Hybrid Intelligence in Evolving Games: Automated Rule Design, Strategy Evolution, and Evaluation Optimisation for Intelligent Societies

PU, JIYAO

Hybrid Intelligence in Evolving Games: Automated Rule Design, Strategy Evolution, and Evaluation Optimisation for Intelligent Societies

Jiyao Pu

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
June 2025

Abstract

Adaptive rule evolution is a fundamental challenge in artificial life and multi-agent systems research, as traditional game environments rely on static, human-designed rules that cannot adapt to emergent behaviours. This dissertation addresses this limitation by exploring automated, dynamic rule creation mechanisms in multi-agent digital environments. It first introduces a structured Strategy-Evaluation-Rule (SER) framework for rule generation that formalises the interplay between agent strategies, evaluative feedback, and rule adaptation. Unlike prior approaches to game rule design, SER does not rely on any prepared datasets or domain knowledge; instead, it generates rules on the fly and refines them through iterative self-play evaluation. The SER framework is implemented in two games, Maze Run and Trust Evolution, demonstrating its effectiveness in driving emergent, complex agent behaviours across disparate domains.

Building on this foundation, the thesis presents the Triadic Reciprocal Dynamics (TRD) system, which establishes a novel closed-loop paradigm linking rule creation, strategy evolution, and performance evaluation. TRD is instantiated as a multi-agent game environment where AI, NPCs, and human players participate together. It employs a neural rule designer and an automated evaluator to continuously generate new game rules and assess their impact on evolving strategies in real time.

Furthermore, principles from Flow Theory are incorporated into the SER framework to enable flow-driven rule design, ensuring that gameplay remains engaging. This extension features dynamic difficulty adjustment (DDA), which dynamically tunes challenge levels in real-time, and a dual-reward scheme that balances extrinsic rewards (e.g., points or achievements) with intrinsic signals of player engagement. A real-time flow visualisation interface is also introduced to monitor players' flow states and guide on-the-fly adjustments to maintain an optimal flow experience. Together, these contributions advance the state of the art in automated game design and adaptive multi-agent systems by enabling rules that evolve autonomously, exemplifying a novel paradigm of dynamic rule evolution and laying the groundwork for future research in lifelike, self-evolving multi-agent environments.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

- **Chapter 3: "Rules for Expectation: Learning to Generate Rules via Social Environment Modeling."**

  Pu J., Duan H., Zhao J., and Long Y. In *IEEE Transactions on Circuits and Systems for Video Technology*, 34(8), pp. 6874–6887, 2023.

- **Chapter 4: "Triadic Reciprocal Dynamics: The AI Framework for Social Rule Evolving."**

  Pu J., Shao M., Sun Y., Li X., Miao X., Ponizovskiy V., and Long Y. Submitted to *Nature Machine Intelligence*, 2025.

- **Chapter 5: "Flow-Centric Rule Design: Evolving Rules for Optimal Difficulty and AI Skill Balance."**

  Pu J., Shao M., Sun Y., Li X., Miao X., and Long Y. Submitted to *ACM International Conference on Multimedia (ACM MM)*, 2025.

Acknowledgements

I would like to express my deepest gratitude to everyone who has enlightened, encouraged, and supported me throughout my PhD journey. Without your discussions, communication, and advice, my research would not have reached this stage.

I am sincerely grateful to my supervisor, Dr. Yang Long, for his unwavering academic and personal support. His timeless words, " 仰望星空脚踏实地 (Look up at the stars, keep your feet on the ground)", " 格物, 致知, 诚意, 正心 (Investigate things, extend knowledge, be sincere in intention, and rectify the mind)", and " 行为本身是行为的奖励 (The reward of your action is the action itself)", have opened new doors in my life and broadened my perspective. The wisdom shared during our midnight talks not only guided me in exploring the domain of rule generation but also inspired me to delve into the realm of metaphysics.

I extend my heartfelt thanks to all the members of HI Lab (Hybrid Intelligence Lab). The engaging discussions, collaborative demonstrations, and shared programming sessions have forged strong connections among us. I am especially grateful to Dr. Haoran Duan, whose support helped me publish my first paper, as well as to Dr. Peng Zhang, Dr. Yang Bai, Dr. Fan Wan, and Dr. Rui Gao, who have served as exemplary role models. I also appreciate the assistance and constructive reviews from my fellow PhD students –Xingyu Miao, Tianyu Zhang, Yuchen Li, Leyuan Zhang, Mingye Shao, Xueqi Qiu, Yueming Sun, Sihan Guo, Xinrun Li, Zixuan Wang, Kaili Sun, Yiheng Wang, and Xiangxin Meng. I will always remember the time when one of us encountered a coding bug, and everyone rallied to help.

I am thankful to Collingwood College for its warm and welcoming environment. The staff' s constant positivity and readiness to help have been like rays of sunshine, offering direct and thoughtful support whenever needed. Although I did not have the chance to live in the College, I always felt like I was part of a caring family. I also appreciate the guidance and support provided by the Counselling & Mental Health Service, which helped me navigate through challenging times.

My deepest gratitude goes to my parents and grandmother—the most precious gifts in my life. Their unconditional love and support have been my anchor, lifting me up in times of failure and guiding me when I lost my way. No matter what happens, my home will always be a place of warmth, with delicious food, clean clothes, a warm bed, and a loving family.

I am thankful for all the friends I met. I especially appreciate Donald Gunton, my skipper, for introducing me to sailing in Scotland and lighting a hope that burns within my heart; David Parker, who taught me the technique of lighting a fireplace and being humble; Runjie Yan, a friend, a trendsetter, and a ”fortune-teller”; Diwen Si, ”airborn hussar”; and Wei Liu, for providing me with a lovely home. Their insights have enriched my understanding of this complex world. I am also sincerely grateful to the Jesmond Parish Church for its support.

Lastly, I thank God for guiding me through these years. In my moments of weakness, His presence has made me whole. I aspire to be a reflection of His salt and light, like a tree planted by streams of water.

Contents

List of Figures

List of Tables

Dedication

To the Creator and the Created.

CHAPTER 1

Introduction

Rules, broadly defined as prescriptive statements that define acceptable behaviour within any system, be it legal, computational, physical, mechanical, or game-based, serve as foundational elements that structure interactions and drive the evolution of intelligent, adaptive systems [1]. In legal and policy domains, rules formalise roles, responsibilities, and accountability through statutes and regulatory mandates, while in computational settings, particularly within machine learning and reinforcement learning (RL), they offer explicability and immediate adaptability by encoding operational constraints [2]. For example, rule-based decision-making policies in self-driving vehicles ensure compliance with traffic laws without necessitating complete retraining [2]. In contrast, RL focuses on learning optimal behaviours from trial-and-error interactions, where rules may function as initial guides or constraints shaping the learning process [3, 4]. This duality is further illustrated in game environments, where clear rules facilitate balanced gameplay by defining allowable moves, rewards, and penalties [5]. Moreover, modern research has increasingly integrated rule-based systems into dynamic, multi-agent frameworks, employing hybrid approaches that combine explicit rules with learning algorithms to enhance interpretability and maintain strategic coherence [3,4,6]. Consequently, the concept of rule-based mechanisms not only underpins diverse systems but also provides the structure and flexibility necessary for the development of adaptive, intelligent systems across multiple disciplines.

AI-based content generation refers to the use of artificial intelligence to produce digital content autonomously—ranging from text and images to audio and video—that traditionally required human creativity and expertise. This domain is rooted in generative modeling, which encompasses a wide array of algorithms and architectures developed over the years, including Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Restricted Boltzmann Machines (RBM), Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), Transformer-based models, and Diffusion Models [7]. Early statistical models like GMM and HMM provided foundational probabilistic frameworks for content generation, while recent deep learning architectures such as GANs and Transformers have dramatically enhanced the fidelity and contextual relevance of the generated outputs [7, 8]. The evolution of these generative AI models has not only expanded the technical repertoire of content creation but also significantly influenced various application domains —from creating immersive digital worlds and interactive metaverse elements [9] to innovating advertising strategies in marketing [10] and enhancing diagnostic imagery in medicine [11]. Moreover, by automating substantial portions of the creative process, AI-based content generation democratises content production, empowering individuals and enterprises to overcome traditional barriers associated with specialised skills and resources, thereby accelerating innovation in interactive media, education, and scientific communication [9, 10]. All these AI-based generative models offer a novel perspective on rule creation and optimisation.

Strategy exploration refers to the autonomous discovery, evaluation, and refinement of decision-making policies within complex environments. Grounded in reinforcement learning (RL), where agents interact with an environment modelled as a Markov Decision Process (MDP) to iteratively maximize cumulative rewards [12,13], this research domain leverages a formal framework that defines states, actions, and rewards to systematically explore and exploit strategic spaces. Early RL approaches, such as Q-learning for discrete action spaces and policy gradient methods for continuous optimisation, laid the foundation for later innovations, including actor-critic models that synergise value estimation with policy improvement [12,14,15]. The advent of deep reinforcement learning further expanded this landscape with algorithms like Deep Q-Networks (DQN), Proximal Policy Optimisation (PPO), Asynchronous Advantage Actor-Critic (A3C), Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Soft Actor-Critic (SAC), enabling high-quality strategy derivation in high-dimensional, dynamic environments [16]. More sophisticated approaches, including Meta-Reinforcement Learning, Inverse Reinforcement Learning, and Multi-Agent Reinforcement Learning (MARL), have further enriched the field by addressing rapid task adaptation, inferring reward structures, and modeling cooperative as well as competitive interactions [17]. Applications across robotics, gaming, and autonomous driving underscore the transformative potential of these methodologies in automating and enhancing decision-making processes, thereby fostering the development of adaptive, intelligent systems capable of navigating complex real-world scenarios [12–14]. By leveraging reinforcement learning or any computer-based models to explore strategies within an environment, a rational and effective foundation for rule evaluation can be established.

Evaluation paradigms represent systematic frameworks and methodologies for assessing the performance, robustness, generalizability, and interpretability across diverse learning approaches, including supervised, unsupervised, semi-supervised, meta learning, reinforcement learning, deep learning, self-supervised learning, symbolic AI, hybrid approaches, and zero shot learning [18]. These paradigms extend beyond traditional outcome-based assessments by incorporating both task-specific and ability-oriented metrics—ranging from accuracy, precision, recall, and F1 scores in supervised settings to clustering quality, anomaly detection rates, and dynamic reward structures within a Markov Decision Process (MDP) framework in reinforcement learning [13]. The evolution of these evaluation frameworks reflects the increasing complexity of AI tasks and interdisciplinary applications, as seen in hybrid models that integrate symbolic reasoning with deep learning to balance transparency and empirical performance [19] and in standardised benchmarks for domains such as protein function prediction [20]. Collectively, multi-dimensional evaluation

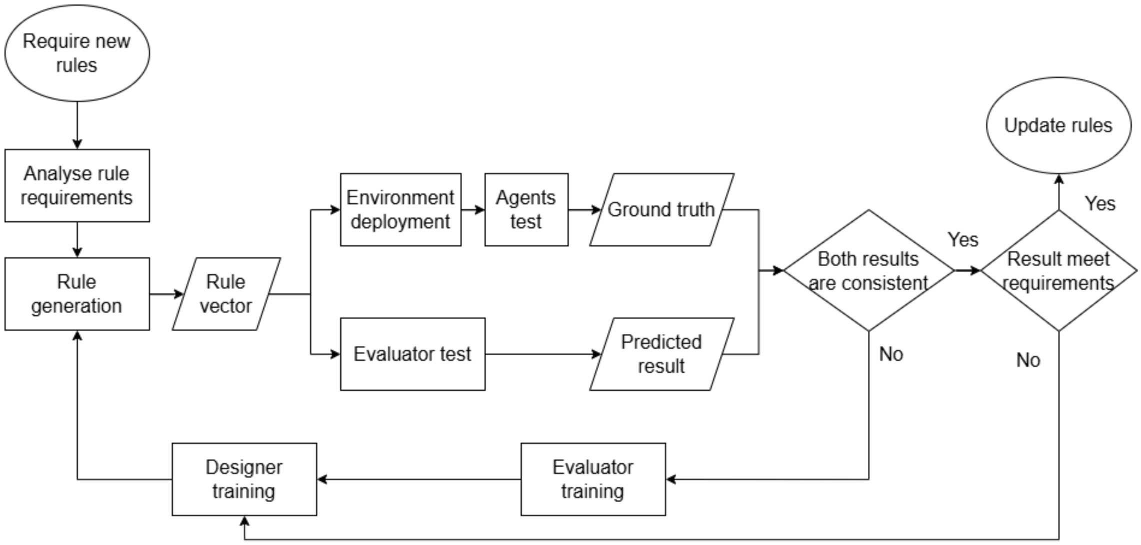


Figure 1.1: SER rule generation process flowchart, illustrating the iterative loop from rule requirement analysis through rule generation, parallel evaluation via environment and evaluator tests, consistency decision, and feedback-driven designer/evaluator retraining to refine rule vectors.

paradigms play a pivotal role in validating AI systems, facilitating methodological comparisons, and driving continuous improvements in algorithm design, thereby ensuring robust, transparent, and adaptable implementations across various real world applications, and are essential for the creation and optimisation of rule-based frameworks.

In this thesis, we proposed the Strategy–Evaluation–Rule (SER) framework to close the loop between gameplay and rule adaptation. AI agents, NPCs, or human players interact with the game environment under a given rule set $\mathcal{R}$, exploring strategies $\mathcal{S}$ that quantify their skill and behaviour. Then, the Rule Designer module generates new rule vectors $\mathcal{R}$ based on desired outcomes E_{target}. After that, the Evaluator estimates actual game results $E = f(\mathcal{R}, \mathcal{S})$ via configurable metrics—either crafted by human experts or learned by AI—which assess each rule' s capacity to balance challenge, skill, and engagement. By iterating these three modules—strategy informing evaluation, evaluation guiding rule design, and rules reshaping strategy—the SER framework continuously tunes game parameters in real time to maintain optimal flow dynamics and maximise player engagement. Fig. 1.1 demonstrate the progress of the rule generation from the rules requirements to the updated rules.

1.1 Motivation

Traditional rule-generation methods—ranging from symbolic induction algorithms such as CN2 [21] to recent neural programme synthesis approaches [22], have demonstrated the feasibility of deriving static rule sets from data or trained models. However, the CN2 algorithm induces a fixed set of classification rules from labelled data without provisions for updating these rules as new samples arrive, while neural programme synthesis approaches generate compositional rules or programs entirely in an offline phase, rather than allowing them to evolve through ongoing interaction. These approaches suffer from two critical bottlenecks: (1) they treat rules as fixed, human-defined and lack mechanisms for agents to adapt or evolve rules in response to changing environments; and (2) they decouple rule creation from strategic exploration and outcome evaluation, relying instead on off-line or one-shot training procedures. Meta reinforcement learning methods [23–25] address adaptation by quickly tuning policy parameters across tasks, but do not extend this adaptivity to the generation of the underlying task specifications themselves. Likewise, generative LLM frameworks such as Auto-GPT [26] can propose novel content or prompts, yet they lack integrated feedback loops that ground rule proposals in concrete environment dynamics and agent performance.

Motivated by these limitations, we introduce the SER (Strategy–Evaluation–Rule) framework and its instantiation in the Triadic Reciprocal Dynamics (TRD) system, which together closes the loop on rule creation, strategic exploration, and outcome evaluation. In our paradigm, agents autonomously propose, test, and refine rule vectors $\mathcal{R}$ based on observed strategy trajectories $\mathcal{S}$ and evaluation outcomes $E = f(\mathcal{R}, \mathcal{S})$. By relinquishing reliance on fixed, human-defined MDP configurations, our approach endows agents with true environmental autonomy and resilience, enabling continuous adaptation to novel, non-stationary, multi-agent settings without manual redesign of states, actions, or objectives.

Traditional rule-generation methods treat the rule set $\mathcal{R} \in \mathbb{R}^{N_r \times D_r}$ as a fixed, human-prescribed artifact, offering no means for agents to adjust or evolve rules when the environment changes. To overcome this limitation, we formally define rule generation as learning a mapping $d : E \longrightarrow \mathcal{R}$, where E is a target evaluation and $\mathcal{R}$ the resulting rule vector. We then embed d within a closed-loop, AI-driven system in which rules are continuously refined in response to observed strategies $\mathcal{S} \in \mathbb{R}^{N_s \times D_s}$ and evaluation outcomes $E = f(\mathcal{R}, \mathcal{S})$. This dynamic paradigm ensures that $\mathcal{R}$ evolves adaptively with environmental feedback, granting agents genuine autonomy to reshape their own operational rules and remain robust in non-stationary, multi-agent settings.

Moreover, our framework jointly evolves rules, strategies, and evaluation metrics in a single feedback loop, blending procedural content generation with human and AI intelligence to achieve controlled, adaptive rule evolution. We further investigate real-time visualisation, platform integration, and interdisciplinary applications in social simulation and computational social science. This unified approach elucidates the reciprocal influences among rule creation, strategic exploration, and evaluation, and paves the way for dynamically redefining social norms through autonomous rule generation.

1.2 Research Questions

Our dissertation is structured around three central research questions: (1) How can AI be designed and trained to achieve controllable, autonomous rule generation? (2) How do AI-designed rules, agent strategies, and evaluation metrics mutually influence each other during the learning process? (3) How can the rule design framework be applied within interdisciplinary and social modelling contexts? Addressing these questions not only strengthens the theoretical foundations of automated rule generation but also enables the development of intelligent systems capable of continuously redefining social norms and maintaining an optimal challenge–skill balance in complex environments.

Achieving controllable, autonomous rule generation that minimises human biases is fundamental to advancing robust and generalizable AI systems. Traditional approaches rely heavily on human-defined rules or labelled datasets, which inherently embed subjective biases and inconsistencies. In contrast, our framework empowers AI to autonomously generate, evaluate, and evolve rules based solely on objective environmental feedback, thereby reducing reliance on human input. This process not only minimises biases but also enhances the adaptability and scalability of the system, as it can dynamically adjust to changing conditions while maintaining consistency and transparency in decision-making. Ultimately, by addressing the challenge of bias through autonomous rule generation, our work establishes a more reliable foundation for subsequent strategy exploration and evaluation, thus significantly contributing to the field of social rule generation.

Capturing dynamic interplay among rule, strategy, and evaluation within a unified platform is the cornerstone of our work. Traditional models often treat rule design, strategy formulation, and outcome evaluation as isolated processes, which can lead to suboptimal adaptations in complex, dynamic environments. By integrating these components into a single, cohesive framework, our approach facilitates a closed-loop feedback mechanism: rules influence agent strategies, which in turn affect game outcomes that are evaluated to iteratively refine the rules. This triadic interaction not only enables more nuanced and responsive rule evolution but also supports the emergence of robust social behaviours across diverse scenarios. Ultimately, this unified platform provides a comprehensive means to simulate and understand how adaptive rules can drive strategic co-evolution, thereby significantly advancing the development of intelligent, socially-aware AI systems.

Interdisciplinary social modelling for adaptive rule evolution bridges the gap between computational intelligence and social sciences. By integrating theories and methodologies from psychology, sociology, and behavioural economics into the SER/TRD framework, we can capture the complexity of human social interactions and norms in a dynamic environment. This interdisciplinary approach enables the system not only to autonomously generate and refine rules based on objective feedback but also to account for nuanced social phenomena such as trust, fairness, and cooperation. Consequently, our platform is capable of adapting rules in a manner that reflects real-world social dynamics, ultimately paving the way for AI systems that are both ethically robust and socially aware.

By addressing these challenges, AI systems could move beyond static rule paradigms and achieve dynamic, adaptive rule generation that is intrinsically aligned with the complexities of real-world social interactions. Overcoming these obstacles enhances the robustness of multi-agent systems, allowing them to continuously refine rules and strategies in response to evolving environments. This, in turn, promotes more intelligent and equitable decision making, facilitating deeper integration between computational models and social dynamics. Ultimately, such advancements pave the way for AI systems that not only optimise performance in diverse applications but also contribute to a richer, interdisciplinary understanding of social behaviour.

To achieve these objectives, we adopt a three-pronged approach. First, we establish a controllable rule generation framework within the SER paradigm, rigorously formalising the processes of rule creation, modification, and deletion. Second, we develop the Triadic Reciprocal Dynamics (TRD) system, a unified platform that integrates rule design, strategy evolution, and evaluation, thereby enabling real-time, multi-dimensional feedback. Third, we extend our framework through interdisciplinary social modelling by incorporating principles from psychology and sociology. This extension facilitates the use of both intrinsic and extrinsic reward signals to guide adaptive rule evolution, allowing our system to dynamically adjust its parameters and strategies based on environmental feedback. Together, these innovations yield an adaptive, robust platform for social rule generation that bridges the gap between artificial intelligence and the broader social sciences.

1.3 Research Objectives

There are three fundamental challenges that emerge in advancing adaptive rule generation within dynamic social environments:

- **Deconstruction of the Rule Generation Process.** It is essential to rigorously deconstruct the procedure into its constituent components—creation, deletion, and modification—to discern the core elements of rule generation. By formally defining a rule set as $\mathcal{R} = [r_{n,d}] \in \mathbb{R}^{N_r \times D_r}$, where N_r denotes the number of rules and D_r their dimensionality, the underlying structure that any AI-based rule designer must replicate is clarified. This challenge necessitates the development of computer-based techniques, such as deep learning and reinforcement learning, to autonomously generate rules without human biases. A precise formalisation of the rule generation process thereby serves as a robust theoretical foundation for the SER framework.

- **Platformisation, Visualisation, and Gamification of the Rule Generation System.** A unified, interactive platform is vital for facilitating interdisciplinary research and enabling human-in-the-loop interventions. The TRD system addresses this challenge by integrating rule generation, strategy exploration, and evaluation into a comprehensive platform featuring dynamic visualisation and gamification elements. Such a platform allows real time tracking of rule evolution and provides a multi-dimensional evaluation paradigm—encompassing metrics such as cooperation rate, individual income, and the Gini coefficient—while permitting domain experts to adjust rule parameters during the process. This integration supports a deeper understanding of complex behavioural dynamics and promotes practical applications across diverse fields.

- **Interdisciplinary Social Modelling for Adaptive Rule Evolution.** Extending the framework to capture real-world social phenomena is imperative. By leveraging the SER/TRD system for cross-domain social modelling, insights from social psychology and computational social science—such as Flow Theory and Self-Determination Theory—can be incorporated into the rule generation process. This challenge involves developing adaptive models that simulate and evaluate the interplay between extrinsic rewards derived from environmental feedback and intrinsic rewards that reflect an agent's engagement or flow state. Such interdisciplinary modelling not only enhances the understanding of emergent social norms but also optimises AI strategies across various applications.

Collectively, these challenges provide a comprehensive basis for a dynamic, adaptive rule generation system that is both theoretically robust and practically versatile, thereby paving the way for significant advancements in artificial intelligence and social modelling.

1.4 Main Contributions

Our research contributions are organised around three principal themes, which together address the challenge of developing adaptive, autonomous rule generation systems that dynamically shape social interactions in multi-agent environments.

Rule Generation Framework

- A structured **Strategy-Evaluation-Rule (SER)** framework is established, wherein rules are defined as a set of principles in a digital environment. This framework systematically organises rule creation, modification, and deletion processes.

- Two digital environments, *Maze Run* and *Trust Evolution*, are developed as testbeds that generate data during training, eliminating the need for pre-existing datasets and enabling a controlled yet dynamic rule generation process.

- The framework addresses critical challenges—namely, the lack of pre-labelled data, the translation of abstract rule vectors into actionable rules, and the impact of impractical design requirements—thus ensuring the controllability of the rule generation task.

Dynamic Social Modelling via Triadic Reciprocal Dynamics

- The **Triadic Reciprocal Dynamics (TRD)** system integrates three core components: a neural network-based rule designer, a game environment simulating interactions among human participants, AI agents, and fixed-strategy NPCs, and a rule evaluator that predicts social metrics such as cooperation rate and individual income.

- This platform captures the dynamic interplay among rule creation, strategy evolution, and evaluation, forming a closed-loop feedback system where evaluation outcomes guide iterative rule refinement.

- The system advances a multi-dimensional evaluation paradigm, enabling comprehensive assessment of both the generated rules and the evolving strategies of participating agents.

Flow-Centric Rule Design

- The SER/TRD framework is extended by incorporating principles from Flow Theory to generate adaptive rules that align task difficulty with agent ability, promoting optimal engagement and performance.
- A dual-reward mechanism is introduced whereby extrinsic rewards, derived from environmental feedback, are combined with intrinsic rewards based on an agent's flow state; this hybrid approach drives both strategic exploration and rule optimisation.
- Real-time flow visualisation and dynamic difficulty modulation are integrated into the platform, facilitating interactive rule tuning and enabling interdisciplinary research that spans AI, psychology, and social modelling.

1.5 Thesis Structure

This thesis is structured to present and evaluate key advances in adaptive rule generation and social modelling for dynamic environments. Figure 1.2 maps the proposed rule generation system onto its corresponding chapters, and Figure 1.3 outlines the overall roadmap, summarising the focus of each chapter.

In Chapter 1, the introductory section establishes the motivation, outlines the research questions and objectives, and delineates the main contributions of this work, which span three interrelated projects: Rules for Expectation, Triadic Reciprocal Dynamics, and Flow-Centric Rule Design.

Chapter 2 provides a comprehensive literature review, discussing existing approaches in rule generation, reinforcement learning, and interdisciplinary social modelling.

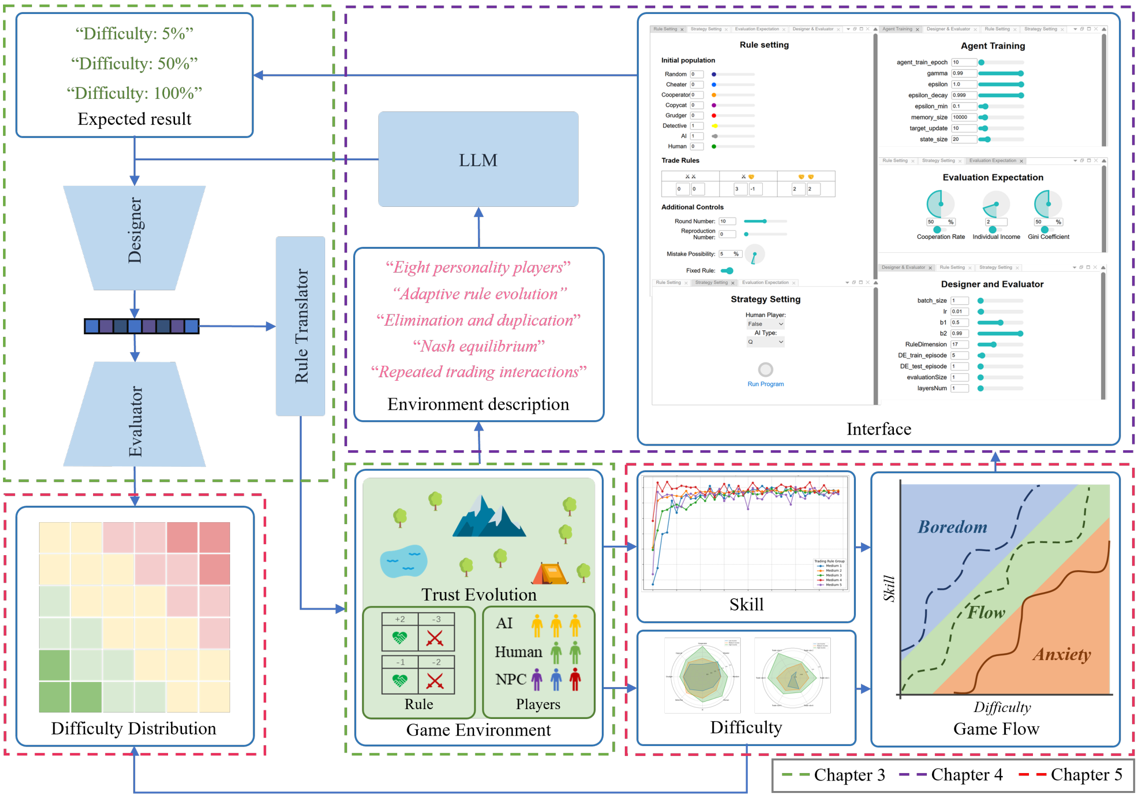


Figure 1.2: Overview of the chapter-to-framework mapping. The green dashed box highlights the SER framework introduced in Chapter 3; the purple dashed box encloses the TRD platform presented in Chapter 4; and the red dashed box denotes the dynamic difficulty flow system developed in Chapter 5.

In Chapter 3, the work titled *Rules for Expectation: Learning to Generate Rules via Social Environment Modelling* is introduced, establishing a controllable framework for rule generation by leveraging digital environments such as Maze Run and Trust Evolution.

Chapter 4 presents *Triadic Reciprocal Dynamics: The AI Framework for Social Rule Evolving*, which focuses on capturing the dynamic interplay among rule creation, strategy evolution, and evaluation through a unified platform with multidimensional evaluation paradigms.

In Chapter 5, *Flow-Centric Rule Design: Evolving Rules for Superior AI Player Immersion and Engagement* is detailed, extending the framework to incorporate adaptive mechanisms inspired by Flow Theory to optimise AI performance and engagement.

Finally, Chapter 6 concludes the thesis by summarising the key contributions, discussing limitations, and outlining potential directions for future research.

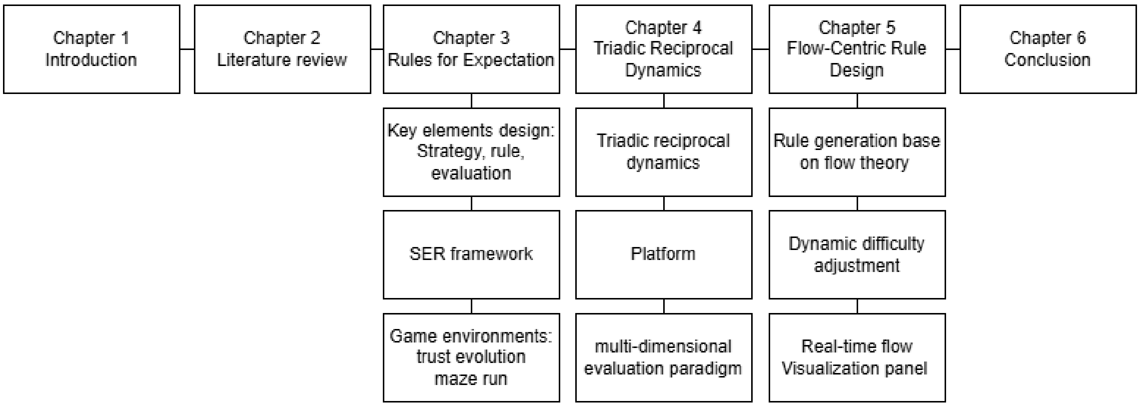


Figure 1.3: Thesis roadmap: each column represents a chapter and its core focus.

CHAPTER 2

Literature Review

This chapter provides a comprehensive review of the literature that underlies the work. It begins with an examination of generative modelling techniques—covering deep neural networks, generative adversarial networks, variational autoencoders, diffusion models, and transformer-based architectures—and their applications in synthetic data and content creation. The discussion then turns to large language models and rule generation frameworks that enable automatic inference and synthesis of game logic and procedural rules. Following this, reinforcement learning methods are surveyed, including value- and policy-based algorithms, multi-agent paradigms, strategy exploration techniques, and artificial life simulations that inform agent behaviour. The chapter proceeds to outline procedural content generation and automated game design, evaluate game engines and social modelling approaches, and present multi-dimensional evaluation metrics for generative, RL, and rule-based systems. Subsequently, multi-agent interaction and social norms are explored through Bandura' s Triadic Reciprocal Determinism, self-determination theory, and flow theory, highlighting their relevance to Trust Evolution. Finally, platformisation and human-in-the-loop methods in interactive experimentation and human–AI collaboration are examined, concluding with an overview of emerging intelligence paradigms—

artificial intelligence, explainable AI, extended intelligence, and hybrid intelligence.

2.1 Generative Models

2.1.1 Deep Neural Networks (DNNs)

Deep Neural Networks (DNNs) have fundamentally transformed machine learning by enabling automatic feature extraction through multi-layered architectures. Building on traditional neural networks, early breakthroughs—such as the advent of deep convolutional neural networks (CNNs) and hierarchical feature representations exemplified by models like LeNet, AlexNet, and ResNet—highlighted the importance of network depth for state-of-the-art performance on large-scale tasks [27]. Subsequent innovations include evolutionary synthesis approaches, which optimise network architectures over successive generations by preserving critical synaptic clusters [28, 29], and Dense Convolutional Networks (DenseNets), which enhance training efficiency and gradient flow through dense layer connectivity [30]. Comprehensive reviews further underscore the dual trends of increasing model depth and architectural diversity—from CNNs and RNNs to hybrid networks—driven by advances in algorithmic frameworks and a deeper understanding of network dynamics [31, 32]. Innovations in parameter selection, evolutionary algorithms, and novel training methodologies have effectively addressed challenges such as vanishing gradients and overfitting, consistently delivering improved performance on complex tasks [33]. These milestones illustrate the dynamic evolution of DNNs and their pivotal role in shaping modern, interpretable systems across diverse applications ranging from image recognition to scientific discovery.

Generative models constitute a broad class of machine learning techniques designed to learn the underlying probability distribution of data and subsequently generate new samples that mirror the training distribution [34, 35]. Unlike discriminative models that focus solely on predicting labels, these models leverage probabilistic frameworks such as Bayesian inference and likelihood maximisation to capture complex dependencies in high-dimensional data. This conceptual foundation has enabled their widespread application across diverse domains—including image synthesis, text generation, and graph structure generation [35]. Over the past decade, significant developments have emerged with the advent of Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and autoregressive models [34, 35]. In particular, GANs employ a dual-network architecture in which a generator and a discriminator engage in a minimax game to produce high-fidelity samples while enhancing the discriminator' s ability to distinguish real from synthetic data [34]. Recent adaptations have further extended these frameworks to non-Euclidean data domains, such as graphs, as detailed in systematic reviews by Guo and Zhao [35]. These advancements illustrate both the conceptual evolution and technical innovations driving the field, underscoring a dynamic research paradigm that continues to expand the applications and robustness of generative models.

The evolution of generative models stands as a defining trend in deep learning research, with significant advances in three main paradigms: Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Diffusion Models. VAEs emerged from the need to ground generative modelling in a probabilistic framework by employing variational inference to approximate intractable posterior distributions over latent variables, thereby enabling principled likelihood-based objectives and a favourable latent space for interpolation and downstream tasks [36]. Despite early challenges in inference, subsequent refinements and hybrid approaches have successfully combined the interpretability of likelihood-based methods with the creative potential of adversarial techniques [37, 38].

In contrast, GANs—introduced by Goodfellow et al. [39]—redefine the generative process as a two-player minimax game, rapidly gaining prominence for producing sharp, high-fidelity images. Architectural innovations such as residual GANs [40], conditional frameworks, and models like Flow-GAN [38, 41] have broadened their applicability to domains ranging from topic modelling [42] to biomedical image augmentation [43], while addressing issues of convergence, sample diversity, and mode coverage. More recently, diffusion models have emerged as a breakthrough approach, utilising score matching and stochastic differential equations to iteratively reverse a data corruption process, which affords refined control over generation and results in high-quality outputs across image, text, and audio synthesis [44]. Although these models typically entail higher computational overhead during sampling, recent efforts have focused on accelerating this process without compromising their stable learning dynamics [45]. The progression from VAEs to GANs and diffusion models illustrates an ongoing interplay between theoretical depth and empirical performance, driving the field toward increasingly robust and versatile generative modelling applications.

Diffusion models are generative architectures that progressively corrupt data with noise and learn to reverse this process, enabling high-fidelity sampling from complex distributions. Ho *et al.* first demonstrated that denoising diffusion probabilistic models (DDPM) can match or exceed GANs in image fidelity by learning a sequence of reverse Gaussian transitions to iteratively remove noise [45]. Song and Ermon then reframed these approaches as score-based generative models via stochastic differential equations, unifying discrete- and continuous-time denoising under a principled mathematical framework and improving sampling diversity [44]. To handle symbolic and categorical data, Austin *et al.* introduced structured diffusion kernels for discrete sequences, showing that diffusion can generate high-quality text and other non-continuous data [46]. In the realm of procedural content generation, Dai et al. (2024) introduced a diffusion-based generative model that, from a single human-designed level, learns dense token-semantic representations and employs a localised latent denoising network to produce tile-based game layouts exhibiting both high stylistic fidelity and structural diversity [47]. Li *et al.* (2022) introduced Diffusion-LM, a non-autoregressive language model that iteratively denoises Gaussian latent sequences into word embeddings, enabling complex, fine-grained control over generated text, such as syntactic structure, via a simple gradient-based algorithm, and demonstrating superior performance on six challenging controllable generation tasks [48]. Recent advances in diffusion-based generative modelling have begun to unlock new possibilities for structured rule synthesis.

Transformer-based generative models have become foundational in sequence modelling by leveraging self-attention to capture long-range dependencies and context. The original Transformer architecture introduced by Vaswani et al. [49] demonstrated that stacking multi-head self-attention layers enables efficient parallelisation and superior performance on machine translation tasks. Building on this, Radford et al. [50] showed that autoregressive pretraining of a Transformer decoder (GPT) on large text corpora yields coherent, high-quality text generation, effectively modelling complex linguistic patterns. Brown et al. [51] scaled this approach to 175 billion parameters (GPT-3), further improving few-shot generation capabilities across diverse domains. Parallel advances in encoder–decoder architectures, such as T5 [52], unified text-to-text tasks under a single Transformer framework and showed remarkable adaptability when fine-tuned on downstream tasks, including code and rule synthesis. In the domain of program and rule generation, Yin and Neubig [53] adapted sequence-to-sequence models with syntax-aware decoding to generate well-formed code from natural language specifications, illustrating Transformers' potential for structured content creation. Transformer-based approaches have been applied to procedural content generation—e.g. transformers conditioned on level representations produce coherent game layouts and rule sets that maintain logical consistency and playability without extensive manual crafting. These works collectively highlight the versatility and power of Transformer architectures for generating complex, structured sequences, laying the groundwork for their application in adaptive, flow-centric rule generation systems.

2.1.2 LLM

The evolution of large language models (LLMs) has been marked by a series of technological breakthroughs that have fundamentally transformed sequential data modelling. Early approaches employed Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which made significant strides in processing language sequences despite challenges like vanishing gradients and limited long-term dependency modelling [54, 55]. A seminal turning point occurred with the introduction of the attention mechanism in the Transformer architecture, which dispensed with recurrence entirely to enable parallel processing and more efficient modelling of long-range dependencies [49]. Building on this framework, pre-trained models such as BERT leveraged bidirectional context to achieve state-of-the-art performance across a variety of NLP tasks [56], while autoregressive models like GPT demonstrated few-shot learning capabilities by pre-training on massive corpora [51]. Recent research continues to refine these architectures by enhancing efficiency, robustness, and adaptability through hybrid approaches that integrate semantic search and other improvements, with comprehensive reviews elucidating trade-offs between classic RNN-based methods and modern Transformer-based models in terms of accuracy, contextual coherence, convergence speed, and interpretability. Collectively, these developments illustrate a robust trajectory from early RNN-based sequence models to the contemporary attention-driven paradigms that underpin today' s LLMs, thereby laying a solid foundation for a wide range of applications from text classification to few-shot learning.

Advances in large language models have enabled the automatic inference and synthesis of formal rules and programs from high-level specifications. OpenAI' s Codex demonstrated that GPT-3.5 can translate natural language prompts into executable code snippets, effectively generating rule logic for diverse tasks [57]. Chain-of-thought prompting further enhances LLMs' step-by-step reasoning, allowing them to decompose complex rule inference problems into manageable subproblems [58]. Self-supervised instruction tuning (Self-Instruct) empowers models to iteratively refine their own generated rules by bootstrapping from curated examples, reducing reliance on human-crafted datasets [59]. Hybrid frameworks such as Program-Aided Language Models (PAL) integrate symbolic solvers with LLMs to validate and optimise generated rule sets in domains requiring strict correctness guarantees [60]. Chengpeng Hu *et al.* proposed an LLM-based framework that, given high-level video game descriptions, simultaneously generates both procedural rules and corresponding level layouts, effectively linking designer intent to concrete environment parameters [61]. These advances highlight LLMs' potential for dynamic, context-aware rule generation, overcoming traditional methods' rigidity and lack of adaptive feedback.

2.1.3 Rule Generation

Rules, defined as explicit or implicit directives that regulate conduct or outline procedural blueprints within a given domain, have long been pivotal across disciplines—from game design and machinery operation to societal law [62–64]. In academic research, rules have been studied both as static constructs and dynamic entities that evolve, serving as mechanisms for causality, fairness, and emergent behaviour [65,66]. Early theoretical frameworks, such as Hatakeyama and Hashimoto' s concept of "Minimum Nomic" [67], laid the foundation for understanding rule evolution as a process akin to natural selection, while investigations in card games by Janssen [68] highlighted how minor modifications can drastically shift strategic interactions.

In the same time, computational approaches have transitioned from hand-crafted rule specifications to algorithmic rule induction; for example, Flynn [69] tackled the inverse problem of deducing game rules from observed play, and Khalifa et al. [70] proposed benchmark frameworks for general video game rule generation emphasizing scalability and reusability. Moreover, Togelius and Schmidhuber' s experiments in automatic game design illustrate that rule generation can be an intrinsic part of the creative process, evolving dynamically via fitness functions based on entertainment and curiosity theories [71]. Formal approaches have also advanced the field; for instance, a comprehensive formal specification of a board wargame [72] and models elucidating the interplay between rules, gameplay, and narrative layers in video games [73, 74] demonstrate how rules serve as connective tissue linking abstract mechanics to user experience. These developments underscore the evolution of rule research—from foundational definitions as mechanisms of causality and policy to advanced computational techniques that drive interactive, dynamic systems in both traditional and digital environments.

Prior research on rule generation was primarily concentrated in fields such as control systems, data quality management, clinical decision support, and autonomous systems, where it focused on three interrelated dimensions: data-free rule generation, automated rule creation/modification/deletion, and controllability. Data-free approaches leverage simulation-based techniques to derive rules directly from intrinsic system behaviours—incorporating metrics for readability and simplicity to enhance interpretability without reliance on annotated data [75, 76]. In parallel, automated rule generation has advanced notably in adaptive control environments, where self-organizing fuzzy controllers dynamically create and refine rule sets to reduce expert dependency [77, 78], and similar techniques have been applied in data quality assessment and association rule mining to efficiently update rule bases in response to evolving datasets [79–81]. Robust controllability mechanisms, crucial for maintaining interpretability and system stability, are achieved by embedding human-understandable constraints into the rule generation process—ensuring that automated modifications remain manageable, as seen in applications ranging from automated data cleaning to ontology learning [82–84]. These developments illustrate a dynamic research landscape that continuously seeks to balance automation with human interpretability and control.

2.2 Reinforcement Learning

2.2.1 Reinforcement Learning Algorithms

Reinforcement Learning (RL) is a foundational framework in machine learning wherein an agent learns to make sequential decisions by interacting with an environment, typically modelled as a Markov Decision Process (MDP) that defines states, actions, transition probabilities, rewards, and a discount factor [85]. Early value-based methods such as Q-learning and SARSA established the basics of RL through iterative reward-based updates [85, 86], paving the way for deep reinforcement learning approaches like the Deep Q-Network (DQN) that integrate deep neural networks to handle high-dimensional sensory inputs and achieve human-level performance in complex tasks [4]. Complementing these, policy-based methods initiated with algorithms have evolved into hybrid Actor-Critic architectures, which combine policy optimisation with value estimation to reduce variance and improve convergence [86,87]. Further advancements include asynchronous methods such as A3C [88] and robust policy optimisation techniques like Trust Region Policy optimisation (TRPO) and Proximal Policy optimisation (PPO), which employ trust-region constraints and clipped probability ratios to stabilise training in high-dimensional environments [16]. This progression—from classical MDP formulations through value- and policy-based methods to scalable deep RL algorithms—demonstrates a dynamic interplay between theoretical advances and empirical successes in applications ranging from robotic control to financial trading [89], inspiring ongoing research toward more efficient, safe, and explainable reinforcement learning models [16].

Reinforcement learning has expanded beyond its classical Markov Decision Process formulation into a diverse array of paradigms that extend its theoretical foundations and practical applications. Multi-Agent RL (MARL) addresses the challenges of interacting agents in non-stationary environments, as demonstrated in wireless network control [90] and reviewed by Yin et al. [91] and Canese et al. [92]. Inverse RL (IRL) shifts the focus from learning policies through trial-and-error to inferring latent reward structures from expert demonstrations, with Bayesian formulations and adversarial techniques enhancing robustness and interpretability [93–95]. Meta RL, exemplified by the RL^2 framework [96], enables agents to rapidly adapt across diverse tasks, while Hierarchical RL (HRL) decomposes complex tasks into subtasks to manage temporal abstraction, as highlighted in curriculum learning [97] and robotics applications [98]. Complementing these approaches, Imitation Learning (IL) leverages expert demonstrations to guide agent behaviour, often serving as an initialisation strategy or a supplement to IRL [98]. Additionally, Distributional RL (DRL) enriches policy evaluation by modelling the full probability distribution over returns, thereby capturing uncertainty more effectively [99]. These paradigms illustrate the evolution of RL from foundational methods to sophisticated frameworks that integrate theoretical innovations with practical solutions for complex, dynamic real-world challenges.

The evolution of Q-learning and its deep variant, the Deep Q-Network (DQN), represents a significant milestone in reinforcement learning, particularly within the game industry. Q-learning, a model-free algorithm that iteratively updates Q-values to estimate an optimal action-value function, laid the foundation for reinforcement learning by operating on discrete representations and handcrafted features [100]. However, its limitations spurred the development of DQN, which integrates convolutional neural networks (CNNs) to approximate Q-values directly from high-dimensional sensory inputs, such as raw pixel data from video games, achieving human-level performance in classic Atari 2600 games [4]. Subsequent advancements, such as Double Q-Learning, addressed inherent issues like overestimation bias by decoupling action selection from evaluation during target updates [101], while further research has refined hyperparameters and adapted these core algorithms to handle temporal dependencies—exemplified by deep recurrent versions of Double Q-Learning in game environments [102] and reinforced by reviews of reinforcement learning applications in intelligent game settings [103]. Beyond gaming, the flexibility of Q-learning and DQN has been demonstrated in robotics, autonomous navigation, and control systems, as illustrated by tutorials on their implementation in simulated environments like the CartPole system [104]. These developments —from traditional table-based Q-learning to sophisticated DQN architectures and their subsequent enhancements—underscore a transformative journey that continues to expand the technological frontier of reinforcement learning in both research and real-world applications.

The evolution of policy optimisation methods in reinforcement learning has advanced the development of agents capable of complex decision-making. Actor-Critic methods—integrating value function estimation with direct policy optimisation—laid the groundwork for handling continuous control tasks in high-dimensional action spaces [105]. Building on these foundations, asynchronous approaches such as A3C leverage parallel agents interacting with separate instances of the environment to enhance training efficiency, exploration, and stability, particularly in challenging game environments [106]. Further refinements in policy optimisation have emerged through algorithms like TRPO and PPO. TRPO employs trust region constraints to limit the magnitude of policy updates, ensuring that new policies remain close to the current ones—a strategy successfully applied in robotics and equilibrium learning in Markov potential games [107, 108]. PPO simplifies these principles using surrogate objective functions and clipping mechanisms, resulting in robust performance with reduced computational overhead [109, 110]. These advancements have significantly impacted the game industry, where deep reinforcement learning algorithms based on actor-critic architectures enable agents to develop complex strategies from high-dimensional sensory inputs with minimal human intervention [103, 111]. The evolution from standard Actor-Critic methods to asynchronous and trust-region-based algorithms illustrates a trajectory of innovation that enhances algorithmic stability and efficiency, thereby broadening the scope of practical applications in dynamic and interactive domains.

2.2.2 Strategy Exploration

Strategy exploration is the process of using computational methods to autonomously discover, evaluate, and refine optimal decision-making strategies in complex environments. The application of reinforcement learning (RL) to strategy exploration has emerged as a powerful paradigm for understanding and developing optimal decision-making behaviours in dynamic environments, particularly within game-based settings. RL methods enable agents to autonomously discover and refine strategies by interacting with complex environments, such as the Iterated Prisoner' s Dilemma, zero-sum stochastic games, and real-time strategy games, thereby providing insights into multifaceted strategic dynamics [112, 113].

Early work demonstrated that parameter tuning and algorithm configuration in temporal difference methods can influence the convergence of meta strategies, laying a foundation for robust strategy formation in repeated game scenarios [112]. Subsequent research has emphasised the importance of efficient exploration in environments with unknown rules, as exemplified by Martín and Sandholm [114], and has further advanced strategy discovery through multi-agent RL approaches. For instance, Wang et al. [115] introduced memory mechanisms in multi-agent settings to uncover sophisticated strategic behaviours such as memory-two bilateral reciprocity, while studies applying algorithms like Q-learning and SARSA in real-time strategy games have refined reward functions and iterative learning schemes to adapt strategies in adversarial settings [116]. Moreover, incorporating models of opponent learning, as explored by Hu et al. [117], has further enhanced the optimisation of strategies in competitive environments. These developments illustrate a dynamic evolution in RL-driven strategy exploration—from early algorithmic foundations to sophisticated multi-agent frameworks—that continues to expand our understanding of strategic interactions in both simulated and real-world domains.

2.2.3 Artificial Life

Artificial life—a field devoted to studying life-like phenomena through computational and synthetic approaches—has evolved into a multifaceted research domain that integrates biological inspiration with advanced computational techniques. At its core, artificial life seeks to simulate, reproduce, and understand complex adaptive systems by leveraging models that range from cellular automata (CA) and self-replication systems to genetic algorithms, agent-based modelling, and artificial neural networks [118]. CA models, which operate on simple, locally defined rules to generate intricate global behaviours, have provided fundamental insights into pattern formation and adaptive interactions, as well as applications such as synthetic musical semiosis [119,120]. The study of self-replication, grounded in von Neumann' s pioneering work, has further advanced our theoretical understanding of replication and evolutionary dynamics in silico [121, 122]. Complementing these approaches, the integration of genetic algorithms, agent-based models, and artificial neural networks has propelled artificial life research by enabling the simulation of autonomous interactions that mirror natural social and ecological behaviours, with applications extending to urban planning, resource management, and governance [123]. These developments illustrate how the convergence of diverse computational tools and methodologies has broadened our understanding of life-like phenomena and spurred innovations that bridge the gap between natural and artificial systems.

2.3 Procedural Content Generation

Procedural Content Generation (PCG) is a versatile framework for the automated creation of digital content, spanning gaming, virtual environments, film, architecture, and urban planning, by leveraging diverse algorithmic paradigms. Early PCG methods relied on noise-based and grammar-driven techniques, such as Perlin noise and context-free or stochastic grammars, to generate textures, terrains, and layouts that capture natural variations and embed design rules [124, 125]. As the field matured, search algorithms, including tree search, heuristic strategies, and multi-objective fitness functions, enabled systematic exploration of large content spaces to assess playability, coherence, novelty, and aesthetics [125–127]. Evolutionary and genetic algorithms further advanced PCG by iteratively refining candidate solutions through selection, recombination, and mutation, proving effective in generating game levels and maps as well as novel architectural and urban planning designs [124, 128–130]. More recently, hybrid approaches that combine evolutionary strategies with local search methods, alongside Experience-Driven PCG frameworks that integrate player experience models, have enhanced convergence speed, solution quality, and personalisation in interactive environments [131–134]. These developments illustrate the evolution of PCG from foundational techniques to sophisticated, context-sensitive systems that complement and enhance human design processes [135].

2.3.1 Automated Game Design

Automated game design (AGD) is a dynamic subfield within game AI that synthesises game mechanics, aesthetics, and dynamics through computational methods. Early efforts, such as Smith et al.'s LUDOCORE [136], focused on rule-based systems that modelled and manipulated game logic for rapid prototyping. Over time, the scope of AGD expanded to incorporate evaluation metrics and creative strategy, as exemplified by Cook et al.'s ANGELINA system [137, 138], which employs probabilistic and evolutionary methods to generate and refine entire games. Subsequent developments have further integrated aesthetic and personalised dimensions into AGD, with systems like Cook's "Puck" [139] and Osborn et al.'s Modular Computational Critics [140] offering real-time evaluative feedback on game mechanics and balance. In parallel, higher-level procedural content generation techniques—such as vision-driven level design [141] and the generation of "juice" effects for enhanced visual and audio appeal [142]—have enriched game experiences by embedding aesthetic considerations into the design process. Finally, the approach of conceptual expansion, as proposed by Guzdial and Riedl [143], marks a shift from manual or crowd-sourced design toward autonomous machine learning techniques that abstract and recombine design elements from existing games. These advancements illustrate the evolution of AGD from foundational rule-based systems to sophisticated, integrated frameworks that holistically address procedural content generation, aesthetic design, and automated evaluation.

2.3.2 Game Engines

Game engines have evolved from simple graphics renderers into comprehensive development ecosystems that serve diverse applications ranging from entertainment and virtual reality to serious simulations and robotics training. Foundational systems like Unity, Unreal Engine, Godot, Gazebo, and CoppeliaSim have been extensively evaluated for their technical capabilities—including high-fidelity real-time rendering, effective resource management [144], and robust simulation performance [145, 146] —which enable the creation of immersive and interactive experiences [147]. Early reviews and comparative studies have highlighted these engines' suitability for both gaming and serious applications, such as virtual simulation platforms for training remotely operated vehicle pilots [148] and industrial training environments [149, 150]. Additionally, the integration of game engines into mixed-reality applications further bridges digital content creation with real-world interaction [151]. In the indie development sphere, lightweight and open-source engines like Godot and CoppeliaSim are favoured for rapid prototyping and iteration, while independent publishing platforms such as itch.io have democratised game distribution and fostered a vibrant, community-driven ecosystem [152, 153]. These developments underscore that game engines not only underpin visually rich, interactive applications but also serve as critical components in simulation and training, with evaluation criteria such as rendering performance, cross-platform capability, efficiency, and integration flexibility driving ongoing innovation across both academic research and grassroots development.

2.3.3 Social Modelling

Social modelling has evolved into a multifaceted discipline that integrates quantitative, computational, and agent-based methods to understand and predict complex social phenomena. Early approaches relied on statistical modelling and descriptive social network analysis—employing techniques from regression and relational event models to exponential random graph models (ERGMs)—to quantify interactions and dependencies within social networks [154–156]. As the field matured, dynamic analyses emerged, with methodologies tracking network evolution over time to elucidate processes such as interhospital patient transfers and influence maximisation in public health campaigns [157, 158]. The advent of agent-based modelling (ABM) and multi-agent systems marked a paradigm shift from aggregate statistical methods to bottom-up simulations, enabling researchers to capture emergent macro-level patterns from individual interactions [159, 160]. These approaches have been particularly effective in domains such as urban planning, epidemiology, and economics, where they simulate competitive behaviours and intervention scenarios [161–163].

More recently, computational social science has emerged as a transdisciplinary field that leverages large-scale data from social media, sensor networks, and administrative records to integrate statistical, network, and agent-based techniques for addressing real-time societal challenges [164–166]. These developments underscore the evolving nature of social modelling—from foundational quantitative approaches to sophisticated, dynamic simulations—establishing it as a critical tool for understanding diverse social phenomena across multiple domains [167–169].

2.3.4 Multi-dimension Evaluation

Evaluation research is essential for assessing the performance of diverse components, such as generative models, reinforcement learning (RL) agents, game environments, and rule systems, across metrics including accuracy, cumulative reward, fairness, efficiency, robustness, and trust. In generative model research, evaluation focuses on quantitative measures like fidelity, consistency with human-designed standards, and classification accuracy when inferring outputs from data [170]. For RL, performance metrics such as cumulative reward, sample efficiency, and convergence stability serve as critical indicators of an agent' s ability to explore and optimise strategies in dynamic, often adversarial, environments [171]. Within game environments, evaluation parameters extend to gameplay quality, adaptive challenge balance, and robustness of decision-making processes, ensuring that agents can operate effectively under varied conditions [172]. In the domain of rule-based systems, evaluation examines the consistency, conformity, and fairness of generated rules against established benchmarks, thereby underpinning system reliability and decision support efficacy [70]. These evaluation frameworks not only advance our theoretical understanding but also drive practical improvements in complex, interactive systems.

2.4 Multi-Agent Interaction and Social Norms

2.4.1 Triadic Reciprocal Determinism

Triadic reciprocal determination is a fundamental concept in Bandura's social cognitive theory that posits a dynamic and bidirectional influence between personal factors (e.g., cognition and emotions), behaviour, and environmental conditions [173, 174]. Originally developed to explain human agency within psychological and social contexts, TRD has since been extended into technology-mediated social modelling. Advances in AI and big data analytics now enable researchers to simulate the complex interactions among cognition, behaviour, and environment. For instance, Zhu et al. [175] applied TRD to study parental attitudes and vaccination intentions during the COVID-19 pandemic, while Huo and Li [176] employed fuzzy-set qualitative comparative analysis to investigate factors influencing the continued use of knowledge payment platforms. Further, Bergman et al. [177] used TRD to analyse commuter mobility patterns, and research in organisational behaviour has integrated TRD to explore how decision-making and environmental dynamics jointly impact employee innovation [178]. These developments illustrate the evolution of TRD from a theoretical framework for understanding human behaviour to a versatile tool in AI-facilitated social modelling, enriching our insight into the interaction between individual predispositions, behaviour and technological and social environments.

2.4.2 Self-determination Theory

Self-determination theory (SDT) is a comprehensive framework for understanding human motivation, development, and well-being that posits individuals possess innate psychological needs for autonomy, competence, and relatedness [179, 180]. Initially formulated from a humanistic perspective to distinguish between autonomous motivation—where actions align with one' s core values—and controlled motivation driven by external pressures [179, 181], SDT has been rigorously validated across domains such as education, organisational studies, psychotherapy, and health interventions [179, 182]. More recently, its integration into computational frameworks —ranging from statistical methods and social network analysis to agent-based and multi-agent simulations—has enriched AI-driven social modelling by embedding motivational dynamics into virtual agents' decision-making processes [180]. By encoding principles of autonomous and controlled motivation, these interdisciplinary efforts enhance the interpretability and fidelity of simulations in urban planning, public health, and market analysis, ultimately fostering more authentic, goal-oriented, and adaptive human-like behaviour in complex systems [179, 180, 182].

2.4.3 Flow Theory

Flow theory, originally developed by Csíkszentmihályi, is a central construct for understanding optimal human experience, defined as a state of complete absorption, intrinsic motivation, and deep engagement in an activity [183]. Initially grounded in positive psychology, early work focused on internal conditions for flow, such as a balance between perceived challenges and personal skills, clear goals, and immediate feedback, which laid the foundation for its application in measuring user engagement and system usability in information systems [183, 184]. In educational contexts, flow has been shown to catalyse enhanced learning experiences, although challenges remain in its automatic detection and real-time assessment [185]. More recently, the integration of Flow Theory with artificial intelligence and social modelling has emerged as a promising research frontier; studies demonstrate that incorporating constructs of perceived competence and flow into task-oriented AI and agent-based simulations not only improves user satisfaction and engagement but also enhances the fidelity of social behaviour models [186–188]. A systematic review by Chong et al. [189] further underscores the evolution of Flow Theory from its roots in leisure and performance to its current applications in digital learning, AI interaction, and computer-mediated communication. These developments illustrate the versatility of Flow Theory as both a theoretical framework and a practical guide for designing interactive, socially intelligent systems in technology-enhanced environments.

Game flow has been redefined as an emergent property resulting from a well-balanced interplay of challenge, skill, clear goals, and immediate feedback that drives player immersion in digital gaming. Early empirical studies extended traditional flow theory by integrating it into models of online gaming behaviour—for example, Hsu and Lu [190] incorporated flow into technology acceptance frameworks, while Cheah et al. [191] identified immersion and flow as key predictors of sustained player involvement. Over time, game flow has evolved from a purely psychological construct into a fundamental design principle. Researchers have empirically linked optimised flow states with enhanced learning outcomes and reduced cognitive load in digital game-based learning environments [192]. More recently, the integration of artificial intelligence and social modelling has further advanced the field: AI-powered systems, such as game-based chatbots, dynamically adapt game difficulty, narrative elements, and feedback to sustain optimal flow states [193], while studies of social network dynamics in games like Team Fortress 2 illustrate how collaborative interactions influence collective flow [194]. These developments reflect a trajectory from foundational psychological theories to sophisticated, computationally driven frameworks that inform the design of more engaging, adaptive, and socially enriched game experiences.

2.5 Platformisation and Human in the loop

2.5.1 Interactive Experimentation and Simulation

Interactive Experimentation and Simulation has evolved into a multifaceted paradigm that bridges educational, social, and computational domains. Initially employed as educational tools to facilitate the transition from concrete to abstract understanding [195], early interactive simulations enabled learners to engage with complex concepts through hands-on experimentation. Over time, the integration of participatory frameworks enhanced collaborative learning in computer science [196], while in the social sciences, computer simulations emerged as pivotal tools for analyzing intricate social phenomena—ranging from social networks to cultural dynamics—thus transforming theoretical models into practical research applications [197, 198].

In the realm of computer science, advancements in agent-based modelling and sophisticated computational frameworks have further expanded the utility of simulations, as evidenced by their application in materials design and workflow management within environments like the Eclipse Integrated Computational Environment [199–201]. Moreover, the adoption of gamified and STEM-oriented simulations has enriched educational initiatives by providing immersive, interactive platforms that promote conceptual understanding and critical thinking [202–204]. These developments underscore a fundamental shift towards interactive, engaging, and contextually relevant methodologies that enhance both theoretical exploration and practical applications in addressing complex real-world challenges.

2.5.2 Human-AI Interaction

Human-AI Interaction (HAII) has evolved from early, rigid automated systems toward dynamic, co-adaptive models that integrate human expertise throughout the AI lifecycle. Early investigations, particularly in healthcare, demonstrated that the clinical impact of AI hinges on harmonising computational capabilities with human judgment [205, 206], thereby laying the groundwork for the human-in-the-loop (HITL) paradigm. Empirical studies, such as those by Maadi et al. [206], further revealed that complex domains like medicine and production planning benefit from HITL approaches, as purely machine learning-based methods often fail to capture nuanced decision-making. Building on these insights, frameworks such as COFI [207] and foundational models by Sreedharan [208] have formalised the cognitive processes mediating human–AI interactions. Concurrently, research has emphasised the importance of socio-technical factors—including trust, predictability, transparency, and explainability—in aligning AI behaviour with human expectations [209, 210]. Recent studies on feedback loops in tasks such as text summarisation [211] and clinical imaging [212] further illustrate that iterative human-AI collaboration enhances operational efficiency, model interpretability, and overall performance. These developments underscore a significant paradigm shift in HAII—from isolated, fully automated systems to designs that integrate human creativity and oversight, thereby redefining the design, deployment, and effectiveness of contemporary AI systems across diverse domains [213].

2.6 Intelligence Paradigms

2.6.1 AI

Artificial Intelligence (AI) has undergone a remarkable evolution from early rule-based and symbolic systems to sophisticated, data-driven models that now underpin diverse technological and societal applications. Foundational texts such as Nilsson' s Principles of Artificial Intelligence [214] and Russell and Norvig [215] established core paradigms—including symbolic reasoning, heuristic search, and agent-based frameworks—that initially sought to mimic human thought processes through explicit programming [216]. Over time, the focus shifted toward machine learning methods that learn from data, giving rise to supervised, unsupervised, and reinforcement learning paradigms [217, 218]. Neural networks, inspired by biological processes, evolved from early perceptrons to deep neural networks capable of modelling complex, non-linear relationships and extracting hierarchical features, thereby revolutionising tasks such as image and speech recognition, natural language processing, and game playing [219]. The advent of deep learning, bolstered by the availability of large datasets and powerful computational resources, has further driven innovations including convolutional neural networks, recurrent neural networks, and generative adversarial networks (GANs) [220]. Recent advancements have also integrated classical machine learning techniques with deep learning frameworks, resulting in hybrid models that leverage the strengths of both approaches. These developments underscore the dynamic evolution of AI from rule-based systems to flexible, robust, and scalable models that continuously expand the boundaries of data representation and pattern recognition across a wide range of application domains.

2.6.2 XAI

The evolution of Explainable Artificial Intelligence (XAI) represents a critical response to the growing complexity and opacity of modern AI models. At its core, XAI emerged from the need to render AI models interpretable and transparent, thereby enhancing trust and facilitating regulatory compliance. Early approaches predominantly relied on inherently interpretable methods, such as decision trees and rule-based models, whose structured design provided immediate clarity into model functioning [220]. Over time, more sophisticated techniques have been developed to elucidate the internal workings of complex, non-linear models. Methods such as Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) explicitly account for feature contributions through importance rankings and sensitivity analysis, while counterfactual explanations offer alternative scenarios that clarify decision boundaries [221]. Additionally, visualisation methods and feature inversion approaches have further demystified deep learning models by enabling stakeholders to scrutinise internal representations and decision pathways [222]. These advancements collectively strive to bridge the gap between high model performance and interpretability, fulfilling both technical and ethical imperatives in large-scale predictive systems.

2.6.3 XI

Extended Intelligence (XI) is an emerging paradigm that augments traditional AI by integrating human expertise with machine computation, fostering iterative human–machine collaboration and enhanced decision support. At its core, XI emphasises the fusion of interactive machine learning and active learning, where systems continually solicit human feedback to guide model refinement [223], a notion reinforced by Holzinger [224]. Concurrently, decision support and augmented analytics frameworks, such as those illustrated by Heart et al. [225], leverage real-time analytical models combined with human insight to dynamically present actionable information, thereby improving outcomes in complex domains like clinical decision-making. Moreover, emerging neuro-symbolic integration aims to merge neural pattern recognition with symbolic reasoning to provide more interpretable and context-aware support [226], while collaborative filtering and recommender systems—exemplified by Alonso [227]—utilise reinforcement learning and interactive techniques to adaptively tailor recommendations based on continuous human feedback. These developments underscore XI' s evolution from early interactive systems to comprehensive, context-aware frameworks that effectively amplify human decision-making and cognitive capabilities across diverse application domains.

2.6.4 HI

Hybrid Intelligence is an emerging paradigm that integrates human insight with machine computation to enhance decision-making and learning processes. It encompasses key technological components such as neuro-symbolic integration, human-in-the-loop machine learning, hybrid decision support systems, ensemble methods augmented by human expertise, and collaborative filtering enriched by human feedback. Neuro-symbolic integration combines the pattern recognition capabilities of neural networks with the interpretability of symbolic reasoning to bridge the gap between data-driven approaches and human-understandable models [228]. Concurrently, human-in-the-loop machine learning has advanced by continuously incorporating expert feedback to refine model parameters and guide error correction in high-stakes domains such as healthcare and finance [206]. Hybrid decision support systems merge algorithmic predictions with rule-based expert systems, resulting in improved accuracy and context-sensitive outcomes in applications like clinical error detection and humanitarian relief management [229, 230]. Additionally, ensemble methods that integrate human expertise optimise performance by aggregating and filtering predictions based on domain-specific knowledge, while collaborative filtering systems leverage iterative human feedback to dynamically adjust recommendations [231]. These advancements illustrate the evolution of Hybrid Intelligence from its foundational concepts to sophisticated, integrated systems that redefine human–

machine collaboration and set new standards for intelligent system design.

CHAPTER 3

Rules for Expectation: Learning to Generate Rules via Social Environment Modelling

Prologue

Chapter 3 introduces a comprehensive framework—Rule Generation Networks (RGN) —for automated rule design, evaluation, and evolution guided by controllable expectations. First, three foundational elements (rules, strategies, and evaluation) are refined and formalised to clarify their interrelationships in rule generation tasks. Next, the RGN architecture is presented, combining generative neural models for rule synthesis with reinforcement learning agents for rule validation. Two custom game environments, Maze Run and Trust Evolution, are developed to demonstrate rule execution and assess generated rules under diverse social and strategic scenarios. Finally, a controllability metric is defined to measure and guide rule evolution, ensuring that the generated rules align with target outcomes. This chapter thus establishes the methodological basis for learning to generate game rules through social environment modelling.

The evolution of natural life is guided by a perpetually adaptive set of rules, encompassing natural laws, human policies, and game mechanics. Automated game design, through the creation of simulated environments populated by AI agents, embodies these rules, aligning with the objectives of artificial life research that seeks to replicate the dynamics of biological life through computational models. This chapter presents a comprehensive framework, the Rule Generation Networks (RGN), devised for automated rule design, evaluation, and evolution in line with controllable expectations. We refine and formalise three cardinal elements - rules, strategies, and evaluation - to elucidate the intricate relationships inherent in rule generation tasks. The RGN integrates generative neural networks for rule design and a suite of reinforcement learning models for rule evaluation. To exemplify rule evolution and adaptation across varying environments, we introduce a controllability metric to gauge game dynamics and evolve the rule designer accordingly. Furthermore, we develop two game environments, Maze Run and Trust Evolution, modelling human exploration and societal trade dynamics, to gamify and evaluate the generated rules.

3.1 Introduction

In diverse contexts, 'rules' are delineated as explicit or implicit directives that regulate conduct or outline a procedural blueprint within a specific activity domain. This scope can range from the operational guidelines governing games [62], to the functional principles directing machinery operation [63], and extend to societal laws that influence our collective behaviour [64]. Rules are typically manifested through alterations in values or the instantiation and annihilation of objects [65]. Carefully constructed rules can nurture equitable environments, fostering cooperation and trust, whereas inefficient ones can undermine societal productivity [66]. The task of rule generation is ubiquitous, featuring prominently in areas such as game development [232], rulemaking processes [233], and legislative procedures [234]. Within the scope of game creation, rule design is considered one of the six core elements [235]. Previous research related to rules in the field of machine learning has primarily focused on rule-based learning [236]. Contrary to formulating new rules, the majority of AI-related research prioritises training models to address specific problems within the constraints of established rules, with applications such as game map gen-

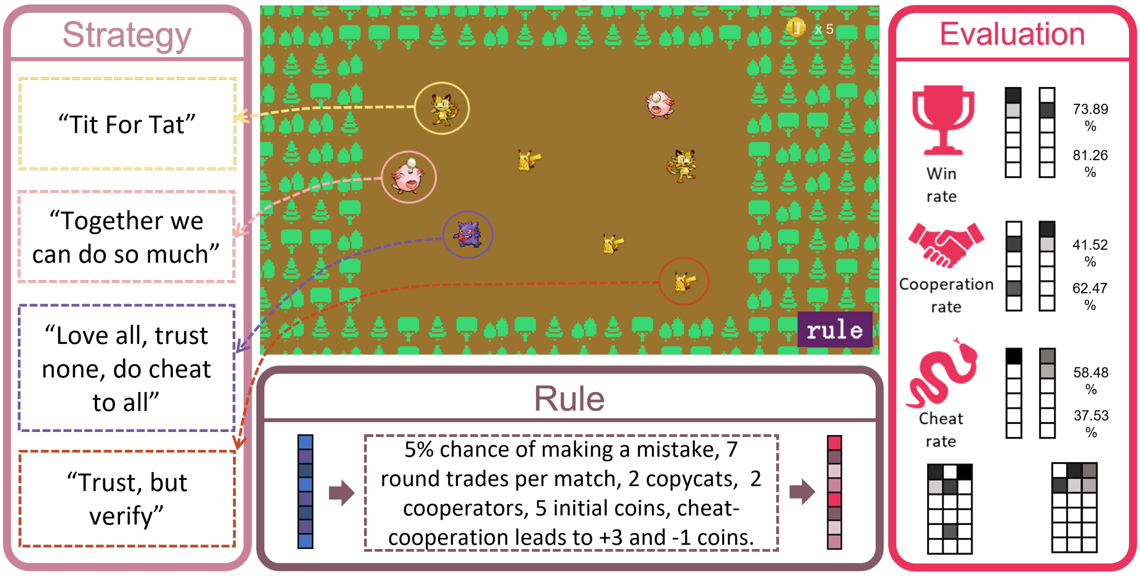


Figure 3.1: Given a conceptual description. Linguistic form rules are translated into a set of rule vectors and implemented in a digital game environment. Players develop strategies based on the current rules and their experience, and the rules are evaluated within the game environment.

eration [237] and elucidation of elementary reactions in chemical kinetics [238]. In well-structured games that accurately reflect real-world scenarios, the potential of automated rule design extends to realms beyond mere gameplay, notably in the fields of medical and electrical engineering. Examples include the automated design of personalised cancer treatment protocols [239] and efficient cooling power modules [240]. Automated rule design can enhance the adaptability of products, increase the efficiency of design processes, and optimise design parameters, materials, and configurations.

Artificial life constitutes the study that explores systems analogous to natural life and evolutionary processes, employing simulations via computer models, robotics, and biochemistry [241], [242], [243]. This discipline has investigated living systems through a synthetic approach, essentially constructing life to gain a deeper understanding of it [244]. Examples of such endeavours include cellular automata [245], machine aquariums [246], and neural MMOs [247]. These projects strive to emulate the evolution of life, aquarium systems, and societal resource changes, thereby augmenting our comprehension of the inherent rules or patterns governing our world.

The metaverse has gained significant interest in industry and academia as a research and technological exploration of the immersive future of the internet. Especially in the gaming domain, it allows users to create avatars and engage in various activities within online virtual worlds, including social interactions, customised environments, and virtual economies [248], [249]. The metaverse encompasses four significant domains: content creation, access and social connectedness, identity and representation, and assessment, validation, and user research [250]. The sustenance of metaverse user experiences depends on two aspects: content experience and content creation. Gamification bridges the gap between content experience and creation, facilitating the construction of virtual worlds by developers and granting player access [251].

Procedural content generation (PCG) represents an algorithmic method for creating elements automatically, leveraging a confluence of human-generated assets, computer-mediated randomness, and computational processing power [252]. It has found extensive application in game design, such as *No Man's Sky* [253], *Minecraft* [254], and *RimWorld* [255]. Within the realm of the gaming industry, a multitude of applications of procedural content generation (PCG) can be categorised as "constructive" techniques, sequentially employing grammar or noise-based algorithms to generate content devoid of evaluation. It also enhances the diversity of game experiences and mitigates the repetitive workload typically associated with design tasks [256]. However, the efficacy of PCG is significantly contingent upon the design and execution of sophisticated algorithms, which often necessitate substantial effort to design and evaluate. While it is common to examine existing game content for inspiration, machine learning methods have far less commonly been used to extract data from existing game content in order to create more content.

Game generation lies at the intersection of a multitude of creative domains, from art and music to rule systems and architecture [65]. Distinct from conventional AI, artificial general intelligence (AGI) is premised on the idea that machines could potentially mimic human cognitive processes in the future [257]. The establishment of General Video Game Artificial Intelligence (GVGAI) was motivated by a desire to steer AI researchers away from an over-reliance on specific tasks or algorithms in game engineering [258]. The use of generative models to construct not only game entities such as maps, characters, audio, and level systems, but also game rules, demonstrates significant potential within the context of GAGAI. Generative models play a vital role in unsupervised learning, offering an efficient means to analyze and comprehend unlabeled data [259]. These models, through learning from the data, develop an understanding of the internal probabilistic distribution necessary for content generation [260]. Significant improvements have been made in generative models such as Variational Autoencoders (VAE) [261], [262], Generative Adversarial Networks (GAN) [263], [264], and flow-based models [265], [266]. Despite these strides, a preponderance of research on generative tasks remains concentrated within the domains of computer vision and natural language processing.

Reinforcement learning (RL) is a mathematical framework for experience-driven autonomous learning [267]. It is designed to learn decision-making and has been employed to address challenges posed by Atari games [268]. Additionally, Multi-Agent Reinforcement Learning (MARL) represents a subfield of reinforcement learning. Multi-agent reinforcement learning concentrates on studying the behaviours of multiple learning agents co-existing in a shared environment [269]. Deep learning facilitated the scalability of reinforcement learning (RL) to address decision-making challenges that were previously deemed intractable, specifically in settings characterised by high-dimensional state and action spaces. Moreover, popular algorithms within deep RL, such as the deep Q-network (DQN) and trust region policy optimisation (TRPO), have garnered extensive utilisation in the realm of game design.

This project aims to establish a framework for rule generation, evaluation, and evolution. To achieve this, two digital environments, Maze Run and Trust Evolution, have been developed. We outline a series of rules that could be translated and illustrated within these environments. Both environments serve as games for AI, non-player characters (NPCs), and humans. We also distinguished the rule generation task in three aspects: 1. There is no pre-existing dataset for model training; all data are generated and collected by the environment during training. 2. The generated rule vector must be translated into rules that the environment can comprehend. 3. Rule design requirements might be impractical and can affect the model's performance. In response to these challenges, we proposed our rule generation framework and summarised our contributions as follows:

- Three core elements, including rule, strategies, and evaluation, are refined and symbolised to clarify relationships inherent in the automated rule generation task. These elements serve as a structured framework for organizing the rule generation process, and they facilitate a deeper understanding of the different components within the RGN framework. Furthermore, this analytical approach has helped us identify three significant challenges associated with rule generation: no dataset, rule translation, and unreasonable requirements, and enlightens us to introduce the controllability for the system evaluation.

- A rule generation framework is proposed based on generative models, digital environments, and reinforcement learning models. This framework integrates neural networks with automated game design and introduces controllability for both rule designers and game environment evaluation. This framework is initialised with default rules and accepts expected results as input. It evaluates the generated rules based on agents' strategies and refines the rule design process by comparing the expectations with the evaluation outcomes.

- Two digital environments, Maze Run and Trust Evolution, are established using Python and Unity as platforms for the demonstration of automated game design. Translators are employed to connect the generated rules to the games. These environments showcase the multi-platform adaptability of the proposed framework, provide opportunities for human participation in rule design, and effectively incorporate both cooperative and competitive social modes, which are vital for game rule evolution.

The remainder of the chapter is structured as follows. Section 2 reviews existing methods for generative models, reinforcement learning, automated game design, and procedural content generation. In Section 3, we present details for the proposed methodology, including environment creation, RGN framework, rule implementation and translation. After that, we demonstrate details about our digital environment and the experimental results in Section 4. Finally, we place important conclusions and discuss possible future works in Section 5.

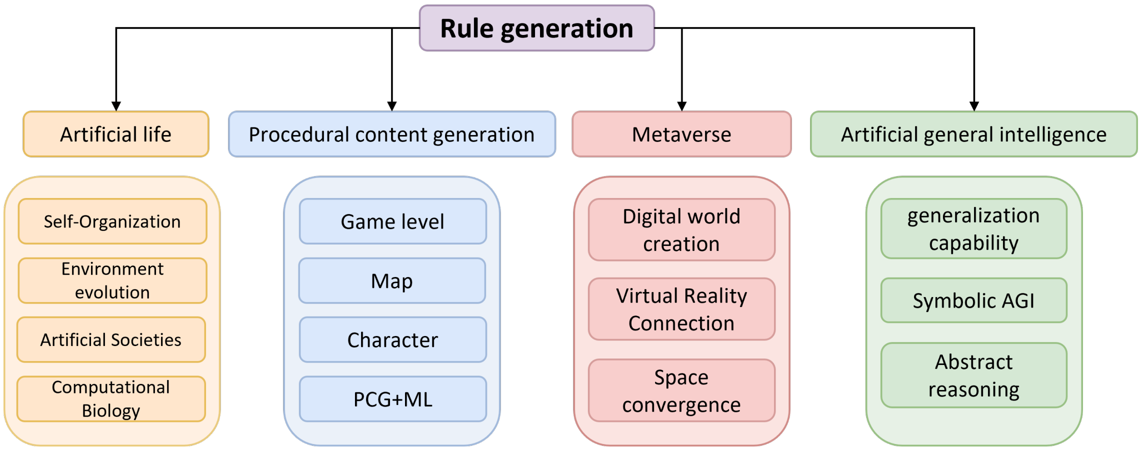


Figure 3.2: Overview of the relationship among rule generation tasks, artificial life, procedural content generation, the concept of the metaverse, and artificial general intelligence. Offering a comprehensive understanding of their interconnectivity within the realm of advanced computational studies.

3.2 Related work

3.2.1 Related tasks and Applications

The concept of Artificial Life (ALife) was first introduced by Langton in 1989, described as *life made by man rather than by nature* [241]. It is occasionally considered synonymous with open-ended skill learning [270]. Present ALife research spans across 14 themes, including computational biology, artificial societies, and adaptation ecology [244], [271]. Supramaniam *et al.* have primarily focused on the microfluidics method, honing in on the molecular and cellular biology domain [272]. Environmental task-driven approaches have also flourished in ALife, often achieved through multi-agent reinforcement learning (RL) [273]. A notable contribution in social modelling is the Neural MMO, a large-scale game environment designed for RL [247]. This research attempts to use AI to simulate social patterns based on the environment or specific tasks, rather than exploring the impact of rule changes. This chapter introduces the concept of artificial life into the game environment, offering the potential to apply automated game design methodologies to real-world scenarios.

Metaverse, a digital twin to the real world, embodies a symbiotic relationship with the gaming industry, modelling technologies, and social computing. The modelling of the Metaverse is supported by the utilisation of game engines such as Cry Engine, Unity Engine, and Unreal Engine [274], [275], [276]. These engines simplify the development process by reducing the requirement from code, thus cultivating a milieu that closely mirrors the real world [277]. In addition, the inherent interactive modalities and scene rendering technologies in games furnish immersive and engaging user experiences, emphasizing the significance of gaming in the formulation of user engagements within the Metaverse. Furthermore, the reliance of the Metaverse on modelling technologies becomes evident in the digital transposition of physical realities and the generation of digital identities via digital twins, identity modelling, and identity addressing [278]. The two environments presented in this chapter serve as representations of the real world, while the evaluator forecasts outcomes as digital twins.

Automated game design can be categorised into two focus areas: generation of game stages, levels, and structures, and generation of game rules, mechanics, and dynamics [143]. Notwithstanding, other facets like visuals, audio, and narrative also play a key role in game design. Procedural content generation (PCG) has been pivotal in creating game structure, for instance, generating levels or puzzles for existing games [279]. A shift towards procedural content generation via machine learning (PCGML) has been observed recently, leveraging existing game content to train models that produce new game content, thus eliminating the need for expert design knowledge [280]. On the other hand, the generation of game rules has seen applications of grammars, optimisation, and constraints to create new rule sets for existing level designs [281], [282], [71], [283], [284], [285]. Competitions like the general video game rule generation track have spurred advancements in this area, demonstrating the effectiveness of both constructive and genetic algorithm approaches [70]. Our automated rule design study begins with identifying key elements of general rule design and establishes a machine learning-based framework, demonstrating the potential for complex rule creation.

Procedural content generation via machine learning (PCGML) has garnered significant attention for its versatility in autonomous generation, co-creative design, data compression, and more, offering innovative solutions in game design [286].

PCGML minimises the need for human input during generation by leveraging representative content for autonomous generation, making it ideal for online content generation, such as in rogue-like games [287]. Moreover, PCGML facilitates co-creation, enabling efficient collaboration between human designers and algorithms in content creation [288], [289]. Notably, PCGML supports content repair by identifying unplayable areas and offering corrective suggestions [290], [291]. In terms of critique and analysis, PCGML outperforms other PCG approaches by providing in-depth analysis and critique of game content [292]. PCGML is also effective in data compression, particularly with autoencoders, allowing for efficient storage of game content [253]. The versatility and efficiency of PCGML in these domains highlight its potential for shaping the future of game design. The proposed framework takes advantage of PCGML to improve the efficiency of designer training and result evaluation.

3.2.2 Related Learning Paradigms

Deep generative models constitute a framework that represents the distribution of generative models using deep neural networks. Recent advancements, such as ChatGPT, BERT, and DALL-E 2 [293], have demonstrated significant potential in recent years. These models can generate text and images based on textual descriptions. Wang *et al.* demonstrated the performance of generative models in image segmentation [294], [295], [296] and noise removal [297]. Ho *et al.* have also validated the performance of diffusion models in video generation [298], [299]. Some research teams have explored causality and relationships using generative models [300], [301], [302]. Additionally, generation tasks in the game domain mainly involve game content generation, such as assets [303] or textual elements [304]. The deep generative neural networks are introduced into the designer for rule creation.

Reinforcement learning has evolved based on Markov decision processes, wherein the selection of actions focuses on the current state and potential reward [267]. Recent advancements in RL demonstrate significant progress in multi-agent [305], reward-free [306], and generation tasks [307]. The essence of RL models lies in acquiring experience based on rewards obtained from various situations. Well-trained models exhibit reliable performance when participating in games, which can assist in evaluating the game environment. Multiple reinforcement learning agents are utilised in the environments to explore strategies and evaluate created rules in this chapter.

The proposed Automated Rule Generation (ARG) will be a new machine learning task in the generative model category. The implementation of the ARG involves deep neural networks as the backbone models. The Deep Generative Model is used for the rule-generation purpose. Reinforcement Learning is not a part of ARG but a learning algorithm paradigm to stimulate AI agent behaviour so that the rules and generated environment can be evaluated and evolve. We do not particularly address the shortcomings of specific models. Instead, the chapter aims to propose a new paradigm and machine learning task.

3.3 Methodology

The rule generation task aims to comprehend the relationship between input rule parameters and game outcomes, and subsequently train a generative model to formulate rules for specific objectives. This necessitates, at a minimum, a rule designer and an environment capable of implementing rules and recording statistics for rule evaluation. Given that the environment functions as a black box, the establishment of an evaluator, acting as a digital twins, can facilitate the designer's optimisation. Gamified rules within a digital environment can be engaged by RL models, NPCs, and humans.

As depicted in Fig. 5.2, our system framework mainly comprises three primary processes: environment development, rule designer training, and rule generation. The environment development involves creating a game with predefined rules represented by a vector. The rule designer training process strives to construct models for rule generation and train them according to the target evaluation metrics. In the rule generation process, the pre-trained generative model designs a set of rules that align with the expected outcomes.

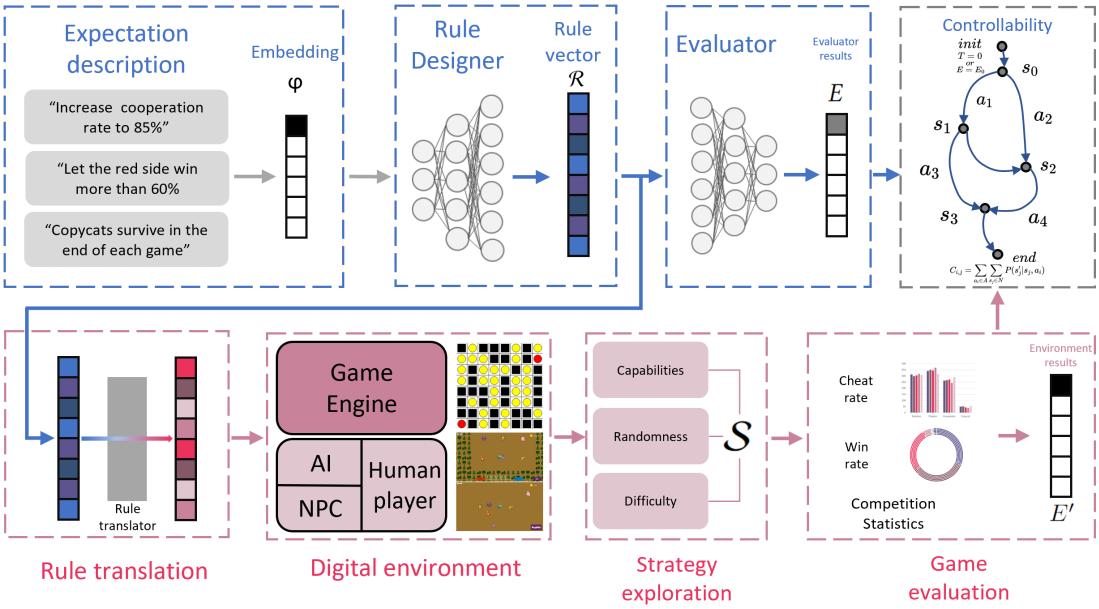


Figure 3.3: An illustration of rule generation framework. The rule designer learns to create rules according to the embedded expectation and evaluates the results. The generated rule vector is then sent directly to the evaluator and translated into an executable parameter for the environment. The evaluator learns to validate the rules individually. The RL model learns strategies by exploring the environment. The controllability is tested on both the environment and the rule designer.

3.3.1 Preliminaries

Extensive-form rule generation tasks involve three critical components: **rule**, **strategy**, **evaluation**. This section presents and explains the definitions and notations below.

Rule. Rules can be regarded as a set of principles in a game, such as players, maps, nodes, functions, natural acts, and decisions [308]. Let $\mathcal{R} = [r_{n,d}] \in \mathbb{R}^{N_r \times D_r}$ represent a set of rules in a digital environment. Here, N_r denotes the quantity of rules and D_r represents the dimension of each rule. The set includes creation, deletion, and modification of rules. Rule creation increments N_r to $N_r + 1$, adds it to $\mathcal{R}$, and increases the rule quantity. Conversely, deletion involves removing r_i where $i \in \{1, ..., N_r\}$ from $\mathcal{R}$, reducing N_r. Rule modification updates $r_{i,d} \in \mathcal{R}$ where $i \in \{1, ..., N_r\}$ to $r_{i,d}'$. The evolved rules can be represented by $\mathcal{R}'$.

Strategies. Game strategies are a series of complete algorithms selected by players according to the situation, in compliance with rules, and determining the result [309], such as the strategy of the prisoner's dilemma. We assume $\mathcal{S} = [s_{n,d}] \in \mathbb{R}^{N_s \times D_s}$ to be strategies developed by players. Here, N_s is the strategy number and D_s is the dimension of each strategy. As each game may have more than one strategy and each strategy may have different stages, D_s may vary. If the game is a complete information game, all strategies can be enumerated and N_s can be a specific number, whereas N_s can be infinite in an incomplete information game. Both humans and AI can play and develop strategies based on the reward.

Evaluation. Game evaluation is associated with high-level heuristics, including spontaneity, interruptability, and continuity [310]. These heuristics are determined by a series of specific game parameters. Let the evaluation result be denoted by $E = [e_{n,d}] \in \mathbb{R}^{N_e \times D_e}$, where N_e and D_e represent the evaluation metric quantity and dimension, respectively. All rules can be assessed using a set of evaluation criteria f to obtain the result E. Each $e \in E$ represents an assessment perspective of the evaluation. Furthermore, the nature of the evaluation is determined by the game rules, while the actual results are demonstrated and recorded by agents' strategies during gameplay in the digital environment. This process can be represented as $E = f(\mathcal{R}, \mathcal{S})$, where f denotes the evaluation metrics.

Automated Rule Design. Automated rule design begins with an initial rule set, $\mathcal{R}$, foundational to the game environment. Players or agents then formulate strategies, denoted as $\mathcal{S}$, which are essentially algorithms or behaviours tailored to optimise outcomes within the confines of $\mathcal{R}$. Subsequent evaluations expressed as $E = f(\mathcal{R}, \mathcal{S})$, represent gameplay metrics such as efficacy and fairness. Drawing from $\mathcal{E}$, the system discerns areas for rule modification in $\mathcal{R}$ to enhance gameplay or meet specific objectives. This cycle of strategy formulation, evaluation, and rule refinement iteratively progresses until the system meets predetermined performance or balance benchmarks.

3.3.2 Frameworks

Figure 5.2 demonstrates that the system consists of three primary components: rule designer, evaluator, and digital environment. Additionally, the digital environment incorporates a set of reinforcement learning agents, non-player characters, and human players. The expectation is a set of linguistic descriptions embedded as a vector φ. The rule designer generates a set of rule vectors $\mathcal{R}$ based on embedding φ. Generated rules are then translated into accessible vectors for game platform implementation. Subsequently, the game is made available to Q-learning agents, NPCs, and human players for strategy exploration. The game, featuring experienced players, simulates the evolution of society under the generated rules. All statistics recorded during gameplay are gathered for rule evaluation. The evaluator serves as a digital twin of the environment, simulating output statistics and sharing the same raw rule vector, created by the rule designer, as input. The evaluation results are utilised to compute controllability, which is then employed to upgrade the rule designer.

The objective of generative tasks is to train a generative model, such as a variational autoencoder (VAE), generative adversarial network (GAN), or diffusion model, to create content according to specific requirements. These models are represented by a function denoted as $g : Z \to X$, which is designed to map random noise vectors Z to high-dimensional output X. Prior to training, it's crucial to curate a labelled dataset, where the labels serve as the model's ground truth. Unlike a traditional generative model, this framework doesn't require a pre-prepared dataset. The rule generation process aims to train a designer $d : E \to \mathcal{R}$ to create rules $\mathcal{R}$. As a data-free training task, the designer takes the expected result E as input, and the output rules are $\mathcal{R} = d(E)$. The created rules $\mathcal{R}$ will be implemented in the environment and the output can be represented as $E' = Env(d(E), S)$, where Env is the evaluation criteria. The training of d is formulated as follows:

$$\min_{d} V(d, Env) = \log(1 - Env(d(E), S)). \tag{3.1}$$

Although $Env(d(E), S)$ represents the ground truth, an evaluator, functioning as a digital twin, aids the designer during the backwards process. The objective of the evaluator is to emulate the environment and predict evaluation outcomes. It accepts $\mathcal{R}$ as input, analogous to the environment, and learns to score rules as $E'' = p(d(E))$, which is denoted by $p : \mathcal{R} \to E''$. Throughout training, the evaluator is refined by minimizing the discrepancy between its own output E'' and the ground truth E'.

$$\min L(E', E'') = \min L(p(d(E)), Env(d(e))). \tag{3.2}$$

Reinforcement learning is based on access to the *Markov Decision Process* that can be defined as the tuple $\{S, \mathcal{A}, \mathcal{T}, R, p(s_0), \gamma\}$. These elements represent states, actions, transition probabilities, rewards, initial state probabilities, and discount factors, respectively. This chapter utilises Q-learning models as RL agents. They share the same action list, reward map, and perceptual field as other players. The actions of well-trained agents can be considered objective, as they maximise the reward. The goal of strategy exploration is to train players $P : State \rightarrow Action$ according to the reward.

3.3.3 Environment and Task

This chapter presents two digital game environments for rule demonstration, referred to as *Maze Run* (MR) and *Trust Evolution* (TE), purposed to demonstrate the practicability of rule generation with judicious utilisation of computational resources. These environments aim to demonstrate the feasibility of rule generation while utilising minimal computing resources. TE serves as a fusion of artificial life and rule-generation tasks, as it simulates cooperative behaviour among individuals in a society.

Maze Run. The maze run provides a 2D map with variable height and width. It is full of reward points and traps that can be modified according to different evaluation metrics. The goal of the game is to survive as long as possible. All agents, including human players and Q-learning agents, try to find a strategy that gets more food and avoids traps. As the MR environment aims to match the rule generation task with minimising parameters, here we fix the number of agents as 2, both height and width are 6 grids, and the initial locations are the left bottom and right top corners separately.

Let M be the set of 2D maps, H be the set of heights, W be the set of widths, G be the set of grid cells, and A be the set of agents. Let $h : M \rightarrow H$ be the height function that maps each map to a height. Let $w : M \to W$ be the width function that maps each map to a width. Let $l : A \to G$ be the location function that maps each agent to a grid cell. Let $f : M \times G \to$ reward point, trap be the contents function that maps each map and grid cell to the contents of the cell. Let $t : M \times A \to R$ be the time function that maps each map and agent to the time they survived. The following constraints hold: (1) For all $m \in M$, $h(m) = 6$ and $w(m) = 6$. (2) For all $m \in M$ and $a \in A$, $l(a)$ is either the left bottom or right top corner of the map m. (3) For all $m \in M$, $g \in G$, $f(m, g)$ is either a reward point or a trap. (4) All agents, including human players and Q-learning agents, try to find a strategy that maximises their time function $t(m, a)$ by collecting reward points and avoiding traps. The goal of the game is to survive as long as possible, so the objective is to maximise the time function $t(m, a)$ for all agents $a \in A$.

Trust Evolution. In the Trust Evolution environment, the rules can be characterised by parameters such as payoff, population size, the round number, reproduction rate, and mistake probability. The mistake probability represents the likelihood of a player choosing the opposite action, while the round number indicates the number of trades each agent conducts. Throughout each game, players attempt to acquire more coins by engaging in trade with others, choosing between two possible actions: *cheat* or *cooperate*. The payoff maps agents' actions to trade outcomes. Six types of NPCs represent various personalities: *random*, *cheater*, *cooperator*, *copycat*, *grudger*, and *detective*. The first three types consistently choose random, cheat, and cooperate, respectively. Copycats initiate cooperation and subsequently mimic others' last actions, while grudgers always cooperate until betrayed. Detectives start with a sequence of cooperation, cheat, cooperation, and cooperation actions; if others never reciprocate cheating, they continue cheating, otherwise, they adopt the copycat strategy. At the conclusion of each match, a selection process eliminates low-performing players and reproduces top performers.

Let R be the set of rounds, N be the set of players, and A be the set of actions (cheating or cooperation). Let p be the mistake rate such that $0 \leq p \leq 1$. Let $f : N \times R \to A$ be the action function that maps each player and round to an action. Let $g : A \times A \to R$ be the payoff function that maps each pair of actions

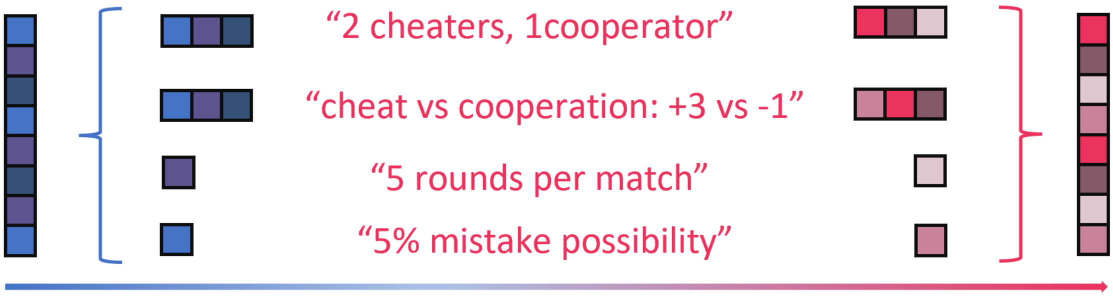


Figure 3.4: An illustration of the rule translation progress. The blue vector is generated by the rule designer and subsequently divided into smaller vectors, each representing a distinct rule. These smaller vectors are then encoded as executable parameters for the environment.

to a reward. Let $T : N \to$ random, cheater, cooperator, copycat, grudger, detective be the type function that maps each player to their personality. Let k be the reproduction number, such that k is a positive integer. The following constraints hold: For all $n \in N$ and $r \in R$: (1) If $T(n) =$ random, then $f(n,r)$ is chosen randomly. (2) If $T(n) =$ cheater, then $f(n,r) =$ cheating. (3) If $T(n) =$ cooperator, then $f(n,r) =$ cooperation. (4) If $T(n) =$ copycat, then $f(n,r) = f(m,r-1)$ for some player $m \in N$ such that $f(m,r-1)$ is the action of m in the previous round. (5) If $T(n) =$ grudger, then: if there exists a player $m \in N$ and a round $s < r$ such that $f(m,s) =$ cheating, then $f(n,r) =$ cheating; otherwise, $f(n,r) =$ cooperation. (6) If $T(n) =$ detective, then: if $r = 1$, then $f(n,r) =$ cooperation; if $r = 2$, then $f(n,r) =$ cheating; if $r = 3$, then $f(n,r) =$ cooperation; if $r = 4$, then $f(n,r) =$ cooperation; if there exists a player $m \in N$ and a round $s < r$ such that $f(m,s) =$ cheating, then $f(n,r) = f(m,r-1)$; otherwise, $f(n,r) =$ cheating. At the end of each game, the k players who won the highest reward are selected for reproduction, and the same number of lowest reward are eliminated.

Rules Translation. Demonstrating game rules in a virtual environment is a cross-platform task. The rules are structured as a set of parameters which can not be understood or accessed by neural networks directly. One potential solution is extensible markup language (XML), which encodes arbitrary information into human-readable and machine-readable formats [311]. It has been introduced into some black-box optimisation tasks as it can be easily translated to inputs [312] [313]. However, the description of the rule includes not only objects but also relationships.

To convert the generated rule vector into accessible game mechanisms within the environment, we employ rule maps. Owing to the relative simplicity of the Maze Run (MR) environment, the rule vector generated by the rule designer merely contains the reward information for each grid cell in the map. This vector takes the form of a tensor with dimensions corresponding to the map size.

The TE environment is relatively complex as the generated rule vector contains multiple rules. TE's rule is a 15-dimensional vector. The first 6 dimensions of the vector represent the population of each personality, and the total number of people is fixed. The following 7th to 12th dimensions map to the trade payoff: cheat-cheat, cheat-cooperate, cooperate-cooperate. The 13th dimension is related to the round number, and the last two dimensions represent the reproduction number and mistake possibility, respectively.

Dimension	0	1	2	3	4	5	12	13	14
Semantics	Random Number	Cheater Number	Cooperator Number	Copycat Number	Grudger Number	Detective Number	Round Number	Reproduction Number	Mistake Probability
Range	[0,6]	[0,6]	[0,6]	[0,6]	[0,6]	[0,6]	[5,20]	[1,10]	[0,0.5]
Type	int	int	int	int	int	int	int	int	float

Table 3.1: Trust Evolution Game rule vector: dimensions 0–5 represent the counts of each agent personality; dimensions 12–14 correspond to round number, reproduction number, and mistake probability.

Dimension	6	7	8	9	10	11
Semantics	Payoff player1 cheat cheat	Payoff player2 cheat cheat	Payoff player1 cheat cooperate	Payoff player2 cheat cooperate	Payoff player1 cooperate cooperate	Payoff player2 cooperate cooperate
Range	[−5,5]	[−5,5]	[−5,5]	[−5,5]	[−5,5]	[−5,5]
Type	float	float	float	float	float	float

Table 3.2: Trust Evolution Game rule vector: dimensions 6–11 correspond to payoff values for the combinations (cheat–cheat, cheat–cooperate, cooperate–cooperate) for both players.

Ethical Analysis. In the realm of automated game design, the generation of rules through artificial intelligence introduces the possibility of inadvertently embedding biases into the gaming experience. This chapter presents an ethical examination of our Rule Generation Network. Throughout the development of the RGN structure, we incorporated two game environments: MR and TE. It is important to highlight that players' roles within these games are devoid of any attributes related to sex, age, race, or similar socio-cultural factors. In MR, traps and rewards are solely tied to reinforcement learning rewards, eliminating potential subjective biases. Similarly, in TE, the six distinct personalities are purely representative of various NPC behavioural patterns, further emphasising our commitment to unbiased game design.

Unreasonable Expectation. In the process of testing the RGN framework, which leverages external expectations to formulate rules, in multiple expectations for future work. We discovered challenges associated with certain expectations that proved to be unreasonable. An illustrative example is the contradiction of expecting both a cheater and a cooperator to win in the same trust evolution game. The contradiction of such unreasonable expectations not only diminishes training efficiency but also adversely impacts the evaluation score of the rule designer within the RGN. Consequently, we highlight the identification of these unreasonable expectations as a hurdle in rule generation. Addressing this challenge may require the application of logical reasoning and structured methodologies.

3.3.4 Controllability

To ensure the rules generated rules of the designer during training are consistent with the input expectation, we introduced controllability from information theory for designer evolution. The controllability matrix is a fundamental concept in control theory, serving as an essential tool for evaluating the controllability of linear and non-linear systems. Controllability represents the ability to guide a system from any initial state to any target state within a finite period using available control inputs. By mathematically assessing this property, the controllability matrix enables researchers and engineers to analyse the efficacy of control inputs in directing a system's state and informs the design of control strategies for various applications in engineering, robotics, and other fields.

Proposition. The linearised system, exemplified by the player's movement, can be expressed as:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.3}$$

$$y(t) = Cx(t), \tag{3.4}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^m$ signifies the control input vector, $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times m}$ represents the input matrix, and $C \in \mathbb{R}^{p\times n}$ is the output matrix. The controllability matrix is defined as:

$$\mathcal{C} = [B, AB, A^2B, \ldots, A^{n-1}B], \tag{3.5}$$

where $\mathcal{C} \in \mathbb{R}^{n\times mn}$, and A^kB corresponds to the k-th power of matrix A multiplied by matrix B. The non-linear system is characterised by the following state-space equation:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \tag{3.6}$$

where $x(t) \in \mathbb{R}^{15}$ is the 15D rule vector, $u(t) \in \mathbb{R}$ represents the 1D win rate input, and $f(x)$ and $g(x)$ are non-linear vector functions. The Lie derivatives of vector fields $f(x)$ and $g(x)$ can be computed as:

$$L_f, L_g = \frac{\partial g(x)}{\partial x} f(x) - \frac{\partial f(x)}{\partial x} g(x). \tag{3.7}$$

Construct the Lie algebra $\mathcal{L}$ generated by the Lie derivatives of vector fields $f(x)$ and $g(x)$:

$$\mathcal{L} = f, g, [f, g], [f, [f, g]], [g, [f, g]], \ldots. \tag{3.8}$$

At a specific point x_0, create the distribution matrix $D(x_0)$ using the vectors in the Lie algebra:

$$D(x_0) = \begin{bmatrix} f(x_0) & g(x_0) & [f, g] & \cdots \end{bmatrix}. \tag{3.9}$$

If the rank of the distribution matrix $D(x_0)$ is equal to the dimension of the state-space ($\text{rank}(D(x_0)) = 15$), then the system is locally controllable at the point x_0.

Lemma. Let N denote a set of players, and A represent actions (cheating or cooperation). To investigate the controllability of the system, we must first define the state-space and action-space. The controllability matrix is a matrix that associates the players' actions with alterations in the system's state. Let C be the controllability matrix, with $C_{i,j}$ signifying the impact of player i's action on player

j's state. The entries in the controllability matrix can be computed as follows:

$$C_{i,j} = \sum_{a_i \in A} \sum_{s_j \in N} P(s'_j | s_j, a_i), \tag{3.10}$$

where s'_j is the state of player j after player i takes action a_i, and $P(s'_j|s_j, a_i)$ is the transition probability from state s_j to state s'_j given action a_i. By calculating the controllability matrix, one can determine the degree to which the players can control the outcome of the game and whether the game is balanced or not. This can help to improve the design of the rule generator and ensure that the generated games are academically sound and enjoyable for players.

As for the designer networks, it provides a mapping from the 1D input to the 15D output space, represented as:

$$x(u) = W_3\sigma(W_2\sigma(W_1u + b_1) + b_2) + b_3. \tag{3.11}$$

3.4 Experiment

3.4.1 Settings

The experiments focus on three primary objectives: (1) implementing the RGN model across multiple environments; (2) demonstrating and gamifying the generated rules; (3) generating rules according to specific evaluation metrics. It requires the RGN model to be capable of generating rules that can be instantiated within diverse environments, allowing AI agents, NPCs, and humans to play. Training the rule designer based on various evaluation metrics, with the expectation that a well-trained designer will produce rules that align with the desired outcomes. It also aims to showcase the generated rules within an environment that provides accessible operations and visual representations of the associated statistical data. This comprehensive experimental approach seeks to validate the effectiveness of the proposed RGN model in generating engaging and meaningful game experiences.

Evaluation Criteria. The evaluation framework for the RGN model is mul-

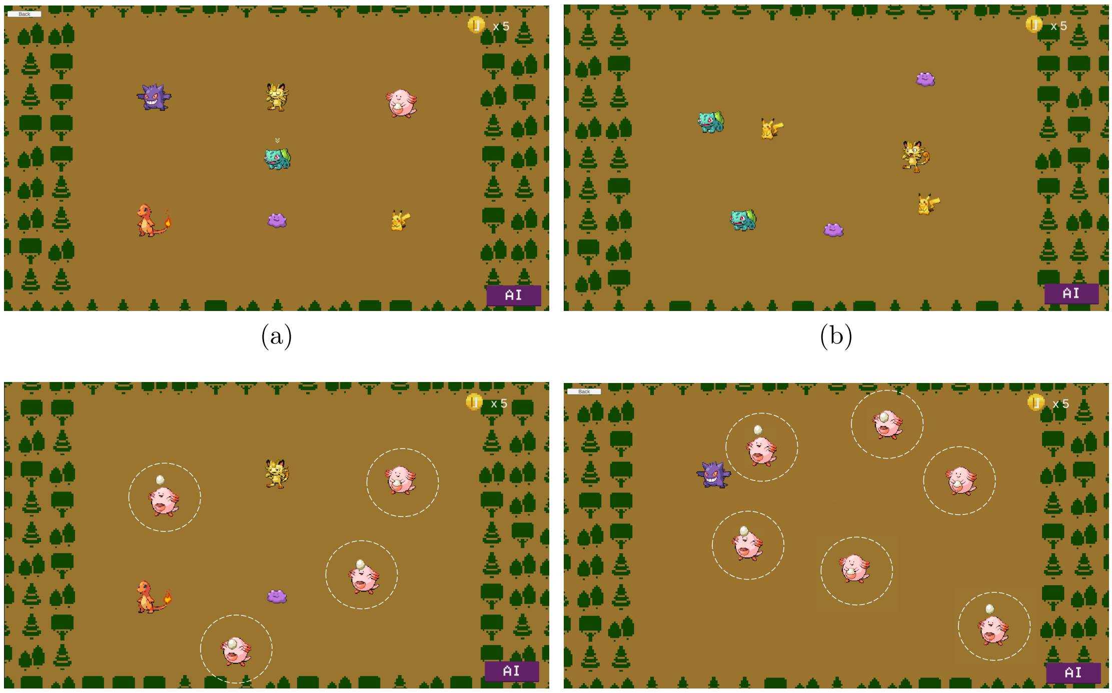
Algorithm 1 RGN training algorithm

Input:
Linguistic description for rule expectation E;
Number of training epochs Ep.
Game environment Env.
Output:
Well-trained rule designer model: $D : E \rightarrow R$.
Well-trained Q-learning model: $Q : S \rightarrow A$.

```
Initialisation: Embed the rule expectation as φ.
Add noise z to the expectation E, resulting in φ = φ + z.
for t = 1 : Ep do
    Rules generation: designer create rules R = D(E).
    Train reinforcement learning model: translate R into R′ = T(R), learning
    strategies S = Env(R′);
    Train evaluator: use E′ = Env(R′) to train the evaluator;
    Train designer: use the evaluator's result E′ and input E to upgrade designer;
    if E′ − E == 0 then
        break;
    end if
end for
Save the well-trained designer model D for rule generation.
Save the well-trained Q-learning model Q for action selection.
```

tifaceted, encompassing a rule designer, an evaluator, Q-learning models, and two distinct virtual environments. Consequently, a diverse array of evaluation methodologies and criteria are implemented. For the rule designer and evaluator, qualitative evaluation is undertaken by integrating the generated rules into the two virtual environments (MR and TE). Quantitatively, Mean Squared Error (MSE) loss and cross-entropy loss functioned as training criteria for the rule designer and evaluator, respectively. Additionally, controllability is employed to gauge the performance of the rule designer and game platform. The distribution of potential game outcomes, ascertained by the random sampling of rules, served as a dataset during RGN training.

In establishing a baseline for the RGN, we separate it into three distinct segments, each according to its specific role and utility in the rule design continuum: the designer, evaluator, and tester. For the RGN designer, the baseline is drawn from both human and random designers. The evaluator's baseline is anchored to actual



(a) (b) (c) (d)

Figure 3.5: Illustration of the different rule designs. The entities encompass six distinct character roles, with different appearances representing unique personalities. The rules are generated by humans, random generation, and RGN. (a) Human design. (b) Random design. (c) RGN design (payoff). (d) RGN design (population).

game results, ensuring an empirical point of reference. Meanwhile, the RGN tester's baseline encompasses a comparative assessment of performances across Q-learning, deep Q-learning (DQN), double DQN, categorical DQN, human participants, and NPCs. This stratified baseline approach provides a comprehensive and multifaceted reference for evaluating the efficacy and robustness of the RGN system. Moreover, the assessment of automated game design systems presents a persistent challenge for scholars in the discipline. The comparisons between individual systems remain infrequent, often limited to qualitative expositions in sections dedicated to related literature. This is primarily due to the distinct nature of each automated game design system, characterised by their unique game engines, technological infrastructures, and design philosophies [139].

Implementation Details. The MR and TE environments are developed using PyPlot and the Unity engine, respectively. In the MR environment, the generated rule vectors encompassed the reward map, whereas in TE, the rule comprised

Table 3.3: Strategies' records of trust evolution game generated by well-trained RGN networks based on 100% cooperation rate expectation. The population for six personalities are fixed to 4, and one of the roles can choose a trade target each time.

Trade id	00		01		02		03		04		05	
Player id	A1	F3	B1	C2	C1	C3	D1	C2	E1	E4	F1	C3
Operation	cheat	cooperate	cheat	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate
Result	win	lose	win	lose	win-win	win-win	win-win	win-win	win-win	win-win	win-win	win-win
Coin	+1.32	-0.34	+1.32	-0.34	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87
Trade id	06		07		08		09		10		11	
Player id	A2	A5	B2	F2	C2	C1	D2	D1	E2	C2	F2	C4
Operation	cheat	cheat	cheat	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate
Result	lose-lose	lose-lose	win	lose	win-win	win-win	win-win	win-win	win-win	win-win	win-win	win-win
Coin	-0.21	-0.21	+1.32	-0.34	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87
Trade id	12		13		14		15		16		17	
Player id	A3	F1	B3	A1	C3	C2	D3	D4	E3	E1	F3	C2
Operation	cheat	cooperate	cheat	cheat	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate	cooperate
Result	win	lose	lose-lose	lose-lose	win-win	win-win	win-win	win-win	win-win	win-win	win-win	win-win
Coin	+1.32	-0.34	-0.21	-0.21	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87	+0.87

various game parameters such as population distribution, payoff structure, round count, reproduction quantity, and mistake probability. Essential metrics in the TE environment, including the Cooperation Rate (CR), Average Coin Increment (ACI), and Cooperator Proportion (CP), represent players' actions, resource fluctuations, and game trends. To maintain consistency and reliability in the TE environment, we drew upon 'The Evolution of Trust' by Nicky Case, excluding Q-learning and human players for demonstration purposes. The rule designer and evaluator are implemented as models with three fully connected layers. Q-learning models functioned as our RL agents, while a fixed response algorithm, providing specific responses to distinct inputs, is utilised for NPCs.

The core RGN module consists of two multilayer perceptrons, the Designer and the Evaluator—both trained using mean-squared error loss and the Adam optimiser (learning rate $\eta = 0.01$, $\beta_1 = 0.5$, $\beta_2 = 0.999$, batch size B). The Designer accepts a batch of evaluation requirements with shape $B \times E$ (where $E = 1$ by default). It comprises three fully connected layers: the first maps the E dimensional input to 8 hidden units, followed by a LeakyReLU activation (negative slope 0.2); the second maps 8 units to 16 units, includes batch-normalisation, and uses a LeakyReLU activation; the final layer maps from 16 units to $L \times R$ outputs (with $L = \texttt{layersNum} = 1$ and $R = \texttt{RuleDimension} = 15$), and applies a Softmax over the feature dimension to produce a probability-simplex "rule vector." The Designer' s forward pass reshapes this output to a tensor of shape $B \times (L\,R)$. The Evaluator takes as input a batch of "rule vectors" with shape $B \times (L\,R)$, flattens each vector, and processes it through

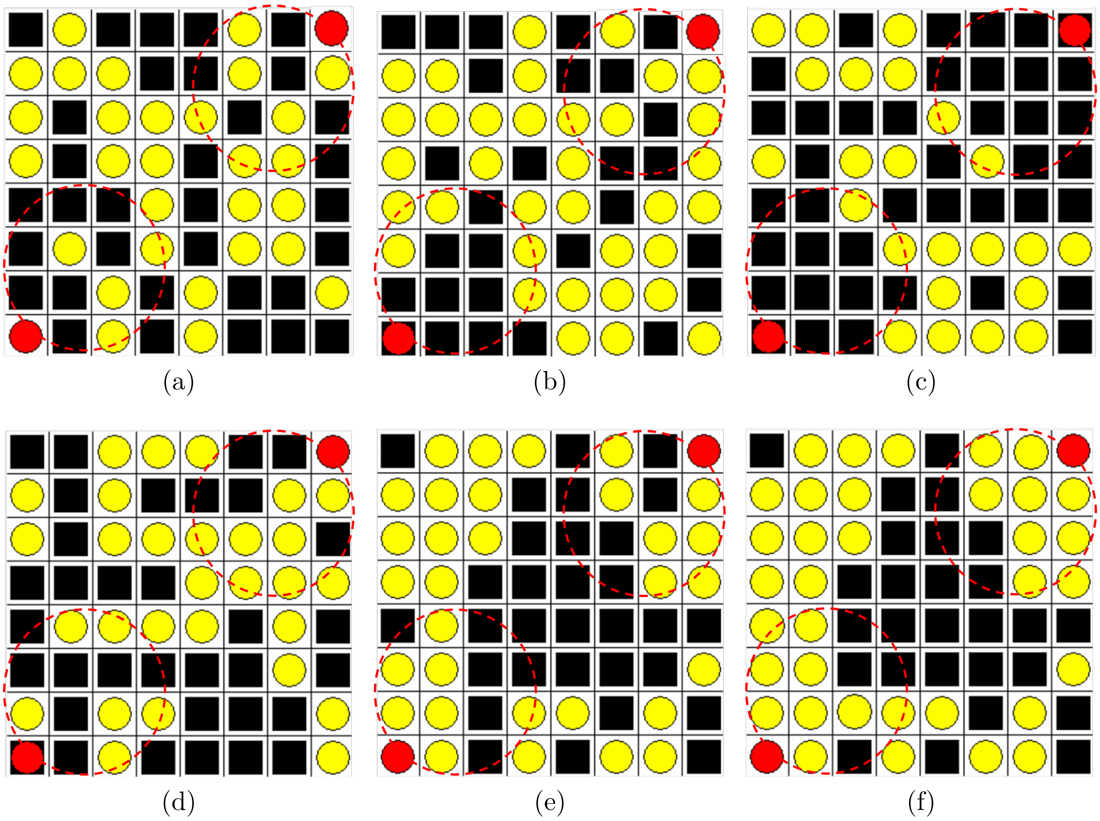
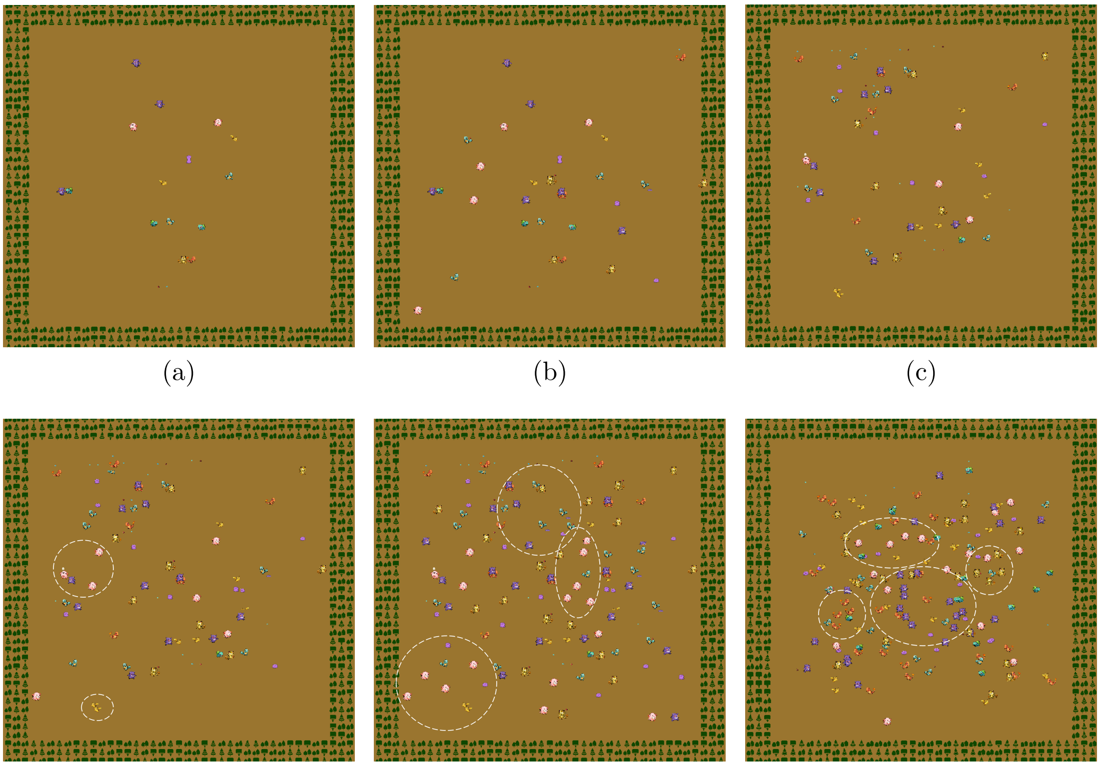


Figure 3.6: Illustration of the created rules from the MR during training. The red circles denote the Q-learning agent, the yellow circles indicate reward points, and the black squares represent traps. The red dotted line highlights the primary changes of rule during training. (a) Expected win rate = 0% : epoch 0. (b) Expected win rate = 0% : epoch 10. (c) Expected win rate = 0% : epoch 100. (d) Expected win rate = 100% : epoch 0. (e) Expected win rate = 100% : epoch 10. (f) Expected win rate = 100% : epoch 100.

three fully connected layers: the first maps $(L\,R)$ inputs to 128 hidden units with a LeakyReLU activation (negative slope 0.2); the second maps 128 to 64 units, again followed by LeakyReLU; the final layer maps 64 units to a single scalar output, passed through a Sigmoid to yield a difficulty prediction in $[0, 1]$. In each iteration, the Designer is updated to minimise the MSE between the Evaluator's prediction on the Designer' s output and the target evaluation requirements. Next, the Evaluator is updated—using the Designer' s generated rule vectors (detached from the gradient)—to minimise the MSE between its prediction and the "ground-truth" difficulty obtained by simulating those rule vectors in the environment. Both networks share identical Adam optimiser settings and batch sizes as described above.



(a) (b) (c) (d) (e) (f)

Figure 3.7: Illustration of six game scenarios with varying player numbers, specifically containing 15, 30, 45, 60, 100, and 150 Q-learning agents as players, respectively. Apart from the player numbers, all other rules remain consistent across the games. (a) 15 agents. (b) 30 agents. (c) 45 agents. (d) 60 agents. (e) 100 agents. (f) 150 agents.

3.4.2 Qualitative Evaluation

The training of the RGN is conducted across multiple game environments, including MR and TE. Due to the single rule accessibility in the MR environment, we will focus on the variations in rule design within the TE environment. The implementation of the designed rules is depicted in Fig. 3.5. The primary design objective sought to encourage a higher survival rate for cooperators. Human-designed rules are easily distinguishable due to their orderly arrangement of roles, as the human designer strategically adjusted the payoff structure to achieve this goal. In contrast, the untrained model generated rules at random. Regarding the rules designed by the well-trained RGN model, a certain level of disorder is apparent in the placement of roles. Both rules designed by the RGN tend to prioritise the initial population over payoff or other parameters. Thus, the proportion of cooperators shows a significant increase, which means the rules are inclined to augment the initial quantity of target winners as a means to ensure they fulfil the expectation.

Fig. 3.6 illustrates the performance evolution of the rule designer for the MR task during the training process. The RGN rule designer's aim is to manipulate the win rate by creating additional traps or rewards in the map, while the Q-learning players attempt to maximise the survival rate and reward coin number. The first row presents the training progression for creating MR game rules where both players have a 0% win rate, whereas the second row focuses on the creation of a game with a 100% win rate. As depicted in Fig. 3.6a, the untrained designer in the MR environment generated rules randomly. However, during training, it started generating more traps around agents' spawn points and increased trap damage as shown in Fig. 3.6b to Fig. 3.6c. In contrast, as illustrated in Fig. 3.6d, Fig. 3.6e, and Fig. 3.6f, the upper right and lower left corners players began to host more reward points, replacing the traps. In summary, the rule designer evolved the rules during training to align with the expected win rate.

In the rule evaluation phase, players, including humans, NPCs, and Q-learning models, endeavoured to devise strategies to maximise their game rewards. Fig. 3.7 presents the environment's capability to handle multi-agent tasks at varying player count levels. As each agent is required to identify their target trading role and execute trading actions, the complexity increases at a rate of $O(n^2)$. Current experiments in TE have supported up to 150 agents learning rules and developing strategies within the game. Furthermore, as the number of agents increased from Fig. 3.7d to Fig. 3.7f, we observed a pattern emerging among agents. As agents tend to trade with those who benefit them the most, some roles exhibit a preference for trading with agents who share the same personality.

Strategies developed by Q-learning players in a structured Maze Run game can be visualised as a map, provided that the learning process is driven by rewards. Figure 3.8 delineates three distinct games, each aiming for distinct win rates of 0%, 50%, and 100%, respectively. The first row of illustrations portrays the reward map, while the second row demonstrates the movement dynamics of two Q-learning agents. The rewards are governed by transitions in states, implying that different cells within

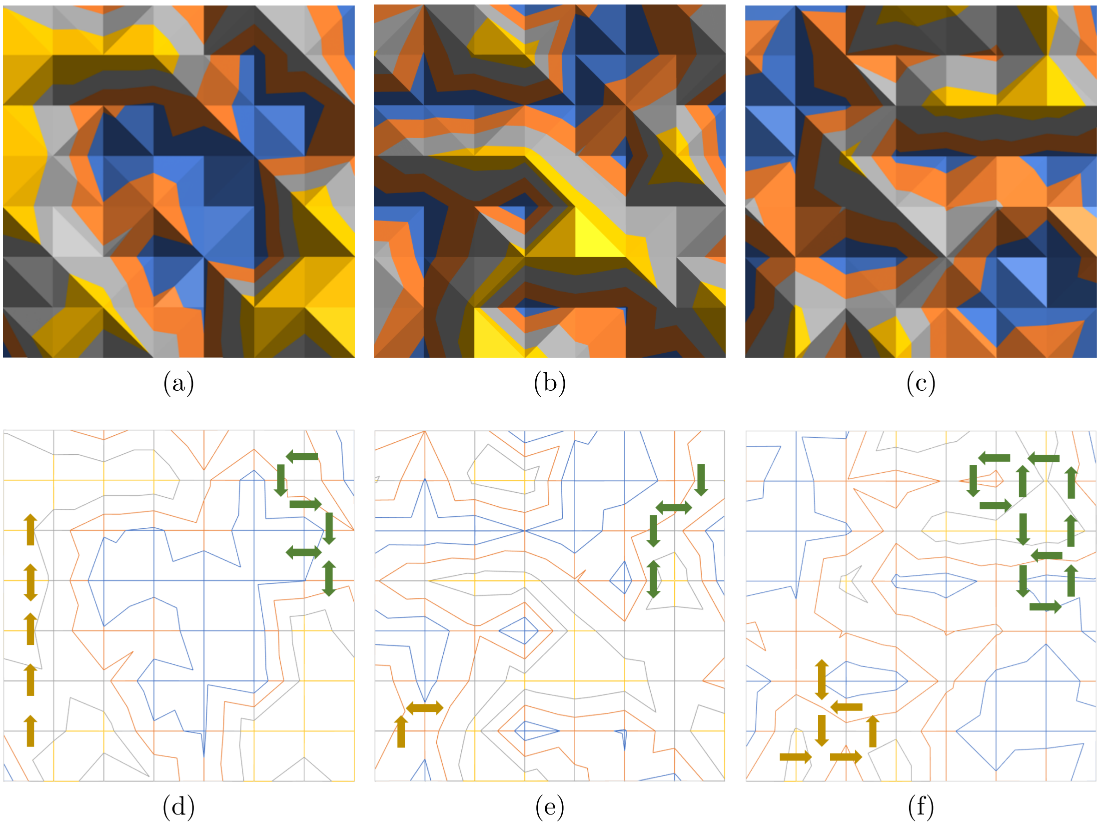


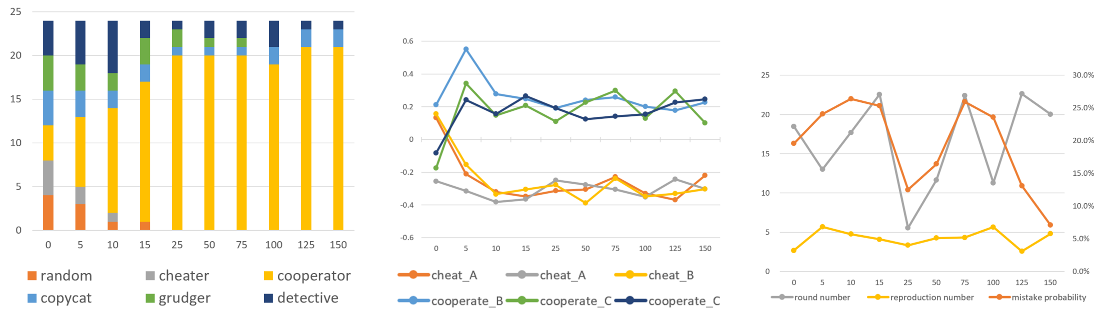
Figure 3.8: Visualisation of the rules and agents' strategies in the maze run game. The blocks coloured in shades of orange and yellow denote regions associated with high rewards, whereas areas signified by grey and blue correspond to relatively lower rewards. The arrows coloured in yellow indicate the potential movements of the agent that start at the bottom left, while the green arrows exemplify the prospective manoeuvres of the agent positioned at the upper right. (a) Reward map: win rate = 0%. (b) Reward map: win rate = 50%. (c) Reward map: win rate = 100%. (d) Agents' strategy: win rate = 0%. (e) Agents' strategy: win rate = 50%. (f) Agents' strategy: win rate = 100%.

the map may not necessarily exhibit continuity or linearity, despite analogous colour representations. As can be observed from Figures 3.8e, 3.8f, and 3.8d, the players investigate regions adjacent to their initiation points, and intermittently retrace their trajectories to maximise rewards. Interestingly, during the training phase, the players also venture into areas associated with lower rewards but display rationality in formulating their ultimate strategies, as depicted in Figure 3.8.

Table 3.3 presents a record of players' strategies encompassed within a series of game rules in the trust evolution. These rules have been generated by an adept designer utilising the RGN designer with the objective of achieving a 100% cooper-

Table 3.4: Performance comparison of four rule design methods for various design requirements. Four distinct rule expectations, including CR=100%, ACI=30, CP=100%, and CP=80%, are fulfilled by four types of designers: human, random, untrained RGN, and well-trained RGN. Three metrics are compared in the table. The best-performing design is bolded.

Expectation	CR = 100%			ACI = 30			CP = 100%			CP = 80%		
Metrics	**CR**	ACI	CP	CR	**ACI**	CP	CR	ACI	**CP**	CR	ACI	**CP**
Human designed rules	**100%**	17	100%	78%	67	64%	60%	41	**100%**	100%	23	52%
Random rules	55%	5	32%	23%	-5	44%	25%	4	0%	42%	11	0%
Untrained RGN	63%	-15	0%	54%	6	58%	0%	-5	0%	68%	8	0%
Well-trained RGN	**100%**	17	100%	100%	**24**	82%	100%	25	**100%**	76%	15	**64%**



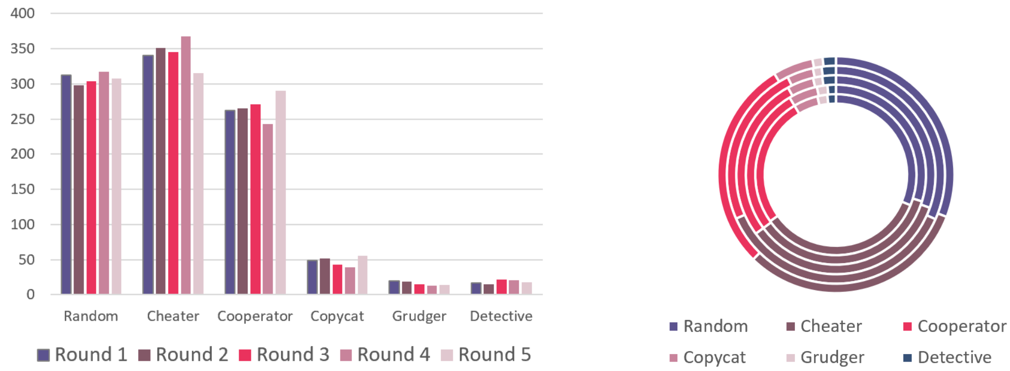
(a) (b) (c)

Figure 3.9: Illustration of rules evolution in the TE during training. The population figure employs six colours to represent six distinct personalities. The symbols A, B, and C in the payoff figure correspond to the 'cheat-cheat,' 'cheat-cooperate,' and 'cooperate-cooperate' scenarios, respectively. The last figure reveals alterations in the round number, reproduction number, and mistake possibility throughout the training process. (a) Population. (b) Payoff. (c) Other parameters.

ation rate. Each trade involves two players, each of whom has the option to choose at least one other player with whom to trade, with the outcome contingent on the payoff outlined in the rules. As the goal is complete cooperation, the designer has amplified the rewards for cooperation while penalising deceit, as demonstrated in the table. Consequently, the majority of players incline towards cooperation, with the exception of those within the 'random' category. Additionally, it has been observed that those who consistently cooperate are widely favoured among players of various personality types.

3.4.3 Quantitative Evaluation

Figure 3.9 represents the evolution of rules during designer training, the objective of which is to formulate rules that align with the anticipated increase in the cooperation



(a) (b)

Figure 3.10: The quantity and proportion of each personality of role wins in five random samplings. In the histogram, columns of the same colour correspond to the same round of sampling, while in the pie chart, circles of the same colour represent the same round of sampling. (a) Size. (b) Ratio.

rate to 100%. As depicted in Figure 3.9a, the designer tends to expand the population of cooperators while reducing the number of cheaters and random players. The population of detectives also decreases, caused by their consistent inclination to scrutinise others. Figure 3.9b portrays modifications in tradeoffs where, regardless of the scenarios being cheat-cheat, cheat-cooperate, or cooperate-cooperate, the designer primarily intensifies the rewards associated with cooperative behaviour. Concerning other parameters in figure 3.9c, such as the number of reproductions, round numbers, and the probability of mistakes, the oscillation visible in their respective curves suggests a relatively minor correlation with the design expectation.

We also demonstrate the distribution of gaming outcomes, which serve as real-time data for training the designer and evaluator. The Monte Carlo method is employed to separately sample 1000 data points 5 times as depicted in Fig. 3.10. Rules within the TE environment are encoded in a 15-dimensional vector, capable of decoding into rules encompassing approximately 3.56265×10^{15} unique cases. As Fig. 3.10a illustrates, there is a preponderance of random game results favouring cheaters, cooperators, and randoms as winners. In comparison, grudgers only possess a 4.764% chance of victory against cheaters. Surprisingly, copycats did not perform better, despite their potential highlighted in Case's model. The pie chart in Fig. 3.10b supports this finding, indicating that it is easier to design rules favouring the victory of cheaters, cooperators, and randoms, which aligns with experimental

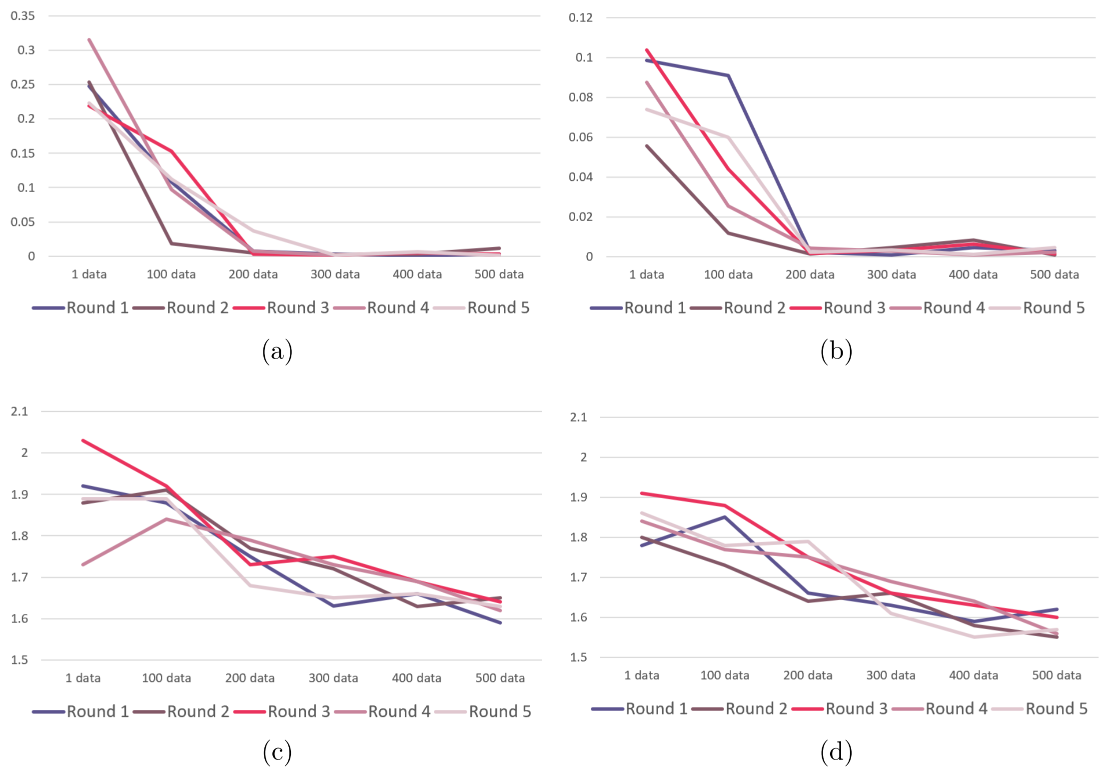


Figure 3.11: Comparison of designers' and evaluators' loss change in multiple environments. Each line chart records five individual results of the same experiments. The first row presents the cross-entropy loss change in the trust evolution environment, while the second row displays the MSE loss change in the maze run environment. The first column represents the designer, followed by the evaluator. (a) Designer cross-entropy loss. (b) Evaluator cross-entropy loss. (c) Designer MSE loss. (d) Evaluator MSE loss.

observations.

The rule designer is flexible in creating rules of varying dimensions, such as the one-dimensional win rate in the MR and the 16-dimensional parameter in the TE. Both MSE loss and cross-entropy loss are applicable to the rule designer and evaluator. Fig. 3.11a and Fig. 3.11b illustrate the progression of cross-entropy loss during the training phase in the TE environment, while Fig. 3.11c and Fig. 3.11d concentrate on the MSE loss in the MR. The decline in all curves attests to the improved performance of both evaluators and designers throughout the training process. Typically, the decrease in evaluators' losses precedes that of designers' losses due to the former's involvement in the backwards pass of the designer. In essence, designers' performance is contingent on that of the evaluators. The MSE

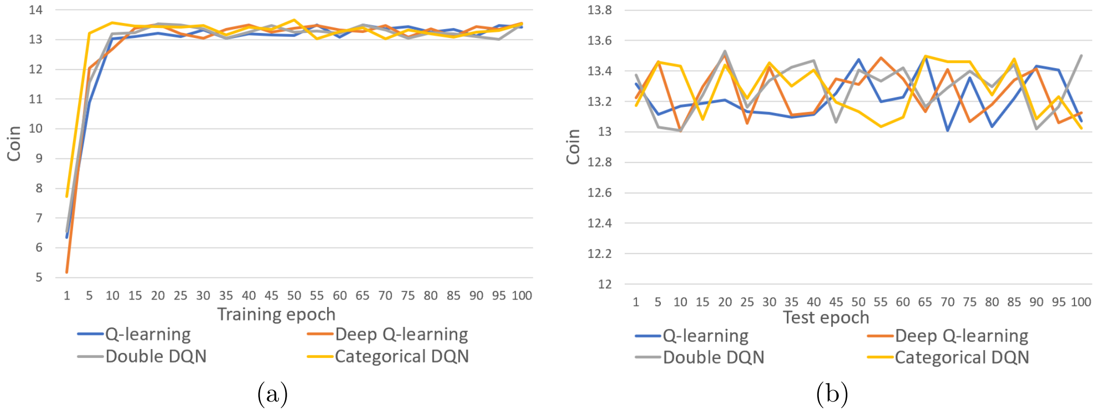


Figure 3.12: Comparison of testers' coin changes in a well-designed MR environment. Four reinforcement learning algorithms participate in the comparison: Q-learning, deep Q-learning, double DQN, and categorical DQN. The first figure records the agents' coin numbers during the training, and the second figure records the testing stage. (a) Tester comparison in training. (b) Tester comparison in testing.

loss in the MR environment undergoes a more substantial change compared to the cross-entropy loss in the TE environment during training, primarily because the designer evolves the rules based solely on the win rate. Consequently, the rule evolution process can be considered a black-box optimisation task. Increasing rule complexity exacerbates the challenge faced by designers in discerning the appropriate rule-generation strategies.

Comparing the performance of human and RGN designers, we established fixed game result expectations in the TE as the design criterion and evaluated their well-designed rules. As evident in Table 3.4, the RGN designer consistently outperforms random generation and untrained models. However, human designers occasionally produce comparable results and can even surpass the RGN designer, particularly when they iteratively test their rules. Increasing the cooperation rate logically entails a decrease in the presence of cheaters, randoms, and grudgers. Designers can also manipulate the payoff value to incentivise cooperation over cheating by offering more coins as rewards. A subtle alteration in the design requirement from 100% to 80% cooperator proportion at the game's conclusion distinguishes the last two sections in the table. This led to the RGN designer emerging victorious. We conclude that the AI designer excels when design requirements are not extreme, more abstract, or when the causality between input rules and output game results is obfuscated.

The performance of RL agents is evaluated in a fixed game rule, which was generated by the well-trained RGN designer. Fig. 3.12a illustrates the efficiency of these agents during the training phase. Notably, the categorical DQN emerges as the most efficient, followed by the double DQN, deep Q-learning, and Q-learning. However, this performance gap diminishes after 10 training epochs. After that, the agents exhibit comparable performance within the game. As evident from Fig. 3.12b, during the game testing phase, the agents' performance converges to a similar value. Because the game environment proposed in this chapter involves only a few rules, even Q-learning agents can achieve reliable performance when testing the generated rules.

3.5 Conclusion

This chapter proposed an innovative rule generation framework, RGN, that leverages generative models, reinforcement learning models, and game environments to address the challenges of no dataset, rule translation, and unreasonable expectations in automated rule design, evaluation, and evolution, in accordance with controllable expectations. We initially refined and notated three fundamental elements of the rule generation task and established two digital environments, maze run and trust evolution, for implementation and demonstration. A well-trained rule designer can generate rules aligned with expectations, except in some unreasonable circumstances. By utilising the environments, AI and NPC, as well as human players, can engage in the generated game rules. Moreover, we observed the rule evolution pattern during training and the competition among Q-learning agents within the game.

For future work, we intend to explore deeper into unreasonable expectations, which were discovered during the research. Unreasonable expectations, such as requiring two different personality groups to win at the same time in TE, can confuse RGN in both the training and evolving stages. It also leads to an unfair evaluation result during the test. We also aspire to investigate the causality between each dimension within the vector and its corresponding rule. Additionally, we are intrigued by the potential effects of sampling data from the environment and pretraining the designer to generate certain unique conditions. Moreover, we intend to scrutinise the relationships among different categories of rules, such as those pertaining to population dynamics and payoff distribution. By introducing some seemingly extraneous rules into the system, we aim to enhance our understanding of the causality at play.

CHAPTER 4

Triadic Reciprocal Dynamics: The AI Framework for Social Rule Evolving

Prologue

Chapter 4 introduces the Triadic Reciprocal Dynamics (TRD) framework, an AI-driven system for dynamic social rule generation and evolution. First, the necessity of modelling continuously changing social rules—beyond static RL scenarios—is established through a review of social science and adaptive learning. Next, the chapter presents how generative neural networks and reinforcement learning agents are integrated to form a triadic cycle of Rule, Strategy, and Evaluation. The TRD architecture is then detailed: a neural-network Rule Designer generates and refines rule parameters; a Game Environment simulates interactions among human, NPC, and AI agents; and an Evaluator predicts key social metrics (e.g., cooperation rate, income distribution). A multi-dimensional evaluation paradigm is introduced to assess rule efficacy, agent performance, and evaluator accuracy. Finally, the chapter demonstrates the TRD platform—Maze Run and Trust Evolution environments—for real-time rule synthesis, strategy exploration, and iterative feedback, establishing a novel methodology for studying and optimising social behaviours in simulated settings.

Rule design plays a critical role in shaping individual behaviour, group strategies, and the dynamics of cooperation and competition in social systems. In addition to traditional approaches such as questionnaires and statistical analyses, there is a growing adoption of social modelling techniques—particularly reinforcement learning (RL)—to explore strategies under fixed-rule conditions in sociology and psychology. However, the study of dynamic environments, where rules, evaluation metrics, and strategies continuously evolve, remains limited. Here, we introduce triadic reciprocal determinism into the Strategy-Evaluation-Rule (SER) framework and develop an AI-driven platform called Triadic Reciprocal Dynamics (TRD) for dynamic rule generation tasks. TRD autonomously generates, evaluates, and evolves rules in dynamic environments, capturing the complex interactions among strategies, evaluations, and rules. The system consists of three core components: the Rule Designer, the Game Environment, and the Evaluator. Additionally, we provide a multi-dimensional evaluation paradigm to assess both rules and agent strategies. This interdisciplinary approach offers a novel tool for understanding and shaping social behaviours in simulated environments, paving the way for significant advancements in social rule design and evaluation.

4.1 Research Background

Understanding the intricate dynamics of human behaviour within social systems has long been a central focus of psychological research [314] [315]. Social modelling, which examines how individuals interact, influence each other, and contribute to collective outcomes, provides invaluable insights into the mechanisms that foster cooperation, competition, and societal cohesion [316]. Self-Determination Theory (SDT) offers a comprehensive framework for understanding the interplay between behaviour, personal factors, and environmental influences [317]. This context-dependent theory articulates how environmental conditions can either facilitate or hinder the fulfilment of these needs, leading to variations in individuals' motivation and subsequent behavioural outcomes [179]. In this context, rewards play a pivotal role in motivating behaviour: intrinsic rewards arise from internal satisfaction

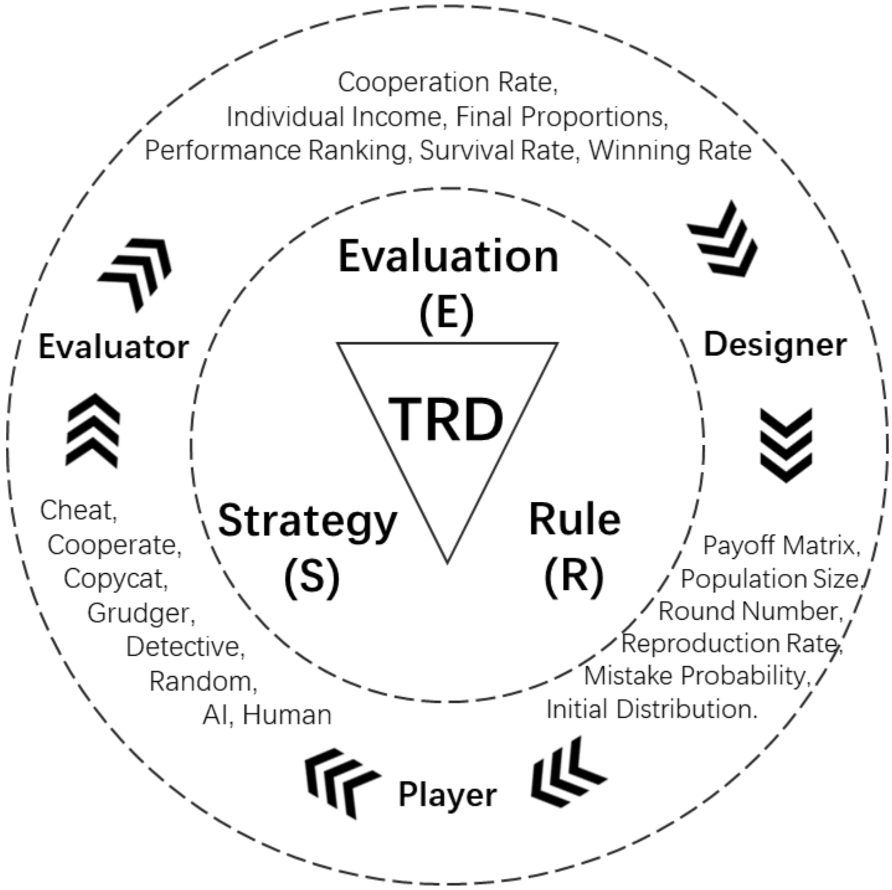


Figure 4.1: Overview of the TRD system. Our approach focus on Rule, Strategy, Evaluation, and their relationships. Rule defines the constraints and principles. Strategy represents the decision-making processes that operate within the given rules. Evaluation, assesses the effectiveness of the strategies, providing feedback that can refine both strategies and rules. The framework forms an iterative cycle, ensuring continuous adaptation and optimization in dynamic environments.

and personal growth, while extrinsic rewards are external incentives imposed by the environment [318]. Balancing these reward types is crucial for understanding how individuals develop strategies to maximise their well-being and navigate complex social landscapes [319]. However, traditional models tend to oversimplify social interactions by not accommodating the nuanced, multifaceted relationships that exist among individuals operating within these networks [320, 321], which highlights the need for a robust framework capable of elucidating these complex interactions.

Advancements in artificial intelligence, particularly in generative neural networks and reinforcement learning, offer promising avenues to address these challenges [322, 323]. Generative neural networks, such as Generative Adversarial Networks and Variational Autoencoders, excel at creating complex data distributions and uncovering latent structures within data [324, 325]. Reinforcement learning (RL), on the other hand, enables agents to learn optimal behaviours through interactions with their environment by maximising cumulative rewards [326]. By integrating these AI methodologies [327, 328], it becomes feasible to develop systems that can autonomously generate and refine social rules, simulate diverse interactions, and predict the outcomes of various rule sets. This cooperation between generative models and RL not only enhances the ability to design sophisticated social simulations but also facilitates the creation of adaptive frameworks capable of evolving alongside changing social dynamics [329].

Reinforcement learning (RL) traditionally relies on fixed rules, assuming a stationary environment and constant rewards. In reality, environments change continuously, making static policies less effective [330, 331]. Fixed-rule RL models can struggle to adapt when conditions shift, leading to suboptimal performance. While methods like those from Yang et al. [332] and Ahmed et al. [333] work well in stable settings, they face challenges in dynamic scenarios. Adaptive strategies such as continual adaptation and meta-learning offer some relief but often require extensive retraining or rely on narrow assumptions [334, 335]. Our Rule Generation Project addresses these limitations by dynamically synthesising new rules in real time, ensuring that the decision-making framework remains aligned with evolving environments. This approach, inspired by lifelong learning research [336] and recent studies on dynamic adaptation, offers a flexible alternative to conventional RL methods.

In response to the limitations of existing social modelling techniques, we introduce the Triadic Reciprocal Dynamics, an innovative AI-driven system designed to autonomously generate and optimise social interaction rules. The triadic reciprocal determinism theory [337] was integrated into the core structure as shown in Fig. 4.1. It comprises three components: a neural network-based rule designer that generates rule parameters based on specific requirements, a game environment that simulates trading interactions among diverse players, including human participants, fixed-strategy NPCs, and AI agents, and a rule evaluator that predicts key social metrics such as cooperation rate and individual income. This integration ensures that strategy, evaluation, and rule formation mutually influence one another, capturing the dynamic interplay inherent in social interactions. Through iterative training, the rule evaluator aligns its predictions with actual game outcomes, thereby guiding the rule designer's optimisation process. We present a multi-dimensional evaluation paradigm designed to assess the performance of distinct system components. The rule designer, evaluator, and AI players are evaluated using metrics tailored to capture their specific roles within the system. This approach not only advances research across diverse domains but also enhances the training efficiency of integrated AI models. By combining these functionalities into a versatile platform, the TRD system not only advances the field of social modelling but also provides a valuable tool for interdisciplinary research in artificial intelligence, psychology, and sociology.

4.2 Result

4.2.1 Task setting

To investigate the impact of rules, strategy, and evaluation within the TRD system, we evaluate our model based on the Trust Evolution Game, which provides a game engine to examine how agents make decisions based on different rules within cooperative, competitive, and adaptive frameworks. The Trust Evolution enables the exploration of trust dynamics and cooperative behaviour within a controlled setting where human researchers from different domains, programmed NPC, and AI players interact. By analysing how different rule parameters influence these interactions, we can gain deeper insights into the mechanisms that drive social cooperation and the balance between self-interest and collective well-being.

Trust Evolution. The Trust Evolution environment simulates decision-making in social interactions, characterised by parameters such as payoff matrix, population size, number of rounds, reproduction rate, and mistake probability. The mistake probability quantifies the likelihood of agents selecting unintended actions, introducing an element of unpredictability into interactions. The number of rounds

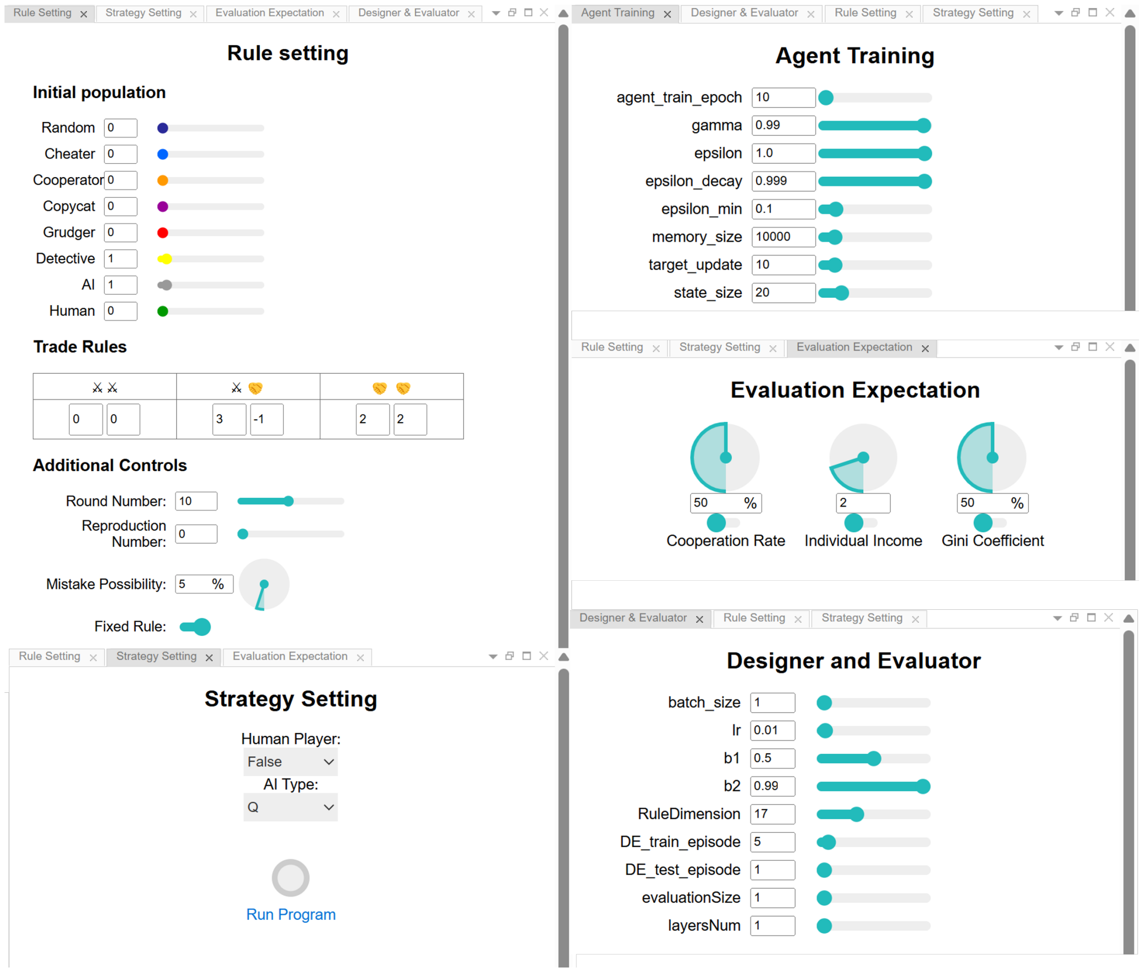


Figure 4.2: Visualisation of the input panel of the TRD interface, showcasing adjustable parameters for initial population and trade rules, agent training configurations, evaluation criteria, and real-time strategy settings. This integrated interface enables users to fine-tune rules, monitor agent performance, and oversee the evolving game environment in a single unified platform.

dictates the frequency and duration of trading interactions among agents. In each interaction, agents choose between two actions: *cheat* or *cooperate*, with outcomes determined by a predefined payoff matrix that rewards or penalises based on the combination of chosen actions. The environment incorporates six types of non-player characters (NPCs), each embodying distinct behavioural patterns: *random*, *cheater*, *cooperator*, *copycat*, *grudger*, and *detective*. These behaviours range from consistent cooperation to strategic retaliation following betrayal, thereby introducing a diverse array of interaction dynamics. For instance, *copycats* mimic the actions of others, *grudgers* cease cooperation after being cheated, and *detectives* employ more sophisticated strategies to uncover cheating behaviours. At the end of each game round, a

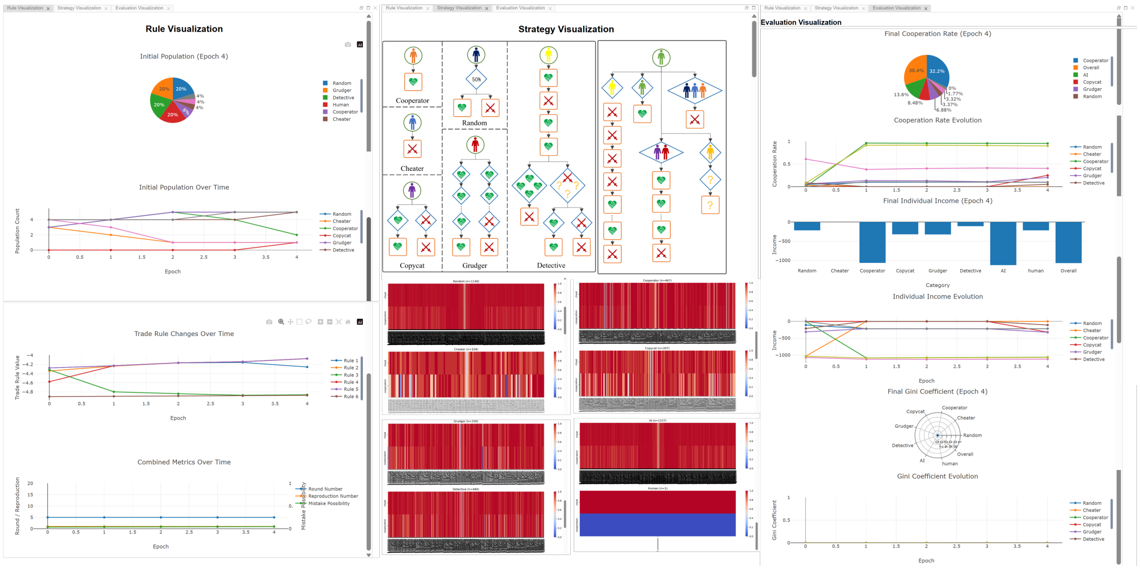


Figure 4.3: An illustration of the TRD interface's output panel, displaying real-time visualisations of rule configurations, strategy distributions, and evaluation metrics. Panels include rule and strategy charts, cooperation rates, final income, and Gini coefficient measurements over multiple epochs, providing immediate feedback and facilitating iterative refinement of game design and agent training.

selection mechanism evaluates agent performance based on predefined metrics such as total income and cooperation rate. Poorly performing agents are eliminated, while the top-performing agents are selected to reproduce, replacing the least successful members of the population. This reproduction rate ensures that successful strategies are propagated, fostering an evolutionary optimization of agent behaviours over successive generations.

4.2.2 TRD platform

The interface of the proposed TRD system consists of two primary components: a settings page and a demonstration page. The setting page serves as the input interface for the backend program and comprises five functional sub-panels: Rule Setting, Strategy Setting, Agent Training, Evaluation Expectation, and Designer & Evaluator. Figure 4.2 presents an exemplary setting page. In the Rule Setting sub-panel, users can define parameters related to rule generation, including the initial populations of the eight personality-based agent types, trade rules, the number of trading rounds, mistake probability, and reproduction number. The Strategy Setting panel enables users to select various AI algorithms, such as Q-learning, DQN, and A2C, and also allows the inclusion of human participation. The Agent Training panel provides settings for training parameters such as the number of epochs, gamma, epsilon, decay, memory size, target update, and state size. The Evaluation Expectation sub-panel facilitates users to define desired societal outcomes by selecting and adjusting evaluation metrics, including Cooperation Rate, Individual Income, and Gini Coefficient, which respectively reflect societal trust, productivity, and fairness. Users can prioritise one or multiple metrics to guide rule evolution iteratively toward desired social objectives. Finally, the Designer & Evaluator panel manages the training parameters of the rule-designer and evaluator AI modules, allowing users to configure evaluation expectations explicitly, such as batch size, learning rate, decay of first-order momentum of gradient, training and testing epoch, Evaluation size, and hidden layer size. After setting these parameters, the TRD system exports user configurations and trains the rule designer, evaluator, and AI agents within the game environment via the underlying SER framework. Experiment results are then visualised and presented interactively in the Demonstration Page, enabling comprehensive analysis and evaluation.

The demonstration page serves primarily to visualise and present experimental outcomes generated by the backend program. Similar to the setting page, it comprises three functional sub-panels: Rule Visualisation, Strategy Visualisation, and Evaluation Visualisation, corresponding precisely to the three core components within our Strategy-Evaluation-Rule framework (Fig. 4.3). The Rule Visualisation panel illustrates the rule generation and modification process through a series of informative charts. Specifically, a pie chart represents the initial population distribution of the eight agent personalities at the latest epoch, while a line chart below tracks these initial populations across all epochs from the beginning to the conclusion of training. Considering that the complete trade rule comprises six individual parameters, we present their evolution jointly through a unified line chart. Additional parameters—including round number, reproduction number, and mistake possibility —are visualised across training epochs, employing dual Y-axes due to the distinct numerical domains of these parameters. In the Strategy visualisation panel, we dis-

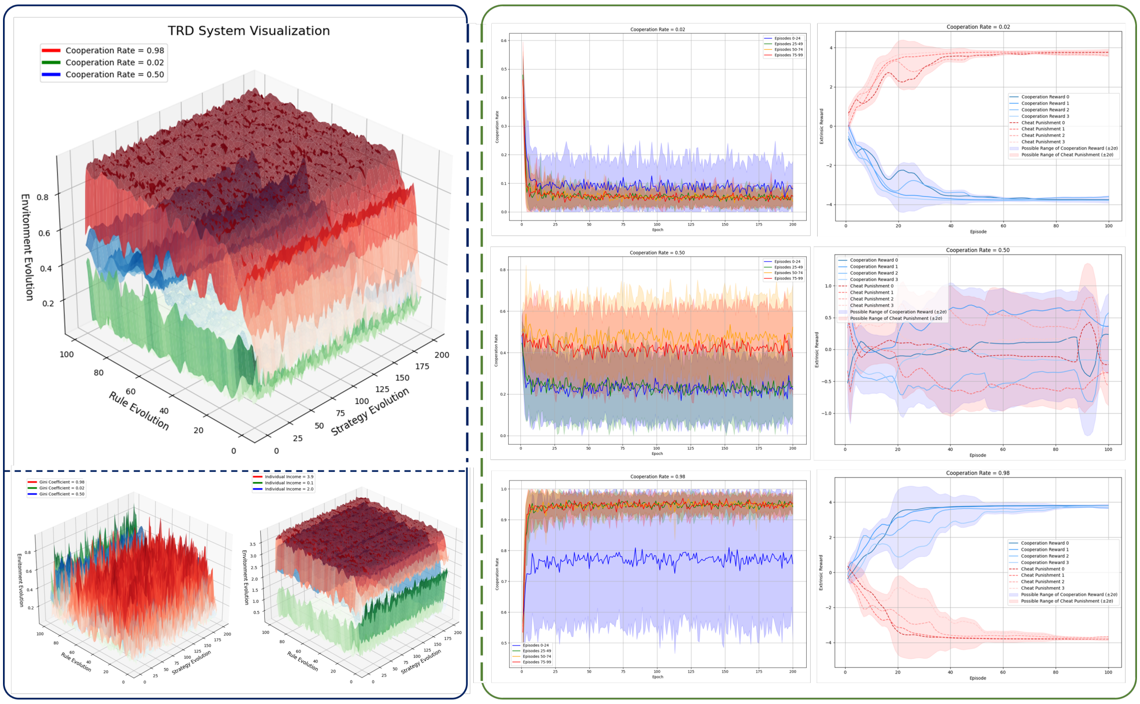


Figure 4.4: TRD System Visualisation: The left panels present three-dimensional surfaces of Q-learning agent performance across varying cooperation rates and training epochs, while the right panels illustrate detailed training trajectories and reward distributions. Shaded regions indicate variability in outcomes.

play the strategies adopted by NPCs, human participants, and AI agents. Our TRD system provides two distinct game environments: an interactive Unity-based version allowing human participation, and a Python-based mini-environment optimised for efficient AI model training. Consequently, we employ behaviour trees—commonly utilised within the game development community—to visualise the decision-making processes of NPCs. Human strategies tailored to specific agent personalities can similarly be expressed through customised behaviour trees. Regarding AI agents, visualisation techniques vary according to the reinforcement learning algorithm employed. For instance, Fig. 4.3 illustrates a Q-learning agent's strategy by converting its Q-table into a heatmap, clearly revealing action preferences across different states. Lastly, the Evaluation visualisation panel provides insights into three primary evaluation metrics: final cooperation rate, individual income, and Gini coefficient, representing societal trust, productivity, and fairness, respectively. This panel includes pie charts depicting the latest epoch's cooperation rate distribution among personalities, line charts tracking these cooperation rates and individual incomes

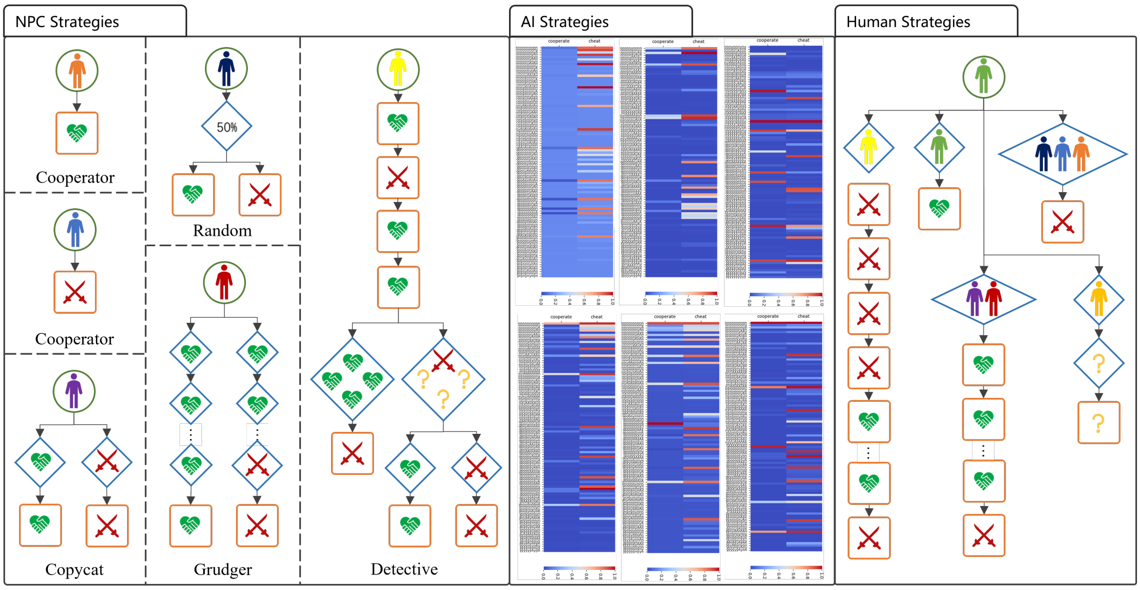


Figure 4.5: visualisation of player decision trees and corresponding strategy distributions. The left panels illustrate how different personality archetypes make sequential decisions, while the right panels depict the distribution of these strategies across multiple training epochs. The heat map illustrates the q-table information and shows how player behaviours evolve.

across training epochs, and charts illustrating both the final Gini coefficient and its evolution throughout the iterative training process.

4.2.3 Triadic Reciprocal Dynamics based model

In our investigation of the interactions among strategy, evaluation, and rule generation, we identified that these three elements form a dynamic and interconnected cycle. Although theoretically independent, these components significantly influence each other throughout the rule-generation process. Specifically, the rules generated by the designer directly influence player strategies within the game environment. These varying strategies lead to different game outcomes, which are then assessed by multiple evaluation metrics. The evaluated results subsequently inform the rule designer, guiding the creation of new rules, thus perpetuating the cycle. Figure 4.4 illustrates this dynamic interplay within the Trust Evolution environment. Here, rules are represented as a 17-dimensional vector, which we convert into a unified extrinsic reward value, representing the reward provided by varying environmental conditions. Additionally, the original fixed trading rule, analogous to the classic

Prisoner' s Dilemma, is treated as a baseline reward reflecting outcomes inherent to each player's interaction. The combination of both reward types is employed to train AI agents.

The left column of Fig. 4.4 visualises the TRD system's training dynamics guided by different evaluation metrics—namely, cooperation rate, Gini coefficient, and individual income. Each plot has three axes representing strategy evolution, rule evolution, and environment evolution, respectively. Using the first plot as an example, the visualisation is segmented into three layers corresponding to target cooperation rates of 98%, 50%, and 2%, depicted in red, blue, and green, respectively. Training starts from an initial state where both rule evolution and strategy evolution are zero. During each training epoch, the rule designer generates rules to train and test AI agents, with the results reflected along the environment evolution dimension. The fully trained designer, indicated by a rule evolution of 100 and a strategy evolution of 200—demonstrates robust capability in generating effective rules aligning with the targeted evaluation metric.

The central column displays the adaptation process of AI players within the game environment following the application of rules generated by the well-trained rule designer, for target cooperation rates of 2%, 50%, and 98% (from top to bottom). During initial epochs, the performance of AI players exhibits considerable fluctuations and strategies misaligned with long-term objectives. With continued training, however, agents gradually converge on strategies that maximise total reward. Notably, rules targeting intermediate cooperation rates, such as 50%, result in significant variability during training, reflecting the inherent complexity of balancing incentives for cooperation and defection. In contrast, rules promoting near-unanimous cooperation or defection require comparatively fewer iterations to stabilise. Finally, the right column illustrates the evolution of extrinsic reward values continuously generated by the rule designer throughout the training process. Across multiple experiments with varying cooperation rate requirements, consistent patterns in extrinsic reward trajectories emerge, reinforcing the TRD system' s reliability and stability in rule generation tasks.

4.2.4 Strategy

In the TRD system, the set of actions that players choose in response to various states is defined as their "Strategy." By aggregating the rational decisions made by all agents in the environment, the system extracts knowledge, identifies patterns, and uses these insights to evaluate the quality of the generated rules. In our Trust Evolution game, three distinct types of players are involved: NPCs, AI, and Humans. NPCs are designed to exhibit fixed, situation-specific responses represented by six distinct personality types, while AI players embody trainable models that continuously adjust their strategies during training. Human players, on the other hand, participate directly either by designing their own behaviour trees or by making choices in each trading interaction.

Figure 4.5 illustrates the strategy representations for these three groups: behaviour trees are used to depict the strategies of NPCs and Humans, whereas heat maps are employed to visualise the strategies of AI players. For example, within the NPC category, the Cooperator consistently opts for cooperation regardless of past interactions, whereas the Cheater invariably defects. The Random type, as implied, makes decisions based on chance. The Copycat starts with cooperation and subsequently mimics its opponent's last action, while the Grudger, though initially cooperative, permanently defects after a single betrayal. The Detective, the most complex among NPCs, follows a probing pattern in its initial rounds before switching to a Copycat-like strategy.

In contrast, human players may leverage visual cues from character appearances to anticipate personalities and adapt their strategies accordingly; for instance, cooperating with other humans, defecting against Random, Cheater, and Cooperator types, cooperating with Copycat and Grudger until the final round (at which point they defect), and following a mixed strategy for Detective opponents. Although the optimal strategy against dynamic AI players remains uncertain, our experiments, which compared Q-learning, DQN, and A2C, show that reinforcement learning, based on a Markov decision process, enables trained models to evaluate the merits of different actions in a given state. In classic fixed-rule conditions, these AI models, as visualised through heat maps that highlight preferred actions (with red indicating higher selection probability), consistently outperform NPCs and Human players in both single-role and multi-role scenarios, and they exhibit improved performance during the rule evolution process.

A key innovation of our TRD system is its multi-dimensional evaluation paradigm, enabling simultaneous assessment of rule efficacy, AI player performance, and the adaptive capacity of rule designer and evaluator modules. In the Trust Evolution game, for instance, rules are primarily measured by cooperation rate, Gini coefficient, and individual income, representing societal trust, fairness, and collective efficiency, respectively. Beyond rule outcomes, the framework also captures player-level metrics such as coins accrued each round, final rankings, and extrinsic rewards, a combination that illuminates how different personalities (e.g., Cheater, Cooperator) and varying AI algorithms (Q-learning, DQN) respond to rule changes. Rule designer performance, in turn, depends on its ability to converge on desired rules within a certain number of epochs, while the evaluator is gauged by how accurately it predicts final outcomes given a rule vector, compared against observed in-game results. By allowing dynamic adjustments across these varied metrics, the system fosters interactive analyses of complex behavioural patterns, thus supporting deeper insights into the dynamic interplay among rules, strategies, and social outcomes. To demonstrate this paradigm, we conducted experiments in which rules and strategies were co-evolved under different target objectives—maximizing cooperation rate, minimising the Gini coefficient, or balancing income distribution. Across multiple epochs, the rule designer generated candidate rules, which were tested in the environment by AI players and, if applicable, human participants. The evaluator then predicted game outcomes based on each rule vector, and these predictions were compared to actual performance metrics—cooperation rates, individual incomes, and fairness indicators—recorded during simulation. Our results consistently showed that the system could converge on rules matching the designated objectives within a moderate number of training epochs. Specifically, in trials aiming to maximise cooperation, the rule designer introduced higher extrinsic rewards for cooperative acts, leading to stable, high-cooperation equilibria. Conversely, attempts to minimise the Gini coefficient promoted equitable strategies, effectively curbing disparities in final incomes. These observations validate the system' s ability to generate and refine rules autonomously while offering comprehensive evaluations of agent behaviour, further underscoring the robustness and utility of our multi-dimensional evaluation paradigm.

4.3 Discussion

Our research introduces the Triadic Reciprocal Dynamics (TRD) framework, an AI-driven approach designed to dynamically generate and optimise rules within complex social systems. Integrating principles from Bandura's Triadic Reciprocal Determinism and Self-Determination Theory, the TRD system incorporates three core innovations: the Strategy-Evaluation-Rule (SER) framework, a novel multi-dimensional evaluation paradigm, and a versatile platform supporting interdisciplinary research. The SER framework consists of a neural-network-based Rule Designer that autonomously creates adaptive rules, a Game Environment simulating diverse social interactions, and an Evaluator module that predicts and assesses social outcomes such as cooperation rates, fairness, and collective productivity. This iterative interaction between rules, strategies, and evaluations enables the system to continuously refine social dynamics, capturing complex feedback loops essential for realistic simulations.

Through experiments in the Trust Evolution Game environment, our platform demonstrated the efficacy of dynamically evolving rules to achieve targeted social objectives, such as promoting cooperation or minimising social inequality. Our innovative multi-dimensional evaluation paradigm simultaneously assesses rule effectiveness, agent performance across multiple personality types and AI models, and the predictive accuracy of evaluation modules. These innovations not only advance theoretical understanding of reciprocal interactions within dynamic social systems but also offer a robust computational tool for real-time adaptive rule generation. Thus, the TRD framework provides significant contributions to computational sociology, psychology, and artificial intelligence, enabling future research to extend beyond simulations toward practical implementations in diverse real-world social contexts.

4.3.1 Further applications and impact

The TRD framework holds significant promise for diverse real-world applications spanning multiple disciplines, from computational sociology and psychology to public policy design and organisational management. In social sciences, the platform provides a powerful tool for modelling complex phenomena, such as trust-building, cooperation emergence, and conflict resolution, offering policymakers and researchers actionable insights for interventions and strategy planning. Organisations can utilise the TRD platform to dynamically design and evaluate internal rules or incentive structures, optimising team performance, fairness, and collaboration. Furthermore, in artificial intelligence research, our approach contributes an effective methodology for lifelong adaptive learning in dynamic environments, which is essential for developing robust autonomous systems capable of sustained adaptation.

The broader impact of this research extends to societal governance and interdisciplinary education, enabling stakeholders to experiment with and refine policies interactively before real-world implementation. For example, cities or governmental bodies could leverage the TRD platform to simulate community responses to new regulations, optimising policies that enhance societal cohesion or economic productivity. Educational institutions could integrate the TRD system into curricula, providing students with hands-on experience in understanding complex social dynamics through computational experiments. By bridging theory and practice across fields, the TRD framework thus stands to fundamentally transform how we understand, predict, and influence the evolution of social behaviours, ultimately guiding societies towards greater cooperation, resilience, and collective well-being.

4.3.2 Limitations

While the TRD framework significantly advances dynamic social modelling, several limitations should be considered. The current evaluation and validation have been conducted primarily within controlled, simulated environments, specifically the Trust Evolution Game. The complexity of real-world social interaction, encompassing diverse cultural, emotional, and contextual factors—exceeds these controlled scenarios, potentially limiting the direct applicability and generalisability of our results. Future studies should extend validations to real-world settings or more complex scenarios involving human populations to confirm ecological validity and robustness. Although the TRD framework effectively models the reciprocal dynamics among rules, strategies, and evaluations, the computational cost of continuously adapting to complex social environments remains considerable. Computational efficiency could become a bottleneck, especially when scaling to larger populations or incorporating richer behavioural representations. Moreover, the dependence on predefined evaluation metrics implies potential biases or oversights, as critical yet unforeseen social dynamics could be inadvertently excluded from the current evaluation paradigm. Addressing these limitations through broader testing, increased computational efficiency, and more inclusive metric design represents essential future directions for enhancing the TRD system's practical and theoretical contributions.

4.3.3 Ethical Analysis

. In the development of our Triadic Reciprocal Dynamics (TRD) system, careful attention has been paid to ensuring an unbiased and ethically sound framework for automated rule generation. Recognising that AI-driven rule generation in digital environments may inadvertently introduce bias, our design explicitly avoids the incorporation of socio-cultural attributes such as sex, age, race, or other sensitive factors. In our experimental game environments, Trust Evolution, player roles are defined solely by distinct behavioural patterns without any demographic markers. In the Trust Evolution environment, for example, reward and penalty structures are entirely derived from reinforcement learning metrics, thereby eliminating subjective biases from the reward system. Similarly, the six distinct NPC personalities represent algorithmic behaviour patterns rather than human socio-cultural identities. Furthermore, the TRD system employs a multi-dimensional evaluation paradigm to continuously monitor and adjust rule outcomes, ensuring that any emergent dis-

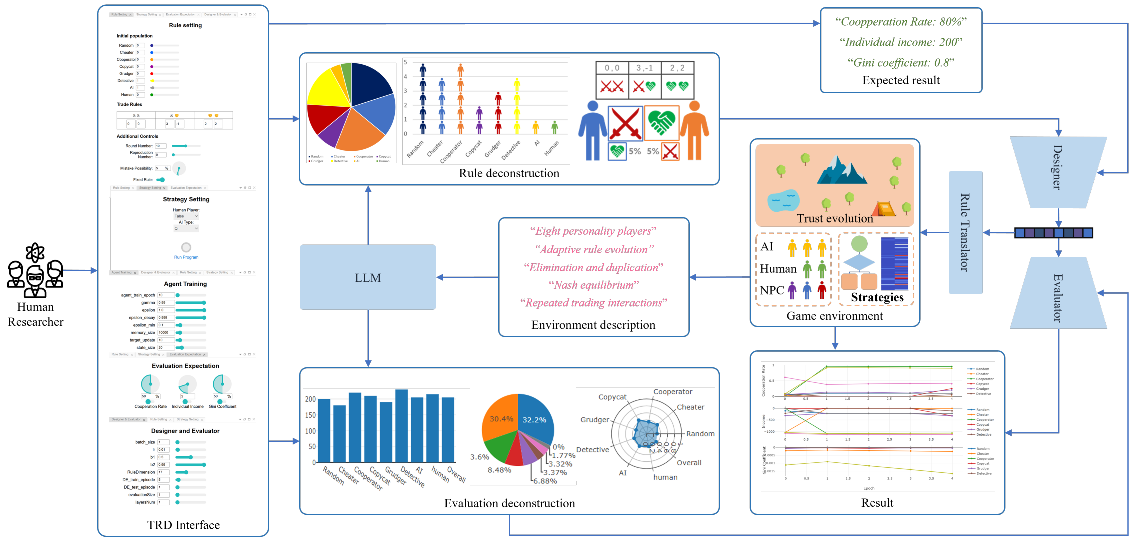


Figure 4.6: An illustration of the TRD system, showcasing the interactions among the rule designer, evaluator, environment, and reinforcement learning model. The rule designer generates rules based on embedded expectations, and the evaluator validates these rules before translating them into executable parameters for the environment. The RL model then explores the environment to learn optimal strategies, while controllability is assessed at both the environment and rule designer levels.

parities are identified and mitigated in real-time. This commitment to neutrality and fairness is fundamental to our approach, aligning our work with the ethical standards expected in the development of socially responsible AI.

4.4 Methodology

4.4.1 Preliminary

We introduce the Strategy-Evaluation-Rule (SER) framework, which comprises three interrelated components essential for automated rule generation in digital environments. **Rules** ($\mathcal{R}$) define the foundational principles and mechanisms of a game, encompassing elements such as players, maps, actions, and decisions [308]. Formally, rules are denoted as $\mathcal{R} = [r_{n,d}] \in \mathbb{R}^{N_r \times D_r}$, where N_r is the number of rules and D_r indicates their dimensionality. Rule dynamics involve creation (adding new rules), deletion (removing rules), and modification (updating existing rules), resulting in evolved rules $\mathcal{R}'$. **Strategies** ($\mathcal{S}$), representing algorithmic decision sequences selected by human or AI agents to maximize rewards under specific game conditions, can be mathematically represented as $\mathcal{S} = [s_{n,d}] \in \mathbb{R}^{N_s \times D_s}$, with N_s and D_s denoting the quantity and dimensionality of strategies, respectively [309]. Strategy enumeration depends on game information completeness, making N_s either finite or infinite. Finally, **Evaluation** (E) quantifies the effectiveness of rules through high-level heuristics (e.g., spontaneity, interruptability, continuity) determined by game parameters, expressed as $E = [e_{n,d}] \in \mathbb{R}^{N_e \times D_e}$, where each evaluation criterion assesses specific rule-strategy interactions [310]. The evaluation process is succinctly captured by $E = f(\mathcal{R}, \mathcal{S})$, indicating that evaluation outcomes depend directly on both rule sets and agent strategies, thus completing the closed-loop interaction among these three key components.

4.4.2 Framework

Figure.4.6 illustrates our TRD Platform, which consists of three primary components: a Designer for rule generation, an Evaluator for rule assessment, and the Environment. Additionally, an interactive Interface is provided for researchers to directly engage with the system and analyse the Environment's large language model (LLM). For instance, when a baseline environment such as the Trust Evolution system is integrated, the game environment description is input into the LLM, which deconstructs elements related to both rules and evaluation. Researchers can then adjust system parameters—including model training settings and environment-specific rules—via the Interface. Upon submission, the SER system incorporates these parameters along with the rule design requirements and iteratively generates refined rules. Each generated rule vector is subsequently transformed by a rule translator into rule parameters that are imported into the Environment, where AI, human, and NPC players participate in strategic exploration. All player data is recorded and analysed by the Environment, forming the ground truth for the Evaluator.

In generative tasks, the objective is to train the rule designer to create content according to specific requirements, represented by a function $g : Z \to X$ that maps random noise vectors Z to high-dimensional outputs X. This framework trains a rule designer $d : E \to \mathcal{R}$ to generate a set of rules $\mathcal{R} \in \mathbb{R}^{N_r \times D_r}$ directly from an expected outcome E. These rules, which can be created, deleted, or modified (yielding an evolved rule set $\mathcal{R}'$), are then implemented in the environment to produce an output $E' = Env\big(d(E), S\big)$, where Env denotes the evaluation function. The training objective for d is formulated as:

$$\min_{d} V(d, Env) = \log(1 - Env(d(E), S)). \tag{4.1}$$

To facilitate this process, an evaluator acts as a digital twin by mapping the rule set to evaluation scores via a function $p : \mathcal{R} \to E''$, and its objective is to minimise the discrepancy between its predictions and the actual environment outcomes:

$$\min L\big(p\big(d(E)\big),\, Env\big(d(E),\, S\big)\big). \tag{4.2}$$

This framework incorporates reinforcement learning within a Markov Decision Process defined as $\{S, \mathcal{A}, \mathcal{T}, R, p(s_0), \gamma\}$, where S, $\mathcal{A}$, $\mathcal{T}$, R, $p(s_0)$, and γ denote the states, actions, transition probabilities, rewards, initial state distribution, and discount factor, respectively. We employ Q-learning agents that share the same action space, reward function, and perceptual domain as other players, and these agents learn a strategy mapping $P : \text{State} \to \text{Action}$ to maximise cumulative reward. This integrated approach enables simultaneous rule generation, strategy exploration, and evaluation in dynamic environments, providing a robust framework for understanding and optimising social interactions.

4.4.3 Rule deconstruction

Figure 4.7 illustrates, via the Trust Evolution environment, the rules visualisation, deconstruction, and evolution processes within our game environment. In Trust Evolution, the core rules are represented by a 17-dimensional vector: dimensions 0–7 specify the initial population, 8–13 define trade rules, and 14–16 correspond to the round number, reproduction number, and mistake possibility. The top row highlights the rule visualisation, where pie and bar charts display the proportions and counts of the initial population for each of the eight personality-based players. Larger populations often hold a higher probability of prevailing until the game' s

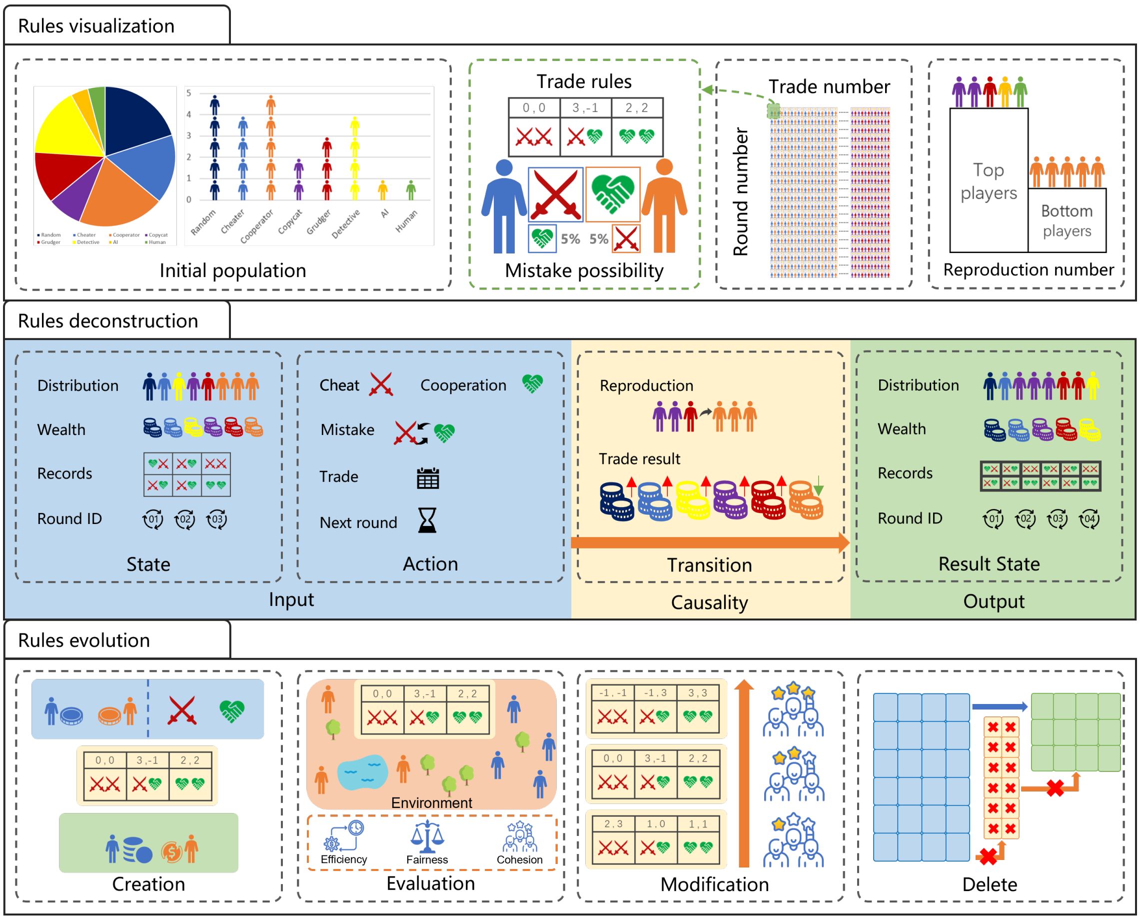


Figure 4.7: High-level schematic of the Trust Evolution environment, illustrating how population distribution, wealth, and round identifiers transition under different rule conditions. Agents choose actions, and their outcomes—captured as trade results—shape final states, including top and bottom players. The diagram highlights how inputs, actions, and rule-driven transitions lead to distinct end states and evolving strategies.

conclusion and can dominate interactions if they exceed a certain threshold. Trade rules can be summarised in a six-value table, reflecting the payoffs for various pairwise scenarios (e.g., both cheating, one cooperating while the other cheats, or both cooperating). Such rules shape AI players' propensities for cooperation or defection. Meanwhile, a nonzero mistake probability can introduce accidental betrayal, occasionally resulting in catastrophic outcomes—for instance, one misstep between two Copycat agents might trigger permanent mutual defection. The round number determines how many rounds will be played before termination, and reproduction indicates the elimination-and-duplication mechanism that replaces the lowest-ranked agents with copies of the top performers at each game's end.

In the second row, we deconstruct the rule structure through a Markov Decision Process (MDP) lens, common in reinforcement learning research. Here, rules emerge as causal mappings between observed states and actions, with transitions producing new states. In the Trust Evolution game, observable states include distribution, wealth, trading records, and round indices, whereas available actions encompass cooperation, defection, or higher-level changes such as introducing mistakes, initiating the next round, or even ceasing further trades. Transitions capture the reproduction procedure and the coin accrual resulting from trades. The AI agents' exploration of different strategies to maximise coin earnings thus becomes, in essence, a process of rule exploration. Finally, the third row presents the rule evolution lifecycle, encompassing creation, evaluation, modification, and deletion. When a new mechanism linking the environment's state, chosen actions, and subsequent state transitions is introduced, a new rule is "created." Next, the environment and AI players "evaluate" that rule by running simulations that yield performance metrics—efficiency, fairness, and cohesion. These outcomes guide the rule designer's training, prompting potential "modifications" to specific parameters (e.g., the six trade-rule values) to achieve, for instance, higher cooperation rates and improved social cohesion. Ultimately, a rule may be "deleted," severing any previously defined causal link for a particular state–action combination, effectively removing that transition from the game's rule set.

4.4.4 Whitney Embedding Theorem and Extrinsic Reward

The agent's initial motivational state is represented by an n-dimensional vector $\mathbf{M} = \{m_0, m_1, \ldots, m_{n-1}\}$, where each m_i signifies distinct motivations. Our objective is to extend this motivational vector into a higher-dimensional space by introducing additional dimensions that capture emergent motivations evolving during the learning process.

We leverage the *Whitney Embedding Theorem* [338], which guarantees that any smooth n-dimensional manifold can be embedded in a Euclidean space such as $\mathbb{R}^{2n}$ without self-intersections. For our purposes, we extend the agent's motivational vector from $\mathbf{M} \in \mathbb{R}^n$ to an augmented vector $\mathbf{M}' \in \mathbb{R}^{n+1}$, where the first n dimensions remain unchanged and the additional dimension captures a novel, dynamically evolving intrinsic motivation.

We embed the original motivation vector $\mathbf{M} = (m_0, m_1, \ldots, m_{n-1})$ into an extended vector $\mathbf{M}' \in \mathbb{R}^{n+1}$ via a smooth mapping $f : \mathbb{R}^n \to \mathbb{R}^{n+1}$, where the new component is $m_n = g(m_0, m_1, \ldots, m_{n-1})$, and g is implemented as a neural network trained by reinforcement learning to maximize long-term rewards (both intrinsic and extrinsic). Concretely, the network takes $\mathbf{M}$ as input and outputs

$$\mathbf{M}' = \big[m_0,\, m_1,\, \ldots,\, m_{n-1},\, g(m_0, \ldots, m_{n-1})\big], \tag{4.3}$$

then learns its parameters θ by minimizing $L(\theta) = \big(r(\mathbf{M}) - f(\mathbf{M}; \theta)\big)^2$ via backpropagation, where $r(\mathbf{M})$ is the reward function. In this way, m_n adapts dynamically through agent–environment interaction, promoting more complex and adaptive behaviours.

Code availability

The source code implementing the TRD system, including the interactive web interface and detailed project description, is publicly accessible via GitHub at `https://github.com/RuleEvo/TRD`.

Game availability

The interactive game environment, including character interactions, AI controls, and model-generated scenarios developed for this research, is publicly accessible on the itch.io platform at `https://glahadrt.itch.io/trust-evolution-for-\artifical-life-research`.

CHAPTER 5

Flow-Centric Rule Design: Evolving Rules for Optimal Difficulty and AI Skill Balance

Prologue

This chapter introduces a flow-centric approach to rule generation that dynamically balances game difficulty and AI skill. It begins by reviewing flow theory' s key constructs—such as challenge–skill balance, concentration, and feedback—and discusses how traditional methods rely on manual rule adjustments and physiological measures, which are labour-intensive and inflexible. We present a novel framework that embeds flow into the Strategy–Evaluation–Rule (SER) cycle, enabling automatic alignment of rule parameters with difficulty and real-time adjustment of AI strategies to match player skill. A Real-Time Flow visualisation panel is implemented to monitor engagement metrics continuously, providing interpretable feedback for adaptive rule tuning. Finally, the chapter outlines a methodology for mapping a 17-dimensional rule vector to discrete difficulty levels via correlation analysis and demonstrates how normalised skill measures guide dynamic difficulty adjustment, thereby ensuring sustained player immersion and optimal learning outcomes.

Flow theory investigates the relationship between participant skill and task difficulty to enhance engagement. Traditional approaches require game designers to manually modify rules, recruit numerous participants, and employ sensors such as EEG to assess player stress and other characteristics, ensuring that game difficulty matches player skill. These methods are labour-intensive and necessitate redesigning rule-difficulty relationships whenever new rules are introduced. Here, we propose a flow-based rule generation system that introduces a novel mechanism that automatically aligns rule parameters with game difficulty and matches strategy to player skill, thereby enabling dynamic difficulty adjustment without extensive manual intervention. Moreover, an integrated real-time Flow visualisation panel continuously monitors engagement metrics, providing immediate and interpretable feedback for adaptive rule tuning. These innovations significantly enhance the system's responsiveness and efficiency, advancing the state-of-the-art in dynamic game design.

5.1 Introduction

Flow theory, originally developed by psychologist Mihály Csíkszentmihályi, describes the state of complete immersion and engagement in an activity where the balance between challenge and skill leads to optimal experiences [339]. Its study is significant as it illuminates the mechanisms behind motivation and engagement, key factors that enhance learning, creativity, and user satisfaction in automated game design and dynamic environments [340, 341]. Contemporary psychological research typically employs interdisciplinary approaches that integrate behavioural observations with quantitative measures to assess flow states in contexts such as virtual environments and video games [342]. However, these methods have limitations, including the subjective variability of flow experiences among individuals and the lack of standardised measures across diverse contexts, which can lead to negative outcomes like frustration when challenges exceed players' capabilities [340] [343]. Integrating flow theory with social modelling and rule generation in dynamic environments addresses these constraints by enabling the design of adaptive systems that adjust challenges in real time to match individual skill levels, thereby enhancing interactive engagement and creating richer, more personalised experiences [344] [345].

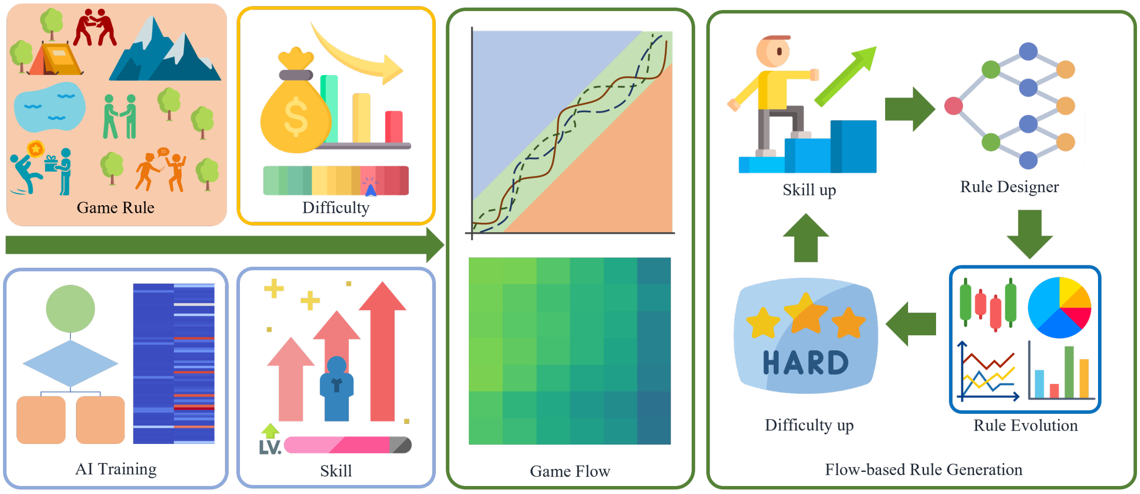


Figure 5.1: Conceptual illustration of flow-based rule evolution. Game environment rules are mapped to difficulty, while AI strategies represent skill; the balance between these factors contributes to game flow, with the rule designer continuously evolving rules to maintain the desired difficulty as players' skill levels change.

This synthesis not only deepens our understanding of flow dynamics in social contexts but also highlights the critical importance of advancing rule generation research to build self-adaptive, efficient systems for complex digital landscapes.

Game flow, an application of flow theory in gaming, emphasises creating immersive experiences that strike a balance between challenge and skill to maximise player engagement and satisfaction [346]. This concept underscores that when games are designed to sustain flow, players experience a deep sense of involvement and enjoyment, often enhanced through elements like personalised avatars [347] and compelling narratives [348]. Continuous feedback and adaptive difficulty further nurture this state [349], although care must be taken to avoid the pitfalls of excessive immersion or "dark flow," which can lead to problematic gaming behaviours [350] [351]. Such insights highlight game flow as a foundational concept in designing engaging, ethically sound gaming experiences with broad applications in education and therapy [352].

Social modelling, which emphasises the influence of societal structures on individual behaviour, is a critical concept in understanding user interactions within digital environments. In social psychology, this is typically achieved through experimental designs and cross-cultural analyses that examine how social conflict, cultural differences, and personalised gamification strategies shape individual responses—approaches exemplified by Trakšelys [353] in his exploration of social conflict theories, and Chow and Huang in their development of expert gamification systems [354]. Common to these methods is the effective use of game mechanics to enhance engagement and satisfaction [355, 356], yet they share limitations, notably the variability in individual responses due to personal and cultural differences, as well as the delicate balance required to sustain flow without triggering excessive competition [357, 358]. While these techniques have proven valuable in advancing our understanding of digital interactions, their inherent constraints underscore the need for a dynamic Rule Generation system that can adapt to evolving user behaviours and social influences, ultimately offering a more responsive and personalised digital environment.

Reinforcement learning (RL) is a mathematical framework for experience-driven autonomous learning in which agents develop decision-making strategies through trial-and-error interactions with their environment [267]. Its inherent characteristics—such as self-improvement, adaptability, and robustness in dynamic settings—render RL particularly effective for testing and refining AI behaviour, as evidenced by its successful application in challenging domains such as Atari games [268]. Deep learning has further scaled RL to address decision-making in high-dimensional state and action spaces, with popular algorithms like the deep Q-network (DQN) and trust region policy optimisation (TRPO) finding extensive use in game design. Moreover, multi-agent reinforcement learning (MARL) extends these principles to settings where multiple learning agents interact within a shared environment [269]. For instance, Park and Kwon [359] demonstrate that RL-enabled AI systems can adjust to social cues and feedback to optimise decision-making processes in educational and interactive contexts, while Burgon et al. [360] underscore the importance of evaluating RL models across diverse decision spaces to ensure their generalizability. Additionally, Wu et al. [361] illustrate that the dynamic testing environments facilitated by RL allow for continuous performance assessment and strategy optimisation, though challenges such as data availability and the need for diverse training datasets remain critical considerations [362, 363].

Flow theory aims to enhance participant engagement in gaming and educational contexts. In particular, game flow research focuses on identifying and maintaining the balance between game difficulty and player skills. The flow zone is dynamic due to the continual change of player performance as the game progresses. Recent work [364] leveraged physiological data from EEG and heart rate sensors, along with in-game behavioural records, to implement dynamic difficulty adjustment (DDA). This research paradigm requires extensive data collection cycles and professional involvement, as individual participant differences significantly influence experimental outcomes. Moreover, when new rules are introduced during game updates, the dynamic difficulty must be recalibrated. This project integrates game flow into a rule generation system by employing reinforcement learning within the game environment to conduct tests, using strategy to measure player skill, and utilising a rule designer for generating and adjusting rules. The system analyses the relationship between rules and difficulty, ultimately training AI to achieve dynamic difficulty adjustment. The contributions can be summarised as follows

- Flow theory is integrated into the rule generation system to model dynamic difficulty by capturing the co-evolution of perceived challenge and agent skill during gameplay, thereby establishing a foundation for adaptive intrinsic motivation mechanisms that sustain optimal engagement.

- The flow-based SER framework incorporates a dynamic difficulty adjustment mechanism via rule evolution, enabling real-time adjustment of game rules in response to agents' learning progress. This flow-based difficulty adjustment approach continuously aligns environmental challenges with evolving skill levels, maintaining the challenge–skill balance and preventing states of boredom or excessive difficulty.

- A real-time Flow visualisation panel is implemented in the interface. This interactive system maps key metrics such as difficulty and agent performance onto a two-dimensional flow space in real time, providing interpretable feedback that facilitates on-the-fly adjustments to optimise both the training process and the end-user experience.

The remainder of the chapter is structured as follows. Section 2 reviews existing methods for game flow, social modelling, generative models, and AI-based strategy exploration. In Section 3, we present details for the proposed methodology, including the flow-based SER framework, difficulty mapping, and skill measurement. After that, we demonstrate details about our digital environment and the experimental results in Section 4. Finally, we place important conclusions and discuss possible future works in Section 5.

5.2 Related work

5.2.1 Related tasks and Applications

Game flow refers to the application of flow theory in game design, where the optimal balance between challenge and skill creates immersive, engaging, and satisfying experiences for players [346]. Early work extended Csikszentmihályi' s framework into digital cooperative contexts, with Kaye [365] demonstrating that cooperative gaming not only promotes individual immersion but also stimulates collective engagement through social mechanisms. Building on these foundations, recent studies have incorporated blockchain technology to enhance gameplay by streamlining secure and transparent in-game transactions [366], while virtual reality has reinvigorated research into immersive experiences, with Wehden et al. [367] and Shelstad et al. [368] highlighting VR' s potential to deepen immersion despite challenges like cybersickness. Concurrently, advances in artificial intelligence have transformed game design by enabling adaptive experiences and dynamic interactions with non-player characters, as evidenced by Jagli et al. [369], which underscores AI's role in maintaining the critical balance between challenge and skill. Moreover, developments in game engine technologies, as detailed have provided robust frameworks for building both 2D and 3D games that incorporate real-time adaptive mechanisms, a trend further supported by educational game research in STEM contexts [192] [370]. Finally, emerging multiplayer platforms [371] illustrate how integrated digital infrastructures facilitate smooth gameplay and effective collaboration, collectively demonstrating that modern game design is increasingly technology-driven and focused on optimising game flow through innovative computational techniques.

Social modelling in social psychology has achieved significant milestones by leveraging advanced computational capabilities and algorithmic techniques that integrate concrete technologies with interdisciplinary approaches. Agent-based modelling (ABM) has become a cornerstone for simulating emergent social phenomena—from micro-level interactions to macro-level dynamics—as demonstrated in early works [372] [373] and further enabled by software toolkits that integrate heterogeneous data and theoretical constructs [374]. Interdisciplinary collaborations have enriched these models by incorporating insights from cognitive science, sociology, and network theory [375], exemplified by frameworks that merge social practice theory with ABM to capture habituality and interconnectivity [376] and socio-cognitive models that integrate cultural and affective dimensions [377]. The application of multiobjective genetic programming has advanced mechanism-based models by systematically exploring causal rules behind social patterns [378], while innovations in machine learning, such as graph convolutional networks, have enabled detailed analyses of friend selection and digital diffusion processes [379]. Additionally, recent developments in modular simulation architectures have standardised the modelling process, facilitating robust comparisons between theoretical predictions and empirical observations—a critical step for policy-oriented research [380]. Finally, studies addressing phenomena like filter bubbles and echo chambers [381] and efforts to bridge computational models with socially interpretable explanations using explainable AI [382] underscore the ongoing evolution of social modelling through concrete technological advances.

Gamification and automated game design demonstrate significant potential by integrating sophisticated algorithms and data-driven techniques to enhance user engagement, personalisation, and system efficiency. Broadly defined as the incorporation of game design elements in non-game contexts [383], gamification has evolved into systems where automated processes adjust motivational triggers and tailor intervention strategies across diverse domains such as education, healthcare, and e-commerce [384] [385]. Notably, the development of recommender systems and automated personalisation engines—exemplified by Rodrigues [386], who proposed a system that adapts gamification elements based on multidimensional user characteristics—has accelerated the move toward dynamic, user-centred design. Additionally, topic modelling approaches have been employed to map trends in gamification research, highlighting an emerging focus on automating design processes and systematically analysing gamification intensity [387] [388]. Early work classifying reward systems in games like World of Warcraft [384] laid the foundation for later studies that extended these classifications into automated contexts, enabling systems to adjust game parameters in real time based on user feedback and performance metrics [389]. This convergence of techniques from computer science, human–computer interaction, and psychology has not only facilitated the creation of adaptive gamified environments but also underscored the scalability and efficiency of automated design in applications ranging from educational simulations to corporate engagement and social marketing initiatives [390] [391] [385]. Our automated rule design study begins with identifying key elements of general rule design and establishes a machine learning-based framework, demonstrating the potential for complex rule creation.

5.2.2 Related Learning Paradigms

Generative models are machine learning architectures that learn the underlying data distribution to synthesise new, high-fidelity samples across diverse modalities, enabling advances in diverse fields like image synthesis and natural language processing. The introduction of GANs by Goodfellow et al. [39] catalysed major innovations in image synthesis and feature manipulation, with contemporary methods in deep convolutional GANs further enhancing high-resolution image quality [392]. Kingma et al. [393] laid the groundwork with semi-supervised deep generative models using explicit density models, setting new standards for model scalability and efficiency. Hybrid approaches such as DiffuseVAE, which combine the latent space advantages of VAEs with the efficient sampling of diffusion models, exemplify the trend towards controllable, high-fidelity generation [394], while Wang et al. [395] extend the versatility of gated convolutional VAEs to non-traditional domains. Moreover, practical applications have broadened into multimodal content generation, as highlighted by Xu [396], with models like DALL-E 3 and Parti enabling cross-modal synthesis, including high-definition video generation. Domain-specific adaptations further underscore the interdisciplinary impact of these technologies, with applications ranging from economic data analysis [397] to natural language processing, where data-to-text generation frameworks integrate content planning for coherent text synthesis [398].

AI strategy emphasises developing adaptive, intelligent systems, with reinforcement learning emerging as a critical methodology to achieve these objectives through foundational algorithmic innovations and diverse, high-impact applications. Early overviews [12] [399] established the integration of deep learning with RL by detailing key techniques, such as learning from demonstration, Monte Carlo tree search, and actor–critic methods—that powered breakthroughs like AlphaGo. Subsequent research has refined these approaches by advancing both value-based and policy-based paradigms [400] and emphasising hierarchical decision-making processes [401]. Domain-specific applications further demonstrate RL's potential, from controlling plasma behaviour in tokamak reactors [402] and formulating tactical strategies in wargames [403] to enabling rapid, adaptive decision-making in real-time strategy games like MicroRTS [404] and enhancing episodic exploration in StarCraft micromanagement [405]. Moreover, innovations in AI game design—including developments in modular architectures for StarCraft II [406], advanced deep RL-based game players [106], and the application of RL in traditional board games such as Gomoku [407]—underscore its versatility in competitive environments. Beyond gaming, RL has been successfully applied to optimize real-world systems, such as regulating distributed energy resources [408]) and driving algorithmic trading strategies through multi-agent frameworks [409]. Collectively, these contributions reflect an enduring research interest in leveraging RL to build adaptive, intelligent, and strategic AI systems across a broad range of high-stakes applications.

The proposed Flow-based SER framework defines a novel machine learning task within the generative model domain. This framework employs deep neural networks as backbone models for rule generation while integrating reinforcement learning to stimulate AI agent behaviour, thereby enabling the evaluation and evolution of both rules and the generated environment. By leveraging flow theory, the system captures the dynamic interplay between perceived challenge and agent skill, facilitating adaptive intrinsic motivation for players and corresponding extrinsic rewards for agents. Rather than focusing on the shortcomings of specific models, this work in-

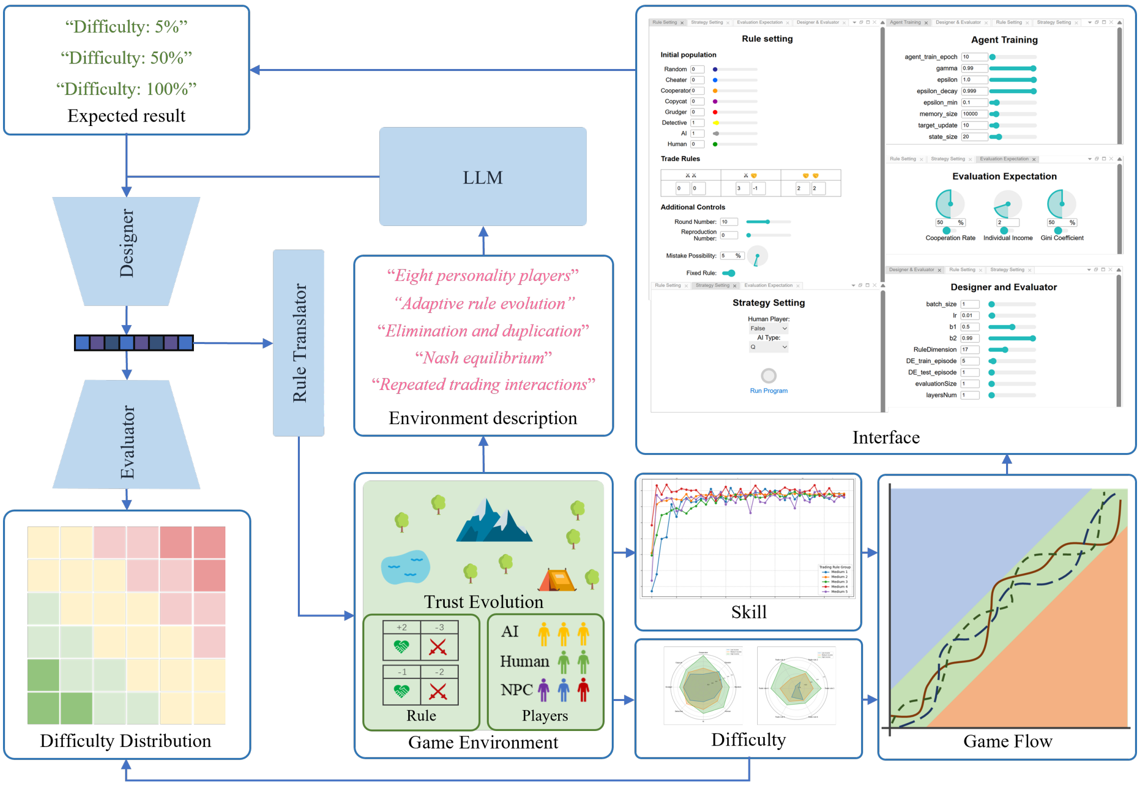


Figure 5.2: An illustration of the rule generation framework. The rule designer learns to create rules according to the embedded expectation and evaluates the results. The generated rule vector is then sent directly to the evaluator and translated into an executable parameter for the environment. The evaluator learns to validate the rules individually. The RL model learns strategies by exploring the environment. The controllability is tested on both the environment and the rule designer.

troduces a new paradigm that incorporates dynamic difficulty modulation via rule evolution and real-time flow visualisation, ultimately redefining rule generation as an adaptive, flow-based process.

5.3 Methodology

The goal of our flow-based rule generation research is to incorporate game flow into the rule generation task to provide players with intrinsic rewards while simultaneously generating rules that align with the agent's flow state to achieve dynamic difficulty adjustment. The core premise of flow theory is that when a player' s skill level matches the game' s difficulty—i.e., when the challenge falls within the "flow zone" —the player naturally enters a state of flow, which enhances performance and efficiency. This immersive experience serves as an intrinsic reward for the player, while the generated game rules act as extrinsic rewards for the agent.

Our flow-based rule generation framework, shown in Fig. 5.2, comprises several key components that form the core of the SER system: a rule designer, an evaluator, and a game Environment. In addition, the framework integrates a large language model (LLM) for rule deconstruction within the environment, a flow-based intrinsic reward manager, and an interactive interface. Together, these modules enable the system to generate rules that dynamically adjust game difficulty in accordance with the player' s flow state, thereby enhancing both user experience and agent performance.

5.3.1 Preliminaries

This section presents and explains the definitions and notations of rule generation below.

Rule: Defined as principles guiding game elements (players, maps, etc.), rules undergo creation, deletion, or modification, impacting the rule set $\mathcal{R}$, a matrix in $\mathbb{R}^{N_r \times D_r}$. These actions influence the quantity (N_r) and dimensions (D_r) of rules, with the updated set represented as $\mathcal{R}'$. **Strategies**: These are comprehensive algorithms ($\mathcal{S} \in \mathbb{R}^{N_s \times D_s}$) players adopt within game rules to determine outcomes. The strategy's scope, determined by the game's information completeness, affects the total strategies (N_s) and their stages (D_s), and can be developed by both humans and AI based on rewards. **Evaluation**: Game evaluation is associated with high-level heuristics, including spontaneity, interruptability, and continuity [310], measured via parameters (E in $\mathbb{R}^{N_e \times D_e}$). Each evaluation metric (e in E) provides an assessment angle, dependent on game rules but manifested through agent strategies in gameplay, formulated as $E = f(\mathcal{R}, \mathcal{S})$.

Automated Rule Design. Automated rule design begins with an initial rule set, $\mathcal{R}$, foundational to the game environment. Players or agents then formulate strategies, denoted as $\mathcal{S}$, which are essentially algorithms or behaviours tailored to optimise outcomes within the confines of $\mathcal{R}$. Subsequent evaluations expressed as $E = f(\mathcal{R}, \mathcal{S})$, represent gameplay metrics such as efficacy and fairness. Drawing from $\mathcal{E}$, the system discerns areas for rule modification in $\mathcal{R}$ to enhance gameplay or meet specific objectives. This cycle of strategy formulation, evaluation, and rule refinement iteratively progresses until the system meets predetermined performance or balance benchmarks.

5.3.2 Frameworks

Figure 5.2 presents the flow-based SER framework, composed of three core components: the designer, the Evaluator, and the Environment. The designer generates rule vectors based on an input expectation φ, expressed as a set of linguistic descriptions. These rule vectors are then passed to the Evaluator for outcome prediction and simultaneously translated into accessible parameters for the game platform. Within the Environment, reinforcement learning agents, non-player characters (NPCs), and human players engage in strategy exploration under the generated rules, producing gameplay statistics that serve as the ground-truth results for rule evaluation. A web-based interface enables researchers to input expectations into the designer and visualise real-time metrics. To further expand rule possibilities, the system integrates a large language model (LLM) that analyses textual descriptions of the Environment, identifies rule elements, and proposes new, meaningful rules, such as extrinsic rewards for specific behaviours, thereby extending the rule vector's dimensionality. Unlike earlier SER implementations, this flow-based version transforms agent strategies into skill levels and rule configurations into difficulty levels, mapping both into the concept of game flow. Consequently, the framework dynamically balances skill and difficulty, ensuring that new rules align with an optimal challenge–skill equilibrium. This design not only supports adaptive game difficulty but also fosters a deeper understanding of how rule evolution and strategy interact to shape the overall gameplay experience.

5.3.3 Environment and Task

This chapter presents a digital game environment for rule demonstration, referred to as *Trust Evolution* (TE), purposed to demonstrate the practicability of rule generation with judicious utilisation of computational resources. TE serves as a fusion of artificial life and rule-generation tasks, as it simulates cooperative behaviour among individuals in a society.

Trust Evolution. In the Trust Evolution environment, game rules are defined by parameters such as payoff, population size, round number, reproduction rate, and mistake probability. The mistake probability quantifies the likelihood that a player will choose an action opposite to their intended behaviour, while the round number indicates the number of trades each agent conducts. In each match, players aim to accumulate coins through trade, choosing between two actions: *cheat* or *cooperate*. The payoff function maps these actions to trade outcomes. Six distinct types of NPCs—*random*, *cheater*, *cooperator*, *copycat*, *grudger*, and *detective*—are incorporated to represent various personality archetypes. Specifically, the first three types consistently exhibit random, cheating, or cooperative behaviour, respectively; copycats initiate cooperation and subsequently mirror the opponent's last action; grudgers continue to cooperate until betrayed; and detectives follow a fixed sequence of cooperation, cheat, cooperation, and cooperation, switching to copycat behaviour if their cheating is reciprocated. At the conclusion of each match, a selection process eliminates low-performing players and replicates top performers.

In addition to these classic rule components, two types of roles—AI and Human—are introduced. AI agents employ reinforcement learning for strategy exploration, whereas human roles may either be player-controlled or guided by human-designed strategies. Furthermore, an LLM (e.g., ChatGPT o3 mini) is integrated to extend the classic rule set by generating an *Extrinsic reward* rule. This extrinsic reward is represented as a two-dimensional vector: the first dimension provides feedback for cooperative behaviour, and the second for cheating behaviour. These rewards are applied directly to all players and are combined with the trade outcomes, thereby influencing overall player behaviour.

5.3.4 Difficulty and Skill

A 17-dimensional rule vector is generated, consisting of three groups of parameters: (1) Initial Population (8 dimensions), (2) Trade Rules (6 dimensions), and (3)

Other parameters—namely, Round Number, Reproduction Number, and Mistake Possibility. Our key idea is to translate these rules into a one-dimensional difficulty measure, which is empirically proxied by the final income obtained by the agents. In our experiments, we maintain a fixed agent skill level so that variations in final income directly reflect changes in environmental difficulty. This approach is consistent with the dynamic difficulty adjustment literature.

Correlation Analysis

For each of the 17 parameters, we compute the Pearson correlation coefficient, r, between the parameter values and the corresponding final income, using the formula:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.1}$$

where x_i and y_i denote the values of the rule parameter and income, respectively, and $\bar{x}$, $\bar{y}$ are their sample means.

Let $d \in [0, 1]$ denote the normalised difficulty level, with $d = 0$ representing the easiest (i.e., highest income) state and $d = 1$ the hardest (i.e., lowest income) state. For each parameter with value range $[a, b]$ and corresponding correlation coefficient r, we define the mapping function $f(d)$ as follows:

$$f(d) = \begin{cases} a + (b - a)\, d, & \text{if } r > \theta, \\ b - (b - a)\, d, & \text{if } r < -\theta, \\ \frac{a+b}{2}, & \text{if } |r| \leq \theta, \end{cases} \tag{5.2}$$

where θ is a small threshold (set to 0.1) to distinguish significant correlations.

We categorize difficulty into six levels: *Very Easy* ($d = 0.0$), *Easy* ($d = 0.2$), *Medium* ($d = 0.4$), *Hard* ($d = 0.6$), *Very Hard* ($d = 0.8$), and *Extreme* ($d = 1.0$). Thus, for each parameter, we obtain a table where each row corresponds to a difficulty level and each column to one of the 17 parameters. This table provides a comprehensive mapping from the rule vector to the difficulty measure.

By evaluating the agent' s performance under a fixed difficulty level, we obtain a proxy for its skill that is decoupled from environmental variations. Formally, let the agent' s strategy be represented by a vector $\mathbf{s} \in \mathbb{R}^n$, and denote its resulting income by $I(\mathbf{s})$. To ensure comparability across different strategies and to eliminate direct dependence on difficulty, we first identify the minimum and maximum incomes observed among all candidate strategies: $I_{\min} = \min_{\mathbf{s}' \in \mathcal{S}} I(\mathbf{s}')$, $I_{\max} = \max_{\mathbf{s}' \in \mathcal{S}} I(\mathbf{s}')$, where $\mathcal{S}$ is the set of strategies under evaluation. We then normalize the raw income $I(\mathbf{s})$ into the interval $[0, 1]$ via

$$\mathrm{Skill}_n(\mathbf{s}) = \frac{I(\mathbf{s}) - I_{\min}}{I_{\max} - I_{\min}} \in [0, 1]. \tag{5.3}$$

Ethical Analysis. In the realm of automated game design, the generation of rules through artificial intelligence introduces the possibility of inadvertently embedding biases into the gaming experience. This chapter presents an ethical examination of our flow-based rule generation system. Throughout the development of the flow-based SER framework, we proposed the Trust Evolution game environment. It is important to highlight that players' roles within these games are devoid of any attributes related to sex, age, race, or similar socio-cultural factors. In TE, the six distinct personalities are purely representative of various NPC behavioural patterns, further emphasising our commitment to unbiased game design.

5.3.5 Game Flow

In the realm of video game experiences, the theory of game flow is characterised by several critical dimensions. At the forefront, we have **Concentration**, represented by C, which embodies the level of focus and engagement a player immerses in during gameplay. Let H denote **the balance of Challenge and Skills**, which come into play when considering a player's perception of the game's optimal challenge level and their personal competence. The **Control** is L, delves into a player's sense of freedom, choices made, and engagement in in-game activities, reflecting the broader theme of autonomy. As for **Clear Goals**, represented by G, encapsulates the existence of well-defined objectives or targets that guide gameplay, while **Feedback** B pertains to the ongoing interaction between the game and the player, characterised by responses and information provision. **Immersion** M broadens the perspective, referencing the depth of a player's presence within the gaming environment. Lastly, **Social Interaction** captures the dimension of engagement with other players, emphasizing the sense of connection and interaction intrinsic to many modern games, denoted as N.

To extract meaningful insights from the interplay of these dimensions, we can model the overall game flow experience as a weighted sum. Given the complexity and relationship of these dimensions, the flow can be represented as:

$$F = \alpha C + \beta H + \gamma L + \delta G + \epsilon B + \zeta M + \eta N. \tag{5.4}$$

Here, α, β, γ, δ, ϵ, ζ, and η are constants representing the relative importance of each dimension. The goal is to understand how varying levels of these factors collectively contribute to the overall player experience, as captured by the model.

Concentration can be measured by how the Q-values for different actions in a given state converge. As the model becomes more confident in its learned values, the Q-values for the optimal action(s) will stabilise, and the difference between the Q-value of the optimal action and other actions will increase. Let $Q(s, a)$ be the Q-value for state s and action a. Let $\pi(a \mid s)$ be the policy derived from Q-values, representing the probability of selecting action a in state s. Then, the concentration CC over time tt can be defined as:

$$C(t) = 1 - \frac{1}{|A|} \sum_{a \in \mathcal{A}} std(\pi(a \mid s)). \tag{5.5}$$

H represents the degree to which an agent's normalised skill and the environment' s normalized difficulty are aligned. Both the skill measure $\text{Skill}_n(\mathbf{s}) \in [0, 1]$ and the difficulty level $d \in [0, 1]$ have been independently mapped into the same unit interval, ensuring that comparisons between them are meaningful. A value of $H = 1$ corresponds to a perfect match—i.e., $\text{Skill}_n(\mathbf{s}) = d$—indicating that the agent' s competence is exactly appropriate for the current challenge. Conversely, as the absolute difference $|\text{Skill}_n(\mathbf{s}) - d|$ increases toward 1, H decreases linearly toward 0, reflecting progressively greater under- or over-matching. This simple linear penalty on the mismatch ensures that small deviations produce only modest reductions in H, while larger mismatches yield proportionally lower balance values, thereby capturing in a single dimension how well the agent's competency aligns with the game's difficulty:

$$H = 1 - \left|\mathrm{Skill}_n(\mathbf{s}) - d\right| \in [0,1]. \tag{5.6}$$

Control L quantifies a player's autonomy by measuring how many actions in each state actually change the game's state. Let S be the set of states, A the set of actions, and $T : S \times A \to S$ the state-transition function. In state s, define the set of effective actions $E(s) = \{\, a \in A \mid T(s,a) \neq s\}$, and let $d(s)$ denote the probability of visiting s. The per-state control is

$$L(s) = \frac{|E(s)|}{|A|} \in [0,1], \tag{5.7}$$

"Clear Goals" G for an RL agent in Trust Evolution is assessed by whether the agent reliably discovers and completes a set of predefined subgoals $\{g_1, \ldots, g_m\}$. Each subgoal g_i is a binary condition on a trajectory (e.g., "sustain mutual cooperation for K consecutive rounds" or "achieve average payoff above threshold P^*"). We associate a success indicator $f_i(\tau) \in \{0,1\}$ with g_i, where $f_i(\tau) = 1$ if trajectory τ satisfies g_i, and 0 otherwise. For a given policy π, we estimate the empirical probability $P_i = \Pr_{\tau\sim\pi}\big[f_i(\tau) = 1\big] \in [0,1]$ by sampling N independent trajectories under fixed environment parameters. We then define

$$G = \frac{1}{m}\sum_{i=1}^{m} P_i, \tag{5.8}$$

so that $G = 1$ if and only if every subgoal is achieved almost surely under π, and lower values of G reflect missing or ambiguous objective signals. This single-scalar metric thus quantifies how clearly the agent "understands" its task structure in Trust Evolution.

Feedback B in Trust Evolution quantifies how strongly and how promptly the game responds to an agent's actions via reward signals. Let r_t denote the immediate payoff received by the agent at time step t, and let $R_{\max}$ be the maximum possible absolute payoff in any single interaction. Over a trajectory of length T generated by policy π, we compute the normalized per-step feedback as $\tilde{r}_t = \frac{|r_t|}{R_{\max}} \in [0, 1]$. The overall feedback measure B is then defined as the time-averaged normalized reward:

$$B = \frac{1}{T} \sum_{t=1}^{T} \tilde{r}_t = \frac{1}{T\, R_{\max}} \sum_{t=1}^{T} |\, r_t|, \quad B \in [0, 1]. \tag{5.9}$$

A higher B indicates that the agent consistently receives clear, sizable payoffs (positive or negative) in response to its actions—i.e., the environment provides strong and timely feedback—whereas a lower B implies that rewards are sparse or minimal, making it harder for the agent to adjust its strategy.

Immersion M quantifies how accurately the agent' s internal predictions align with actual outcomes across the entire state–action space. Let $\hat{R}(s, a)$ denote the agent' s expected immediate reward for taking action a in state s, and let $Q(s, a)$ be the corresponding learned action-value. We then define

$$M = 1 - \frac{1}{|S|\,|A|} \sum_{s \in S} \sum_{a \in A} |\, \hat{R}(s, a) - Q(s, a)|, \tag{5.10}$$

where $|S|$ and $|A|$ are the numbers of states and actions, respectively. Because the absolute prediction error $|\hat{R}(s, a) - Q(s, a)|$ measures the discrepancy between expected and actual returns, a smaller average error indicates that the agent has formed an accurate model of the environment. Consequently, a higher value of M (closer to 1) implies stronger alignment between expectation and experience, suggesting deeper engagement and immersion in the game.

Social Interaction N measures the extent of cooperative engagement among agents in Trust Evolution. In this setting, each interaction between two agents yields a binary decision: cooperate or defect. For each agent n, let $C_i(n) = \frac{\text{Number of trust decisions made by } n}{\text{Total decisions made by } n} \in [0, 1]$. We then compute the population-level social interaction metric N as the average cooperation index:

$$N = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} C_i(n) = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \frac{\text{Trust actions by } n}{\text{Total actions by } n}. \tag{5.11}$$

Here, $|\mathcal{N}|$ denotes the total number of agents. A value of $N = 1$ indicates that every agent always chooses to trust, whereas $N = 0$ indicates no cooperative behaviour at all. In Trust Evolution, a higher N reflects stronger social connectivity through cooperation.

Translating flow-theory constructs into quantitative metrics for RL agents allows us to create objective experience indicators that parallel human "flow" dimensions without subjective self-reports, improve algorithmic interpretability by linking behavioural dynamics (e.g., Q-value convergence) to established psychological constructs, and establish a benchmark for multi-agent and human–AI collaboration studies. These efforts build on foundational flow-theory research (e.g., Csikszentmihalyi [410]), extend game-industry flow assessment models (e.g., GameFlow [411], empirical gaming studies [412]), and provide a rigorous framework for evaluating immersion, control, clear goals, feedback, challenge–skill balance, concentration, and social interaction in Trust Evolution. This approach grounds RL evaluation in validated psychological theory and facilitates future comparisons of human and AI engagement in interactive environments.

5.4 Experiment

5.4.1 Settings

The experiments target three main objectives: (1) elucidating the relationship between rule dimensions and game difficulty, (2) investigating the interplay between strategic adaptations and player skill, and (3) dynamically generating flow-based rules to balance skill and difficulty. The proposed flow-based SER is deployed in diverse environments to instantiate rules for AI agents, NPCs, and human players. The rule designer is trained using evaluation metrics that quantify both performance outcomes and the impact of rule parameters on difficulty. Simultaneously, the framework examines how evolving strategies correlate with player skill. The dynamically generated, flow-based rules are demonstrated in an interactive environment that provides real-time visualisations of key statistical data.

Evaluation Criteria.

The evaluation framework adopts a multi-dimensional approach to assess the rule

Algorithm 2 SER training algorithm

Input:
Linguistic description for rule expectation E;
Number of training epochs Ep.
Game environment Env.
Output:
Well-trained rule designer model: $D : E \to R$.
Well-trained Q-learning model: $Q : S \to A$.

Initialisation: Embed the rule expectation as φ.
Add noise z to the expectation E, resulting in $\varphi = \varphi + z$.
for $t = 1 : Ep$ **do**
  Rules generation: designer create rules $R = D(E)$.
  Train reinforcement learning model: translate R into $R' = T(R)$, learning strategies $S = Env(R')$;
  Train evaluator: use $E' = Env(R')$ to train the evaluator;
  Train designer: use the evaluator's result E' and input E to upgrade designer;
  if $E' - E == 0$ **then**
    break;
  end if
end for
Save the well-trained designer model D for rule generation.
Save the well-trained Q-learning model Q for action selection.

designer, evaluator, and AI player (using RL methods such as Q-learning) alongside the stability of the game environment. Quantitatively, the rule designer is evaluated using Mean Squared Error and the evaluator via cross-entropy loss, with the distribution of game outcomes from random rule sampling serving as the training dataset. Controllability is measured by the system's ability to dynamically adjust game difficulty based on generated rules, and agent performance is assessed by the rate of cumulative reward maximisation. Qualitatively, integrating the generated rules into two virtual environments provides insight into user engagement, flow experience, and game balance. Together, these metrics ensure that rule generation aligns with the evolving challenge–skill balance, thereby optimising both agent performance and player engagement.

Implementation Details.

The Trust Evolution (TE) environment, which drew upon The Evolution of Trust' by Nicky Case [413], is developed using the Unity engine and defines game

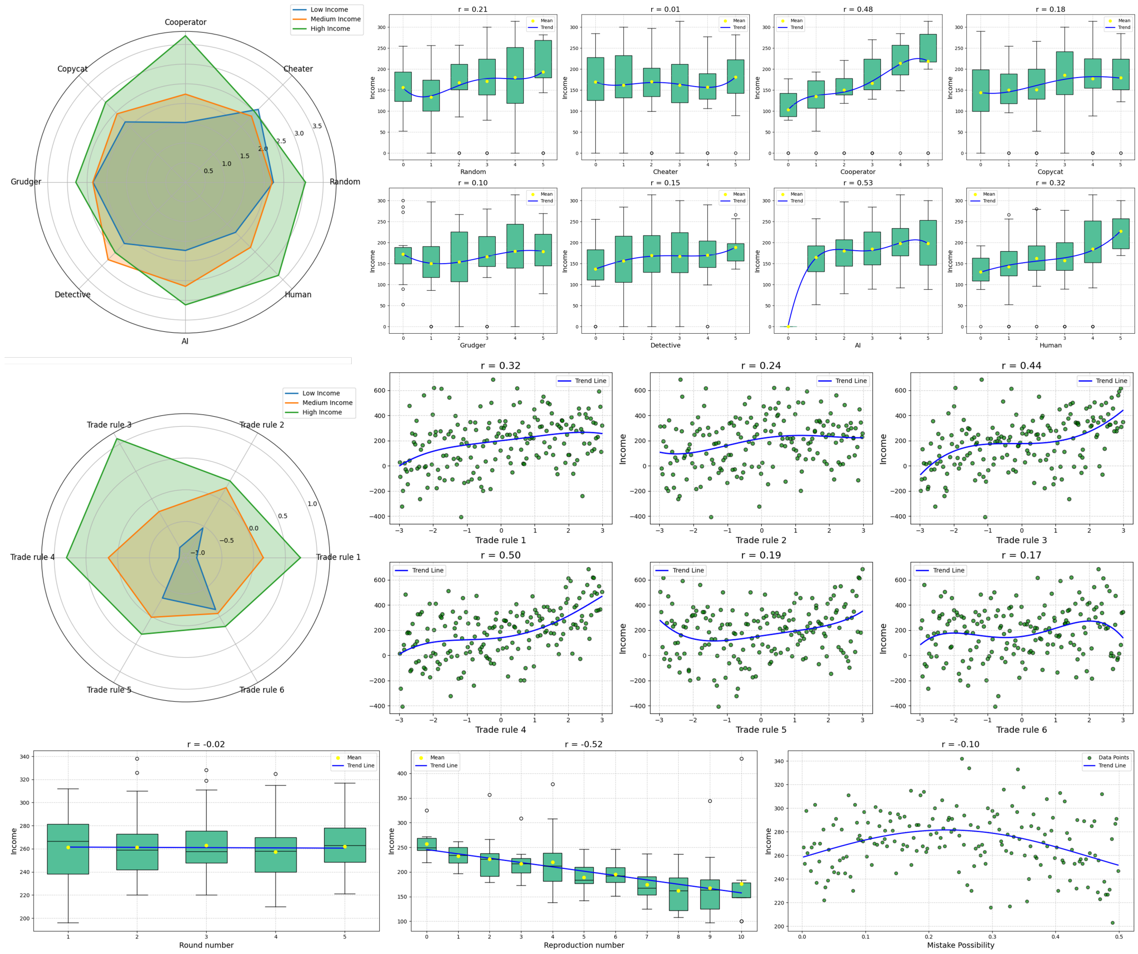


Figure 5.3: visualisation of how each rule vector dimension correlates with game difficulty, where r denotes the Pearson correlation coefficient for each dimension. The radar charts (top left) depict the impact of key parameters, while the box plots (top right) and scatter plots (middle and bottom rows) illustrate variations in difficulty as individual dimensions change.

rules through parameters such as population distribution, payoff structure, round count, reproduction quantity, and mistake probability, along with an additional extrinsic reward. The system leverages a ChatGPT o3 mini-based LLM for rule deconstruction, while reinforcement learning agents are implemented using Q-learning and DQN. Both the rule designer and evaluator are realised as three-layer fully connected neural networks. A web-based interface provides user interaction, and a backend Python program seamlessly integrates Unity, Python modules, and the interface to support end-to-end evaluation. This integrated framework facilitates the systematic analysis and evolution of game rules, ensuring dynamic adjustment of game difficulty in response to agent performance and environmental feedback.

Table 5.1: Mapping of a 17-dimensional rule to six difficulty levels. Each row represents a specific difficulty, with rule parameters.

Difficulty	Rand	Cheat	Coop	Copy	Grudg	Detec	AI	Human	T1	T2	T3	T4	T5	T6	Rounds	Reprod	Mistake
Very Easy	0	2	0	0	3	0	0	0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	3	10	0.25
Easy	1	2	1	1	3	1	1	1	-1.8	-2.04	-1.8	-1.8	-2.04	-2.04	3	8	0.25
Medium	2	2	2	2	3	2	2	2	-0.6	-1.08	-0.6	-0.6	-1.08	-1.08	3	6	0.25
Hard	2	2	3	2	3	2	3	3	0.6	-0.12	0.6	0.6	-0.12	-0.12	3	4	0.25
Very Hard	3	2	4	3	3	3	4	4	1.8	0.84	1.8	1.8	0.84	0.84	3	2	0.25
Extreme	4	2	5	4	3	4	5	5	3.0	1.8	3.0	3.0	1.8	1.8	3	0	0.25

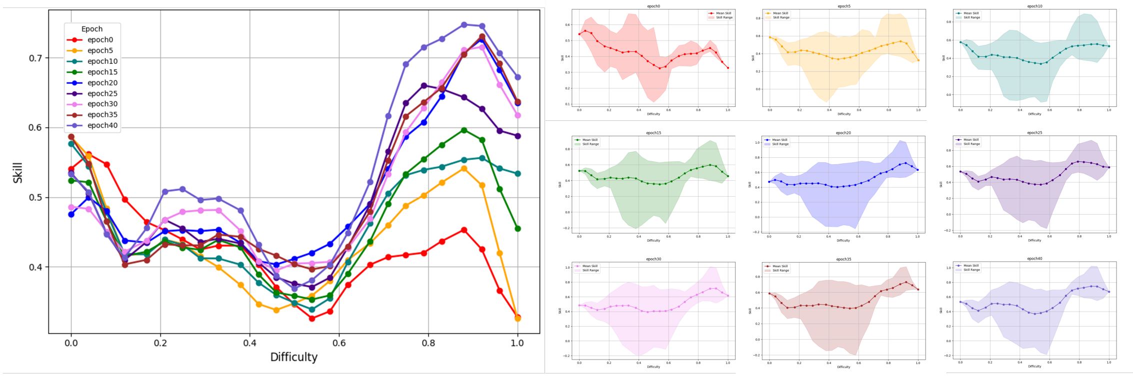


Figure 5.4: Performance comparison of AI players at different training epochs for extrinsic rewards. The left panel depicts how skill varies with difficulty as training progresses, showing each epoch' s performance trend. The right panel presents the corresponding skill responses under distinct rule conditions, with shaded regions indicating performance variability across multiple runs.

5.4.2 Dynamic Difficulty Adjustment

In game design, difficulty settings are typically implemented through a set of discrete modes (e.g., easy, medium, hard) rather than requiring players to fine-tune numerous game parameters. Fig. 5.3 visualises the relationship between rules and difficulty, which is a prerequisite for exploring game flow. In the Trust Evolution Game Environment, five classic rules are defined: initial population, payoff matrix, round number, reproduction number, and mistake possibility. The initial population is represented by an 8-dimensional vector, the payoff matrix by a 6-dimensional vector, and the remaining rules by single-dimensional values, forming a 17-dimensional rule vector that is mapped to a one-dimensional difficulty metric, ultimately influencing the performance of the flow-based SER framework.

The radar charts in Fig. 5.3 illustrate how each dimension of the initial population and trade rule affects difficulty. For simplicity, difficulty is divided into three levels—low, medium, and high—based on AI player income under equivalent skill

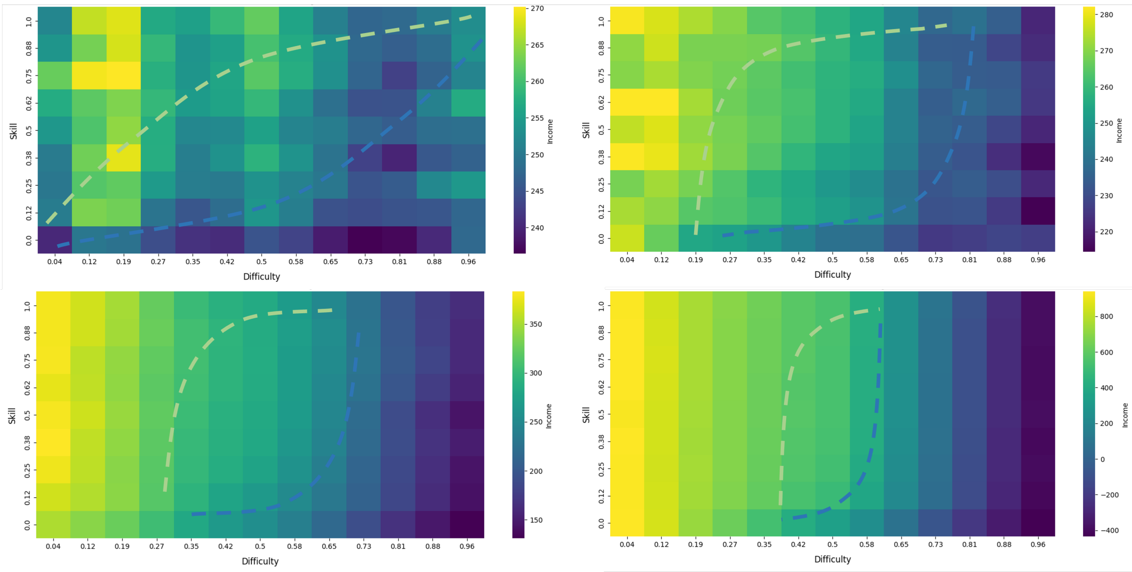


Figure 5.5: Heat maps illustrating the interplay between difficulty (horizontal axis) and skill (vertical axis) under four distinct weighting scenarios. Each panel highlights how overall performance (color scale) shifts when either skill or difficulty dominates, with brighter regions indicating lower stress (or boredom) and darker regions signifying higher stress (or anxiety). The dashed lines approximate the flow zone where difficulty and skill achieve an optimal balance.

conditions, with higher difficulty naturally leading to lower income. It is evident that the number of Cooperators, Humans, and AI players in the initial population significantly impacts difficulty; a similar effect is observed in the first, third, and fourth dimensions of the trade rule, where rewards and penalties for initiating cooperation or mutual cheating have a pronounced effect on AI skill.

The box plot on the right side of the initial population radar chart further shows the impact of each personality type on difficulty as their numbers vary. An increase in AI player count results in lower difficulty, particularly noticeable in the range from 0 to 1, with a similar trend observed for Humans, indicating that AI tends to benefit from interactions with human-designed personalities. In contrast, the influence of Cheater and Grudger types on difficulty is minimal, likely because, upon encountering cheating, both Grudger and Cheater behave similarly by isolating other players and choosing to cheat.

The final row of Fig. 5.3 presents the effects of round number, reproduction number, and mistake possibility on difficulty. While the round number affects the total income players obtain, it has a negligible impact on per-round income, resulting

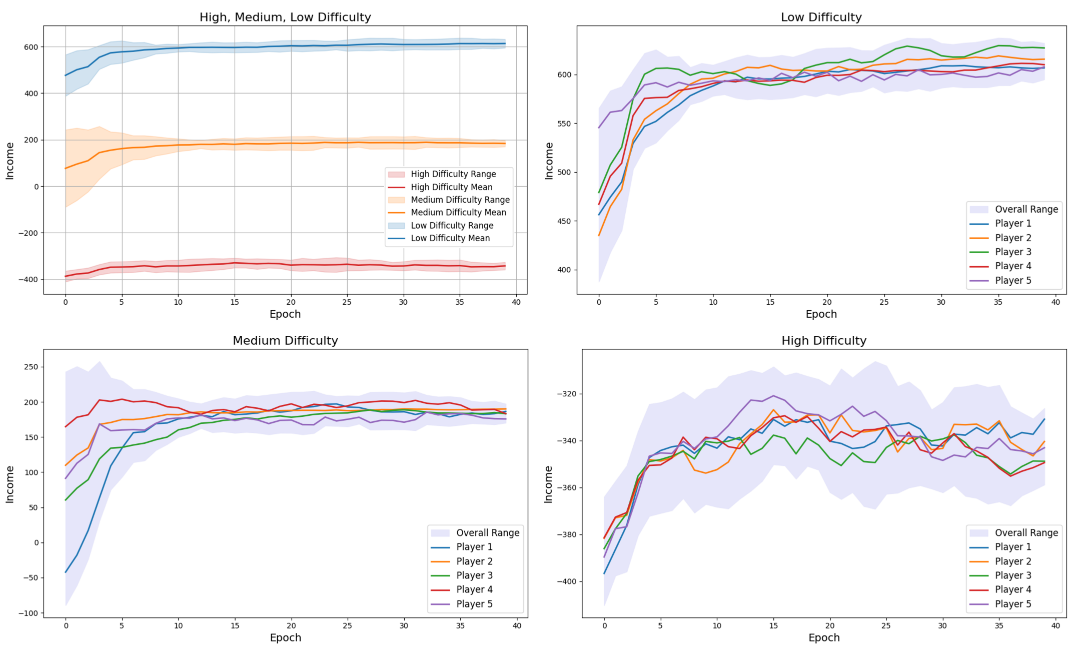


Figure 5.6: Illustration of training trajectories of Q-learning agents at three difficulty levels (High, Medium, and Low). The top-left panel compares mean performance across these difficulties, while the other panels detail individual player incomes for each difficulty setting. Shaded regions indicate performance variability over multiple runs, illustrating how difficulty modulates both learning speed and final reward outcomes.

in almost zero correlation. Reproduction number, however, influences the selection process, as a higher reproduction rate results in greater turnover among players, which, in early rounds where AI players are undertrained, correlates with higher difficulty. Mistake possibility exhibits a weak overall correlation; however, when divided into two stages—0–25% and 25–50%—a stronger relationship emerges. In the first stage, as mistake possibility increases, Cheater and Grudger types have a higher chance of cooperation, and Detective types are more inclined to cooperate, leading to increased income and lower difficulty. In the second stage, further increases in mistake possibility cause player behaviour to shift from rational to random, thereby reducing income.

Table 5.1 presents the mapping between a 17-dimensional rule vector and six discrete difficulty levels (ranging from Very Easy to Extreme) as determined by the SER system' s Evaluator. The rule vector comprises eight behavioural or personality dimensions (Rand, Cheat, Coop, Copy, Grudg, Detec, AI, Human), six trade-related parameters (T1 through T6), and three game-specific settings (Rounds, Reprod, and

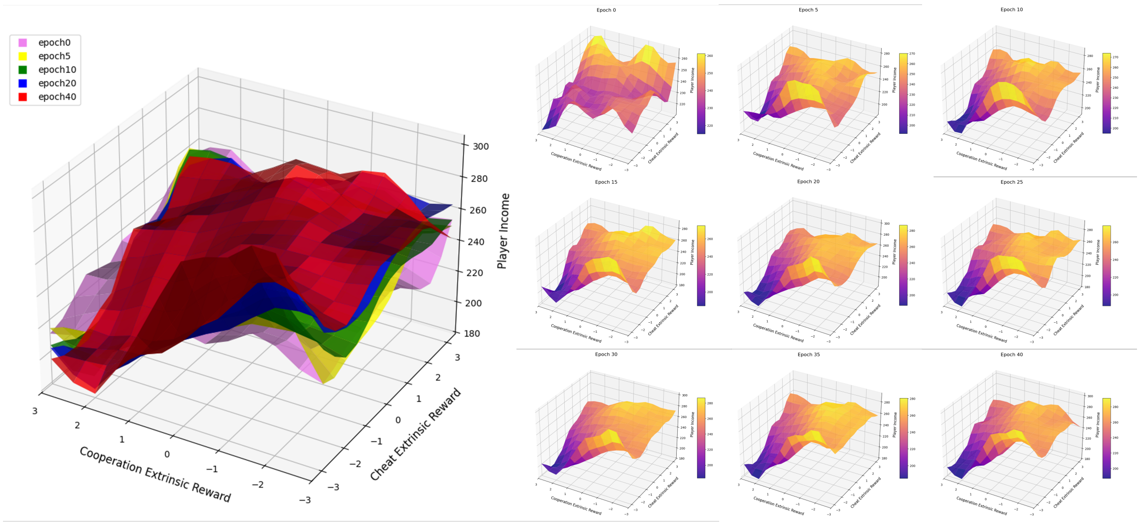


Figure 5.7: visualisation of how extrinsic reward parameters for cooperation and cheating influence player income across multiple training epochs. The left panel depicts a 3D surface where each colored layer represents the evolving relationship between cooperation/cheat rewards and player income. The right panel shows epoch-specific surfaces, illustrating how the environment' s sensitivity to extrinsic rewards shifts as AI agents gain experience.

Mistake). This table illustrates how variations in each rule dimension collectively influence the overall game difficulty. The mapping serves as the foundation for dynamic difficulty adjustment within the flow-based SER framework by ensuring that the generated rules appropriately reflect the balance between game challenge and player skill. In our experiments, these mappings are utilised to systematically modulate the game environment, thereby optimising player engagement and performance.

5.4.3 Game Flow

Different training levels of AI players were visualised to examine the transformation from strategy to skill under varying difficulty settings. In the left panel of Fig. 5.3, when difficulty trends remain consistent, AI players of different training levels exhibit similar skill variation patterns. At extremely low difficulty levels, an increase in difficulty leads to a decline in overall performance; however, once the difficulty reaches approximately 0.5, performance begins to improve until about 0.9, after which it rapidly declines again. Notably, players with higher training consistently achieve superior performance. The initial decline in performance is primarily attributed to low difficulty levels that yield high rewards regardless of the actions taken, thereby impeding the discovery of more optimal strategies. As the difficulty increases and the overall environmental rewards decrease, the punishment potential becomes more pronounced—especially beyond a difficulty of 0.5—prompting AI players to enhance their strategies during training. Consequently, the performance gap between highly trained and less trained players widens. The right panel displays the detailed performance of players with fixed training iterations across different difficulty levels. Due to the game environment being composed of multiple rules and the influence of AI players' greedy behaviours, each scenario was tested ten times, with the results presented as a range of skill levels.

Fig. 5.5 illustrates the dynamic interplay between difficulty and skill across four distinct scenarios, each representing a different balance of influences. In the top-left panel, skill exerts a far greater impact than difficulty, leading to predominantly high-performance regions even under moderate difficulty. In contrast, the top-right panel shows a near-equal influence from both factors, producing a more evenly distributed heat map. The bottom-left panel depicts a slight dominance of difficulty over skill, where performance tends to decline rapidly once difficulty surpasses a certain threshold. Finally, in the bottom-right panel, difficulty overwhelmingly dictates outcomes, causing a pronounced drop in performance once difficulty rises above minimal levels.

In each panel, the colour scale (ranging from yellow to deep purple) corresponds to the level of player comfort. Brighter yellow regions indicate relatively low stress or even boredom, while deeper purple zones signal heightened anxiety. Notably, the flow zone—an area reflecting an optimal balance between skill and difficulty—appears as a smooth transition region in each heat map. Observing the shape and extent of this zone provides insights into how the game environment might adapt: under conditions where skill dominates, even modest difficulty increments can maintain player engagement, whereas when difficulty is dominant, small increases may quickly lead to overwhelming challenges.

The Q-learning training trajectories for AI agents under three difficulty levels (High, Medium, and Low) are demonstrated in Fig. 5.6. The upper-left chart consolidates the overall training results, showing that agents in Low and Medium difficulty environments accumulate higher incomes over time compared to those facing High difficulty. Notably, the High difficulty agents exhibit greater variability in performance, reflecting a more random or exploratory strategy. The remaining three panels break down the performance for each difficulty level individually. Under Low difficulty (top-right), agents quickly converge to relatively high income values, whereas in High difficulty (bottom-right), reward accumulation remains modest with considerable oscillations. Medium difficulty (bottom-left) demonstrates a pronounced inflexion point where income sharply increases once the agent discovers an effective strategy, aligning with the flow theory notion that moderate challenges can drive more pronounced skill development. The shaded regions in each plot represent the variance or range of outcomes across multiple runs, indicating that both difficulty level and training progress significantly influence the consistency of agent performance. Overall, these results underscore that the balance between challenge and skill level—particularly evident in the Medium difficulty setting—facilitates more robust learning and higher eventual incomes, mirroring the core principles of flow theory.

The proposed flow-based SER framework not only adjusts difficulty but also creates new rules and incorporates flow mechanisms. Fig. 5.7 illustrates the impact of the newly introduced extrinsic reward on player income. The left panel compares the performance of AI agents at various training levels as the extrinsic reward trends vary, while the right panel displays the responses of agents at different training stages under distinct rule conditions. In the game, the AI player's objective is to maximise its cumulative reward, where the extrinsic reward generated by the flow-based rule designer acts as a form of reward and punishment for cooperative and cheating behaviours. Notably, higher extrinsic reward values correspond to lower game difficulty and improved agent performance. Furthermore, the figure demonstrates that agents with higher training levels achieve greater rewards under the same difficulty settings, reflecting an increase in their skill. These results indicate that the proposed flow-based SER framework effectively generates controllable, flow-based parameters that significantly influence player performance.

5.5 Conclusion

This work presents a novel flow-based SER framework that integrates flow theory into automated rule generation for dynamic difficulty adjustment. By capturing the co-evolution of perceived challenge and agent skill during gameplay, the framework establishes adaptive intrinsic motivation mechanisms that sustain optimal engagement in gaming and educational contexts. The proposed system dynamically adjusts game rules in real time through rule evolution, ensuring that environmental challenges continuously align with evolving skill levels and thereby maintaining an effective challenge–skill balance. Furthermore, the implementation of a real-time Flow visualisation panel enables interpretable, on-the-fly feedback by mapping key metrics onto a two-dimensional flow space, which facilitates both system training and end-user interaction. These contributions significantly advance the state-of-the-art in adaptive game design and rule generation, offering a robust methodology for dynamically modulating game difficulty to enhance player engagement and performance.

Future research will further enhance the flow-based SER framework by investigating the variability of physiological signals and their mapping to player engagement. This involves refining the integration of EEG and heart rate data to better capture dynamic flow boundaries. Additionally, we will extend the framework to cover a wider range of game genres and rules, enabling a comprehensive evaluation of adaptive difficulty adjustments. Integration of real-time sentiment analysis and contextual behavioural cues is also planned to improve reward calibration and strategy optimisation. Finally, we will examine long-term adaptation scenarios where the system evolves during extended gameplay sessions to assess its scalability and sustainability. These directions aim to refine the balance between game difficulty and player skill, ultimately contributing to more engaging and personalised interactive systems.

CHAPTER 6

Conclusions

In the domain of adaptive game design and automated rule generation, the integration of advanced generative models, reinforcement learning techniques, and flow theory has been pivotal in enhancing dynamic difficulty adjustment and player engagement. The contributions made in this doctoral research include: (1) the development and rigorous evaluation of a novel Rule Generation Networks (RGN) framework that automatically synthesizes and refines game rules without relying on pre-existing datasets (Chapter 3); (2) the implementation of a Triadic Reciprocal Dynamics (TRD) system that captures the complex interactions among rule generation, strategy evolution, and evaluation within dynamic environments (Chapter 4); and (3) the incorporation of flow-based methodologies to enable real-time, adaptive difficulty adjustment, thereby aligning game challenges with evolving player skills through intrinsic and extrinsic reward mechanisms (Chapter 5). These innovations collectively advance the capability of automated systems to generate and adjust game rules, maintain the optimal challenge–skill balance, and provide interpretable feedback via real-time visualisation, thereby laying a robust interdisciplinary foundation for intelligent and engaging interactive systems.

6.1 Summary and Contribution

This thesis makes three key contributions to adaptive rule generation and dynamic difficulty adjustment in interactive environments.

A novel Rule Generation Networks (RGN) framework is presented, which integrates generative neural networks with reinforcement learning to automatically design, evaluate, and evolve game rules. The framework addresses several challenges: it eliminates the need for pre-existing datasets by generating and collecting data within digital environments; it translates abstract rule vectors into executable game parameters through a dedicated rule translator; and it introduces controllability metrics to manage impractical design requirements. These innovations enable the system to autonomously adjust rules in accordance with evolving environmental and agent behaviours.

An AI-driven platform, termed Triadic Reciprocal Dynamics (TRD), is developed to capture the complex interplay among rule creation, strategy evolution, and evaluation in dynamic, multi-agent environments. TRD leverages a neural network-based rule designer and evaluator to simulate social interactions among AI agents, non-player characters, and human participants. The system implements a multi-dimensional evaluation paradigm to quantify social metrics such as cooperation rate and individual income, ensuring that iterative feedback is used to refine rule generation continuously. This approach enhances the fidelity and adaptability of rule evolution in simulated social settings.

The flow-based SER framework extends these systems by incorporating flow theory to achieve dynamic difficulty adjustment. By mapping rule parameters to game difficulty and aligning strategic responses with player skill, the framework maintains an optimal challenge–skill balance. A Real-Time Flow Visualisation panel is integrated into the interface, providing immediate, interpretable feedback on engagement metrics and facilitating on-the-fly rule tuning. This integration not only enhances system responsiveness but also deepens the understanding of the relationship between game flow, player performance, and rule evolution.

6.2 Limitations

The proposed flow-based SER framework, while innovative, incurs significant computational costs. The system relies on iterative training cycles in which every newly generated rule must be evaluated within a simulated environment. Each evaluation requires all participating AI agents to undergo reinforcement learning, using deep neural network models for both rule generation and performance assessment. This approach leads to an exponential increase in training time and resource usage as the number of agents and complexity of the environment grow. The inherent computational burden is a critical challenge for scaling the system to larger, more complex scenarios or for achieving real-time rule adjustments in resource-constrained settings.

Another constraint concerns the generalizability and interpretability of the system. The mapping between a high-dimensional rule vector and game difficulty is highly domain-specific, meaning that the current configurations may not readily transfer to different types of game environments or simulation scenarios. Variability in agent behaviour and environmental parameters can yield unexpected outcomes, reducing the robustness of the system. Moreover, while the real-time flow visualisation panel provides some insight into the relationship between game difficulty and player skill, the decision-making process, driven by deep neural networks and reinforcement learning, remains largely opaque. This lack of transparency complicates the task of interpreting rule adjustments and understanding the underlying causal mechanisms, which is a common challenge in multi-agent deep reinforcement learning. Additionally, the absence of effective human-in-the-loop intervention limits the opportunity to incorporate designer expertise for refining rules and mitigating emergent behaviours that may be counterproductive or ethically questionable.

A further concern is related to the dynamic adaptation and stability of the difficulty adjustment process. The system aims to continuously align environmental challenges with evolving player skills; however, rapid or excessive rule modifications can result in instability in the training dynamics. Oscillatory behaviours may occur when the system overreacts to short-term fluctuations in agent performance, leading to inconsistent game flow that undermines player engagement. The evaluation metrics currently employed—such as cumulative reward and win rate—serve as proxies for assessing the quality of the induced game flow, but they do not fully capture the nuanced and subjective nature of human engagement. This discrepancy between objective measures and the subjective experience of optimal challenge remains a significant challenge in dynamic difficulty adjustment research. Addressing these issues will likely require incorporating advanced control theory, meta-learning techniques, or additional qualitative assessments that better reflect human perceptual and emotional responses.

While the flow-based SER framework demonstrates a promising approach to automated rule generation and adaptive difficulty adjustment, these limitations highlight the need for further research. Enhancements in computational efficiency, improved generalizability and interpretability, and more robust dynamic adaptation mechanisms are essential for advancing the practical utility of adaptive multi-agent systems. Future work in this area should focus on developing more efficient training methods, integrating human-in-the-loop mechanisms, and designing evaluation techniques that bridge the gap between quantitative performance metrics and the qualitative aspects of human engagement.

6.3 Future Directions

Enhancing computational efficiency and scalability remains a primary direction for future research. The current flow-based SER framework demands extensive computational resources due to its iterative rule evaluation in multi-agent environments, which poses challenges for real-time adaptation and large-scale deployment. Future work should explore more sample-efficient reinforcement learning algorithms, parallelised simulation techniques, and optimised neural architectures to reduce training time and resource consumption. Additionally, developing model compression and transfer learning approaches may enable the system to generalise across more complex and varied environments without sacrificing performance. Expanding the framework to accommodate more abstract rule representations could further enhance its cross-domain generalizability. By incorporating meta-learning techniques, the system could potentially adapt its rule generation process to new environments with minimal retraining, thereby improving its versatility and robustness in diverse application scenarios.

Improving interpretability and integrating human-in-the-loop mechanisms are also critical future research directions. Although the current system incorporates a real-time flow visualisation panel to monitor engagement metrics, the underlying decision-making processes of the deep neural networks and reinforcement learning agents remain opaque. Future efforts should focus on integrating explainable AI (XAI) techniques to extract human-readable rationales for rule adjustments and strategy adaptations, thus increasing the transparency and trustworthiness of the system. Moreover, a more interactive framework that allows human designers to intervene by approving, modifying, or vetoing generated rules could harness domain expertise to guide the rule generation process. Such an approach would bridge the gap between automated optimisation and human creativity, ensuring that the system' s outputs align with higher-level design objectives and ethical standards.

Refining evaluation methodologies constitutes another essential future direction. While current metrics, such as cumulative reward, win rate, and standard performance statistics, offer useful proxies for assessing system performance, they do not fully capture the subjective aspects of player experience, such as engagement and flow. Future research should develop holistic evaluation paradigms that integrate quantitative performance metrics with qualitative measures, including physiological sensor data and user feedback. This comprehensive approach would provide a more nuanced understanding of how adaptive rule generation affects long-term engagement and satisfaction in interactive environments. Addressing these challenges will not only enhance the framework' s performance but also broaden its applicability across various domains, ultimately advancing the state-of-the-art in adaptive, multi-agent systems.

Bibliography

[1] M. Zalnieriute, L. B. Moses, and G. Williams, "The rule of law and automation of government decision-making," *The Modern Law Review*, vol. 82, no. 3, pp. 425–455, 2019. 1

[2] J. Lin, W. Zhou, H. Wang, Z. Cao, W. Yu, C. Zhao, D. Zhao, D. Yang, and J. Li, "Road traffic law adaptive decision-making for self-driving vehicles," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2034–2041, IEEE, 2022. 1

[3] Y. B. Ma, "Robot reinforcement learning based on lcs-gdm," *Applied Mechanics and Materials*, vol. 347, pp. 416–420, 2013. 1

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015. 1, 2.2.1

[5] P. Sweetser and M. Aitchison, "Do game bots dream of electric rewards? the universality of intrinsic motivation," in *Proceedings of the 15th International Conference on the Foundations of Digital Games*, pp. 1–7, 2020. 1

[6] G. Sileno, A. Boer, and T. Van Engers, "A constructivist approach to rule bases," in *International Conference on Agents and Artificial Intelligence*, vol. 2, pp. 540–547, SCITEPRESS, 2015. 1

[7] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," *arXiv preprint arXiv:2303.04226*, 2023. 1

[8] S. Adams and P. A. Beling, "A survey of feature selection methods for gaussian mixture models and hidden markov models," *Artificial Intelligence Review*, vol. 52, pp. 1739–1779, 2019. 1

[9] H. X. Qin and P. Hui, "Empowering the metaverse with generative ai: Survey and future directions," in *2023 IEEE 43rd international conference on distributed computing systems workshops (ICDCSW)*, pp. 85–90, IEEE, 2023. 1

[10] N. Kshetri, Y. K. Dwivedi, T. H. Davenport, and N. Panteli, "Generative artificial intelligence in marketing: Applications, opportunities, challenges, and research agenda," 2024. 1

[11] P. Zhang and M. N. Kamel Boulos, "Generative ai in medicine and healthcare: promises, opportunities and challenges," *Future Internet*, vol. 15, no. 9, p. 286, 2023. 1

[12] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: an overview," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2*, pp. 426–440, Springer, 2018. 1, 5.2.2

[13] G. Li, S. Li, S. Li, and X. Qu, "Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient," *IET Intelligent Transport Systems*, vol. 16, no. 12, pp. 1669–1681, 2022. 1

[14] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable deep reinforcement learning for robotic manipulation," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6244–6251, IEEE, 2018. 1

[15] D. Dutta and S. R. Upreti, "A survey and comparative evaluation of actor-critic methods in process control," *The Canadian Journal of Chemical Engineering*, vol. 100, no. 9, pp. 2028–2056, 2022. 1

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 1, 2.2.1

[17] B. Zhao, W. Jin, Z. Chen, and Y. Guo, "A semi-independent policies training method with shared representation for heterogeneous multi-agents reinforcement learning," *Frontiers in Neuroscience*, vol. 17, p. 1201370, 2023. 1

[18] J. Hernández-Orallo, "Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement," *Artificial Intelligence Review*, vol. 48, pp. 397–447, 2017. 1

[19] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022. 1

[20] F. Boadu, A. Lee, and J. Cheng, "Deep learning methods for protein function prediction," *Proteomics*, vol. 25, no. 1-2, p. 2300471, 2025. 1

[21] P. Clark and T. Niblett, "The cn2 induction algorithm," *Machine learning*, vol. 3, no. 4, pp. 261–283, 1989. 1.1

[22] M. Nye, A. Solar-Lezama, J. Tenenbaum, and B. M. Lake, "Learning compositional rules via neural program synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10832–10842, 2020. 1.1

[23] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, "Meta-reinforcement learning of structured exploration strategies," *Advances in neural information processing systems*, vol. 31, 2018. 1.1

[24] J. Beck, R. Vuorio, E. Z. Liu, Z. Xiong, L. Zintgraf, C. Finn, and S. Whiteson, "A survey of meta-reinforcement learning," *arXiv preprint arXiv:2301.08028*, 2023. 1.1

[25] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*, pp. 1094–1100, PMLR, 2020. 1.1

[26] H. Yang, S. Yue, and Y. He, "Auto-gpt for online decision making: Benchmarks and additional opinions," *arXiv preprint arXiv:2306.02224*, 2023. 1.1

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 2.1.1

[28] M. J. Shafiee and A. Wong, "Evolutionary synthesis of deep neural networks via synaptic cluster-driven genetic encoding," *arXiv preprint arXiv:1609.01360*, 2016. 2.1.1

[29] M. J. Shafiee, E. Barshan, and A. Wong, "Evolution in groups: a deeper look at synaptic cluster driven evolution of deep neural networks," *arXiv preprint arXiv:1704.02081*, 2017. 2.1.1

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017. 2.1.1

[31] A. Shahroudnejad, "A survey on understanding, visualizations, and explanation of deep neural networks," *arXiv preprint arXiv:2102.01792*, 2021. 2.1.1

[32] J. H. Yousif and M. J. Yousif, "Critical review of neural network generations and models design," *Preprints*, p. 2023091149, 2023. 2.1.1

[33] F. Altenberger and C. Lenz, "A non-technical survey on deep convolutional neural network architectures. arxiv," *arXiv preprint arXiv:1803.02129*, 2018. 2.1.1

[34] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, W. Li, X. Xie, and M. Guo, "Learning graph representation with generative adversarial nets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3090–3103, 2019. 2.1.1

[35] X. Guo and L. Zhao, "A systematic survey on deep generative models for graph generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5370–5390, 2022. 2.1.1

[36] D. P. Kingma, M. Welling, *et al.*, "Auto-encoding variational bayes," 2013. 2.1.1

[37] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017. 2.1.1

[38] A. Grover, M. Dhar, and S. Ermon, "Flow-gan: Combining maximum likelihood and adversarial learning in generative models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018. 2.1.1

[39] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 2.1.1, 5.2.2

[40] H. Tu, T. Zhan, and D. Hu, "Human behavior recognition based on residual generative adversarial networks," in *International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2022)*, vol. 12288, pp. 175–182, SPIE, 2022. 2.1.1

[41] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE access*, vol. 7, pp. 36322–36333, 2019. 2.1.1

[42] X. Hu, R. Wang, D. Zhou, and Y. Xiong, "Neural topic modeling with cycle-consistent adversarial training," *arXiv preprint arXiv:2009.13971*, 2020. 2.1.1

[43] M. Di Giammarco, A. Santone, M. Cesarelli, F. Martinelli, and F. Mercaldo, "Evaluating deep learning resilience in retinal fundus classification with generative adversarial networks generated images," *Electronics*, vol. 13, no. 13, p. 2631, 2024. 2.1.1

[44] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020. 2.1.1

[45] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020. 2.1.1

[46] J. Austin *et al.*, "Structured denoising diffusion models in discrete spaces," *arXiv preprint arXiv:2101.05543*, 2021. 2.1.1

[47] S. Dai, X. Zhu, N. Li, T. Dai, and Z. Wang, "Procedural level generation with diffusion models from a single example," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 10021–10029, 2024. 2.1.1

[48] X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *Advances in neural information processing systems*, vol. 35, pp. 4328–4343, 2022. 2.1.1

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 2.1.1, 2.1.2

[50] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018. 2.1.1

[51] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. 2.1.1, 2.1.2

[52] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020. 2.1.1

[53] P. Yin and G. Neubig, "A syntactic neural model for general-purpose code generation," *arXiv preprint arXiv:1704.01696*, 2017. 2.1.1

[54] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015. 2.1.2

[55] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, vol. 15, no. 9, p. 517, 2024. 2.1.2

[56] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019. 2.1.2

[57] C.-C. Chen, H.-H. Huang, and H.-H. Chen, "Evaluating the rationales of amateur investors," in *Proceedings of the Web Conference 2021*, pp. 3987–3998, 2021. 2.1.2

[58] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022. 2.1.2

[59] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," *arXiv preprint arXiv:2212.10560*, 2022. 2.1.2

[60] K. Ellis, L. Wong, M. Nye, M. Sable-Meyer, L. Cary, L. Anaya Pozo, L. Hewitt, A. Solar-Lezama, and J. B. Tenenbaum, "Dreamcoder: growing generalizable, interpretable knowledge with wake–sleep bayesian program learning," *Philosophical Transactions of the Royal Society A*, vol. 381, no. 2251, p. 20220050, 2023. 2.1.2

[61] C. Hu, Y. Zhao, and J. Liu, "Game generation via large language models," in *2024 IEEE Conference on Games (CoG)*, pp. 1–4, IEEE, 2024. 2.1.2

[62] K. S. Tekinbas and E. Zimmerman, *Rules of play: Game design fundamentals*. MIT press, 2003. 2.1.3, 3.1

[63] M. A. Nowak, "Five rules for the evolution of cooperation," *science*, vol. 314, no. 5805, pp. 1560–1563, 2006. 2.1.3, 3.1

[64] B. Z. Tamanaha *et al.*, *On the rule of law: History, politics, theory*. Cambridge University Press, 2004. 2.1.3, 3.1

[65] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, "Orchestrating game generation," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2018. 2.1.3, 3.1, 3.1

[66] R. A. Posner, *Law, pragmatism, and democracy*. Harvard University Press, 2009. 2.1.3, 3.1

[67] M. Hatakeyama and T. Hashimoto, "Minimum nomic: a tool for studying rule dynamics," *Artificial Life and Robotics*, vol. 13, pp. 500–503, 2009. 2.1.3

[68] M. A. Janssen, "The evolution of rules in shedding-type of card games," *Advances in Complex Systems*, vol. 13, no. 06, pp. 741–754, 2010. 2.1.3

[69] A. Flynn, "Inducing game rules from varying quality game play," *arXiv preprint arXiv:2008.01664*, 2020. 2.1.3

[70] A. Khalifa, M. C. Green, D. Perez-Liebana, and J. Togelius, "General video game rule generation," in *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 170–177, IEEE, 2017. 2.1.3, 2.3.4, 3.2.1

[71] J. Togelius and J. Schmidhuber, "An experiment in automatic game design," in *2008 IEEE Symposium On Computational Intelligence and Games*, pp. 111–118, IEEE, 2008. 2.1.3, 3.2.1

[72] T. By, "Formalizing game-play," *Simulation & Gaming*, vol. 43, no. 2, pp. 157–187, 2012. 2.1.3

[73] C. S. Ang, "Rules, gameplay, and narratives in video games," *Simulation & Gaming*, vol. 37, no. 3, pp. 306–325, 2006. 2.1.3

[74] S. Yan, "Design your rules-a roguelike designe," in *Proceedings of the 2nd International Conference on Culture, Design and Social Development (CDSD 2022)*, pp. 357–363, Atlantis Press, 2023. 2.1.3

[75] H. Yahagi, S. Shimizu, T. Ogata, T. Hara, and J. Ota, "Simulation-based rule generation considering readability," *International Scholarly Research Notices*, vol. 2015, no. 1, p. 159289, 2015. 2.1.3

[76] H. Yahagi, M. Takehisa, S. Shimizu, T. Hara, and J. Ota, "Simulation-based simple and robust rule generation for motion coordination of multi-agent system," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 421–426, IEEE, 2013. 2.1.3

[77] C.-H. Lee and S.-D. Wang, "A self-organizing adaptive fuzzy controller," *Fuzzy Sets and Systems*, vol. 80, no. 3, pp. 295–313, 1996. 2.1.3

[78] M.-C. Hwang and X. Hu, "Dynamical multi-rule generation for self-organizing adaptive fuzzy controller," in *Proceedings of 6th International Fuzzy Systems Conference*, vol. 2, pp. 1015–1020, IEEE, 1997. 2.1.3

[79] G. Lee, K. Lee, and A. L. Chen, "Efficient graph-based algorithms for discovering and maintaining association rules in large databases," *Knowledge and Information Systems*, vol. 3, pp. 338–355, 2001. 2.1.3

[80] S.-L. Wang, K.-W. Huang, T.-C. Wang, and T.-P. Hong, "Maintenance of informative ruler sets for predictions," *Intelligent Data Analysis*, vol. 11, no. 3, pp. 279–292, 2007. 2.1.3

[81] D. W. Cheung, S. D. Lee, and B. Kao, "A general incremental technique for maintaining discovered association rules," in *Database Systems For Advanced Applications' 97*, pp. 185–194, World Scientific, 1997. 2.1.3

[82] L. Bradji and M. Boufaida, "A rule management system for knowledge based data cleaning," *Intelligent Information Management*, vol. 3, no. 6, pp. 230–239, 2011. 2.1.3

[83] A. Ozaki, "Learning description logic ontologies: Five approaches. where do they stand?," *KI-Künstliche Intelligenz*, vol. 34, no. 3, pp. 317–327, 2020. 2.1.3

[84] A. Prince and P. Smolensky, "Optimality theory: Constraint interaction in generative grammar," *Optimality Theory in phonology: A reader*, pp. 1–71, 2004. 2.1.3

[85] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996. 2.2.1

[86] M. A. Wiering and H. Van Hasselt, "Two novel on-policy reinforcement learning algorithms based on td (λ)-methods," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 280–287, IEEE, 2007. 2.2.1

[87] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1125–1139, 2019. 2.2.1

[88] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, pp. 1928–1937, PmLR, 2016. 2.2.1

[89] R.-J. Park, K.-B. Song, and B.-S. Kwon, "Short-term load forecasting algorithm using a similar day selection method based on reinforcement learning," *Energies*, vol. 13, no. 10, p. 2640, 2020. 2.2.1

[90] C. Farquhar, P. S. P. V. Kumar, A. Jagannath, and J. Jagannath, "Distributed transmission control for wireless networks using multi-agent reinforcement learning," in *Big Data IV: Learning, Analytics, and Applications*, vol. 12097, pp. 77–96, SPIE, 2022. 2.2.1

[91] Q. Yin, T. Yu, S. Shen, J. Yang, M. Zhao, W. Ni, K. Huang, B. Liang, and L. Wang, "Distributed deep reinforcement learning: A survey and a multi-player multi-agent learning toolbox," *Machine Intelligence Research*, vol. 21, no. 3, pp. 411–430, 2024. 2.2.1

[92] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, and S. Spanò, "Multi-agent reinforcement learning: A review of challenges and applications," *Applied Sciences*, vol. 11, no. 11, p. 4948, 2021. 2.2.1

[93] C. A. Rothkopf and C. Dimitrakakis, "Preference elicitation and inverse reinforcement learning," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part III 22*, pp. 34–48, Springer, 2011. 2.2.1

[94] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017. 2.2.1

[95] D. Brown and S. Niekum, "Efficient probabilistic performance bounds for inverse reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018. 2.2.1

[96] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl ^2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016. 2.2.1

[97] S. Narvekar, "Curriculum learning in reinforcement learning.," in *IJCAI*, pp. 5195–5196, 2017. 2.2.1

[98] L. Tai, J. Zhang, M. Liu, J. Boedecker, and W. Burgard, "A survey of deep network solutions for learning control in robotics: From reinforcement to imitation," *arXiv preprint arXiv:1612.07139*, 2016. 2.2.1

[99] K. Miyamoto, M. Suzuki, Y. Kigami, and K. Satake, "Convergence of q-value in case of gaussian rewards," in *Progress in Intelligent Decision Science: Proceeding of IDS 2020*, pp. 153–165, Springer, 2021. 2.2.1

[100] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013. 2.2.1

[101] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016. 2.2.1

[102] F. Moreno-Vera, "Performing deep recurrent double q-learning for atari games," in *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–4, IEEE, 2019. 2.2.1

[103] I. Szita, "Reinforcement learning in games," in *Reinforcement Learning: State-of-the-art*, pp. 539–577, Springer, 2012. 2.2.1

[104] S. Kumar, "Balancing a cartpole system with reinforcement learning–a tutorial," *arXiv preprint arXiv:2006.04938*, 2020. 2.2.1

[105] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015. 2.2.1

[106] H. Wang, "Artificial intelligence gamers based on deep reinforcement learning," *vol*, vol. 81, pp. 469–472, 2024. 2.2.1, 5.2.2

[107] F. Didier, S. Laghrouche, and D. Depernet, "Deep reinforcement learning-based pitch control for floating offshore wind turbines," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1–6, IEEE, 2023. 2.2.1

[108] S. V. Macua, J. Zazo, and S. Zazo, "Learning parametric closed-loop policies for markov potential games," *arXiv preprint arXiv:1802.00899*, 2018. 2.2.1

[109] L. Bisi, L. Sabbioni, E. Vittori, M. Papini, and M. Restelli, "Risk-averse trust region optimization for reward-volatility reduction," *arXiv preprint arXiv:1912.03193*, 2019. 2.2.1

[110] T. Zheng, "Dynamic difficulty adjustment using deep reinforcement learning: A review," *Applied and Computational Engineering*, vol. 71, pp. 157–162, 2024. 2.2.1

[111] L. Zhang, "Application of artificial intelligence technology in game npc," in *ICSETPSD 2023: Proceedings of the First International Conference on Science, Engineering and Technology Practices for Sustainable Development, ICSETPSD 2023, 17th-18th November 2023, Coimbatore, Tamilnadu, India*, p. 26, European Alliance for Innovation, 2024. 2.2.1

[112] W. Barfuss, J. F. Donges, and J. Kurths, "Deterministic limit of temporal difference reinforcement learning for stochastic games," *Physical Review E*, vol. 99, no. 4, p. 043305, 2019. 2.2.2

[113] M. Harper, V. Knight, M. Jones, G. Koutsovoulos, N. E. Glynatsi, and O. Campbell, "Reinforcement learning produces dominant strategies for the iterated prisoner' s dilemma," *PloS one*, vol. 12, no. 12, p. e0188046, 2017. 2.2.2

[114] C. Martin and T. Sandholm, "Efficient exploration of zero-sum stochastic games," *arXiv preprint arXiv:2002.10524*, 2020. 2.2.2

[115] L. Wang, Q. Su, H. Wang, and Y. Xia, "Exploring dominant strategies in iterated and evolutionary games: a multi-agent reinforcement learning approach," 2024. 2.2.2

[116] H. Sethy and A. Patel, "Reinforcement learning approach for real time strategy games battle city and s3," *arXiv preprint arXiv:1602.04936*, 2016. 2.2.2

[117] J. Hu, J.-R. Gaglione, Y. Wang, Z. Xu, U. Topcu, and Y. Liu, "Reinforcement learning with reward machines in stochastic games," in *ECAI 2023*, pp. 1068–1075, IOS Press, 2023. 2.2.2

[118] M. Bedau, "The scientific and philosophical scope of artificial life," *Leonardo*, vol. 35, no. 4, pp. 395–400, 2002. 2.2.3

[119] M. Dascălu, *Cellular automata and randomization: A structural overview*. IntechOpen, 2018. 2.2.3

[120] E. Bilotta and P. Pantano, "Synthetic harmonies: an approach to musical semiosis by means of cellular automata," *Leonardo*, vol. 35, no. 2, pp. 153–159, 2002. 2.2.3

[121] B. McMullin, "John von neumann and the evolutionary growth of complexity: Looking backward, looking forward," *Artificial life*, vol. 6, no. 4, pp. 347–361, 2000. 2.2.3

[122] C. Salzberg and H. Sayama, "Complex genetic evolution of artificial self-replicators in cellular automata," *Complexity*, vol. 10, no. 2, pp. 33–39, 2004. 2.2.3

[123] A. Iliopoulos, "Complex systems: Phenomenology, modeling, analysis," *Int. J. Appl. Exp. Math*, vol. 1, p. 105, 2016. 2.2.3

[124] N. Shaker, J. Togelius, and M. J. Nelson, "Procedural content generation in games," 2016. 2.3

[125] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 172–186, 2011. 2.3

[126] D. Bhaumik, A. Khalifa, M. Green, and J. Togelius, "Tree search versus optimization approaches for map generation," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, pp. 24–30, 2020. 2.3

[127] D.-F. H. Adrian and S.-G. C. A. Luisa, "An approach to level design using procedural content generation and difficulty curves," in *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, pp. 1–8, IEEE, 2013. 2.3

[128] A. Liapis, G. N. Yannakakis, and J. Togelius, "Neuroevolutionary constrained optimization for content creation," in *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pp. 71–78, IEEE, 2011. 2.3

[129] A. Liapis, G. N. Yannakakis, and J. Togelius, "Constrained novelty search: A study on game content generation," *Evolutionary computation*, vol. 23, no. 1, pp. 101–129, 2015. 2.3

[130] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 4, pp. 317–335, 2014. 2.3

[131] W. Wan and J. B. Birch, "An improved hybrid genetic algorithm with a new local search procedure," *Journal of Applied Mathematics*, vol. 2013, no. 1, p. 103591, 2013. 2.3

[132] O. A. C. Cortes and J. C. da Silva, "A local search algorithm based on clonal selection and genetic mutation for global optimization," in *2010 Eleventh Brazilian Symposium on Neural Networks*, pp. 241–246, IEEE, 2010. 2.3

[133] A. Isaacs, T. Ray, and W. Smith, "A hybrid evolutionary algorithm with simplex local search," in *2007 IEEE Congress on Evolutionary Computation*, pp. 1701–1708, IEEE, 2007. 2.3

[134] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011. 2.3

[135] G. Smith, "The future of procedural content generation in games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 10, pp. 53–57, 2014. 2.3

[136] A. M. Smith, M. J. Nelson, and M. Mateas, "Ludocore: A logical game engine for modeling videogames," in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pp. 91–98, IEEE, 2010. 2.3.1

[137] M. Cook, S. Colton, and J. Gow, "The angelina videogame design system—part i," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 2, pp. 192–203, 2016. 2.3.1

[138] M. Cook, S. Colton, and J. Gow, "The angelina videogame design system—part ii," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 254–266, 2016. 2.3.1

[139] M. Cook, "Puck: A slow and personal automated game designer," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 18, pp. 232–239, 2022. 2.3.1, 3.4.1

[140] J. Osborn, A. Grow, and M. Mateas, "Modular computational critics for games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 9, pp. 163–169, 2013. 2.3.1

[141] M. Cook, "Would you look at that! vision-driven procedural level design," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 11, pp. 9–14, 2015. 2.3.1

[142] M. Johansen and M. Cook, "Challenges in generating juice effects for automatically designed games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, pp. 42–49, 2021. 2.3.1

[143] M. Guzdial and M. Riedl, "Automated game design via conceptual expansion," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 14, pp. 31–37, 2018. 2.3.1, 3.2.1

[144] J. Gregory, *Game engine architecture*. AK Peters/CRC Press, 2018. 2.3.2

[145] P. Skop, "Comparison of performance of game engines across various platforms," *Journal of Computer Sciences Institute*, vol. 7, pp. 116–119, 2018. 2.3.2

[146] B. Szabat and M. Plechawska-Wójcik, "Comparative analysis of selected game engines," *Journal of Computer Sciences Institute*, vol. 29, pp. 312–316, 2023. 2.3.2

[147] L. G. Almeida, N. V. d. Vasconcelos, I. Winkler, and M. F. Catapan, "Innovating industrial training with immersive metaverses: a method for developing cross-platform virtual reality environments," *Applied Sciences*, vol. 13, no. 15, p. 8915, 2023. 2.3.2

[148] C. S. Chin, X. Zhong, R. Cui, C. Yang, and V. Mohan, "Virtual simulation platform for training semi-autonomous robotic vehicles' operators," in *Autonomous Vehicles*, IntechOpen, 2018. 2.3.2

[149] P. Petridis, I. Dunwell, D. Panzoli, S. Arnab, A. Protopsaltis, M. Hendrix, and S. de Freitas, "Game engines selection framework for high-fidelity serious applications," *International Journal of Interactive Worlds*, 2012. 2.3.2

[150] A. Ciekanowska, A. Kiszczak-Gliński, and K. Dziedzic, "Comparative analysis of unity and unreal engine efficiency in creating virtual exhibitions of 3d scanned models," *Journal of Computer Sciences Institute*, vol. 20, pp. 247–253, 2021. 2.3.2

[151] A. Jungherr and D. B. Schlarb, "The extended reach of game engine companies: How companies like epic games and unity technologies provide platforms for extended reality applications and the metaverse," *Social Media+ Society*, vol. 8, no. 2, p. 20563051221107641, 2022. 2.3.2

[152] A. Y. Ayas, H. Aydın, A. Çetinkaya, and Z. Güney, "Artificial intelligence (ai)-based self-deciding character development application in two-dimensional video games," *Bilgi ve İletişim Teknolojileri Dergisi*, vol. 5, no. 1, pp. 1–19, 2023. 2.3.2

[153] R.-H. Tunnel and U. Norbisrath, "A survey of estonian video game industry needs.," *Journal of Education and Learning*, vol. 11, no. 5, pp. 183–192, 2022. 2.3.2

[154] S. Chen, A. Mira, and J.-P. Onnela, "Flexible model selection for mechanistic network models," *Journal of complex networks*, vol. 8, no. 2, p. cnz024, 2020. 2.3.3

[155] A. J. O' malley and P. V. Marsden, "The analysis of social networks," *Health services and outcomes research methodology*, vol. 8, pp. 222–269, 2008. 2.3.3

[156] Q. Wu, Z. Zhang, J. Waltz, T. Ma, D. Milton, and S. Chen, "Predicting latent links from incomplete network data using exponential random graph model with outcome misclassification," *bioRxiv*, p. 852798, 2019. 2.3.3

[157] D. Vu, A. Lomi, D. Mascia, and F. Pallotti, "Relational event models for longitudinal network data with an application to interhospital patient transfers," *Statistics in medicine*, vol. 36, no. 14, pp. 2265–2287, 2017. 2.3.3

[158] Q. Zhan, W. Zhuo, and Y. Liu, "Social influence maximization for public health campaigns," *IEEE Access*, vol. 7, pp. 151252–151260, 2019. 2.3.3

[159] R. Conte and M. Paolucci, "On agent-based modeling and computational social science," *Frontiers in psychology*, vol. 5, p. 668, 2014. 2.3.3

[160] J. M. Epstein, *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, 2012. 2.3.3

[161] A. M. El-Sayed, P. Scarborough, L. Seemann, and S. Galea, "Social network analysis and agent-based modeling in social epidemiology," *Epidemiologic Perspectives & Innovations*, vol. 9, no. 1, pp. 1–9, 2012. 2.3.3

[162] B. D. Marshall, M. M. Paczkowski, L. Seemann, B. Tempalski, E. R. Pouget, S. Galea, and S. R. Friedman, "A complex systems approach to evaluate hiv prevention in metropolitan areas: preliminary implications for combination intervention strategies," 2012. 2.3.3

[163] Y. Nakai, Y. Koyama, and T. Terano, *Agent-based approaches in economic and social complex systems VIII*. Springer, 2013. 2.3.3

[164] R. Vatrapu, R. R. Mukkamala, A. Hussain, and B. Flesch, "Social set analysis: A set theoretical approach to big data analytics," *Ieee Access*, vol. 4, pp. 2542–2571, 2016. 2.3.3

[165] S. A. Matei, M. G. Russell, and E. Bertino, *Transparency in social media*. Springer, 2015. 2.3.3

[166] W. An, R. Beauvile, and B. Rosche, "Causal network analysis," *Annual Review of Sociology*, vol. 48, no. 1, pp. 23–41, 2022. 2.3.3

[167] J. Zhang and D. Centola, "Social networks and health: New developments in diffusion, online and offline," *Annual Review of Sociology*, vol. 45, no. 1, pp. 91–109, 2019. 2.3.3

[168] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, 2003. 2.3.3

[169] A. O. Yüksel and S. Somyürek, "The use of social network analysis in educational sciences studies," *Journal of Educational Technology and Online Learning*, vol. 6, no. 4, pp. 1128–1145, 2023. 2.3.3

[170] R. S. Nasr and S. El-Deeb, "Generative ai and its potential impact on humanity," *International Journal of Consumer Studies*, vol. 49, no. 1, p. e70024, 2025. 2.3.4

[171] O. Gottesman, F. Johansson, J. Meier, J. Dent, D. Lee, S. Srinivasan, L. Zhang, Y. Ding, D. Wihl, X. Peng, *et al.*, "Evaluating reinforcement learning algorithms in observational health settings," *arXiv preprint arXiv:1805.12298*, 2018. 2.3.4

[172] J. Zhang, Z. Meng, J. He, Z. Wang, and L. Liu, "Uav air game maneuver decision-making using dueling double deep q network with expert experience storage mechanism," *Drones*, vol. 7, no. 6, p. 385, 2023. 2.3.4

[173] A. Bandura, "Human agency in social cognitive theory.," *American psychologist*, vol. 44, no. 9, p. 1175, 1989. 2.4.1

[174] A. Bandura, "Social cognitive theory: An agentic perspective," *Asian journal of social psychology*, vol. 2, no. 1, pp. 21–41, 1999. 2.4.1

[175] Y. Zhu, M. Beam, Y. Ming, N. Egbert, and T. C. Smith, "A social cognitive theory approach to understanding parental attitudes and intentions to vaccinate children during the covid-19 pandemic," *Vaccines*, vol. 10, no. 11, p. 1876, 2022. 2.4.1

[176] H. Huo and Q. Li, "Influencing factors of the continuous use of a knowledge payment platform—fuzzy-set qualitative comparative analysis based on triadic reciprocal determinism," *Sustainability*, vol. 14, no. 6, p. 3696, 2022. 2.4.1

[177] Z. Bergman, M. M. Bergman, and A. Thatcher, "Agency and bandura's model of triadic reciprocal causation: An exploratory mobility study among metrorail commuters in the western cape, south africa," *Frontiers in psychology*, vol. 10, p. 411, 2019. 2.4.1

[178] X. Zhou, Y. Wang, X. Zhang, and L. Li, "The influence of decision-making logic on employees' innovative behaviour: The mediating role of positive error orientation and the moderating role of environmental dynamics," *Psychology Research and Behavior Management*, pp. 2297–2313, 2023. 2.4.1

[179] E. L. Deci and R. M. Ryan, "Self-determination theory: A macrotheory of human motivation, development, and health.," *Canadian psychology/Psychologie canadienne*, vol. 49, no. 3, p. 182, 2008. 2.4.2, 4.1

[180] E. L. Deci, R. M. Ryan, *et al.*, "Motivation, personality, and development within embedded social contexts: An overview of self-determination theory," *The Oxford handbook of human motivation*, vol. 18, no. 6, pp. 85–107, 2012. 2.4.2

[181] E. L. Deci and R. M. Ryan, "Facilitating optimal motivation and psychological well-being across life's domains.," *Canadian psychology/Psychologie canadienne*, vol. 49, no. 1, p. 14, 2008. 2.4.2

[182] R. M. Ryan and E. L. Deci, "A self-determination theory approach to psychotherapy: The motivational basis for effective change.," *Canadian psychology/Psychologie canadienne*, vol. 49, no. 3, p. 186, 2008. 2.4.2

[183] J. Nakamura and M. Csikszentmihalyi, "Flow theory and research," 2009. 2.4.3

[184] M. C. Bölen, H. Calisir, and Ü. Özen, "Flow theory in the information systems life cycle: The state of the art and future research agenda," *International Journal of Consumer Studies*, vol. 45, no. 4, pp. 546–580, 2021. 2.4.3

[185] W. O. dos Santos, I. I. Bittencourt, S. Isotani, D. Dermeval, L. B. Marques, and I. F. Silveira, "Flow theory to promote learning in educational systems: Is it really relevant?," *Revista Brasileira de Informática na Educação*, vol. 26, no. 02, p. 29, 2018. 2.4.3

[186] Y. Yang, J. Luo, and T. Lan, "An empirical assessment of a modified artificially intelligent device use acceptance model—from the task-oriented perspective," *Frontiers in psychology*, vol. 13, p. 975307, 2022. 2.4.3

[187] J. Mökander and R. Schroeder, "Ai and social theory," *AI & society*, vol. 37, no. 4, pp. 1337–1351, 2022. 2.4.3

[188] L. Fan, M. Xu, Z. Cao, Y. Zhu, and S.-C. Zhu, "Artificial social intelligence: A comparative and holistic view," *CAAI Artificial Intelligence Research*, vol. 1, no. 2, pp. 144–160, 2022. 2.4.3

[189] S.-E. Chong, S.-I. Ng, and K. Norazlyn, "A systematic review of studies on flow experience from 2010-2022. insights and directions for future research," *NUST Business Review*, vol. 4, no. 2, pp. 1–21, 2023. 2.4.3

[190] C.-L. Hsu and H.-P. Lu, “Why do people play on-line games? an extended tam with social influences and flow experience,” *Information & management*, vol. 41, no. 7, pp. 853–868, 2004. 2.4.3

[191] I. Cheah, A. S. Shimul, and I. Phau, “Motivations of playing digital games: A review and research agenda,” *Psychology & Marketing*, vol. 39, no. 5, pp. 937–950, 2022. 2.4.3

[192] C.-C. Chang, C. A. Warden, C. Liang, and G.-Y. Lin, “Effects of digital game-based learning on achievement, flow and overall cognitive load,” *Australasian Journal of Educational Technology*, vol. 34, no. 4, 2018. 2.4.3, 5.2.1

[193] Y. Xu, J. Zhu, M. Wang, F. Qian, Y. Yang, and J. Zhang, “The impact of a digital game-based ai chatbot on students’academic performance, higher-order thinking, and behavioral patterns in an information technology curriculum,” *Applied Sciences*, vol. 14, no. 15, p. 6418, 2024. 2.4.3

[194] I. Tollman and M. Yaffe, “A qualitative analysis of a team fortress 2 social network,” 2015. 2.4.3

[195] M.-P. Chen, “The effects of prior computer experience and gender on high school students’ learning of computer science concepts from instructional simulations,” in *2010 10th IEEE International Conference on Advanced Learning Technologies*, pp. 610–612, IEEE, 2010. 2.5.1

[196] C. Yin, H. Ogata, and Y. Yano, “Participatory simulation framework to support learning computer science,” *International Journal of Mobile Learning and Organisation*, vol. 1, no. 3, pp. 288–304, 2007. 2.5.1

[197] I. Douven, “Introduction: Computer simulations in social epistemology,” *Episteme*, vol. 6, no. 2, pp. 107–109, 2009. 2.5.1

[198] E. Winsberg, “Simulations, models, and theories: Complex physical systems and their representations,” *Philosophy of science*, vol. 68, no. S3, pp. S442–S454, 2001. 2.5.1

[199] S. P. You, “Application of computer simulation in materials science,” *Applied Mechanics and Materials*, vol. 189, pp. 453–456, 2012. 2.5.1

[200] A. J. Abdellatif and B. MacCollum, “A proposed framework for simulation based learning of inheritance,” in *2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim)*, pp. 75–78, IEEE, 2016. 2.5.1

[201] J. J. Billings, A. R. Bennett, J. Deyton, K. Gammeltoft, J. Graham, D. Gorin, H. Krishnan, M. Li, A. J. McCaskey, T. Patterson, *et al.*, “The eclipse integrated computational environment,” *SoftwareX*, vol. 7, pp. 234–244, 2018. 2.5.1

[202] N. Rutten, W. R. Van Joolingen, and J. T. Van Der Veen, "The learning effects of computer simulations in science education," *Computers & education*, vol. 58, no. 1, pp. 136–153, 2012. 2.5.1

[203] G. Shamir, D. Tsybulsky, and I. Levin, "Introducing computational thinking practices in learning science of elementary schools [research-in-progress]," in *InSITE 2019: Informing Science+ IT Education Conferences: Jerusalem*, pp. 187–205, 2019. 2.5.1

[204] U. Rivera-Ortega, "Interactive projectile motion stem simulation and game, based on scratch (s4a) and arduino," *Physics Education*, vol. 56, no. 6, p. 065029, 2021. 2.5.1

[205] O. Asan and A. Choudhury, "Research trends in artificial intelligence applications in human factors health care: mapping review," *JMIR human factors*, vol. 8, no. 2, p. e28236, 2021. 2.5.2

[206] M. Maadi, H. Akbarzadeh Khorshidi, and U. Aickelin, "A review on human–ai interaction in machine learning and insights for medical applications," *International journal of environmental research and public health*, vol. 18, no. 4, p. 2121, 2021. 2.5.2, 2.6.4

[207] J. Rezwana and M. L. Maher, "Designing creative ai partners with cofi: A framework for modeling interaction in human-ai co-creative systems," *ACM Transactions on Computer-Human Interaction*, vol. 30, no. 5, pp. 1–28, 2023. 2.5.2

[208] S. Sreedharan, "Human-aware ai—a foundational framework for human–ai interaction," *AI Magazine*, vol. 44, no. 4, pp. 460–466, 2023. 2.5.2

[209] S. Humr and M. Canan, "Intermediate judgments and trust in artificial intelligence-supported decision-making," *Entropy*, vol. 26, no. 6, p. 500, 2024. 2.5.2

[210] M. Kolomaznik, V. Petrik, M. Slama, and V. Jurik, "The role of socio-emotional attributes in enhancing human-ai collaboration," *Frontiers in Psychology*, vol. 15, p. 1369957, 2024. 2.5.2

[211] R. Cheng, A. Smith-Renner, K. Zhang, J. R. Tetreault, and A. Jaimes, "Mapping the design space of human-ai interaction in text summarization," *arXiv preprint arXiv:2206.14863*, 2022. 2.5.2

[212] R. Fogliato, S. Chappidi, M. Lungren, P. Fisher, D. Wilson, M. Fitzke, M. Parkinson, E. Horvitz, K. Inkpen, and B. Nushi, "Who goes first? influences of human-ai workflow on decision making in clinical imaging," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1362–1374, 2022. 2.5.2

[213] G. Pisoni, N. Díaz-Rodríguez, H. Gijlers, and L. Tonolli, "Human-centered artificial intelligence for designing accessible cultural heritage," *Applied Sciences*, vol. 11, no. 2, p. 870, 2021. 2.5.2

[214] N. J. Nilsson, *Principles of artificial intelligence.* Morgan Kaufmann, 2014. 2.6.1

[215] S. Russell and P. Norvig, "A modern, agent-oriented approach to introductory artificial intelligence," *Acm Sigart Bulletin*, vol. 6, no. 2, pp. 24–26, 1995. 2.6.1

[216] M. Haenlein and A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence," *California management review*, vol. 61, no. 4, pp. 5–14, 2019. 2.6.1

[217] F. Undie, S. Hameed, J. Ikwen, M. Okache, and R. Adebayo, "Artificial intelligence and machine learning: A review of state-of-the-art trends, global developments, and practical implications," *J Curr Trends Comp Sci Res*, vol. 3, no. 1, pp. 1–6, 2024. 2.6.1

[218] Y. Xu, X. Liu, X. Cao, C. Huang, E. Liu, S. Qian, X. Liu, Y. Wu, F. Dong, C.-W. Qiu, *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," *The Innovation*, vol. 2, no. 4, 2021. 2.6.1

[219] N. Anantrasirichai and D. Bull, "Artificial intelligence in the creative industries: a review," *Artificial intelligence review*, vol. 55, no. 1, pp. 589–656, 2022. 2.6.1

[220] R. T. Hughes, L. Zhu, and T. Bednarz, "Generative adversarial networks–enabled human–artificial intelligence collaborative applications for creative and design industries: A systematic review of current approaches and trends," *Frontiers in artificial intelligence*, vol. 4, p. 604234, 2021. 2.6.1, 2.6.2

[221] P. Gohel, P. Singh, and M. Mohanty, "Explainable ai: current status and future directions," *arXiv preprint arXiv:2107.07045*, 2021. 2.6.2

[222] C. Sharma, S. Sharma, K. Sharma, G. K. Sethi, and H.-Y. Chen, "Exploring explainable ai: a bibliometric analysis," *Discover Applied Sciences*, vol. 6, no. 11, pp. 1–23, 2024. 2.6.2

[223] A. Alamäki, L. Aunimo, H. Ketamo, and L. Parvinen, "Interactive machine learning: Managing information richness in highly anonymized conversation data," in *Collaborative Networks and Digital Transformation: 20th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2019, Turin, Italy, September 23–25, 2019, Proceedings 20*, pp. 173–184, Springer, 2019. 2.6.3

[224] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?," *Brain informatics*, vol. 3, no. 2, pp. 119–131, 2016. 2.6.3

[225] T. Heart, R. Padman, O. Ben-Assuli, D. Gefen, and R. Klempfner, "On intelligence augmentation and visual analytics to enhance clinical decision support systems," 2022. 2.6.3

[226] D. D. Schmorrow and A. Kruse, "Augmented cognition," *Berkshire encyclopedia of human-computer interaction*, vol. 1, pp. 54–59, 2004. 2.6.3

[227] M. Alonso, "Learning user preferences via reinforcement learning with spatial interface valuing," in *Universal Access in Human-Computer Interaction. Multimodality and Assistive Environments: 13th International Conference, UAHCI 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part II 21*, pp. 403–418, Springer, 2019. 2.6.3

[228] S.-A. N. Alexandropoulos, C. K. Aridas, S. B. Kotsiantis, and M. N. Vrahatis, "Stacking strong ensembles of classifiers," in *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15*, pp. 545–556, Springer, 2019. 2.6.4

[229] J. Corny, A. Rajkumar, O. Martin, X. Dode, J.-P. Lajonchère, O. Billuart, Y. Bézie, and A. Buronfosse, "A machine learning–based clinical decision support system to identify prescriptions with a high risk of medication error," *Journal of the American Medical Informatics Association*, vol. 27, no. 11, pp. 1688–1694, 2020. 2.6.4

[230] N. Sahebjamnia, S. A. Torabi, and S. A. Mansouri, "A hybrid decision support system for managing humanitarian relief chains," *Decision Support Systems*, vol. 95, pp. 12–26, 2017. 2.6.4

[231] X. Liu, T. Suel, and N. Memon, "A robust model for paper reviewer assignment," in *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 25–32, 2014. 2.6.4

[232] Z. Wu, Y. Mao, and Q. Li, "Procedural game map generation using multi-leveled cellular automata by machine learning," in *Proceedings of the 2nd International Symposium on Artificial Intelligence for Medicine Sciences*, pp. 168–172, 2021. 3.1

[233] S. W. Yackee, "The politics of rulemaking in the united states," *Annual Review of Political Science*, vol. 22, pp. 37–55, 2019. 3.1

[234] J. P. da Costa, C. Mouneyrac, M. Costa, A. C. Duarte, and T. Rocha-Santos, "The role of legislation, regulatory initiatives and guidelines on the control of plastic pollution," *Frontiers in Environmental Science*, vol. 8, p. 104, 2020. 3.1

[235] J. Togelius and J. Schmidhuber, "An experiment in automatic game design," in *2008 IEEE Symposium On Computational Intelligence and Games*, pp. 111–118, IEEE, 2008. 3.1

[236] A. Farizawani, M. Puteh, Y. Marina, and A. Rivaie, "A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches," in *Journal of Physics: Conference Series*, vol. 1529, p. 022040, IOP Publishing, 2020. 3.1

[237] Z. Wu, Y. Mao, and Q. Li, "Procedural game map generation using multi-leveled cellular automata by machine learning," in *Proceedings of the 2nd International Symposium on Artificial Intelligence for Medicine Sciences*, pp. 168–172, 2021. 3.1

[238] M. Liu, A. Grinberg Dana, M. S. Johnson, M. J. Goldman, A. Jocher, A. M. Payne, C. A. Grambow, K. Han, N. W. Yee, E. J. Mazeau, *et al.*, "Reaction mechanism generator v3. 0: advances in automatic mechanism generation," *Journal of Chemical Information and Modeling*, vol. 61, no. 6, pp. 2686–2696, 2021. 3.1

[239] F. Angaroni, A. Graudenzi, M. Rossignolo, D. Maspero, T. Calarco, R. Piazza, S. Montangero, and M. Antoniotti, "An optimal control framework for the automated design of personalized cancer treatments," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 523, 2020. 3.1

[240] Z. Zeng, K. Ou, L. Wang, and Y. Yu, "Reliability-oriented automated design of double-sided cooling power module: A thermo-mechanical-coordinated and multi-objective-oriented optimization methodology," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 3, pp. 584–595, 2020. 3.1

[241] C. G. Langton, *Genetic Algorithms and Artificial Life*, pp. 267–289. 1997. 3.1, 3.2.1

[242] S. G. Ficici and J. B. Pollack, "Challenges in coevolutionary learning: Arms-race dynamics," in *Artificial Life VI: Proceedings of the sixth international conference on artificial life*, vol. 6, p. 238, MIT Press, 1998. 3.1

[243] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," 3.1

[244] W. Aguilar, G. Santamaría-Bonfil, T. Froese, and C. Gershenson, "The past, present, and future of artificial life," *Frontiers in Robotics and AI*, vol. 1, p. 8, 2014. 3.1, 3.2.1

[245] B. Chopard and M. Droz, "Cellular automata," *Modelling of Physical*, 1998. 3.1

[246] N. S. Lachenmyer and S. Akasha, "An aquarium of machines: A physically realized artificial life simulation," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 5, no. 4, pp. 1–11, 2022. 3.1

[247] J. Y. P. IgorMordatch, "Neural mmo: A massively multiagent game environment for training and evaluating intelligent agents," 3.1, 3.2.1

[248] S. Mystakidis, "Metaverse," *Encyclopedia*, vol. 2, no. 1, pp. 486–497, 2022. 3.1

[249] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, "A fast and accurate one-stage approach to visual grounding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4683–4693, 2019. 3.1

[250] P. Mirza-Babaei, R. Robinson, R. Mandryk, J. Pirker, C. Kang, and A. Fletcher, “Games and the metaverse,” in *Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play*, pp. 318–319, 2022. 3.1

[251] E. Shin and J. H. Kim, “The metaverse and video games: merging media to improve soft skills training,” *Journal of Internet Computing and Services*, vol. 23, no. 1, pp. 69–76, 2022. 3.1

[252] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, “Procedural content generation via machine learning (pcgml),” *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018. 3.1

[253] S. Risi and J. Togelius, “Procedural content generation: from automatically generating game levels to increasing generality in machine learning,” *arXiv preprint arXiv:1911.13071*, 2019. 3.1, 3.2.1

[254] C. Schifter and M. Cipollone, “Minecraft as a teaching tool: One case study,” in *Society for Information Technology & Teacher Education International Conference*, pp. 2951–2955, Association for the Advancement of Computing in Education (AACE), 2013. 3.1

[255] P. Mawhorter, “Efficiency, realism, and representation in generated content: a case study using family tree generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pp. 1–4, 2017. 3.1

[256] D. Perez-Liebana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, “General video game ai: A multitrack framework for evaluating agents, games, and content generation algorithms,” *IEEE Transactions on Games*, vol. 11, no. 3, pp. 195–214, 2019. 3.1

[257] A. Strong, “Applications of artificial intelligence & associated technologies,” *Science [ETEBMS-2016]*, vol. 5, no. 6, 2016. 3.1

[258] D. Perez-Liebana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, “General video game ai: A multitrack framework for evaluating agents, games, and content generation algorithms,” *IEEE Transactions on Games*, vol. 11, no. 3, pp. 195–214, 2019. 3.1

[259] A. Oussidi and A. Elhassouny, “Deep generative models: Survey,” in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1–8, IEEE, 2018. 3.1

[260] M. Abukmeil, S. Ferrari, A. Genovese, V. Piuri, and F. Scotti, “A survey of unsupervised generative models for exploratory data analysis and representation learning,” *Acm computing surveys (csur)*, vol. 54, no. 5, pp. 1–40, 2021. 3.1

[261] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *stat*, vol. 1050, p. 1, 2014. 3.1

[262] X. Li, H. Zhang, and R. Zhang, "Adaptive graph auto-encoder for general data clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9725–9732, 2022. 3.1

[263] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. 3.1

[264] F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen, "Deep learning fault diagnosis method based on global optimization gan for unbalanced data," *Knowledge-Based Systems*, vol. 187, p. 104837, 2020. 3.1

[265] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3617–3621, IEEE, 2019. 3.1

[266] X. Sheng, L. Li, D. Liu, Z. Xiong, Z. Li, and F. Wu, "Deep-pcac: An end-to-end deep lossy compression framework for point cloud attributes," *IEEE Transactions on Multimedia*, vol. 24, pp. 2617–2632, 2021. 3.1

[267] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017. 3.1, 3.2.2, 5.1

[268] V. Mnih, K. Kavukcuoglu, D. Silver, A. G. I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 3.1, 5.1

[269] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021. 3.1, 5.1

[270] L. Yaeger *et al.*, "Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or poly world: Life in a new context," Citeseer. 3.2.1

[271] B. Zhao, M. Ye, L. Stankovic, and V. Stankovic, "Non-intrusive load disaggregation solutions for very low-rate smart meter data," *Applied Energy*, vol. 268, p. 114949, 2020. 3.2.1

[272] P. Supramaniam, O. Ces, and A. Salehi-Reyhani, "Microfluidics for artificial life: techniques for bottom-up synthetic biology," *Micromachines*, vol. 10, no. 5, p. 299, 2019. 3.2.1

[273] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, "Emergent complexity via multi-agent competition," in *International Conference on Learning Representations*, 2018. 3.2.1

[274] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet of Things Journal*, 2023. 3.2.1

[275] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks for moment localization with natural language," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12870–12877, 2020. 3.2.1

[276] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, *et al.*, "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018. 3.2.1

[277] G. E. Raptis, C. Fidas, and N. Avouris, "Effects of mixed-reality on players' behaviour and immersion in a cultural tourism game: A cognitive processing perspective," *International Journal of Human-Computer Studies*, vol. 114, pp. 69–79, 2018. 3.2.1

[278] H. Ning, Z. Zhen, F. Shi, and M. Daneshmand, "A survey of identity modeling and identity addressing in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4697–4710, 2020. 3.2.1

[279] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 9, no. 1, pp. 1–22, 2013. 3.2.1

[280] A. Summerville, J. Osborn, and M. Mateas, "Charda: causal hybrid automata recovery via dynamic analysis," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2800–2806, 2017. 3.2.1

[281] B. Pell, "Metagame in symmetric chess-like games," *Computational Intelligence*, vol. 12, no. 1, pp. 177–198, 1996. 3.2.1

[282] V. Hom and J. Marks, "Automatic design of balanced board games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 3, pp. 25–30, 2007. 3.2.1

[283] C. Browne and F. Maire, "Evolutionary game design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 1–16, 2010. 3.2.1

[284] M. Cook, S. Colton, A. Raad, and J. Gow, "Mechanic miner: Reflection-driven game mechanic discovery and level design," in *Applications of Evolutionary Computation: 16th European Conference, EvoApplications 2013, Vienna, Austria, April 3-5, 2013. Proceedings 16*, pp. 284–293, Springer, 2013. 3.2.1

[285] L. Chiariglione, M. Choi, G. Chollet, F. Damiani, J. Kang, D. Schultens, M. Seligman, G. Torta, and F. Yassa, "Towards a standard for human interaction with connected autonomous vehicles," in *2022 Fifth International Conference on Connected and Autonomous Driving (MetroCAD)*, pp. 63–71, IEEE, 2022. 3.2.1

[286] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, "Procedural content generation via machine learning (pcgml)," *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018. 3.2.1

[287] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation," in *Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part I*, pp. 141–150, Springer, 2010. 3.2.1

[288] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Transactions on computational intelligence and AI in games*, vol. 3, no. 3, pp. 201–215, 2011. 3.2.1

[289] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 9, pp. 215–216, 2013. 3.2.1

[290] A. J. Summerville and M. Mateas, "Super mario as a string: Platformer level generation via lstms," 2016. 3.2.1

[291] R. Jain, A. Isaksen, C. Holmgård, and J. Togelius, "Autoencoders for level generation, repair, and recognition," in *Proceedings of the ICCC workshop on computational creativity and games*, vol. 9, 2016. 3.2.1

[292] A. Summerville, J. R. Mariño, S. Snodgrass, S. Ontañón, and L. H. Lelis, "Understanding mario: an evaluation of design metrics for platformers," in *Proceedings of the 12th international conference on the foundations of digital games*, pp. 1–10, 2017. 3.2.1

[293] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 3.2.2

[294] C. Wang, W. Pedrycz, J. Yang, M. Zhou, and Z. Li, "Wavelet frame-based fuzzy c-means clustering for segmenting images on graphs," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3938–3949, 2019. 3.2.2

[295] C. Wang, W. Pedrycz, Z. Li, and M. Zhou, "Residual-driven fuzzy c-means clustering for image segmentation," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 876–889, 2020. 3.2.2

[296] C. Wang, W. Pedrycz, M. Zhou, and Z. Li, "Sparse regularization-based fuzzy c-means clustering incorporating morphological grayscale reconstruction and wavelet frames," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 7, pp. 1826–1840, 2020. 3.2.2

[297] C. Wang, Z. Yan, W. Pedrycz, M. Zhou, and Z. Li, “A weighted fidelity and regularization-based method for mixed or unknown noise removal from images on graphs,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5229–5243, 2020. 3.2.2

[298] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” *arXiv preprint arXiv:2204.03458*, 2022. 3.2.2

[299] Z. Chen, X. Tan, K. Wang, S. Pan, D. Mandic, L. He, and S. Zhao, “Infergrad: Improving diffusion models for vocoder by considering inference in training,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8432–8436, IEEE, 2022. 3.2.2

[300] X. Zhang, Y. Wong, X. Wu, J. Lu, M. Kankanhalli, X. Li, and W. Geng, “Learning causal representation for training cross-domain pose estimator via generative interventions,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11270–11280, 2021. 3.2.2

[301] W. Lin, H. Lan, H. Wang, and B. Li, “Orphicx: A causality-inspired latent variable model for interpreting graph neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13729–13738, 2022. 3.2.2

[302] S. Li, X. Zhao, L. Stankovic, and D. Mandic, “Demystifying cnns for images by matched filters,” *arXiv preprint arXiv:2210.08521*, 2022. 3.2.2

[303] R. Karp and Z. Swiderska-Chadaj, “Automatic generation of graphical game assets using gan,” in *2021 7th International Conference on Computer Technology Applications*, pp. 7–12, 2021. 3.2.2

[304] X. Sheng, L. Xu, Y. Xu, D. Jiang, and B. Ren, “Semantic-preserving abstractive text summarization with siamese generative adversarial net,” in *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2121–2132, 2022. 3.2.2

[305] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021. 3.2.2

[306] W. ZHANG, D. Zhou, and Q. Gu, “Reward-free model-based reinforcement learning with linear function approximation,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 1582–1593, Curran Associates, Inc., 2021. 3.2.2

[307] R. Raileanu, M. Goldstein, D. Yarats, I. Kostrikov, and R. Fergus, “Automatic data augmentation for generalization in reinforcement learning,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 5402–5415, Curran Associates, Inc., 2021. 3.2.2

[308] B. H. Zhang and T. Sandholm, "Finding and certifying (near-) optimal strategies in black-box extensive-form games," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 5779–5788, 2021. 3.3.1, 4.4.1

[309] P. D. Straffin, *Game Theory and Strategy*, vol. 36. American Mathematical Society, 2023. 3.3.1, 4.4.1

[310] J. Paavilainen, "Critical review on video game evaluation heuristics: social games perspective," in *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, pp. 56–65, 2010. 3.3.1, 4.4.1, 5.3.1

[311] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (xml)," *World Wide Web Journal*, vol. 2, no. 4, pp. 27–66, 1997. 3.3.3

[312] J. Wu, J. B. Tenenbaum, and P. Kohli, "Neural scene de-rendering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 699–707, 2017. 3.3.3

[313] X. Lu, J. Gonzalez, Z. Dai, and N. D. Lawrence, "Structured variationally auto-encoded optimization," in *International conference on machine learning*, pp. 3267–3275, PMLR, 2018. 3.3.3

[314] S.-L. Shaw, X. Ye, M. Goodchild, and D. Sui, "Human dynamics research in giscience: challenges and opportunities," *Computational Urban Science*, vol. 4, no. 1, p. 31, 2024. 4.1

[315] R. de Filippis and A. Al Foysal, "Insights unveiled: Harnessing ai to explore human behaviour in social sciences," *Authorea Preprints*, 2024. 4.1

[316] J. Cowls, T. King, M. Taddeo, and L. Floridi, "Designing ai for social good: Seven essential factors," *Available at SSRN 3388669*, 2019. 4.1

[317] J. Y. Ng, N. Ntoumanis, C. Thøgersen-Ntoumani, E. L. Deci, R. M. Ryan, J. L. Duda, and G. C. Williams, "Self-determination theory applied to health contexts: A meta-analysis," *Perspectives on psychological science*, vol. 7, no. 4, pp. 325–340, 2012. 4.1

[318] F. B. Gillison, P. Rouse, M. Standage, S. J. Sebire, and R. M. Ryan, "A meta-analysis of techniques to promote motivation for health behaviour change from a self-determination theory perspective," *Health psychology review*, vol. 13, no. 1, pp. 110–130, 2019. 4.1

[319] M. D. Young, R. Plotnikoff, C. Collins, R. Callister, and P. Morgan, "Social cognitive theory and physical activity: a systematic review and meta-analysis," *Obesity reviews*, vol. 15, no. 12, pp. 983–995, 2014. 4.1

[320] O. D. Apuke and B. Omar, "What drives news sharing behaviour among social media users? a relational communication model from the social capital perspective," *International Sociology*, vol. 36, no. 3, pp. 339–361, 2021. 4.1

[321] O. Pesamaa, J. F. Hair Jr, and A. Haahti, "Motives, partner selection and establishing trust reciprocity and interorganisational commitment," *International Journal of Tourism Policy*, vol. 3, no. 1, pp. 62–77, 2010. 4.1

[322] S. Mirbakhsh, "Artificial intelligence-based real-time traffic management review article," *Journal of Electrical and Electronics Engineering*, vol. 2, no. 4, pp. 368–373, 2023. 4.1

[323] D. Ha and Y. Tang, "Collective intelligence for deep learning: A survey of recent developments," *Collective Intelligence*, vol. 1, no. 1, p. 26339137221114874, 2022. 4.1

[324] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *2018 International conference on intelligent systems and computer vision (ISCV)*, pp. 1–8, IEEE, 2018. 4.1

[325] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–49, 2021. 4.1

[326] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, 2022. 4.1

[327] M. Nelson and M. Mateas, "Recombinable game mechanics for automated design support," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 4, pp. 84–89, 2008. 4.1

[328] J. Pu, H. Duan, J. Zhao, and Y. Long, "Rules for expectation: Learning to generate rules via social environment modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 8, pp. 6874–6887, 2023. 4.1

[329] A. K. Mackay, L. Riazuelo, and L. Montano, "Rl-dovs: Reinforcement learning for autonomous robot navigation in dynamic environments," *Sensors*, vol. 22, no. 10, p. 3847, 2022. 4.1

[330] M. Korshunova, N. Huang, S. Capuzzi, D. S. Radchenko, O. Savych, Y. S. Moroz, C. I. Wells, T. M. Willson, A. Tropsha, and O. Isayev, "Generative and reinforcement learning approaches for the automated de novo design of bioactive compounds," *Communications Chemistry*, vol. 5, no. 1, p. 129, 2022. 4.1

[331] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications*, vol. 231, p. 120495, 2023. 4.1

[332] L. Yang, X. Li, M. Sun, and C. Sun, "Hybrid policy-based reinforcement learning of adaptive energy management for the energy transmission-constrained island group," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 11, pp. 10751–10762, 2023. 4.1

[333] M. Ahmed, A. Abobakr, C. P. Lim, and S. Nahavandi, "Policy-based reinforcement learning for training autonomous driving agents in urban areas with affordance learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12562–12571, 2021. 4.1

[334] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021. 4.1

[335] N. B. Ødegaard, H. T. Myrhaug, T. Dahl-Michelsen, and Y. Røe, "Digital learning designs in physiotherapy education: a systematic review and meta-analysis," *BMC medical education*, vol. 21, pp. 1–18, 2021. 4.1

[336] J. Staddon, "The dynamics of behavior: Review of sutton and barto: Reinforcement learning: An introduction .," *Journal of the Experimental Analysis of Behavior*, vol. 113, no. 2, 2020. 4.1

[337] M. L. Schiavo, B. Prinari, I. Saito, K. Shoji, and C. C. Benight, "A dynamical systems approach to triadic reciprocal determinism of social cognitive theory," *Mathematics and Computers in Simulation*, vol. 159, pp. 18–38, 2019. 4.1

[338] G. S. Alberti and Y. Capdeboscq, "Combining the runge approximation and the whitney embedding theorem in hybrid imaging," *International Mathematics Research Notices*, vol. 2022, no. 6, pp. 4387–4406, 2022. 4.4.4

[339] K. Kiili and T. Lainema, "Evaluations of an experiential gaming model: The realgame case," in *EdMedia+ Innovate Learning*, pp. 2343–2350, Association for the Advancement of Computing in Education (AACE), 2006. 5.1

[340] S. P. Sithungu and E. M. Ehlers, "Adaptive game ai-based dynamic difficulty scaling via the symbiotic game agent," in *International Conference on Intelligent Information Processing*, pp. 107–117, Springer, 2020. 5.1

[341] Y. S. Chee, "Embodiment, embeddedness, and experience: Game-based learning and the construction of identity," *Research and Practice in Technology Enhanced Learning*, vol. 2, no. 01, pp. 3–30, 2007. 5.1

[342] A. Yessad, J.-M. Labat, and F. Kermorvant, "Segae: A serious game authoring environment," in *2010 10th IEEE International Conference on Advanced Learning Technologies*, pp. 538–540, IEEE, 2010. 5.1

[343] K. Merrick, "Modeling motivation for adaptive nonplayer characters in dynamic computer game worlds," *Computers in Entertainment (CIE)*, vol. 5, no. 4, pp. 1–32, 2008. 5.1

[344] Y. Huang, J. Chen, L. Huang, and Q. Zhu, "Dynamic games for secure and resilient control system design," *National Science Review*, vol. 7, no. 7, pp. 1125–1141, 2020. 5.1

[345] H. Zhang, "The application of artificial intelligence technology in game design," *Applied and Computational Engineering*, vol. 19, pp. 197–204, 2023. 5.1

[346] S.-Y. Chou, W. Jang, S. C. Ma, C.-H. Chang, and K. K. Byon, "Is mobile gaming a new pillar of esports? exploring players' in-game purchases in pc and mobile platforms by using flow and clutch," *International Journal of Sports Marketing and Sponsorship*, vol. 24, no. 2, pp. 311–332, 2023. 5.1, 5.2.1

[347] G.-Y. Liao, T. Cheng, and C.-I. Teng, "How do avatar attractiveness and customization impact online gamers' flow and loyalty?," *Internet Research*, vol. 29, no. 2, pp. 349–366, 2019. 5.1

[348] A. Alexiou, M. C. Schippers, I. Oshri, and S. Angelopoulos, "Narrative and aesthetics as antecedents of perceived learning in serious games," *Information Technology & People*, vol. 35, no. 8, pp. 142–161, 2022. 5.1

[349] M. Palaus, E. M. Marron, R. Viejo-Sobera, and D. Redolar-Ripoll, "Neural basis of video gaming: A systematic review," *Frontiers in human neuroscience*, vol. 11, p. 248, 2017. 5.1

[350] M. J. Dixon, M. Stange, C. J. Larche, C. Graydon, J. A. Fugelsang, and K. A. Harrigan, "Dark flow, depression and multiline slot machine play," *Journal of gambling studies*, vol. 34, pp. 73–84, 2018. 5.1

[351] S. G. Spicer, J. Close, L. L. Nicklin, M. Uther, B. Whalley, C. Fullwood, J. Parke, J. Lloyd, and H. Lloyd, "Exploring the relationships between psychological variables and loot box engagement, part 2: exploratory analyses of complex relationships," *Royal Society Open Science*, vol. 11, no. 1, p. 231046, 2024. 5.1

[352] M. Alizadeh, F. Heidari, A. Mirzazadeh, S. S. Peighambardoust, F. Beheshtizadeh, A. Angouraj Taghavi, A. Saramad, L. Janani, and G. Hassanzadeh, "Comparing flow experience of medical students in cognitive, behavioral, and social educational games: A quasi-experimental study," *Journal of Medical Education Development*, vol. 15, no. 47, pp. 36–42, 2022. 5.1

[353] K. Trakšelys, "Education sociological paradigms importance of education sciences," *Pedagogika/Pedagogy*, vol. 129, no. 1, pp. 5–14, 2018. 5.1

[354] I. Chow and L. Huang, "An expert gamification system with psychological theories for virtual and cross-cultural software teams," *International Journal of Software Engineering & Applications*, vol. 8, no. 2, pp. 1–16, 2017. 5.1

[355] W. Oliveira, O. Pastushenko, L. Rodrigues, A. M. Toda, P. T. Palomino, J. Hamari, and S. Isotani, "Does gamification affect flow experience? a systematic literature review," *arXiv preprint arXiv:2106.09942*, 2021. 5.1

[356] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work?–a literature review of empirical studies on gamification," in *2014 47th Hawaii international conference on system sciences*, pp. 3025–3034, Ieee, 2014. 5.1

[357] D. Li, H. Yang, and Z. Hu, “Exploring the ineffectiveness of gamification health management: a u-shaped relationship between competition and technological exhaustion,” *Information Technology & People*, vol. 37, no. 3, pp. 1229–1250, 2024. 5.1

[358] M. A. Merhabi, P. Petridis, and R. Khusainova, “Gamification for brand value co-creation: A systematic literature review,” *Information*, vol. 12, no. 9, p. 345, 2021. 5.1

[359] W. Park and H. Kwon, “Implementing artificial intelligence education for middle school technology education in republic of korea,” *International journal of technology and design education*, vol. 34, no. 1, pp. 109–135, 2024. 5.1

[360] A. Burgon, N. Petrick, B. Sahiner, G. Pennello, K. H. Cha, and R. K. Samala, “A tool for the assessment of ai generalizability via decision space composition,” in *Medical Imaging 2024: Computer-Aided Diagnosis*, vol. 12927, pp. 337–342, SPIE, 2024. 5.1

[361] Q. Wu, M.-Q. Li, and J.-H. Wang, “Behavioral intentions in metaverse tourism: An extended technology acceptance model with flow theory,” *Information*, vol. 15, no. 10, p. 632, 2024. 5.1

[362] L. Hadjiiski, K. Cha, H.-P. Chan, K. Drukker, L. Morra, J. J. Näppi, B. Sahiner, H. Yoshida, Q. Chen, T. M. Deserno, *et al.*, “Aapm task group report 273: recommendations on best practices for ai and machine learning for computer-aided diagnosis in medical imaging,” *Medical physics*, vol. 50, no. 2, pp. e1–e24, 2023. 5.1

[363] Y. Mao, X. Luo, S. Wang, Z. Mao, M. Xie, and M. Bonaiuto, “Flow experience fosters university students’ well-being through psychological resilience: A longitudinal design with cross-lagged analysis,” *British Journal of Educational Psychology*, vol. 94, no. 2, pp. 518–538, 2024. 5.1

[364] N. Fisher and A. K. Kulshreshth, “Exploring dynamic difficulty adjustment methods for video games,” in *Virtual Worlds*, vol. 3, pp. 230–255, MDPI, 2024. 5.1

[365] L. K. Kaye, “Exploring flow experiences in cooperative digital gaming contexts,” *Computers in Human Behavior*, vol. 55, pp. 286–291, 2016. 5.2.1

[366] A. Bhand, R. Bhand, and M. Mali, “Mage runner: The game using blockchain technology,” *Journal of Innovations in Business and Industry*, vol. 2, no. 2, pp. 69–78, 2024. 5.2.1

[367] L.-O. Wehden, F. Reer, R. Janzik, W. Y. Tang, and T. Quandt, “The slippery path to total presence: How omnidirectional virtual reality treadmills influence the gaming experience,” *Media and Communication*, vol. 9, no. 1, pp. 5–16, 2021. 5.2.1

[368] W. J. Shelstad, D. C. Smith, and B. S. Chaparro, "Gaming on the rift: How virtual reality affects game user satisfaction," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 61, pp. 2072–2076, SAGE Publications Sage CA: Los Angeles, CA, 2017. 5.2.1

[369] D. Jagli, D. S. Chandra, S. R. Dhanikonda, and N. Laxmi, "Artificial intelligence usage in game development," *Artificial Intelligence Usage in Game Development (August 19, 2024)*, 2024. 5.2.1

[370] M. Ninaus, K. Moeller, J. McMullen, and K. Kiili, "Acceptance of game-based learning and intrinsic motivation as predictors for learning success and flow experience," *International Journal of Serious Games*, vol. 4, no. 3, pp. 15–30, 2017. 5.2.1

[371] B. M. Alom, C. Scoular, and N. Awwal, "Multiplayer game design: performance enhancement with employment of novel technology," *International Journal of Computer Applications*, vol. 145, no. 1, pp. 27–32, 2016. 5.2.1

[372] C. Cioffi-Revilla, "A methodology for complex social simulations," *Journal of Artificial Societies and Social Simulation*, vol. 13, no. 1, p. 7, 2010. 5.2.1

[373] P. Hedström and G. Manzo, "Recent trends in agent-based computational research: A brief introduction," *Sociological Methods & Research*, vol. 44, no. 2, pp. 179–185, 2015. 5.2.1

[374] J. Kozlak and A. Zygmunt, "Agent-based modelling of social organisations," in *2011 International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 467–472, IEEE, 2011. 5.2.1

[375] S. Song and S.-H. Choi, "Modeling dynamic organizational network structure," in *Reshaping Society through Analytics, Collaboration, and Decision Support: Role of Business Intelligence and Social Media*, pp. 191–203, Springer, 2014. 5.2.1

[376] R. Mercuur, V. Dignum, and C. M. Jonker, "Integrating social practice theory in agent-based models: A review of theories and agents," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 5, pp. 1131–1145, 2020. 5.2.1

[377] D. Rato and R. Prada, "Towards social identity in socio-cognitive agents," *Sustainability*, vol. 13, no. 20, p. 11390, 2021. 5.2.1

[378] T. M. Vu, C. Buckley, H. Bai, A. Nielsen, C. Probst, A. Brennan, P. Shuper, M. Strong, and R. C. Purshouse, "Multiobjective genetic programming can improve the explanatory capabilities of mechanism-based models of social systems," *Complexity*, vol. 2020, no. 1, p. 8923197, 2020. 5.2.1

[379] Y. Wei, D. Chen, and H. Xu, "A social friend selection method based on interest and social behavior," in *Fifth International Conference on Artificial Intelligence and Computer Science (AICS 2023)*, vol. 12803, pp. 274–288, SPIE, 2023. 5.2.1

[380] E. Bruch and J. Atwell, “Agent-based models in empirical social research,” *Sociological methods & research*, vol. 44, no. 2, pp. 186–221, 2015. 5.2.1

[381] D. Geschke, J. Lorenz, and P. Holtz, “The triple-filter bubble: Using agent-based modelling to test a meta-theoretical framework for the emergence of filter bubbles and echo chambers,” *British Journal of Social Psychology*, vol. 58, no. 1, pp. 129–149, 2019. 5.2.1

[382] T. Miller and Z. Jing, “Explanation in artificial intelligence: Insights from the social sciences,” *Digital Humanities Research*, vol. 4, no. 2, p. 90, 2024. 5.2.1

[383] J. Hamari and J. Koivisto, “Why do people use gamification services?,” *International journal of information management*, vol. 35, no. 4, pp. 419–431, 2015. 5.2.1

[384] A. Rapp, “From games to gamification: A classification of rewards in world of warcraft for the design of gamified systems,” *Simulation & Gaming*, vol. 48, no. 3, pp. 381–401, 2017. 5.2.1

[385] I. A. Bogoslov, E. A. STOICA, M. R. Georgescu, and A. E. LUNGU, “Gamification in e-commerce: Advantages, challenges, and future trends.,” *Revista Economică*, vol. 75, no. 2, 2023. 5.2.1

[386] L. Rodrigues, A. M. Toda, W. Oliveira, P. T. Palomino, J. Vassileva, and S. Isotani, “Automating gamification personalization to the user and beyond,” *IEEE Transactions on Learning Technologies*, vol. 15, no. 2, pp. 199–212, 2022. 5.2.1

[387] A. Ayaz, O. Ozyurt, W. M. Al-Rahmi, S. A. Salloum, A. Shutaleva, F. Alblehai, and M. Habes, “Exploring gamification research trends using topic modeling,” *IEEE Access*, vol. 11, pp. 119676–119692, 2023. 5.2.1

[388] T. Bohné, I. Heine, F. Mueller, P.-D. J. Zuercher, and V. M. Eger, “Gamification intensity in web-based virtual training environments and its effect on learning,” *IEEE Transactions on Learning Technologies*, vol. 16, no. 5, pp. 603–618, 2022. 5.2.1

[389] S. Dreimane, “Gamification for education: Review of current publications,” *Didactics of smart pedagogy: Smart pedagogy for technology enhanced learning*, pp. 453–464, 2019. 5.2.1

[390] R. F. Mulcahy, R. Russell-Bennett, N. Zainuddin, and K.-A. Kuhn, “Designing gamified transformative and social marketing services: An investigation of serious m-games,” *Journal of Service Theory and Practice*, vol. 28, no. 1, pp. 26–51, 2018. 5.2.1

[391] A. Pakinee and K. Puritat, “Designing a gamified e-learning environment for teaching undergraduate erp course based on big five personality traits,” *Education and Information Technologies*, vol. 26, no. 4, pp. 4049–4067, 2021. 5.2.1

[392] R. Nandhini Abirami, P. Durai Raj Vincent, K. Srinivasan, U. Tariq, and C.-Y. Chang, "Deep cnn and deep gan in computational visual perception-driven image analysis," *Complexity*, vol. 2021, no. 1, p. 5541134, 2021. 5.2.2

[393] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, vol. 27, 2014. 5.2.2

[394] K. Pandey, A. Mukherjee, P. Rai, and A. Kumar, "Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents," *arXiv preprint arXiv:2201.00308*, 2022. 5.2.2

[395] J. Wang and Z. Lu, "A controllable ipv6 address generation based on conditional variational autoencoder," in *Proceedings of the 2023 2nd International Conference on Networks, Communications and Information Technology*, pp. 78–81, 2023. 5.2.2

[396] Y. Xu, "Evolution and future directions of artificial intelligence generated content (aigc): A comprehensive review," *Applied and Computational Engineering*, vol. 95, pp. 1–13, 2024. 5.2.2

[397] M. Joanis, A. Lodi, and I. Sadoune, "On deep generative modeling in economics: An application with public procurement data," *Available at SSRN 4193922*, 2022. 5.2.2

[398] M. Wang, J. Cao, X. Yu, and Z. Nie, "A data-to-text generation model with deduplicated content planning," in *CCF Conference on Big Data*, pp. 92–103, Springer, 2022. 5.2.2

[399] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017. 5.2.2

[400] H. Byeon, "Advances in value-based, policy-based, and deep learning-based reinforcement learning," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, 2023. 5.2.2

[401] M. Al-Emran, "Hierarchical reinforcement learning: a survey," *International journal of computing and digital systems*, vol. 4, no. 02, 2015. 5.2.2

[402] E. Kolemen, J. Seo, R. Conlin, A. Rothstein, S. Kim, J. Abbate, K. Erickson, J. Wai, R. Shousha, and A. Jalalvand, "Avoiding tokamak tearing instability with artificial intelligence," 2023. 5.2.2

[403] J. Zhang and Q. Xue, "Actor–critic-based decision-making method for the artificial intelligence commander in tactical wargames," *The Journal of Defense Modeling and Simulation*, vol. 19, no. 3, pp. 467–480, 2022. 5.2.2

[404] S. Manandhar and B. Banerjee, "Reinforcement actor-critic learning as a rehearsal in microrts," *The Knowledge Engineering Review*, vol. 39, p. e6, 2024. 5.2.2

[405] N. Usunier, G. Synnaeve, Z. Lin, and S. Chintala, "Episodic exploration for deep deterministic policies: An application to starcraft micromanagement tasks," *arXiv preprint arXiv:1609.02993*, 2016. 5.2.2

[406] D. Lee, H. Tang, J. Zhang, H. Xu, T. Darrell, and P. Abbeel, "Modular architecture for starcraft ii with deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 14, pp. 187–193, 2018. 5.2.2

[407] W. Liang, C. Yu, B. Whiteaker, I. Huh, H. Shao, and Y. Liang, "Alphazero gomoku," *arXiv preprint arXiv:2309.01294*, 2023. 5.2.2

[408] T. Xu, T. Chen, C. Gao, M. Song, Y. Wang, and H. Yuan, "Distributed flexible resource regulation strategy for residential communities based on deep reinforcement learning," *IET Generation, Transmission & Distribution*, vol. 18, no. 21, pp. 3378–3391, 2024. 5.2.2

[409] S. Sarin, S. K. Singh, S. Kumar, S. Goyal, B. B. Gupta, W. Alhalabi, and V. Arya, "Unleashing the power of multi-agent reinforcement learning for algorithmic trading in the digital financial frontier and enterprise information systems.," *Computers, Materials & Continua*, vol. 80, no. 2, 2024. 5.2.2

[410] M. Csikzentimihalyi, "Beyond boredom and anxiety: Experiencing flow in work and play," *San Francisco/Washington/London*, 1975. 5.3.5

[411] P. Sweetser and P. Wyeth, "Gameflow: a model for evaluating player enjoyment in games," *Computers in Entertainment (CIE)*, vol. 3, no. 3, pp. 3–3, 2005. 5.3.5

[412] L. E. Nacke and C. A. Lindley, "Affective ludology, flow and immersion in a first-person shooter: Measurement of player experience," *arXiv preprint arXiv:1004.0248*, 2010. 5.3.5

[413] N. Case, "The evolution of trust," 2017. 5.4.1