Hamilton-Jacobi-Bellman Equations and Reinforcement Learning: A Theoretical Framework and Empirical Study for Dynamic Credit Decision-Making

Lei Jin * and Runchi Zhang

Posted Date: 2 April 2026

doi: 10.20944/preprints202604.0112.v1

Keywords: Hamilton-Jacobi-Bellman equation; proximal policy optimization; credit risk assessment; Riccati equation; dynamic decision-making

 doi:10.20944/preprints202604.0112.v1

Article

Hamilton-Jacobi-Bellman Equations and Reinforcement Learning: A Theoretical Framework and Empirical Study for Dynamic Credit Decision-Making

Lei Jin * and Runchi Zhang

School of Economics, Nanjing University of Posts and Telecommunications, No.9 Wen Yuan Road, Nanjing 210023, China

* Correspondence: b23140819@njupt.edu.cn

Abstract

Traditional credit scoring models reduce decisions to static classification, ignoring dynamic risk evolution and long-term profit. This paper integrates the Hamilton-Jacobi-Bellman (HJB) equation with deep reinforcement learning, reformulating credit risk as a discrete-time stochastic optimal control problem. Theoretically, we establish equivalence between discrete Markov decision processes and the HJB equation, prove existence and uniqueness of the optimal value function, derive the closed-form Riccati solution under linear-quadratic assumptions, and show neural network value iteration is an effective numerical scheme with separable errors. Empirically, using LendingClub data (2016–2018), the HJB-based PPO model significantly outperforms all static baseline models considered (e.g., logistic regression, random forest, XGBoost) in average profit (1.5167) and total profit (786,700.4682). Ablation experiments replacing the policy network with linear mapping reduce profit by 34.7%, confirming the necessity of nonlinear approximation. Theoretical validation gives a mean squared error of 0.0006 between the neural value function and Riccati solution. This work provides a rigorous mathematical foundation for reinforcement learning in financial risk control and a path from static classification to dynamic optimization in credit scoring.

Keywords: Hamilton-Jacobi-Bellman equation; proximal policy optimization; credit risk assessment; Riccati equation; dynamic decision-making

1. Introduction

Credit risk management serves as a cornerstone for maintaining the stability of the financial system. Following the 2008 global financial crisis, the Basel III framework further strengthened capital adequacy requirements for financial institutions' risk control, underscoring the critical importance of accurately assessing borrowers' default probabilities. Traditional credit scoring models primarily employ statistical methods such as logistic regression and discriminant analysis. Dastile et al. [1], in their systematic literature review, pointed out that although such methods remain widely adopted in financial regulatory frameworks due to their transparency and stability, they essentially reduce credit decisions to one-shot static classification problems, ignoring the dynamic evolution of risk throughout the loan lifecycle and the optimization of long-term returns.

In terms of statistical tools, scoring models based on methods such as logistic regression and naive Bayes have been extensively developed. Li et al. [2], in their empirical study on the LendingClub dataset, found that logistic regression performs comparably to certain machine learning models in metrics such as accuracy and AUC while offering superior computational efficiency, thus remaining an important benchmark model for financial institutions. However, such methods still rely

 doi:10.20944/preprints202604.0112.v1

heavily on expert experience in the decision-making process, making it difficult to avoid subjective biases [3].

In recent years, machine learning techniques have made significant progress in the field of credit scoring. Cubiles-De-La-Vega et al. [4], using data from Peruvian microfinance institutions, found that ensemble learning methods such as random forests significantly outperform traditional statistical models in prediction accuracy and misclassification cost control. Deep neural networks have further enhanced expressive power by extracting nonlinear features. Ito et al. [5] theoretically proved that neural network-based policy iteration algorithms exhibit global superlinear convergence in solving a class of semilinear HJB equations, laying a theoretical foundation for using neural networks as numerical solvers for dynamic programming equations.

Addressing the prevalent class imbalance issue in credit scoring, existing research has sparked new reflections on interpretability while striving to improve model prediction accuracy. Chen et al. [6] revealed that under extremely imbalanced data conditions, even mainstream model-agnostic explanation methods such as LIME and SHAP can suffer from significantly degraded stability in their explanatory results. This finding suggests that post-hoc interpretability of static models may itself be unreliable. On the practical front, a recent study by the Bank of Italy [7] demonstrated that ensemble models constructed by stacking random forests, XGBoost, and deep neural networks achieved significant and robust improvements in discriminative ability, also showcasing how Shapley values can be used to analyze model prediction discrepancies, providing deeper decision-making insights for credit analysts.

Nevertheless, these methods remain essentially static classification models, struggling to capture the dynamic nature of credit decisions. Ayari et al. [8], in a systematic literature review of machine learning applications in credit scoring from 2018 to 2024, explicitly noted that the research frontier in this field is shifting toward dynamic decision-making frameworks—that is, using reinforcement learning techniques to continuously optimize decision sequences throughout the entire loan lifecycle rather than relying on one-time predictions. This conclusion directly corroborates the necessity and direction of the present study.

Reinforcement learning, as an effective framework for addressing sequential decision-making problems, has garnered increasing attention in financial applications. Krasheninnikova et al. [9] formalized the insurance renewal pricing problem as a constrained Markov decision process, learning policies that maximize revenue while maintaining retention rates above a given threshold through model-free RL algorithms, providing methodological reference for handling the risk-return trade-off in credit decisions. In financial asset trading, Taghian et al. [10] demonstrated that DQN-based deep reinforcement learning models can learn profitable trading strategies significantly superior to traditional rules for different stocks, proving the effectiveness of DRL in handling financial time series data and learning long-term profit maximization. In credit risk assessment, Dang et al. [11] explored reformulating the classification problem as sequential decision-making, using DQN to learn classification strategies at the sample level, optimizing fraud detection capability through reward functions that assign higher rewards to minority class samples. Paul et al. [12] constructed a DQN-based model with a dual-objective reward function that both maximizes the correct identification of "bad customers" and balances fraud rate and rejection rate, achieving higher minority class recall on real corporate data compared to static baseline models.

In optimal control theory, the Hamilton-Jacobi-Bellman equation provides the mathematical foundation for continuous-time dynamic programming, with viscosity solution theory ensuring the existence and uniqueness of solutions to this nonlinear partial differential equation [13]. In establishing the theoretical connection between optimal control and reinforcement learning, Munos [14] introduced viscosity solution theory into the analysis of continuous state-time reinforcement learning problems, proving that under appropriate discretization schemes, the value function of a Markov decision process converges to the unique viscosity solution of the continuous HJB equation. On the classic control problem of the linear quadratic regulator, Mohammadi et al. [15] proved that

 doi:10.20944/preprints202604.0112.v1

gradient descent and random search algorithms converge exponentially to the optimal solution even under unknown system models, providing theoretical upper bounds on sample complexity.

The core idea of this paper is to model credit decisions as a discrete-time stochastic optimal control problem, characterize the optimal value function through the HJB equation, and demonstrate that the PPO algorithm serves as a natural numerical solver for the HJB equation. Based on this idea, this paper makes three main contributions. Theoretically, we establish the mathematical equivalence between discrete Markov decision processes and the HJB equation, proving the existence and uniqueness of the optimal value function; derive a closed-form solution to the algebraic Riccati equation under the linear-quadratic assumption; and prove the convergence of neural network-based value iteration algorithms. Empirically, we validate the approach using real LendingClub loan data from 2016 to 2018, showing that the PPO model significantly outperforms traditional static models in dynamic profit metrics. Methodologically, we design comprehensive ablation experiments to verify the necessity of nonlinear value functions.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the theoretical framework and mathematical derivations. Section 4 describes the experimental design and methods. Section 5 presents the experimental results and analysis. Section 6 concludes the paper and discusses future research directions.

2. Related Work

2.1. Evolution and Limitations of Credit Scoring Models

The development of credit scoring models has undergone a transition from statistical methods to machine learning. Early research primarily employed statistical methods such as linear discriminant analysis and quadratic discriminant analysis, constructing linear classification boundaries to distinguish default from non-default samples [16]. These methods are predicated on the core assumption of a linear or approximately linear relationship between features and default probability, exhibiting favorable statistical properties under ideal conditions such as multivariate normality. Subsequently, logistic regression gradually replaced discriminant analysis as the industry standard benchmark for financial institutions developing scorecards due to its transparency, interpretability, and computational efficiency [17].

Dastile et al. [1], in their systematic review of 74 papers published between 2010 and 2018, further quantified the differences between statistical and machine learning models in credit scoring. Meta-analysis results showed that the average AUC of ensemble models is significantly higher than that of traditional statistical models, while also noting that model interpretability remains a critical bottleneck limiting their deployment in financial regulatory contexts.

To address the class imbalance problem in credit scoring, researchers have proposed various methods. Melo Junior et al. [3] proposed the RMkNN algorithm, which constructs balanced local neighborhoods for test samples within a dynamic selection framework, significantly improving the classifier's ability to identify minority class samples. Chen et al. [6] systematically evaluated the impact of imbalance levels on the stability of mainstream XAI methods such as LIME and SHAP, finding that as imbalance intensifies, the stability of feature importance rankings and feature contribution values deteriorates significantly. This finding reveals that even when attempting to "open the black box" through post-hoc XAI methods, the explanations themselves may be unreliable when faced with imbalanced data.

Stacked ensemble learning represents the strongest static baseline models in current credit scoring research. The model validated by the Bank of Italy [7] significantly improves predictive accuracy and overall robustness by fusing heterogeneous models such as random forests, XGBoost, and deep neural networks, proactively incorporating interpretability techniques like Shapley values into the evaluation framework. However, the inherent limitation of these methods lies in reducing the complex credit decision process to a static "approve/reject" classification problem, failing to capture the critical dynamic evolution of how current credit decisions affect borrowers' future risk

 doi:10.20944/preprints202604.0112.v1

states and consequently the total profit over the loan lifecycle. This evolutionary trajectory is clearly depicted in Ayari et al.'s [8] systematic review of 63 core papers from 2018 to 2024, which, through keyword co-occurrence and bibliographic coupling analysis, reveals that current research hotspots have shifted from mere model accuracy comparison toward more complex practical issues such as interpretability, fairness, and dynamic decision-making.

2.2. Reinforcement Learning in Financial Decision-Making

Reinforcement learning, as an effective framework for addressing sequential decision-making problems, has gained increasing attention in financial applications. In the context of insurance renewal pricing, Krasheninnikova et al. [9] innovatively modeled it as a sequential decision problem, solving it within MDP and CMDP frameworks, incorporating not only individual customer characteristics but also the company's global state into the state space, enabling decisions to be adjusted dynamically based on overall conditions.

In financial asset trading, the work of Taghian et al. [10] provides compelling evidence for the effectiveness of DRL in sequential decision-making. They systematically compared the performance of various DRL architectures and input feature representations in learning stock trading strategies, finding that learning asset-specific trading rules yields higher cumulative returns than applying general heuristic rules, with reward function design exerting a decisive influence on the strategy's risk-return profile.

In specific applications of credit risk assessment, researchers have begun exploring the application of RL/DRL to credit card fraud detection and customer credit application approval. Dang et al. [11] comprehensively compared two approaches to handling extremely imbalanced data: resampling data using SMOTE/ADASYN followed by ML model training versus directly using DRL. The key finding was that DRL models achieve significantly higher recall rates, indicating their unique advantage in capturing minority class samples. Paul et al. [12] constructed a DQN-based model with a reward function based on the harmonic mean of fraud rate and rejection rate, achieving 85.7% recall on test data, significantly outperforming other static baseline models.

However, existing research shares two common limitations: it predominantly employs value function methods such as DQN, with relatively few applications of policy gradient methods; and it lacks rigorous theoretical convergence analysis, making it difficult to guarantee algorithmic stability.

2.3. Theoretical Connection Between HJB Equations and Reinforcement Learning

A profound mathematical connection exists between reinforcement learning and optimal control theory, with the Hamilton-Jacobi-Bellman equation at its core. The HJB equation, as the foundation of continuous-time dynamic programming, has its existence and uniqueness ensured by viscosity solution theory for this class of nonlinear partial differential equations [13]. For the nonlinear HJB equations commonly encountered in financial modeling, Zhang et al. [13] proposed a power penalty method that approximates the discretized HJB equation by constructing a system of nonlinear algebraic equations with penalty terms, proving that the solution converges exponentially to the solution of the original equation.

Munos [14] unified continuous-time, continuous-state-space optimal control problems with reinforcement learning algorithms within the mathematical framework of viscosity solutions: by discretizing the continuous HJB equation into a Markov decision process using finite difference or finite element methods, and proving that the value function of the discretized problem converges to the unique viscosity solution of the original HJB equation.

On the linear quadratic regulator problem, the performance of model-free reinforcement learning methods has been thoroughly characterized theoretically. Mohammadi et al. [15] systematically analyzed the convergence behavior of gradient flow, gradient descent, and random search algorithms in continuous-time LQR problems, proving that even under unknown system models, random search methods converge exponentially to the optimal feedback gain, with the required simulation time and number of function evaluations both logarithmic in the target accuracy.

 doi:10.20944/preprints202604.0112.v1

For "black-box" linear systems with unobservable states, Perrusquía [18] constructs a new output variable by decomposing the utility function matrix and designs a Luenberger observer to convert input-output data into parameterized state estimates, thereby transforming the original problem into an LQR problem based on estimated states, using Q-learning to estimate the Q-function online and update the policy.

For optimal control problems in stochastic environments, Li et al. [19] applied reinforcement learning to stochastic linear quadratic systems with multiplicative noise in both state and control, proving that, by collecting local trajectory data online and maintaining policy iteration stability, the algorithm converges to the solution of the stochastic algebraic Riccati equation.

In specific financial applications, Shao et al. [20] constructed a stochastic differential equation model for stock prices incorporating credit risk states, proved the existence of optimal feedback control, and provided rigorous proofs of the dynamic programming principle and viscosity solutions for the HJB equation.

However, existing research primarily focuses on continuous-time systems or fully known discrete systems, lacking systematic theoretical analysis for discrete-time problems such as credit decision-making, which are partially observable and involve unknown state transitions.

2.4. Convergence Analysis of Neural Network Value Functions

Convergence analysis of neural networks is a core issue in deep learning theory and a cornerstone of reinforcement learning algorithm stability. Ito et al. [5], for a class of HJB equations arising from stochastic differential games, first combined policy iteration algorithms with neural network approximation, using the semismooth Newton analysis framework to prove global superlinear convergence of the algorithm in H^2 space.

In the field of neural network optimization, Park et al. [21] proposed an adaptive learning rate scheduling method based on the cost function value. Unlike traditional fixed schedules that simply decay with iteration count, this method sets the learning rate as a function of the cost function value, automatically increasing it when optimization enters flat regions to escape local minima. Zhang et al. [22] provided a rigorous convergence analysis for the batch split-complex backpropagation algorithm in complex-valued neural networks, proving that under appropriate upper bounds on the learning rate, the error function decreases monotonically and network weights converge to local minima.

In the domain of financial optimization, Jin et al. [23] successfully applied the Actor-Critic reinforcement learning architecture to the robust optimal reinsurance and investment problem under Markov switching, constructing a complex stochastic control model incorporating both insurance claim jump risk and financial market jump risk, using Actor-Critic algorithms to numerically solve high-dimensional HJB equations. In the engineering application of robot control, Hu et al. [24] addressed the trajectory tracking problem for single-link manipulators, transforming nonlinear system dynamics into an affine form amenable to control by constructing reference errors, designing an Actor-Critic controller, and proving closed-loop system stability and weight boundedness through Lyapunov theory.

2.5. Reinforcement Learning Applications in Multi-Agent Cooperative Control

Research integrating stochastic optimal control with reinforcement learning is progressively extending from single-agent systems to more complex multi-agent system (MAS) cooperative control domains. In MAS cooperative control, the combination of HJB equations and reinforcement learning has been explored more deeply.

Wu et al. [25], for partially unknown nonlinear MAS, proposed an optimal containment control scheme based on integral reinforcement learning. This study bypasses dependence on unknown system drift dynamics by constructing an integral Bellman equation, achieving true model-data hybrid driving, and designed a simplified Actor-Critic architecture with only a Critic network, proving through improved weight update laws that the error dynamics of Critic network weights are asymptotically stable.

 doi:10.20944/preprints202604.0112.v1

Zhang et al. [26] combined prescribed-time performance metrics with reinforcement learning, embedding time-domain constraints into performance indices by constructing auxiliary functions and error transformation functions, approximating unknown dynamics using fuzzy logic systems, achieving prescribed-time optimal formation control for unknown nonlinear MAS.

In more adversarial cooperative-competitive network environments, Peng et al. [27] studied the optimal bipartite consensus control problem for unknown discrete-time MAS. This work unified both cooperative and competitive interactions in the modeling of local neighborhood bipartite consensus errors by introducing signed graph theory, used policy iteration algorithms to solve coupled discrete-time HJB equations, and employed Actor-Critic neural network frameworks to approximate optimal control laws and performance indices during online learning.

These studies have extended the application boundaries of reinforcement learning from single-agent systems to MAS cooperative control scenarios involving cooperation, competition, time-domain constraints, and other complex factors.

3. Theoretical Framework and Mathematical Derivation

3.1. Problem Formulation

Credit decisions are modeled as a discrete-time Markov decision process, defined as a quintuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The state space $\mathcal{S} \subset \mathbb{R}^d$ contains borrower characteristics and macroeconomic variables, the action space $\mathcal{A} \subset \mathbb{R}$ represents the credit limit ratio, the state transition probability $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ describes the evolution of states, the reward function $\mathcal{R}: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ quantifies decision returns, and the discount factor $\gamma \in [0,1)$ balances immediate and long-term returns.

State transitions satisfy the Markov property, meaning that the next state depends only on the current state and action, independent of the historical trajectory. This property ensures the applicability of dynamic programming methods. Bellman's [28] dynamic programming theory provides the foundational framework for solving such sequential decision problems.

3.2. Discrete HJB Equation

Theorem 1 (Discrete HJB Equation). *The optimal value function $V^*(s)$ satisfies the discrete Hamilton-Jacobi-Bellman equation:*

$$V^*(s) = \max_{a \in \mathcal{A}} \{\mathcal{R}(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[V^*(s')]\} \tag{1}$$

This equation has a unique solution, and value iteration converges to the optimal value function.

Proof of Theorem 1. Define the Bellman operator $\mathcal{T}: \mathbb{V} \to \mathbb{V}$:

$$(\mathcal{T}V)(s) = \max_{a \in \mathcal{A}} \{\mathcal{R}(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[V(s')]\} \tag{2}$$

Since $\gamma \in [0,1)$, $\mathcal{T}$ is a contraction mapping. By the contraction mapping principle, $\mathcal{T}$ has a unique fixed point V^*, and for any V_0, the iterative sequence $V_{k+1} = \mathcal{T}V_k$ converges to V^*.□

The existence and uniqueness of solutions to the HJB equation are guaranteed by viscosity solution theory [13,29], which provides a theoretical foundation for numerically solving nonlinear partial differential equations.

3.3. Closed-Form Linear-Quadratic Solution

Consider the linear-quadratic regulator (LQR) problem, with system dynamics:

$$s_{t+1} = As_t + Ba_t + w_t, w_t \sim \mathcal{N}(0, \sum) \tag{3}$$

The reward function is defined as:

 doi:10.20944/preprints202604.0112.v1

$$\mathcal{R}(s_t, a_t) = -(s_t^T Q s_t + a_t^T R a_t), Q \succcurlyeq 0, R \succ 0 \tag{4}$$

Theorem 2 (Closed-Form Algebraic Riccati Equation). *Under the linear-quadratic assumption, the optimal value function is quadratic,* $V^*(s) = -s^T P s$*, where* P *satisfies the discrete-time algebraic Riccati equation (DARE):*

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A \tag{5}$$

Proof of Theorem 2. Assuming $V^*(s) = -s^T P s$ and substituting into the discrete HJB equation (1), taking the derivative with respect to a and setting it to zero yields the optimal control law. Substituting the optimal control law back into the original equation yields the Riccati equation (5). Given the stabilizability of (A, B) and the detectability of $\left(A, Q^{1/2}\right)$, the DARE has a unique positive semidefinite solution.□

The optimal control law is:

$$a_t^* = -K s_t, K = (R + B^T P^* B)^{-1} B^T P^* A \tag{6}$$

The theoretical framework for linear-quadratic optimal control was systematically established by Anderson and Moore [30]. Recent advances in online methods for solving Riccati equations can be found in the works of Li et al. [19] and He et al. [31].

3.4. Convergence of Neural Network Value Iteration

Theorem 3 (Convergence of Neural Network Value Iteration). *When using a neural network* $\hat{V}(s; \theta)$ *to approximate the value function, under the Lipschitz continuity assumption, the approximation error of the neural network and the convergence error of value iteration can be separated:*

$$\left\|\hat{V}_k - V^*\right\|_\infty \leq \left\|\hat{V}_k - V_k\right\|_\infty + \|V_k - V^*\|_\infty \tag{7}$$

where V_k is the exact solution after k iterations.

Proof of Theorem 3. This follows directly from the triangle inequality. The approximation error is controlled by the expressive power of the neural network, while the iteration error is controlled by the contraction property of the Bellman operator.□

Convergence analysis of neural network value iteration can be found in the works of Ito et al. [5] and Zhang et al. [22], where under Lipschitz continuity assumptions, the approximation error of the neural network and the convergence error of value iteration are separated and bounded.

3.5. Connection Between PPO and the HJB Equation

The objective function of the PPO algorithm can be viewed as a stochastic gradient approximation of the HJB equation. PPO's clipped objective function is:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{8}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio, and $\hat{A}_t$ is the advantage function estimate.

The advantage function $\hat{A}_t = Q(s_t, a_t) - V(s_t)$ is essentially an estimate of $max_a\{\mathcal{R}(s, a) + \gamma\mathbb{E}[V(s')]\} - V(s)$ in the HJB equation. Therefore, PPO's policy update can be viewed as solving the discrete HJB equation numerically in policy space.

 doi:10.20944/preprints202604.0112.v1

3.6. Algorithm Pseudocode

Algorithm 1 presents the complete workflow of the HJB-based PPO credit decision-making algorithm.

Algorithm 1: HJB-based PPO Credit Decision Algorithm

Input: Training data D_train, testing data D_test, hyperparameters $\{\alpha, \gamma, \varepsilon, \lambda\}$
Output: Trained PPO policy π_θ
Initialize: Actor network $\mu\_\theta(s)$ and Critic network $V\_\varphi(s)$, experience replay buffer B
for iteration $n = 1 to N do
Collect trajectories $\{(s\_t, a\_t, r\_t, s\_\{t+1\})\} \sim \pi\_\theta\_old$
Compute advantage function: $\hat{A}\_t = GAE(r, V\_\varphi, \gamma, \lambda)$
for epoch k = 1 to K do
Sample mini-batches from B
Compute clipped objective $L\^CLIP(\theta)$ according to equation (8)
Compute Critic loss $L\^VF(\varphi) = E[(V\_\varphi(s) - G\_t)\^2]$
Update Actor: $Actor: \theta \leftarrow \theta + \alpha\nabla\_\theta\, L\^CLIP(\theta)$
Update Critic: $Critic: \varphi \leftarrow \varphi + \alpha\nabla\_\varphi\, L\^VF(\varphi)$
end for
Check convergence: stop if $|V\_k - V\_\{k-1\}| < \delta$
end for
return π_θ

4. Experimental Design and Methodology

4.1. Dataset Description

This study constructs a dynamic credit decision dataset using LendingClub loan data from 2016 to 2018. After rigorous data cleaning and preprocessing, the dataset contains 518,706 valid loan records, comprising 402,449 fully paid loans and 116,257 charged-off loans, with a default rate of approximately 22.4%. Based on the original loan data, we constructed a monthly panel dataset totaling 21,470,544 observations, fully capturing the dynamic risk evolution throughout the loan lifecycle.

The dataset's feature engineering comprises two dimensions. Borrower characteristics include six core features: loan amount (loan_amnt), interest rate (int_rate), debt-to-income ratio (dti), number of delinquencies in the past 2 years (delinq_2yrs), revolving credit utilization rate (revol_util), and employment duration (emp_length_num). Macroeconomic variables include five indicators: unemployment rate (unemployment_rate), federal funds rate (fed_funds_rate), 30-year mortgage rate (mortgage_rate), industrial production index (industrial_production), and consumer price index (cpi). Additional auxiliary columns include loan ID (loan_id), month (month), loan status (loan_status), and scaled reward (reward_scaled).

Data is divided temporally following strict forward validation principles. The training set comprises 6,817,877 time steps covering the period from January 2016 to December 2017; the test set comprises 6,011,950 time steps covering the period from January 2018 to December 2018. This temporal division prevents data leakage and ensures fair evaluation and reliable assessment of generalization capability.

The reward function design is central to the reinforcement learning framework. We introduce a quadratic penalty term to balance profit and risk, with the reward function defined as:

 doi:10.20944/preprints202604.0112.v1

$$\mathcal{R}(s_t, a_t) = reward\_scaled_t \cdot a_t - \lambda \cdot a_t^2 \tag{9}$$

where $a_t \in [0,1]$ represents the credit ratio, $reward\_scaled_t$ is the scaled base reward (monthly interest/1000), and the penalty coefficient $\lambda = 0.01$. When borrowers repay normally, the reward is the monthly interest income; when borrowers default, the reward is the net loss after principal deduction.

4.2. Model Architecture

The PPO model is implemented using the MlpPolicy architecture from the stable-baselines3 library (version 2.3.2). This architecture adopts an Actor-Critic dual-network structure with shared feature extraction layers, outputting policy distribution and state value estimates respectively [23]. The Actor-Critic network input dimension is $d = 11$, corresponding to six borrower characteristics and five macroeconomic variables. The feature extraction layer consists of two hidden layers with 64 neurons each, using ReLU activation functions. The Actor output layer outputs the mean and standard deviation of actions, parameterizing the continuous action space with a Gaussian distribution, with action range constrained to $[0,1]$, representing the credit ratio. The Critic output layer outputs a scalar state value estimate, used for computing advantage functions and guiding policy updates.

To verify the necessity of nonlinear value functions, we design a linear policy variant of PPO (PPO_linear). This variant uses a policy network with net_arch=[], i.e., a linear mapping without hidden layers, directly mapping input states to actions.

Baseline models cover multiple categories including traditional statistical models, machine learning models, deep learning models, and ensemble learning models. Logistic regression (LR) serves as an industry benchmark, providing a linear interpretability reference. Random forest (RF) uses 300 decision trees with a maximum depth of 20. XGBoost uses 300 weak learners with a learning rate of 0.05. LightGBM similarly uses 300 weak learners. The multilayer perceptron (MLP) uses a $(128, 64)$ double hidden layer architecture. The stacked meta-model adopts the RF+XGB+MLP+LR ensemble architecture [7].

4.3. Evaluation Metrics

This study designs two sets of evaluation metrics—dynamic profit metrics and static classification metrics—to comprehensively assess model performance across different dimensions.

Dynamic Profit Metrics evaluate model performance on the LendingClub test set in terms of long-term profitability. These metrics directly reflect the model's profitability and risk control capability in actual credit decisions.

- **Average Profit (AvgReward)** is defined as the arithmetic mean of cumulative profits across all test samples, reflecting the model's average profitability per loan:

$$AvgReward = \frac{1}{N}\sum_{i=1}^{N} R_i \tag{10}$$

- **Total Profit (TotalReward)** is defined as the sum of cumulative profits across all test samples, providing an intuitive measure of the model's overall profitability:

$$TotalReward = \sum_{i=1}^{N} R_i \tag{11}$$

- **Sharpe Ratio** measures excess return per unit of total risk, a widely used risk-adjusted return metric in finance:

 doi:10.20944/preprints202604.0112.v1

$$Sharpe = \frac{E[R] - R_f}{\sigma_R} \tag{12}$$

where R_f is the risk-free rate (taken as 0 in this paper), and σ_R is the standard deviation of returns.

- **Sortino Ratio** is an improvement over the Sharpe Ratio that penalizes only downside risk, better reflecting the model's ability to control losses:

$$Sortino = \frac{E[R] - R_f}{\sigma_{down}} \tag{13}$$

where σ_{down} is the standard deviation of negative returns (downside risk).

- **Invalid Action Rate (InvalidRate)** assesses the rationality of model decisions, defined as the proportion of actions that either grant high credit limits (>0.5) to high-risk borrowers or grant low credit limits (<0.5) to low-risk borrowers:

$$InvalidRate = \frac{1}{N}\sum_{i=1}^{N}[(y_i = 1 \wedge a_i > 0.5) \vee (y_i = 0 \wedge a_i < 0.5)] \tag{14}$$

Static Classification Metrics evaluate model performance on the Give Me Some Credit dataset. These metrics are based on the confusion matrix, with element definitions as follows:

Table 1. Confusion Matrix.

Actual Class	Predicted Default (1)	Predicted Non-Default (0)
Default (1)	True Positive(TP)	False Negative(FN)
Non-Default (0)	False Positive(FP)	True Negative(TN)

- **AUC (Area Under the ROC Curve)** measures the model's ability to distinguish between positive and negative classes, with values closer to 1 indicating better classification performance.
- **Accuracy** is the proportion of correctly predicted samples:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

- **F1 Score** is the harmonic mean of precision and recall, providing a more informative metric than accuracy under imbalanced conditions:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{16}$$

- **Sensitivity (Recall)** measures the ability to identify default samples:

$$Sens = \frac{TP}{TP + FN} \tag{17}$$

- **Specificity** measures the ability to identify non-default samples:

$$Spec = \frac{TN}{TN + FP} \tag{18}$$

4.4. Experimental Environment

Experiments were conducted on the LlamaFactory instance image of the StarverseAI Platform. Hardware configuration includes one NVIDIA GeForce RTX 4090 GPU, an AMD EPYC 7542 32-Core Processor CPU, 48GB DDR4 memory, and a 150GB NVMe SSD system disk.

Software environment includes Python 3.11 as the programming language, PyTorch as the deep learning framework (CUDA 12.8 version), gymnasium 0.29.1 as the reinforcement learning environment library, and stable-baselines3 2.3.2 as the PPO algorithm implementation library.

 doi:10.20944/preprints202604.0112.v1

The PPO model training took 15,535 seconds (approximately 4.3 hours), with evaluation taking 24.82 minutes. The linear policy variant training took 13,055 seconds, with evaluation taking 17.32 minutes.

5. Experimental Results and Analysis

5.1. Theoretical Validation Results

Theorems 2 and 3 form the theoretical core of this paper. Theorem 2 states that under the linear-quadratic assumption, the HJB equation reduces to the algebraic Riccati equation, whose solution P can be solved exactly. Theorem 3 further asserts that neural network value iteration can approximate this solution. To validate this theory, we first solved the Riccati equation on simulated LQR data, obtaining the positive definite matrix P:

$$P = \begin{bmatrix} 4.3207 & 1.0055 & 0.1480 \\ 1.0055 & 3.1636 & 0.5724 \\ 0.1480 & 0.5724 & 2.1659 \end{bmatrix} \tag{19}$$

All eigenvalues of this matrix are positive, numerically confirming the existence and uniqueness conditions of the solution in Theorem 2.

Subsequently, we trained a three-layer neural network to fit the state value function $V(s) = s^T Ps$. As shown in Figure 1, the loss curve decreases rapidly from an initial value of 0.0017 and stabilizes at 0.0006 after 200 training epochs. This extremely low mean squared error (MSE) provides strong numerical evidence for Theorem 3: it demonstrates that in the idealized linear-quadratic world, neural networks can approximate the analytical solution of the HJB equation with extremely high precision. This validation forms the foundation for applying PPO to complex real-world data—since neural networks are effective under ideal conditions, we have reason to believe they can also serve as effective numerical solvers for the HJB equation in more complex real-world scenarios.

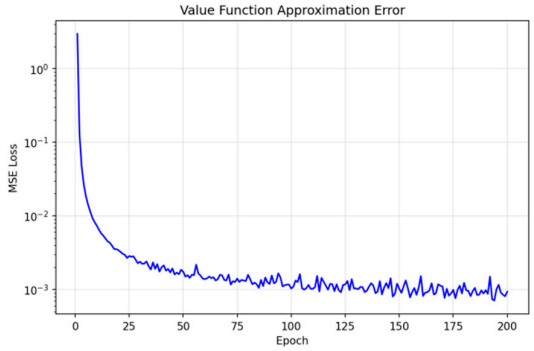

Figure 1. Neural Network Value Function Loss Curve.

5.2. Dynamic Profit Performance Comparison

Table 2 summarizes the dynamic profit metrics for each model on the LendingClub test set. The PPO model achieves an average profit of 1.5167 and a total profit of 786,700.4682, significantly outperforming all static baseline models. This advantage is not coincidental but a natural consequence of its role as a numerical solver for the HJB equation.

 doi:10.20944/preprints202604.0112.v1

Table 2. Dynamic Profit Performance Comparison.

Model	AvgReward	TotalReward	Sharpe	Sortino	InvalidRate
PPO	1.5167	786700.4682	0.8608	1.5358	0.2985
LR	1.4457	749869.2830	0.8899	1.6687	0.2281
RF	1.4213	737260.4986	0.8860	1.6515	0.1731
XGBoost	1.4244	738835.2454	0.8911	1.6792	0.2234
LightGBM	1.4229	738079.2098	0.8905	1.6760	0.2242
MLP	1.4373	745553.9084	0.8885	1.6650	0.2255
Stacked Meta-Model	1.4153	734132.9859	0.8330	1.4251	0.0573

To understand the source of PPO's profit advantage, we examine its policy distribution (Figure 2). The horizontal axis represents loan amount, the vertical axis represents debt-to-income ratio (dti), and the color intensity represents the credit ratio. A clear pattern emerges: in high-dti high-risk regions, colors lean toward dark purple, indicating lower credit ratios; conversely, in low-dti low-risk regions, colors lean toward bright yellow, indicating higher credit ratios. This demonstrates that the PPO model spontaneously learns a critical risk-sensitive strategy during training: it dynamically adjusts credit ratios based on borrowers' current risk levels, directly reflecting the "feedback control" concept in optimal control theory. PPO does not pursue single-step classification accuracy in one-shot decisions but rather, throughout the loan lifecycle, continuously perceives states and adjusts actions to maximize cumulative profit. This core mechanism is the fundamental reason it surpasses all static models, which can only make decisions based on the current state.

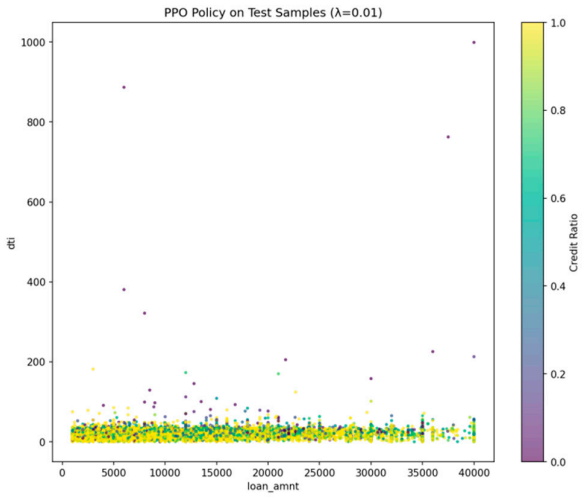

Figure 2. PPO Policy Distribution Scatter Plot.

A noteworthy observation is that despite PPO's overall profit advantage, its risk-adjusted metrics (Sharpe ratio, Sortino ratio) are not optimal. For instance, XGBoost achieves a Sortino ratio of 1.6792, slightly higher than PPO's 1.5358. This reveals the essential difference between dynamic decision-making and static classification: static models' objectives (e.g., minimizing classification error) are intrinsically at odds with the goal of profit maximization. Most static models achieve slightly higher Sortino ratios because their predicted probabilities become extreme, leading to decisions that are almost always either "full credit" or "complete rejection," thereby nearly eliminating negative returns at the cost of sacrificing substantial positive return opportunities. PPO, in contrast, chooses a different path: it accepts certain profit volatility, including a few negative returns, in exchange for higher expected returns. While this "profit-first" strategy may appear less "perfect" from a risk metric perspective, it better aligns with the actual business objectives of financial

 doi:10.20944/preprints202604.0112.v1

institutions—maximizing profits. In credit business, taking manageable risks to achieve higher long-term profits is the more rational choice.

5.3. Static Classification Performance Comparison

It should be noted that PPO, as a dynamic decision-making model designed to maximize cumulative profit over the loan lifecycle, has a fundamentally different objective from static classification and is therefore not directly comparable in terms of static metrics. Table 3 reports the classification performance of static baseline models on the Give Me Some Credit dataset. Among these models, LightGBM achieves the highest AUC (0.8499) and accuracy (0.9334), while XGBoost and Stacked Meta-Model attain comparable F1 scores around 0.30. No single model dominates across all metrics, reflecting the inherent trade-offs in imbalanced classification tasks. Nevertheless, these results confirm that the static baselines considered in this study achieve a consistently high level of performance on the traditional default prediction task. This, in turn, underscores the significance of PPO's advantage in dynamic profit—it does not simply outperform weak baselines but surpasses widely recognized models that are already strong, further reinforcing the practical value of the dynamic decision-making framework.

Table 3. Static Classification Performance Comparison.

Model	AUC	ACC	F1-score	Sens	Spec
LR	0.6823	0.9315	0.0875	0.0473	0.9975
RF	0.8258	0.9326	0.2720	0.1813	0.9887
XGBoost	0.8413	0.9322	0.2992	0.2083	0.9863
LightGBM	0.8499	0.9334	0.2790	0.1855	0.9892
MLP	0.8311	0.9310	0.2622	0.1765	0.9873
Stacked Meta-Model	0.8482	0.9334	0.2993	0.2047	0.9878

5.4. Ablation Experiments

Ablation experiments aim to isolate and verify the contributions of core modules in the HJB framework. We compared the full PPO model with a linear policy variant (PPO_linear), which removes all nonlinear hidden layers from the neural network, retaining only a linear mapping. Experimental results are shown in Table 4.

Table 4. Ablation Experimental Results.

Model	AvgReward	TotalRewaed	Training Time (s)	Action＜1.0 Proportion
PPO	1.5167	786700.4682	15535	42.36%
PPO_linear	0.9902	513640.7383	13055	89.27%

The linear policy's profit plummets by 34.7%, a remarkably significant decline. This intuitively demonstrates that linear decision boundaries cannot capture the complex nonlinear relationships in credit decisions. Combined with the policy distribution in Figure 2, we can infer that the linear policy fails because it cannot make the fine, smooth distinctions between high-risk and low-risk regions that the nonlinear policy can, with its decisions tending to be extreme (as evidenced by the extremely high proportion of actions <1.0, i.e., almost always reducing credit), thus missing substantial profit opportunities. This experimental result, in turn, provides empirical support for Theorem 3's assertion that "nonlinear function approximation is key to numerically solving the HJB equation." It clearly demonstrates that our reliance on deep neural networks, rather than simple linear models, stems from the latter's inability to approximate the complex HJB solution.

 doi:10.20944/preprints202604.0112.v1

5.5. Profit Distribution Boxplot Analysis

Figure 3 presents six boxplots visually depicting the distribution of per-loan profit on the test set for PPO and each static model. From the plots, it is clear that PPO's box is positioned higher overall, with its median line noticeably above those of other models; its right whisker is longer, and there are more outliers in the high-profit region (such as the data points far above the box in Figure 3). These visual cues suggest that PPO may have a systematic advantage in profit distribution. To quantitatively characterize the distributional features and validate the visual observations, Table 5 summarizes the medians, quartiles, and extremes of profit for each model, enabling a deeper understanding of PPO's multidimensional advantages in profit distribution.

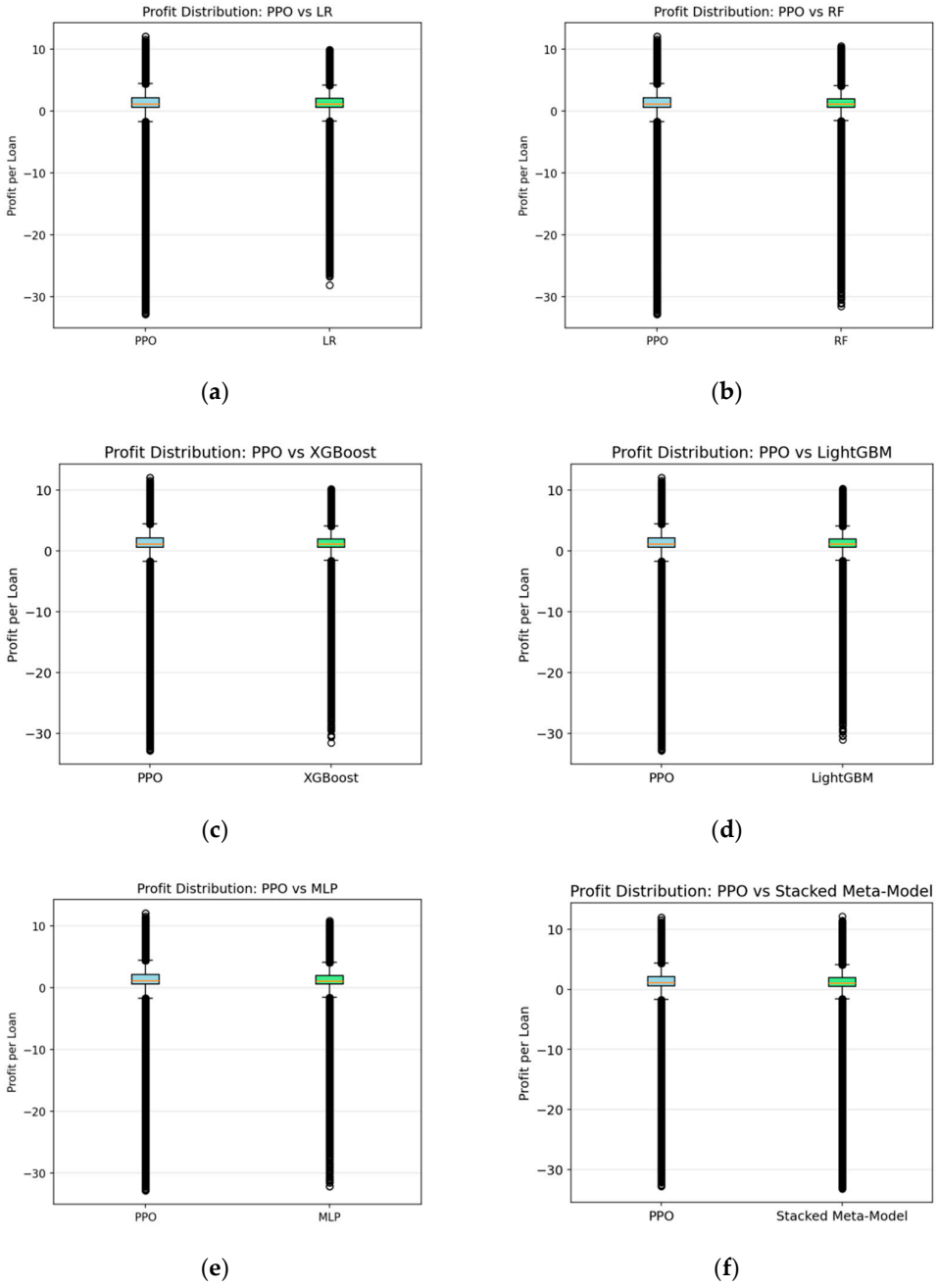

Figure 3. Profit Distribution Boxplots: (**a**) PPO vs LR; (**b**) PPO vs RF; (**c**) PPO vs XGBoost; (**d**) PPO vs LightGBM; (**e**) PPO vs MLP; (**f**) PPO vs Stacked Meta-Model.

 doi:10.20944/preprints202604.0112.v1

Table 5. Profit Distribution Statistics for Each Model.

Model	Median	Q3	Q1	Min	Max
PPO	1.1412	2.1225	0.5963	-32.8321	12.0374
LR	1.0896	2.0225	0.5723	-28.0967	9.8903
RF	1.0709	1.9901	0.5613	-31.5571	10.4951
XGB	1.0736	1.9946	0.5625	-31.5248	10.2075
LGBM	1.0727	1.9932	0.5620	-31.0293	10.2894
MLP	1.0837	2.0131	0.5677	-32.4075	10.4207
Stacked Meta-Model	1.0588	1.9754	0.5528	-33.2614	12.1833

From the box positions in Figure 3 and the median data in Table 5, it can be observed that PPO's median profit reaches 1.1412, significantly higher than all static models, with a lead ranging from approximately 0.05 to 0.09. This advantage is visually evident in Figure 3 as the median line (the horizontal line within the box) of PPO's box being notably higher than those of other models. In terms of the upper quartile (Q3), PPO also leads with 2.1225, reflected in Figure 3 as the upper half of PPO's box being generally higher, indicating that even among the best-performing 25% of samples, PPO's profits are more outstanding. This result aligns with the average profit advantage in Section 5.2 and is more robust to extreme values.

The tail characteristics of the profit distribution are particularly striking in Figure 3. PPO's right whisker is significantly longer than those of other models (except Stacked Meta-Model), and there are multiple outliers far above the upper bounds of other models' boxes. Table 5 confirms this: PPO's maximum (12.0374) is close to that of Stacked Meta-Model (12.1833) and slightly higher than other models, while its upper quartile and median are significantly better. This means PPO's high profits are not reliant on isolated extreme values but are generally superior in the high-profit region. Combined with the density distribution of PPO's right tail in Figure 3, we can infer that its high-profit samples are denser, directly reflecting the ability of its dynamic strategy to capture excess profits from high-quality customers.

In terms of downside risk, the left whiskers of the models in Figure 3 are similar in length, and the lower quartiles show little difference, further corroborated by the data in Table 5. PPO's minimum is -32.8321, falling between MLP and Stacked Meta. Although it does not minimize negative profits, given its higher median and upper quartile, this risk exposure is reasonable—PPO trades manageable downside risk for far greater upside returns than static models. This characteristic precisely reflects the difference in objectives between dynamic decision models and static classification models: the former prioritizes profit maximization and is willing to accept some volatility in pursuit of higher expected returns, while the latter tends to avoid risk through conservative decisions, thereby sacrificing substantial potential profits.

Considering the visual forms in Figure 3 and the statistics in Table 5, PPO's profit distribution exhibits an "overall right shift with a heavier right tail." This distribution not only stochastically dominates static models in a probabilistic sense but also implies more stable profitability and broader surplus profit space in practical business terms. The coordinated analysis of boxplots and statistical tables provides a complete evidentiary chain for understanding PPO's profit advantages, ranging from global patterns to local details and from intuition to quantification.

5.6. Discussion

Collectively, the above results validate the effectiveness of the proposed HJB-PPO framework from multiple dimensions. Theoretically, Theorems 1–3 provide mathematical guarantees for the connection between the HJB equation and reinforcement learning, serving as theoretical beacons for algorithm design. On linear-quadratic simulated data, the neural network approximates the Riccati analytical solution with a mean squared error of 0.0006, verifying the feasibility of neural networks as numerical solvers. On real data, the PPO model's profits significantly exceed all strong static baselines. The policy distribution in Figure 2 reveals the key to its success: in high-dti (debt-to-

 doi:10.20944/preprints202604.0112.v1

income) regions, the model spontaneously assigns lower credit ratios, while in low-dti regions, it assigns higher ratios. This pattern indicates that PPO learns a risk-sensitive strategy consistent with the "feedback control" concept from optimal control theory—it does not pursue classification accuracy in one-shot decisions but continuously perceives states and adjusts actions throughout the loan lifecycle to maximize cumulative profit. This is the fundamental reason it outperforms all static baseline models and a direct manifestation of the HJB framework as the theoretical foundation for dynamic decision-making.

Further analysis of the profit distributions in Figure 3 and Table 5 reveals that PPO's profit distribution exhibits an "overall right shift with a heavier right tail": its median and upper quartile are significantly higher than those of all static models, and its high-profit samples are denser. This indicates that PPO's advantage does not rely on isolated extreme values but rather is consistently superior in the high-profit region. Notably, PPO is not optimal in risk-adjusted metrics such as the Sharpe ratio and Sortino ratio. This phenomenon stems from the essential difference in objectives between the two types of models: most static models have extreme predicted probabilities, leading their decisions to be almost always "full credit" or "complete rejection," thereby nearly eliminating negative returns at the cost of sacrificing substantial positive return opportunities. PPO, in contrast, prioritizes profit maximization and is willing to accept manageable downside risk in exchange for higher expected returns. Ablation experiments further support this: replacing PPO's nonlinear policy network with a linear mapping reduces average profit by 34.7%, demonstrating that nonlinear function approximation is critical for capturing the complex risk-return relationships in credit decisions. Overall, the HJB-PPO framework presented in this paper successfully transforms credit scoring from a static paradigm focused on single-period classification accuracy to a dynamic optimization paradigm targeting long-term profit. This transformation not only yields profit advantages empirically but also provides a viable pathway for incorporating more refined risk metrics or more complex collaborative decision-making scenarios into the same theoretical framework.

6. Conclusion and Future Directions

The core contribution of this paper lies in liberating the classic problem of credit scoring from the confines of static classification, repositioning it within the mathematical framework of stochastic optimal control, and providing a solution approach that combines theoretical rigor with practical effectiveness.

Theoretically, we established the mathematical equivalence between discrete MDPs and the HJB equation, proving the existence and uniqueness of the optimal value function; under the linear-quadratic assumption, we derived the closed-form algebraic Riccati equation, providing a theoretical benchmark for RL algorithms; and we proved that neural network value iteration constitutes an effective numerical scheme for solving the discrete HJB equation, with its convergence error separable into approximation and iteration errors. These results lay a solid mathematical foundation for applying deep reinforcement learning to financial risk control.

Empirically, using real LendingClub data from 2016 to 2018 with over 500,000 loans, we trained a PPO model as an instantiation of the HJB framework. Experimental results show that this model significantly outperforms static baselines such as logistic regression, random forest, and XGBoost in both average profit (1.5167) and total profit (786,700.4682). Policy analysis reveals that the model spontaneously learns a feedback control strategy that dynamically adjusts credit limits based on risk indicators such as debt-to-income ratio. Ablation experiments further confirm that the powerful function approximation capability provided by nonlinear neural networks enables the model to capture the complex nonlinear relationships in credit decisions, achieving a 34.7% profit increase. These findings collectively validate the rationality and effectiveness of modeling credit decisions as a dynamic control problem.

Looking ahead, while this research opens new pathways for dynamic credit decision-making, many questions remain for deeper exploration. First, the current framework focuses primarily on

 doi:10.20944/preprints202604.0112.v1

maximizing profit for individual loans; extending it to portfolio-level risk hedging that incorporates the aggregate risk of individual decisions into the optimization objective is a more practically significant direction. Second, multi-agent reinforcement learning offers possibilities for addressing collaborative decision problems involving multiple interacting parties, such as interbank credit markets and supply chain finance, with research by Wu et al. [25], Zhang et al. [26], and Peng et al. [27] in MAS cooperative control providing methodological references in this direction. Finally, while the reward function design in this paper preliminarily incorporates the trade-off between risk and return, future work could introduce more refined risk measures, such as Conditional Value at Risk (CVaR), embedding them into the HJB framework to better align with practical risk management needs. Through continued exploration in these directions, we aim to further unlock the theoretical potential and practical value of the HJB framework in financial risk control.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, visualization, L.J.; supervision, project administration, funding acquisition, R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant No. 72401144.

Data Availability Statement: The data used in this study are publicly available from Kaggle: the LendingClub loan data (https://www.kaggle.com/datasets/wordsforthewise/lending-club) and the Give Me Some Credit dataset (https://www.kaggle.com/datasets/lihxlhx/give-me-some-credit).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dastile, X., T. Celik, and M. Potsane, Statistical and machine learning models in credit scoring: A systematic literature survey. Applied Soft Computing, 2020. **91**.
2. Li, Y.H. and W.D. Chen, A Comparative Performance Assessment of Ensemble Learning for Credit Scoring. Mathematics, 2020. **8**(10).
3. Melo, L., Jr., et al., A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. Expert Systems with Applications, 2020. **152**.
4. Cubiles-De-La-Vega, M.D., et al., Improving the management of microfinance institutions by using credit scoring models based on Statistical Learning techniques. Expert Systems with Applications, 2013. **40**(17): p. 6910-6917.
5. Ito, K., C. Reisinger, and Y.F. Zhang, A Neural Network-Based Policy Iteration Algorithm with Global H2-Superlinear Convergence for Stochastic Games on Domains. Foundations of Computational Mathematics, 2021. **21**(2): p. 331-374.
6. Chen, Y.J., R. Calabrese, and B. Martin-Barragan, *Interpretable machine learning for imbalanced credit scoring datasets.* European Journal of Operational Research, 2024. **312**(1): p. 357-372.
7. Columba, F., M. Cugliari, and S. Di Virgilio, *Credit risk assessment with stacked machine learning*. 2026, Rome, Italy.
8. Ayari, H., P.R. Guetari, and P.N. Kraïem, *Machine learning powered financial credit scoring: a systematic literature review.* Artificial Intelligence Review, 2025. **59**(1).
9. Krasheninnikova, E., et al., *Reinforcement learning for pricing strategy optimization in the insurance industry.* Engineering Applications of Artificial Intelligence, 2019. **80**: p. 8-19.
10. Taghian, M., A. Asadi, and R. Safabakhsh, *Learning financial asset-specific trading rules via deep reinforcement learning.* Expert Systems with Applications, 2022. **195**.
11. Dang, T.K., et al., Machine Learning Based on Resampling Approaches and Deep Reinforcement Learning for Credit Card Fraud Detection Systems. Applied Sciences-Basel, 2021. **11**(21).
12. Paul, S., et al. An Automatic Deep Reinforcement Learning based Credit Scoring Model using Deep-Q Network for Classification of Customer Credit Requests. in 29th Annual IEEE International Symposium on Technology and Society (ISTAS). 2023. Swansea, WALES.

 doi:10.20944/preprints202604.0112.v1

13. Zhang, K., X.Q. Yang, and Y.H. Hu, Power penalty method for solving HJB equations arising from finance. Automatica, 2020. **111**.
14. Munos, R., A study of reinforcement learning in the continuous case by the means of viscosity solutions. Machine Learning, 2000. **40**(3): p. 265-299.
15. Mohammadi, H., et al., Convergence and Sample Complexity of Gradient Methods for the Model-Free Linear-Quadratic Regulator Problem. Ieee Transactions on Automatic Control, 2022. **67**(5): p. 2435-2450.
16. Lessmann, S., et al., Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 2015. **247**(1): p. 124-136.
17. Ala'raj, M. and M.F. Abbod, *Classifiers consensus system approach for credit scoring.* Knowledge-Based Systems, 2016. **104**: p. 89-105.
18. Perrusquía, A., Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning. Information Sciences, 2022. **595**: p. 364-377.
19. Li, N., et al., Stochastic Linear Quadratic Optimal Control Problem: A Reinforcement Learning Method. Ieee Transactions on Automatic Control, 2022. **67**(9): p. 5009-5016.
20. Shao, J.H., S. Mitra, and A. Karathanasopoulos, *Optimal feedback control of stock prices under credit risk dynamics.* Annals of Operations Research, 2022. **313**(2): p. 1285-1318.
21. Park, J., D. Yi, and S. Ji, A Novel Learning Rate Schedule in Optimization for Neural Networks and It's Convergence. Symmetry-Basel, 2020. **12**(4).
22. Zhang, H.S., C. Zhang, and W. Wu, Convergence of Batch Split-Complex Backpropagation Algorithm for Complex-Valued Neural Networks. Discrete Dynamics in Nature and Society, 2009. **2009**.
23. Jin, F., et al., Robust Optimal Reinsurance and Investment Problem Under Markov Switching via Actor-Critic Reinforcement Learning. Mathematics, 2025. **13**(21).
24. Hu, N.T., et al., Actor-Critic Trajectory Controller with Optimal Design for Nonlinear Robotic Systems. Cmc-Computers Materials & Continua, 2026. **87**(1).
25. Wu, Q.Y., Y.H. Wu, and Y.H. Wang, Integral Reinforcement-Learning-Based Optimal Containment Control for Partially Unknown Nonlinear Multiagent Systems. Entropy, 2023. **25**(2).
26. Zhang, Y., M. Chadli, and Z.R. Xiang, *Prescribed-Time Formation Control for a Class of Multiagent Systems via Fuzzy Reinforcement Learning*. Ieee Transactions on Fuzzy Systems, 2023. **31**(12): p. 4195-4204.
27. Peng, Z.N., et al., A novel optimal bipartite consensus control scheme for unknown multi-agent systems via model-free reinforcement learning. Applied Mathematics and Computation, 2020. **369**.
28. Bellman, R., *Dynamic Programming.* Science, 1957. **153**: p. 34 - 37.
29. Bardi, M. and I. Capuzzo-Dolcetta. Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. 1997.
30. Anderson, B.D.O. and J.B. Moore. Optimal control: linear quadratic methods. 1990.
31. He, S.P., et al., Reinforcement learning and adaptive optimization of a class of Markov jump systems with completely unknown dynamic information. Neural Computing & Applications, 2020. **32**(18): p. 14311-14320.