Distilling Gaming Strategy through Explainability in Tetris

Eleftheria Lagiokapa*
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, 57001, Greece
elagio@iti.gr

Georgios Loupas*
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, 57001, Greece
loupgeor@iti.gr

Makrina Viola Kosti
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, 57001, Greece
mkosti@iti.gr

Nefeli Valeria Georgakopoulou
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, 57001, Greece
nefeli.valeria@iti.gr

Sotiris Diplaris
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, 57001, Greece
diplaris@iti.gr

Stefanos Vrochidis
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, 57001, Greece
stefanos@iti.gr

Abstract

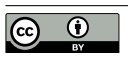
As artificial intelligence (AI) systems become increasingly integrated into game design, the demand for transparent and adaptive decision-making grows. While Explainable AI (XAI) has illuminated the internal reasoning of AI agents, most explanation-based training methods have traditionally prioritized alignment with a teacher model over the exploration of strategic diversity. In this paper, we introduce a novel framework that leverages explanation-based knowledge distillation to modulate agents' internal reasoning, yielding both convergent and divergent behavioral strategies. To demonstrate this approach, we conducted experiments in a Tetris environment comparing baseline agents trained with standard reinforcement learning to agents whose training was modified by incorporating explainability losses. Our dynamic framework integrates a feedback mechanism that adjusts the influence of the explainability term based on performance and strategic utility. This work demonstrates the potential of employing explainability not only as an interpretative tool but also as a means to actively diversify and refine strategies in complex, dynamic environments.

Keywords

Explainable AI (XAI), Reinforcement Learning (RL), Knowledge Distillation (KD), LIME, Feature Attribution, Emergent Behaviors, Game AI Agents, Tetris

How to cite this paper:
Eleftheria Lagiokapa, Georgios Loupas, Makrina Viola Kosti, Nefeli Valeria Georgakopoulou, Sotiris Diplaris, and Stefanos Vrochidis. 2025. Distilling Gaming Strategy through Explainability in Tetris. In *Proceedings of ACM IMX Workshops, June 3 - 6, 2025, Niterói, Brazil*. SBC, Porto Alegre/RS, Brazil, 8 pages. https://doi.org/10.5753/imxw.2025.4388

*Both authors contributed equally to this research.



ACM IMX Workshops, June 3 - 6, 2025, Niterói, Brazil

https://doi.org/10.5753/imxw.2025.4388

1 Introduction

Artificial intelligence (AI) continues to transform the landscape of digital games, powering increasingly sophisticated non-player character (NPC) behavior, adaptive systems, and procedural content generation. As these systems grow more autonomous and complex, the need for transparency and strategic adaptability becomes critical—not only to maintain trust but also to support richer design and debugging workflows. Explainable AI (XAI) has played a key role in meeting this demand, offering post-hoc insights into model behavior through feature attribution and surrogate modeling. However, the integration of XAI into training remains focused largely on interpretability through alignment, rather than diversification through controlled divergence.

In this paper, we present a novel training framework that reimagines the role of XAI in reinforcement learning (RL)-based game agents. Rather than using explanations to enforce similarity between teacher and student agents—as is common in explanation-based knowledge distillation (KD)—we introduce an explanation divergence loss that penalizes overlapping feature attributions. This approach encourages student agents to develop alternative strategies while still benefiting from the teacher's high-level guidance. To ensure this divergence is both meaningful and performant, we embed the loss within a dynamic feedback loop that adjusts its weight over time based on strategic utility. In addition, we explore a complementary training paradigm that leverages the explainability loss to promote convergence toward the teacher's balanced, robust strategy, while still allowing the agents to incorporate meaningful strategic nuances.

We evaluate our framework in the domain of Tetris, a game well-suited to analyzing structural variation and long-term planning. While full experimental results are ongoing, early trials indicate that our approach can lead to emergent behaviors not seen in the teacher policy, without sacrificing performance. These preliminary outcomes suggest that encouraging divergent reasoning—or, alternatively, fostering strategic convergence through explainability—may be a viable path toward designing more adaptable and robust agents, particularly in game environments where creative strategy and dynamic balance are central.

2 Background and Related Work

Artificial Intelligence (AI) has emerged as a transformative force across various fields, with gaming standing out as a domain where its impact is both profound and diverse. Within the gaming industry, AI-driven innovations are redefining how non-player characters (NPCs) behave, revolutionizing procedural content generation, and enabling adaptive systems that create dynamic, personalized player experiences. Central to these advancements are Reinforcement Learning (RL) and Explainable Artificial Intelligence (XAI), which not only enhance decision-making capabilities but also promote transparency and trust in AI systems. This section reviews related efforts in explainability, reinforcement learning, and game balancing, with a focus on how these inform our novel approach: a knowledge distillation framework that encourages strategy divergence through explanation-based loss functions and dynamic feedback.

2.1 Knowledge Distillation and Explanation Transfer

2.1.1 Classical Knowledge Distillation. Knowledge distillation (KD) emerged as a key framework to compress the knowledge of a complex model (teacher) into a smaller, more efficient model (student). The foundational work by Hinton et al. (2015) [8] introduced soft-label matching, where the teacher's output distributions (soft targets) are used to train the student. This soft-label approach transfers richer information than one-hot labels, encapsulating inter-class relationships that enable the student to mimic the teacher's predictive capability effectively. The method has been instrumental in reducing computational costs while maintaining high performance, making it particularly valuable for deploying models in resource-constrained environments.

2.1.2 Extensions to Explanation-Based KD. Although classical KD [8] excels at compressing predictive knowledge, it does not inherently transfer the internal reasoning of the teacher. To bridge this gap, explanation-based KD methods were developed. These approaches, such as DiXtill [6], Exp-KD [13], and XDistillation [1], incorporate explainability into the distillation process by aligning internal representations or feature attributions between teacher and student:

- **DiXtill** [6] leverages XAI techniques like SHAP (SHapley Additive Explanations) to extract interpretable knowledge from teacher models and transfer it to compact student architectures. Its utility lies in low-resource applications, such as deploying large language models (LLMs) or vision models on edge devices, where maintaining both interpretability and computational efficiency is critical. By embedding explainable features into the student, DiXtill enhances trust and usability in constrained environments.
- **Exp-KD** [13] extends the KD framework by enforcing alignment in feature importance scores, ensuring that the student not only mirrors the teacher's outputs but also replicates its attribution maps. This approach is especially relevant for high-stakes applications like medical image analysis, where interpretability is paramount.
- **XDistillation** [1], on the other hand, utilizes convolutional autoencoders to compress the teacher's explanations and align them with the student's representations. Such models have been applied to tasks like chest X-ray classification, demonstrating improved model trustworthiness and diagnostic consistency.

2.1.3 Divergence Through Explanation Losses. : While conventional explanation-based knowledge distillation (KD) methods focus on aligning teacher and student reasoning, our approach extends this paradigm by leveraging explanations as a versatile training signal. In our framework, we incorporate two complementary strategies. First, an explanation divergence loss penalizes similarity between the local feature attributions of the teacher and the student, thereby encouraging the development of alternative, risk-prone strategies. Simultaneously, we explore a convergence approach in which an additive explanation loss rewards alignment of internal decision-making with the teacher. This dual-objective setting represents a novel application of explanation-based KD in the gaming domain, where both strategic diversity and adherence to a well-calibrated baseline can be valuable depending on the context.

2.2 Explainable AI in Gaming

Explainable AI (XAI) techniques have gained traction in gaming as tools for improving transparency, debugging, and strategy discovery in game AI systems. By providing insights into how AI agents make decisions, XAI fosters trust and enables developers to refine AI behaviors dynamically.

2.2.1 Local Explanation Tools. LIME and SHAP

This section focuses specifically on local explanation methods that have been successfully applied in gaming environments. To date, our review indicates that LIME and SHAP are the primary methods leveraged in this field due to their model-agnostic nature and ease of implementation in game AI systems.

LIME (Local Interpretable Model-Agnostic Explanations), introduced by Ribeiro et al. (2016) [11], provides interpretable and faithful explanations for individual model predictions by approximating the model locally with simpler surrogate models. Initially applied to tasks like text classification and image recognition, LIME demonstrated its utility in debugging, trust-building, and identifying critical features in machine learning workflows. By framing the problem as a submodular optimization task, LIME can effectively summarize model behavior for both simulated and real-world scenarios.

SHAP (SHapley Additive Explanations) [10], presented by Lundberg and Lee (2017), unified multiple feature attribution methods under a single framework of additive feature importance. Its theoretical foundations in Shapley values from cooperative game theory ensure that SHAP assigns consistent, intuitive importance scores to each feature. SHAP has been widely adopted in areas such as medical diagnosis and financial decision-making, where the combination of interpretability and accuracy is essential.

2.2.2 Applications in Gaming. Applications of LIME and SHAP have expanded beyond their initial uses in decision-making and vision tasks, proving highly valuable in gaming contexts. For instance, LIME has been instrumental in detecting problematic regions in procedurally generated game levels, allowing faster and more targeted

repairs [4]. These contributions highlight the growing importance of XAI in optimizing gaming systems and fostering strategic adaptability.

2.3 Reinforcement Learning and Strategy Discovery in Games

The use of RL and XAI has enabled significant advancements in NPC adaptability and realism. Traditional NPC behavior often relied on Finite State Machines (FSMs), which were rigid and predictable. Recent efforts, such as Dynamic NPC AI Using Reinforcement Learning [12], integrate RL with XAI to create NPCs capable of real-time decision-making and strategic adaptation. By analyzing NPC decisions with tools like SHAP, developers can refine their behaviors to reduce predictability and increase engagement, transforming NPCs into intelligent, interactive entities.

Beyond NPC control, RL agents have been employed in automated playtesting frameworks, as seen in Exploring Gameplay With AI Agents [7]. By simulating thousands of gameplay scenarios, these agents can identify imbalances, evaluate rewards, and expose design flaws. This data-driven approach has informed adjustments that improve player experiences and enhance the strategic depth of game systems.

Building on these strategy-discovery paradigms, we extend RL-based exploration through a novel explanation-driven mechanism: our framework builds on these RL foundations by introducing an explanation divergence loss that promotes reasoning divergence in student agents, and also an explanation convergence loss that encourages student agents to align their reasoning with proven, optimal strategies.

For our testbed, we have chosen Tetris—not solely for its widespread recognition, but for its unique combination of simplicity in gameplay and computational complexity. Tetris is fundamentally straightforward: players must arrange falling tetrominoes, aiming to complete horizontal lines while managing an ever-increasing pace. However, beneath its simple rules lies a highly dynamic and stochastic environment where the sequence of pieces is random, and decision making requires balancing immediate actions with long-term planning. This dual demand is further underscored by the fact that determining the optimal sequence of moves in Tetris is NP-complete, marking it as a computationally challenging task. Such features make Tetris an excellent platform for our experiments; it provides a clear, well-understood framework while also posing complex decision-making challenges that are ideal for evaluating the emergence of novel and strategically diverse behaviors. By applying our explanation-driven divergence loss to Tetris, our agents discover strategies that emphasize different aspects of gameplay, such as long-term structural flexibility over immediate reward, thereby demonstrating the method's potential for uncovering emergent strategic behaviors that are not encoded in the teacher's logic.

2.4 Dynamic Feedback Loops for Balancing and Training

Dynamic feedback loops have been widely adopted to refine AI behavior and balance complex systems in real time. Feedback mechanisms such as curriculum learning (Bengio et al., [2009]) [5] and reward shaping (Hu et al., [2009]) [9] have significantly informed the refinement of reinforcement learning (RL) training objectives, aligning agent behavior with evolving task demands. Curriculum learning formalizes the idea of presenting training examples in a meaningful order, progressing from simpler to more complex concepts. This strategy not only improves the generalization capabilities of RL models but also accelerates convergence to optimal solutions by smoothing the learning trajectory in non-convex optimization landscapes [5]. Similarly, reward shaping introduces domain knowledge into RL systems by providing additional reward signals to guide agent behavior. Adaptive shaping approaches have demonstrated the ability to transform suboptimal reward functions into beneficial ones, ensuring agents effectively balance exploration and exploitation, even in sparse-reward environments (Hu et al., [2009]) [9].

In gaming contexts, dynamic feedback loops have proven invaluable for refining AI-driven systems. For instance, SHAP-based feedback mechanisms have enabled real-time adjustments to gameplay mechanics in real-time strategy (RTS) games, allowing developers to address imbalances and create more equitable and engaging player experiences. Furthermore, feedback loops have been employed to adapt RL training objectives dynamically, ensuring that agents remain aligned with performance targets while maintaining interpretability. Such techniques have been particularly effective in aligning agent behavior with broader gameplay objectives, fostering fairness and engagement.

Our framework incorporates a similar feedback mechanism to refine both explanation divergence and convergence during training. By iteratively adjusting the respective losses, the mechanism ensures that each student agent's reasoning evolves meaningfully while maintaining competitive performance. This feedback-driven approach dynamically steers divergent strategies to uncover novel tactics and convergent strategies to reinforce proven behaviors, thereby aligning both exploration and refinement with the broader objectives of adaptability and robustness in gaming AI. Through this integration, we aim to enable agents to balance the discovery of innovative strategies with adherence to the principles of efficient and interpretable reinforcement learning.

3 Methodology

The central hypothesis under investigation is whether differences in local explainability between models can lead to the adoption of distinct strategic behaviors over the course of a game. In other words, we examine if variations in the locally interpretable decision-making processes translate into divergent gameplay strategies. To assess this hypothesis, we trained four agents on Tetris.

3.1 Agent training

Two baseline agents, AgentS and AgentS_T, were trained following a standard reinforcement learning (RL) paradigm using a Deep Q-Network (DQN). In this framework, the agents interact with the Tetris environment by sampling state–action pairs, computing a performance loss (typically the mean squared error between predicted Q-values and target values), and updating their parameters accordingly. These agents serve as controls, providing a robust reference for standard RL behavior.

To investigate the impact of local explainability on strategic decision-making, we also trained AgentX in two variations by modifying the total loss function with a local explainability term. The local explainability is quantified by computing feature importance values for game states using techniques such as LIME, which indicate the sensitivity of the agents' predictions to individual features.

The first variant, denoted as $\text{AgentX}_{\text{div}}$, is trained by adding the inverse explainability loss to the conventional RL loss. This approach is designed to encourage divergence in the internal decision-making process, effectively pushing $\text{AgentX}_{\text{div}}$ away from the baseline strategy by forcing a discrepancy in the feature importance profiles when compared to the teacher or reference agent.

Conversely, the second variant, $\text{AgentX}_{\text{cov}}$, is trained by adding the explainability loss to the RL loss. This technique promotes convergence of $\text{AgentX}_{\text{cov}}$'s decision-making process with that of the teacher, thereby aligning its internal feature attributions closely with those observed in the standard RL agents.

For each training batch, the feature importance for every state was computed independently for both the baseline agents and the AgentX variants. The mean squared error (MSE) between the feature importance values of the teacher model (or the chosen baseline) and that of the AgentX variant was then calculated and incorporated into the total loss.

By analyzing the resulting behavior and performance metrics, we aim to determine whether enforcing divergence or convergence in local model interpretability drives the adoption of alternative strategic approaches during gameplay.

3.2 Explanation Divergence Loss Function

The explainability loss is defined as

$$L_{\text{EXPL}} = \left\| \text{EXPL}_{\text{AgentX}} - \text{EXPL}_{\text{AgentS}_\text{T}}) \right\|_2^2 .$$

In this term, $\text{EXPL}_{\text{AgentX}}$ and $\text{EXPL}_{\text{AgentS}_\text{T}}$ represent the explainability outputs—typically in the form of feature attribution vectors—from two different agents. This loss measures the squared L_2-norm of the difference between these attribution vectors, thereby quantifying how similar or dissimilar their internal decision processes are. By minimizing L_{EXPL} one can enforce that the explanation of decisions provided by the two agents becomes more alike. Conversely, if one wishes to encourage divergence in explanations, similar formulations can be incorporated into the overall loss in a manner that incentivizes distinct feature attributions.

To incorporate the explainability aspect into the overall training objective, two alternative formulations of the total loss can be defined. In the case where the goal is to promote divergence in the explanation signals—encouraging the agent to develop a distinct strategy from the reference—the total loss is defined as

$$L_{\text{total\_div}} = L_{\text{RL}} + \lambda_{\text{div}} \frac{1}{L_{\text{EXPL}} + \epsilon}$$

where L_{RL} is the conventional reinforcement learning loss capturing the prediction error or TD error associated with the value function or policy; L_{EXPL} denotes the explainability loss that measures the similarity between the agent's internal representations and those of the reference; λ_{div} is a regularization parameter that controls the influence of the explainability loss on the overall objective; and ϵ is a small constant (e.g., 10^{-6}) added for numerical stability and to prevent division by zero.

This formulation works by penalizing the agent for being too similar to the reference. When the agent's explanation loss L_{EXPL} is very low—indicating that its internal representations are closely aligned with those of the reference—the inverse term $\frac{1}{L_{\text{EXPL}}+\epsilon}$ becomes very large, thereby significantly increasing the total loss. This higher loss discourages convergence toward the reference behavior and pushes the agent to develop divergent, novel strategies. Conversely, when L_{EXPL} is higher (indicating that the agent's representations differ from the reference), the penalty is reduced, leading to a lower overall loss. Additionally, this formulation bounds the overall loss from below.

Alternatively, in scenarios where one wishes the explanations between the agents to converge—ensuring that the agent's internal decision-making process is aligned with that of a reference— the total loss is defined as

$$L_{\text{total\_cov}} = L_{\text{RL}} + \lambda_{\text{cov}} L_{\text{EXPL}}$$

In this formulation, the conventional L_{RL} loss is augmented by the explainability loss and λ_{cov} is a regularization parameter that controls the influence of the explainability loss L_{EXPL} on the overall loss $L_{\text{total\_cov}}$. Because L_{EXPL} is non-negative and increases as the difference between $\text{EXPL}_{\text{AgentX}}$ and $\text{EXPL}_{\text{AgentS}_\text{T}}$ grows, minimizing $L_{\text{total\_cov}}$ enforces not only high performance as measured by L_{RL}, but also drives the model toward reducing the disparity between the agents' explanation signals. This encourages the agent to develop internal representations that are similar to those of the reference, fostering convergence in feature attributions and a more interpretable decision-making process.

Thus, by choosing either a divergent formulation ($L_{\text{total\_div}}$) or a convergent formulation ($L_{\text{total\_cov}}$), one can steer the training process to either accentuate differences or promote alignment in the learned explainability metrics, providing a dual-objective loss function that balances performance with explicit control over interpretability.

4 Experimental Setup

Our experimental setup evaluates both task performance and the interpretability of internal decision processes in a challenging reinforcement learning (RL) setting. We selected the game Tetris which is an NP-complete game [3] with high-dimensional state space. Inspired by Ashry's 2020 work [2], which demonstrated that deep Q-networks applied on high-level state spaces (instead of raw board pixels) can significantly reduce state complexity and speed up learning, our environment uses a state representation derived from key game-level metrics. The implementation of Tetris is based on an open source code[1] adjusted to our experiments.

A Xavier uniform weight initialization is applied and the training was performed using the Adam optimizer with a learning rate of 0.001 for several epochs until the agent reaches a satisfactory score, using a batch size of 512. The learning process employed a decay-driven exploration strategy, starting with an initial epsilon of 1.0 and gradually reducing it to a final value of 0.001 over 2000 epochs

[1] https://github.com/vietnh1009/Tetris-deep-Q-learning-pytorch.git

to balance exploration and exploitation. The Q-learning update was governed by a discount factor of 0.99. For our experiments, we used a NVIDIA GeForce RTX 3090 GPU.

4.1 Tetris Game Environment

Board and State Representation: The game board is modeled as a 10×20 grid. Instead of using raw pixel data, we summarize the board using a four-dimensional feature vector capturing:

- Cleared Lines: Number of rows completely filled and removed.
- Holes: Count of empty cells under blocks.
- Bumpiness: Variation in the heights of columns.
- Aggregate Height: Sum of the heights of all columns.

This abstraction, as proposed in Ashry's thesis, reduces the state space significantly, thereby speeding up the learning process while still capturing the critical aspects of the game dynamics.

Reward Structure and Grouped Actions: Rewards are designed to reflect the score changes based on the number of cleared lines—with bonuses for clearing multiple lines simultaneously.

4.2 Model Architecture: Deep Q-Network (DQN)-Based Model

Both AgentS, as AgentS_T and the explanation-infused AgentX's share the following network backbone:

- Input: A 4-dimensional feature vector (as described above) representing the board state.
- Hidden Layers: Two fully connected layers: Layer 1: Maps the 4-dimensional input to 64 nodes with ReLU activation. Layer 2: Processes the 64-dimensional representation further into another 64-dimension vector (again with ReLU).
- Output Layer: A final linear layer outputs a single scalar representing the Q-value estimate for the selected action.

4.3 Training Protocol

Training is divided into two stages: first, we train AgentS and AgentS_T (the teacher) using a standard DQN approach; then, we transfer knowledge to AgentX's (the explanation-infused students) using a composite loss that includes both performance and explanation-based terms computed with on-the-fly LIME attributions.

4.4 Experimental Design and Evaluation Metrics

The primary objective of the experiments was to analyze how differences in local explainability influenced the strategic behavior of agents during gameplay. To achieve this, two deep reinforcement learning agents (AgentS and AgentS_T) were trained under identical environmental conditions using a standard RL paradigm, serving as baseline models. In addition, two modified agents—AgentX_div and AgentX_cov—were trained with an explainability-driven term integrated into their loss functions, aimed at promoting controlled reasoning divergence and convergence relative to the baseline agents. AgentS, in particular, functions as a critical control; by evaluating its strategy—developed solely from conventional RL loss—we can assess whether a second agent trained only with RL loss adopts a different strategy from the teacher and serves as a reliable benchmark for determining if AgentX_cov indeed converges toward the teacher's balanced approach.

In this setup, the loss function for AgentX_div was augmented by adding the inverse of the mean squared error (MSE) between its feature importance scores and those of the teacher model (AgentS_T), computed using on-the-fly LIME (Local Interpretable Model-Agnostic Explanations) attributions for each state during training. This encouraged AgentX_div to prioritize decisions differently from the teacher, fostering a divergent strategic approach. Conversely, the loss function for AgentX_cov incorporated an additional term that added the MSE between feature importance values, incentivizing the agent to align its internal decision-making with that of the teacher, resulting in a convergent strategy.

The hypothesis driving this experiment is that variations in local interpretability can directly shape the way reinforcement learning agents evaluate board states and select optimal moves, influencing the emergence of distinct or aligned long-term gameplay patterns. By training these four agents under these conditions, we aim to systematically investigate the dual impacts of divergence and convergence on strategic behavior while maintaining high levels of performance.

All agents were trained until they consistently achieved a high Tetris score—specifically, about 500,000 points—ensuring each model reached a robust level of performance before evaluation. For these experiments, the regularization parameter was fixed at $\lambda_{\text{cov}} = 1$, $\lambda_{\text{div}} = 1$ across all runs. While this setting provided valuable initial insights into the impact of our explanation-based modifications, further investigation is needed to rigorously examine how varying the regularization constant influences the training dynamics and the emergent strategies.

To evaluate the capability of distilling strategies from agents with differing learning objectives, we conducted two experiments under controlled conditions. In the first experiment, we introduced variability by starting each game from a pre-selected random board state. We collected a diverse set of 3000 board states from a medium-strength independent agent that was not part of our primary experimental setup. This agent played multiple games, with board states randomly sampled at various points throughout its gameplay. The collected states were then shuffled to eliminate any ordering bias before being used as initial conditions for all four agents. Each game began with all agents starting from the same randomly selected board state and performing 200 moves, using identical Tetris piece sequences.

In the second experiment, the agents played 3000 fresh games, with each game consisting of 200 moves and using identical Tetris piece sequences. This standardized setup ensured that every game began from a uniform baseline, enabling us to capture sustained performance and strategic adaptation over numerous independent trials.

To systematically compare the four agents, we measured the following key gameplay metrics:

- Mean lines cleared
- Multiline Clear Frequency (MLCF)
- Mean number of holes
- Mean bumpiness

- Mean board height
- Mean score
- Number of moves per game
- Final score

For each game played by the agents, the per-move metrics were first computed by averaging the values recorded at each move (e.g., the number of lines cleared per move), while the total metrics (moves and score) were recorded in aggregate. In particular, the MLCF metric quantifies the frequency at which an agent clears more than one line in a single move. A higher MLCF indicates that the agent is more often executing high-value moves that remove multiple lines simultaneously, which can be indicative of a more aggressive or risk-tolerant strategy. Subsequently, for every pairwise combination among the four strategies, we computed the per-game differences in each key metric. In doing so, we compared the modified agents ($\text{AgentX}_{\text{cov}}$ and $\text{AgentX}_{\text{div}}$) against the standard RL baselines (AgentS and AgentS_{T}) by generating distributions of the per-game differences for each feature. These differences were then averaged over all games, and paired t-tests were performed on the resulting distributions to assess statistical significance. In doing so, we sought to determine whether the explainability modifications drive agents either towards convergence with the teacher's strategy or towards a divergent, risk-tolerant approach.

Significant differences in features such as mean board height, bumpiness, and the number of holes indicate meaningful variations in stacking and placement strategies. For instance, when a set of gameplay metrics consistently shows substantially higher board heights, increased bumpiness, and elevated hole counts—in conjunction with a higher MLCF, which reflects a greater frequency of multi-line clears—this pattern can be interpreted as evidence of a high-risk, high-reward strategy aimed at rapid, explosive scoring. Conversely, if these metrics exhibit only modest deviations and the board structure remains closer to a stable baseline, this suggests a more controlled and balanced approach. Additionally, variations in the total number of moves and final scores help to further elucidate the strategic trade-offs being made, revealing whether a strategy prioritizes aggressive, potentially destabilizing maneuvers for the sake of scoring or favors a more sustainable, stability-oriented design. Together, these metrics provide a robust framework for discerning divergent strategic behaviors in gameplay.

5 Results and Discussion

In this section, we present the experimental results that quantify the strategic performance and board characteristics of four reinforcement learning agents trained to play Tetris. The primary objective of these experiments was to determine whether the incorporation of explainability loss leads to measurable differences in gameplay strategy and overall performance. The results are shown in Tables 1, 2.

In both experimental setups—all games lasting 200 moves with identical Tetris piece sequences—the paired t-test statistics (with p values indicating significance when $p<0.05$, and in our case mostly $p=0.000$) reveal robust differences among the agents. These metrics include Mean Lines Cleared, Multiline Clear Frequency (MLCF), Mean Holes, Mean Bumpiness, Mean Height, Mean Score, Total Score and Total Moves. Together, they help us infer that subtle modifications to the loss function—in terms of adding versus subtracting an explainability loss—systematically steer the agents' strategic behaviors.

In the experiment where games begin from an initial random board state (Table 1) the teacher agent (AgentS_{T}) consistently demonstrates a conservative playstyle. For instance, its Mean Lines Cleared per Move is significantly higher than that of the baseline AgentS ($t=7.89$, $p=0.000$) and even more pronounced compared to the divergent variant ($t=11.65$, $p=0.000$). In addition, its Multiline Clear Frequency (MLCF) is markedly lower than that of the other agent—evidenced by t-values of -56.39 ($p=0.000$) when compared to the divergent agent, and -32.90 ($p=0.000$) relative to the convergent agent—which indicates that the teacher avoids rapid, multi-line clearances that are typically associated with riskier tactics. Furthermore, comparisons in Mean Bumpiness show that AgentS_{T} builds notably smoother boards (with $t=-27.57$ and -37.92, both $p=0.000$, when compared to AgentS and $\text{AgentX}_{\text{cov}}$, respectively), and its lower Mean Height ($t=-4.41$ vs. AgentS and $t=-27.75$ vs. $\text{AgentX}_{\text{div}}$, both $p=0.000$) confirms that it maintains lower stacking levels, reducing the risk of premature game termination. While there is some ambiguity in the Mean Holes metric—for example, the comparison between AgentS_{T} and AgentS yields a positive t-value ($t=17.82$, $p=0.000$), suggesting a higher hole count for the teacher relative to AgentS—the overall picture provided by the lower bumpiness, lower height, and the fact that AgentS_{T} accumulates more Total Moves ($t=9.27$, $p=0.000$ vs. AgentS) supports the conclusion that this agent favors long-term board stability over explosive, high-risk maneuvers. Its scoring metrics (Mean and Total Score) tend to be lower compared to the aggressive strategies, reinforcing the idea that the teacher's conservative approach is geared toward sustainable play rather than immediate rewards.

In contrast, the divergent agent ($\text{AgentX}_{\text{div}}$), trained by subtracting the explainability loss, exhibits a markedly aggressive, high-risk, high-reward strategy. Its MLCF is significantly elevated relative to AgentS_{T}, as shown by a t-value of -56.39 ($p=0.000$) when comparing the two, meaning that $\text{AgentX}_{\text{div}}$ clears multiple lines in a single move far more frequently—a key indicator of an aggressive style. Moreover, the Mean Height comparison ($t=-27.75$, $p=0.000$ for AgentS_{T} vs. $\text{AgentX}_{\text{div}}$) reveals that $\text{AgentX}_{\text{div}}$ consistently builds much taller stacks, which is a sign of risky board management since high stacks are more prone to collapsing in Tetris. Although its Mean Lines Cleared figures are lower than the teacher's (and even than those of AgentS in some cases), the combination of a high MLCF with significantly greater board height and an increased number of holes (e.g., AgentS vs. $\text{AgentX}_{\text{div}}$ shows $t=-26.25$, $p=0.000$ for Mean Holes) indicates that $\text{AgentX}_{\text{div}}$ is willing to sacrifice board stability for rapid, high-impact clearances. Notably, this agent also executes a lower Total Moves count ($t=8.35$, $p=0.000$ when compared to AgentS_{T}), implying that its aggressive tactics often lead to an earlier end of the game.

The convergent agent ($\text{AgentX}_{\text{cov}}$), trained by adding the explainability loss to the RL loss, shows behaviors that place it between the teacher and the divergent agent. Its MLCF is significantly lower than that of $\text{AgentX}_{\text{div}}$ ($t=27.94$, $p=0.000$ for $\text{AgentX}_{\text{div}}$ vs. $\text{AgentX}_{\text{cov}}$), suggesting that it does not pursue multi-line clearances as aggressively. In terms of Mean Height, the difference between

Table 1: Paired t-test Statistics for Tetris Metrics (Initial Random Board States)

Metric	AgentS$_T$ vs AgentS	AgentS$_T$ vs AgentX$_{div}$	AgentS$_T$ vs AgentX$_{cov}$	AgentS vs AgentX$_{div}$	AgentS vs AgentX$_{cov}$	AgentX$_{div}$ vs AgentX$_{cov}$
Mean Lines Cleared	$t = 7.89,\ p = 0.000$	$\mathbf{t = 11.65,\ p = 0.000}$	$t = 4.81,\ p = 0.000$	$t = 4.40,\ p = 0.000$	$t = -3.82,\ p = 0.000$	$t = -7.83,\ p = 0.000$
MLCF	$\mathbf{t = -58.66,\ p = 0.000}$	$t = -56.39,\ p = 0.000$	$t = -32.90,\ p = 0.000$	$t = 2.26,\ p = 0.024$	$t = 31.24,\ p = 0.000$	$t = 27.94,\ p = 0.000$
Mean Holes	$t = 17.82,\ p = 0.000$	$t = -6.42,\ p = 0.000$	$t = 13.59,\ p = 0.000$	$\mathbf{t = -26.25,\ p = 0.000}$	$t = -6.38,\ p = 0.000$	$t = 18.83,\ p = 0.000$
Mean Bumpiness	$t = -27.57,\ p = 0.000$	$t = -8.30,\ p = 0.000$	$\mathbf{t = -37.92,\ p = 0.000}$	$t = 19.73,\ p = 0.000$	$t = -2.83,\ p = 0.005$	$t = -24.19,\ p = 0.000$
Mean Height	$t = -4.41,\ p = 0.000$	$\mathbf{t = -27.75,\ p = 0.000}$	$t = -2.30,\ p = 0.022$	$t = -26.88,\ p = 0.000$	$t = 2.64,\ p = 0.008$	$t = 27.58,\ p = 0.000$
Mean Score	$\mathbf{t = -42.03,\ p = 0.000}$	$t = -38.50,\ p = 0.000$	$t = -25.99,\ p = 0.000$	$t = 1.35,\ p = 0.177$	$t = 19.76,\ p = 0.000$	$t = 17.83,\ p = 0.000$
Total Score	$\mathbf{t = -17.91,\ p = 0.000}$	$t = -17.42,\ p = 0.000$	$t = -11.11,\ p = 0.000$	$t = -0.22,\ p = 0.822$	$t = 8.28,\ p = 0.000$	$t = 8.16,\ p = 0.000$
Total Moves	$\mathbf{t = 9.27,\ p = 0.000}$	$t = 8.35,\ p = 0.000$	$t = 6.03,\ p = 0.000$	$t = -0.59,\ p = 0.558$	$t = -4.36,\ p = 0.000$	$t = -3.17,\ p = 0.002$

Table 2: Paired t-test Statistics for Tetris Metrics (Initial Empty Board States)

Metric	AgentS$_T$ vs AgentS	AgentS$_T$ vs AgentX$_{div}$	AgentS$_T$ vs AgentX$_{cov}$	AgentS vs AgentX$_{div}$	AgentS vs AgentX$_{cov}$	AgentX$_{div}$ vs AgentX$_{cov}$
Mean Lines Cleared	$\mathbf{t = 9.25,\ p = 0.000}$	$t = 8.23,\ p = 0.000$	$t = 4.95,\ p = 0.000$	$t = -0.94,\ p = 0.347$	$t = -3.35,\ p = 0.001$	$t = -2.51,\ p = 0.012$
MLCF	$\mathbf{t = -64.58,\ p = 0.000}$	$t = -60.97,\ p = 0.000$	$t = -29.41,\ p = 0.000$	$t = 5.58,\ p = 0.000$	$t = 31.02,\ p = 0.000$	$t = 26.98,\ p = 0.000$
Mean Holes	$t = 25.48,\ p = 0.000$	$t = -3.19,\ p = 0.001$	$t = 13.38,\ p = 0.000$	$\mathbf{t = -28.65,\ p = 0.000}$	$t = -12.19,\ p = 0.000$	$t = 17.04,\ p = 0.000$
Mean Bumpiness	$t = -30.41,\ p = 0.000$	$t = -7.73,\ p = 0.000$	$\mathbf{t = -35.40,\ p = 0.000}$	$t = 23.13,\ p = 0.000$	$t = -6.27,\ p = 0.000$	$t = -30.21,\ p = 0.000$
Mean Height	$t = -5.54,\ p = 0.000$	$\mathbf{t = -39.91,\ p = 0.000}$	$t = -5.21,\ p = 0.000$	$t = -35.33,\ p = 0.000$	$t = 0.29,\ p = 0.775$	$t = 35.90,\ p = 0.000$
Mean Score	$t = -42.05,\ p = 0.000$	$\mathbf{t = -45.04,\ p = 0.000}$	$t = -21.62,\ p = 0.000$	$t = -3.16,\ p = 0.002$	$t = 18.08,\ p = 0.000$	$t = 21.25,\ p = 0.000$
Total Score	$t = -20.91,\ p = 0.000$	$\mathbf{t = -26.60,\ p = 0.000}$	$t = -12.00,\ p = 0.000$	$t = -4.38,\ p = 0.000$	$t = 8.85,\ p = 0.000$	$t = 13.58,\ p = 0.000$
Total Moves	$\mathbf{t = 10.04,\ p = 0.000}$	$t = 5.10,\ p = 0.000$	$t = 5.25,\ p = 0.000$	$t = -4.74,\ p = 0.000$	$t = -4.36,\ p = 0.000$	$t = 0.29,\ p = 0.776$

AgentS_T and AgentX_cov is modest (t=−2.30, p=0.022), indicating that its stacking is more in line with the teacher's controlled approach. Moreover, while its Mean Bumpiness remains higher than that of the teacher (as seen in t=−37.92, p=0.000 for AgentS_T vs. AgentX_cov), it is lower than that produced by the baseline AgentS in some comparisons. The scoring data further reinforce the convergent strategy: although its Mean and Total Scores are significantly lower than those of AgentX_div (with t-values such as 17.83 and 8.16, respectively, p=0.000), they exceed those of the teacher in certain cases, suggesting that AgentX_cov balances risk and reward by partially emulating the teacher's conservative style while still capturing some of the scoring potential observed in more aggressive strategies.

In the fresh games experiment (Table 2)—where each game starts from an empty board—the trends are corroborated by similar per-move metrics. The teacher continues to clear lines effectively (Average Lines Cleared per Move t=9.25, p=0.000 vs. AgentS) and maintains the lowest MLCF (t=−64.58, p=0.000 when compared to AgentS_T vs. AgentX_div), reinforcing its focus on safety and board longevity. Consistency in Mean Bumpiness and Mean Height per Move further validates that its board management remains stable, while the divergent agent again differentiates itself by exhibiting much higher stacking (t=−39.91, p=0.000 when comparing AgentS_T vs. AgentX_div) and increased multi-line clearances. In contrast, the convergent agent tends to produce boards and scores that are intermediate, thereby demonstrating a measured compromise between the teacher's risk-averse approach and the divergent agent's aggressive tactics.

Overall, the extremely low p-values (typically p=0.000, indicating a probability of less than 0.1% that the observed differences are due to chance) confirm that the variations in all these metrics are statistically significant. These detailed statistical findings lend support to our hypothesis that incorporating the explainability loss may indeed shape the agents' strategic behavior. In the divergent case, incorporating the inverse of the explainability loss into the RL loss encourages an agent to pursue rapid, multi-line clearances and aggressive stacking (as observed with AgentX_div), albeit at the expense of increased board irregularity and a tendency toward earlier game termination. Conversely, adding the explainability loss guides an agent toward a balanced, teacher-like strategy, characterized by controlled stacking, smoother boards, and prolonged play (as demonstrated by AgentX_cov). The teacher agent (AgentS_T), in its conservative performance, serves as a benchmark for sustained, risk-averse play. These strategic differentiations, as quantified by the t-test statistics across every key metric, affirm that nuanced adjustments in the training objective can systematically and predictably influence an agent's in-game strategy.

6 Conclusions and Future Work

In this work, we introduced an explanation divergence loss into our reinforcement learning framework and conducted preliminary experiments in Tetris. Our initial findings modestly indicate that incorporating an explainability-based loss may lead to subtle shifts in strategic behavior among agents. Our initial results reveal modest yet consistent differences in strategic behavior. The variations in metrics such as board height, bumpiness, number of holes, moves, and total score reveal a promising tendency for agents to adopt different strategies through slight modifications to the loss function. However, these early indications call for more extensive experimentation to fully understand the long-term impact and robustness of these effects across diverse game scenarios. In particular, the divergent agent tends to adopt a more aggressive, risk-prone strategy, while the convergent agent shows a more stable, balanced approach. These early findings support the notion that integrating explainability into the training process can nudge agents to explore alternative strategies, thereby enriching their overall gameplay dynamics.

The differences in board management and scoring metrics provide early indications that even a slight modification in the loss function can encourage agents to explore alternative strategies.

Looking ahead, more extensive and systematic experiments are needed to robustly validate these early observations. Future work will explore whether agents can eventually develop converging strategies that could be beneficial for player control. In addition, we plan to investigate the incorporation of alternative loss functions and perform detailed regularization tuning to optimize the balance between interpretability and performance. Additionally, we plan to investigate whether incorporating a richer set of features could foster greater diversity in strategies, and whether assigning additional importance to domain-specific features can further refine the agents' decision-making in targeted contexts. Specifically:

- **Expanding the Agent Pool:** Incorporating a greater number and diversity of agents will help determine if the observed effects are consistent and robust across different architectures and training scenarios.
- **Sequential and Extended Gameplay Studies:** Investigating how these strategic divergences evolve over longer sequences of play could provide additional insights into the long-term implications of the explainability loss.
- **Integration with Additional Interpretability Techniques:** Combining our approach with other interpretability measures, such as SHAP, may help further elucidate the relationship between feature attributions and emergent strategic behaviors.
- **Balancing Risk and Stability:** Fine-tuning the interplay between aggressive actions and game stability remains a promising direction. Future work might involve dynamically adjusting the loss weight to better manage this trade-off.

Nevertheless, using explainable AI (XAI) in a dynamic feedback loop for gaming agents is a novel and exciting prospect. This approach could pave the way for agents that not only learn distinct strategies but also adaptively contribute to adjusting the game's balance and difficulty—potentially leading to more engaging and robust player experiences.

7 Acknowledgments

This research was funded by the EC-funded research and innovation programme Horizon Europe i-Game under the grant agreement No. 101132449

References

[1] Raed Alharbi, Minh N. Vu, and My T. Thai. 2021. Learning Interpretation with Explainable Knowledge Distillation. arXiv:2111.06945 [cs.LG] https://arxiv.org/abs/2111.06945

[2] Mohamed Ashry. 2020. *Applying Deep Q-Networks (DQN) to the game of Tetris using high-level state spaces and different reward functions.* Ph. D. Dissertation. doi:10.13140/RG.2.2.27533.56807

[3] Sualeh Asif, Michael J. Coulombe, Erik D. Demaine, Martin L. Demaine, Adam Hesterberg, Jayson Lynch, and Mihir Singhal. 2020. Tetris is NP-hard even with $O(1)$ rows or columns. *CoRR* abs/2009.14336 (2020). arXiv:2009.14336 https://arxiv.org/abs/2009.14336

[4] Mahsa Bazzaz and Seth Cooper. 2024. Guided game level repair via explainable AI. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 20. 139–148.

[5] Y. Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. *Journal of the American Podiatry Association* 60, 6. doi:10.1145/1553374.1553380

[6] Riccardo Cantini, Alessio Orsino, and Domenico Talia. 2024. Xai-driven knowledge distillation of large language models for efficient deployment on low-resource devices. *Journal of Big Data* 11 (05 2024). doi:10.1186/s40537-024-00928-3

[7] Fernando de Mesentier Silva, Igor Borovikov, John Kolen, Navid Aghdaie, and Kazi A. Zaman. 2018. Exploring Gameplay With AI Agents. *CoRR* abs/1811.06962 (2018). arXiv:1811.06962 http://arxiv.org/abs/1811.06962

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[9] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. 2020. Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping. *CoRR* abs/2011.02669 (2020). arXiv:2011.02669 https://arxiv.org/abs/2011.02669

[10] Scott Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs.AI] https://arxiv.org/abs/1705.07874

[11] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938 (2016). arXiv:1602.04938 http://arxiv.org/abs/1602.04938

[12] Chathuranga Senanayake. 2025. DYNAMIC NPC AI USING REINFORCEMENT LEARNING FOR AN ENHANCED GAMING EXPERIENCE. doi:10.5281/zenodo.15024259

[13] Tianli Sun, Haonan Chen, Guosheng Hu, and Cairong Zhao. 2025. Explainability-based knowledge distillation. *Pattern Recognition* 159 (2025), 111095. doi:10.1016/j.patcog.2024.111095