009, pp. 9–16.

[6] A. Bhatt, S. Lee, F. de Mesentier Silva, C. W. Watson, J. Togelius, A. K. Hoover, Exploring the hearthstone deck space, in: Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG '18, Association for Computing Machinery, New York, NY, USA, 2018. URL: https://doi.org/10.1145/3235765.3235791. doi:10.1145/3235765.3235791.

[7] P. I. Cowling, C. D. Ward, E. J. Powley, Ensemble determinization in monte carlo tree search for the imperfect information card game magic: The gathering, IEEE Transactions on Computational Intelligence and AI in Games 4 (2012) 241–257.

[8] M. Świechowski, T. Tajmajer, A. Janusz, Improving hearthstone ai by combining mcts and supervised learning algorithms, in: 2018 IEEE Conference on Computational Intelligence and Games (CIG), 2018, pp. 1–8.

[9] A. Santos, P. A. Santos, F. S. Melo, Monte carlo tree search experiments in hearthstone, in: 2017 IEEE Conference on Computational Intelligence and Games (CIG), 2017, pp. 272–279.
[10] E. Bursztein, I am a legend: Hacking hearthstone using statistical learning methods, in: 2016 IEEE Conference on Computational Intelligence and Games (CIG), 2016, pp. 1–8.
[11] A. R. da Silva, L. F. W. Goes, Hearthbot: An autonomous agent based on fuzzy art adaptive neural networks for the digital collectible card game hearthstone, IEEE Transactions on Games 10 (2018) 170–181.
[12] A. Stiegler, K. P. Dahal, J. Maucher, D. Livingstone, Symbolic reasoning for hearthstone, IEEE Transactions on Games 10 (2018) 113–127.
[13] C. Xiao, A. Primanita, M. N. A. Khalid, H. Iida, Analysis of card collection game 'hearthstone'*, in: 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2019, pp. 1–6.
[14] A. Churchill, S. Biderman, A. Herrick, Magic: The gathering is turing complete, CoRR abs/1904.09828 (2019). URL: http://arxiv.org/abs/1904.09828. arXiv:1904.09828.