

# Learning Optimal Stopping Policies via High-Precision Teacher Labels

Lanqiong Lou

Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, USA

Email: llou3@jh.edu

Manuscript received July 21, 2026; accepted October 12, 2026; published October 27, 2026.

**Abstract**—This paper discusses the effect of labeling on classification-based optimal stopping. Labels generated naively from one continuation payoff realization contain a lot of pathwise noise since the label does not capture the conditional expectation of continuation. In order to resolve this problem, a High-Precision LSM Teacher generates labels for exercises or continues while a quadratic Logistic Regression Student learns from these labels. Comparing labels directly, it turns out that using Teacher labels instead of pathwise labels decreases the pricing error of the Student relative to the Teacher from 16.09% to 0.51%. Using SVMs for comparison, there appear no significant improvements in using more complicated classifiers.

**Keywords**—American option pricing, optimal stopping, least-squares monte Carlo, teacher–student learning, logistic regression

## I. INTRODUCTION

### A. Background

An American option is a type of financial derivative. It gives the holder the right to exercise the option before or at maturity. This means that investors can exercise the option at any time before maturity according to the evolution of the underlying asset price. Compared with a European option, an American option is more flexible because it can be exercised early. Options play an important role in risk management and portfolio hedging. Accurate option pricing is a core issue in the fields of financial mathematics and quantitative finance. Therefore, developing accurate and efficient numerical methods for American option pricing remains an important research topic.

### B. Limitation of LSM

Longstaff and Schwartz (2001) introduced the Least Squares Monte Carlo (LSM) method for pricing American options through simulation. At each possible exercise time, the option holder should compare the immediate exercise payoff with the conditional expected continuation payoff. Because the value of continuing to hold the option cannot be predicted directly, LSM estimates the discounted future cash flows by regressing them on functions of the current state variables. The fitted values obtained from the regression approximate the conditional expected continuation value. When the immediate exercise payoff is greater than or equal to the estimated continuation value, the option can be exercised; otherwise, the option holder continues to hold the option.

Even though this approach has many benefits, there is still some estimation error associated with it. The reason for this is that since the conditional continuation value is estimated from a limited number of simulation paths, the obtained estimate depends on the simulated sample, the choice of the

basis functions, and the specification of the regression model. The wrong exercise boundary will be obtained due to the error in the estimation of the continuation value.

Another problem that can be mentioned is related to the transformation of the LSM approach into the binary classification task. Naively designed labels can compare the immediate exercise payoff and the discounted future cash flow observed along a simulated path. While this cash flow serves as a regression target in the original LSM approach, it does not equal the conditional continuation value itself. Since each future payoff has randomness at the path level, such labels generated by the direct comparison could be noisy and might end up assigning different decisions to observations that share the same present state. The problem with label generation is the primary driving force behind the Teacher–Student framework presented here.

### C. Classification View

In the traditional Least Squares Monte Carlo (LSM) method, the optimal stopping strategy is determined by comparing the immediate exercise payoff with the conditional expectation of the continuation value. Therefore, the core task is to estimate the continuation value. However, in the Teacher–Student framework proposed in this paper, the Teacher already generates a stopping decision label for each training sample, where Exercise is labeled as 1 and Continue is labeled as 0. This means that the Student does not estimate the continuation value directly; instead, it predicts the stopping decision category for each sample based on the current state variables. After this process, the optimal stopping problem becomes a binary classification problem.

Logistic Regression is one of the most widely used binary classification models. It learns the relationship between input features and categories, outputs the probability that a sample belongs to a certain category, and completes the final classification using a decision boundary. Due to its advantages of a simple model structure, strong interpretability, and high computational efficiency, this paper chooses Logistic Regression as the Student classifier to learn the optimal stopping strategy generated by the Teacher.

### D. Main Idea

In the classification-based optimal stopping framework, the quality of the training labels plays a crucial role in determining the performance of the learned stopping policy. Although previous studies mainly focus on improving the flexibility and complexity of learning models, the effect of label construction has received relatively less attention.

The main issue with the original pathwise label design is that it compares the immediate exercise payoff with the realized future payoff from a single simulated path. However,

the theoretically relevant quantity for an optimal stopping decision is the conditional expected continuation value given the current state. Since a single future realization contains substantial Monte Carlo randomness, the generated labels may not accurately represent the true exercise-or-continue decision.

As a result, observations with similar current states may be assigned different labels because their future paths are different. This inconsistency adds noise to the classification problem and makes it more difficult for the Student to learn the stopping boundary accurately.

To reduce this noise, this paper uses a Teacher–Student framework. First, a High-Precision LSM Teacher is constructed using a large number of simulated paths to estimate the conditional continuation value. The stopping decisions produced by the Teacher are then used as training labels for the Student classifier. Rather than estimating the continuation value itself, the Student learns to reproduce the Teacher’s exercise decisions and approximate its stopping policy.

The proposed framework separates the high-precision estimation of the stopping policy from the subsequent learning process. This design allows us to investigate whether improving label quality can achieve better performance without relying on increasingly complex classification models.

### E. Contributions

First, this paper examines an important source of error in classification-based optimal stopping: the construction of training labels. Previous studies have mainly focused on improving learning algorithms, with less attention paid to label generation. The results show that labels based on realized pathwise continuation payoffs contain considerable Monte Carlo noise, since a realized payoff from a single path is not the same as the conditional expected continuation value.

Second, this paper develops a Teacher–Student framework that treats label generation and policy learning as two separate steps. A High-Precision LSM Teacher produces the stopping labels, and a simpler Student classifier learns the stopping policy from them.

Third, the results indicate that label quality may matter more than classifier complexity. With Teacher-generated labels, the Logistic Regression Student produces prices close to those obtained from more flexible Support Vector Machine classifiers. This finding suggests that using more reliable labels can improve pricing performance without requiring a more complicated learning model.

## II. RELATED LITERATURE

### A. Traditional Numerical Methods for American Option Pricing

American options allow the holder to exercise them at any permitted time before the expiration date, therefore, the pricing problem is essentially an optimal stopping problem. Cox *et al.* (1979) introduced the binomial tree model, which determines the optimal decision by comparing the immediate payoff with the discounted expected continuation value at each time point. This method has a clear structure, so it is widely used in American option pricing. However, as the

number of time steps or the number of state variables increases, the computational cost increases extremely quickly.

To improve feasibility in complex models, researchers started to combine simulation methods with regression methods. Carriere (1996) estimated the conditional expected continuation value using nonparametric regression and made the decision based on the comparison between the immediate payoff and the estimated continuation value. Tsitsiklis and Van Roy (1999) developed this method by using linear combinations of basis functions to approximate the value function in the optimal stopping problem and proved the convergence of the related approximation algorithms. These studies show that function approximation can reduce the computational difficulties associated with traditional dynamic programming in high-dimensional state spaces.

Broadie and Glasserman (1997) studied American option pricing using a simulation method and constructed price estimators with upward and downward biases. Their study shows that estimation errors in the stopping strategy directly affect the final pricing results and that Monte Carlo estimates with finite samples usually cannot completely avoid these biases. However, the computational cost of this method increases rapidly as the number of exercise times increases.

Longstaff and Schwartz (2001) proposed the Least Squares Monte Carlo (LSM) method, which estimates the conditional expected continuation value through regression and compares it with the immediate exercise payoff to determine whether to exercise or continue at each time point. LSM combines Monte Carlo simulation with least squares regression and is suitable for handling path-dependent and high-dimensional American options. Therefore, it has become one of the most widely used simulation-based pricing methods. However, the LSM results still depend on the number of simulated paths, the choice of basis functions, and the accuracy of the continuation-value estimation. Estimation errors in the continuation value may further lead to incorrect stopping decisions and pricing bias. This limitation provides the motivation for using a High-Precision LSM Teacher to generate more stable stopping-decision labels in this paper.

Mean-reverting processes have also been applied in financial modeling. Vasicek (1977) uses a mean-reverting Gaussian process to model short-term interest rates, providing a classical financial application of the Ornstein–Uhlenbeck structure. Carr and Itkin (2021) develop a semi-analytical method for pricing American and barrier options under a time-dependent Ornstein–Uhlenbeck process. Azze, D’Auria, and García-Portugués (2024) formulate American option exercise under a time-dependent Ornstein–Uhlenbeck process as an optimal stopping problem and study the corresponding exercise boundary. These studies support the use of the Ornstein–Uhlenbeck process in the numerical experiments of this paper.

Overall, existing studies mainly focus on improving option price approximation, continuation value estimation, or the characterization of the optimal exercise boundary. However, relatively little attention has been given to the quality of the stopping-decision labels used to train classification models. If these labels contain estimation errors or Monte Carlo noise, the classifier may learn these errors, which can affect the learned stopping policy and the final option price. This

limitation motivates the Teacher-generated label design proposed in this paper.

### B. Deep Learning and Neural Network Methods

With the development of machine learning, neural networks have increasingly been used to approximate the nonlinear relationship between American option prices and model parameters. Training these models can be computationally expensive. Once trained, however, they can produce option prices quickly, which makes them useful when a large number of options must be priced repeatedly.

Gaspar *et al.* (2020) applied neural networks to the pricing of American put options under empirical market settings and compared their results with prices obtained from the Longstaff–Schwartz LSM method. They found that deeper networks using a broader set of option and firm characteristics produced lower out-of-sample pricing errors. Their results demonstrate the ability of neural networks to capture complex nonlinear pricing relationships. However, their approach predicts the option price directly rather than learning the continuation value or the optimal exercise decision.

Kohler *et al.* (2010) used neural networks to estimate the continuation value in high-dimensional American option pricing problems. Similarly, Lapeyre and Lelong (2021) used neural network regression to estimate the continuation value of Bermudan options within a Regression Monte Carlo framework, replacing traditional basis functions with neural networks. These studies show that neural networks provide a flexible method for continuation value estimation, especially in high-dimensional problems. However, the stopping decision still depends on a comparison between the estimated continuation value and the immediate exercise payoff. In contrast, the Student in this paper does not estimate the continuation value but directly learns the stopping-decision labels generated by a High-Precision LSM Teacher.

Jang and Lee (2019) proposed a generative Bayesian neural network that combines the risk-neutral pricing structure and prior information in American index option pricing. The model showed greater stability in deep in-the-money and deep out-of-the-money regions, where market data are limited. This indicates that combining financial theory with neural networks can improve price prediction. However, this method also focuses on fitting market prices rather than learning the exercise-or-continue decision at each state.

Anderson and Ulrych (2023) first used a finite difference method to generate American option prices under the Heston model and then trained a deep neural network to learn the mapping between model parameters and option prices. After training, the neural network could approximate the prices generated by traditional numerical methods at high speed. This study adopts the idea of using a high-precision traditional model to generate training data and then using a learning model for fast approximation, which is somewhat similar to the Teacher–Student framework in this paper. However, its neural network learns the final option price rather than the optimal stopping strategy.

Overall, existing machine learning studies mainly focus on improving the learning model, while relatively little attention has been given to the design of the training labels. This paper

argues that the quality of the generated stopping-decision labels is equally important. Therefore, instead of developing a more complex learning model, this paper proposes a High-Precision LSM Teacher to generate more reliable stopping-decision labels and then trains a simple Student classifier to learn the optimal stopping strategy.

### C. Machine Learning Methods for Optimal Stopping

Gramacy and Ludkovski (2015) point out that the main goal of Regression Monte Carlo is not necessarily to obtain an accurate global estimate of the continuation value over the entire state space, but rather to correctly identify the boundary between the stopping and continuation regions. Instead, the exercise decision mainly depends on correctly determining the sign of the difference between the continuation value and the immediate exercise payoff. Therefore, the state space can be divided into a stopping region and a continuation region, with the boundary between these two regions defining the optimal stopping decision.

Based on this view, the accuracy of the estimated stopping boundary may be more important than the accuracy of the continuation-value estimate over the entire state space. Gramacy and Ludkovski (2015)] formulate the central step of Regression Monte Carlo as a contour-finding problem and use active learning to add observations near the stopping boundary. The results show that machine learning methods can focus the learning process on the decision boundary between the stopping and continuation regions.

Becker *et al.* (2019) proposed a related deep learning method that directly learns the optimal stopping rule from Monte Carlo samples. Their approach represents an optimal stopping time as a sequence of zero–one stopping decisions and learns these decisions recursively using neural networks. At each possible exercise date, the decision function selects either stopping or continuation based on the current state. The method uses a neural network to generate a stopping probability between zero and one, which is then converted into a final stopping decision using a threshold of 0.5. This shows that the optimal stopping problem can be represented and learned as a sequence of binary decisions, rather than being solved only through the estimation of the final option price.

Machine learning models have also been used within the LSM framework to improve the estimation of the continuation value. Djagba and Ndizihiwe (2024) compared several machine learning models, including Logistic Regression, Decision Trees, Random Forest, XGBoost, LightGBM, and  $k$ -nearest neighbors, within the LSM framework. These models were trained using the current state variables and the corresponding discounted option values at the next time step to estimate the continuation value. The predicted continuation value was then compared with the immediate exercise payoff to determine whether the option should be exercised or continued. The results also reflect the trade-off between model complexity and computational efficiency: Logistic Regression has relatively high computational efficiency, whereas Random Forest requires substantially more computational time.

Support Vector Machines (SVMs) are another classical supervised classification method. Cortes and Vapnik (1995) introduced the Support-Vector Network for binary

classification problems. The method maps the input variables into a high-dimensional feature space and constructs an optimal decision boundary by maximizing the margin between the two classes.

Through kernel functions, SVMs can also form nonlinear decision boundaries. Therefore, SVMs are used as comparison models for the Logistic Regression Student to examine whether more flexible classification boundaries can learn the stopping decisions generated by the High-Precision LSM Teacher more accurately.

#### D. Label Generating Gap

Existing machine learning studies mainly focus on directly predicting American option prices, estimating the continuation value, or using complex neural networks to learn an optimal stopping strategy. Most studies focus on improving the learning model, while relatively little attention has been given to how the stopping-decision labels are generated. In particular, labels generated from pathwise realized continuation payoffs may contain substantial Monte Carlo noise because they do not represent the conditional expected continuation value.

Therefore, the main innovation of the proposed framework does not lie in the development of a new classification algorithm. Instead, it lies in the Teacher-generated label design, where a High-Precision LSM Teacher generates more reliable stopping-decision labels before the learning process. The Student then directly learns these labels to predict the exercise-or-continue decision. SVMs are used as comparison models to examine whether greater model complexity provides additional improvements once high-quality labels are available.

### III. PROBLEM FORMULATION

Throughout this paper,  $N$  represents the number of time steps,  $M$  represents the number of simulated paths in one experiment,  $R$  represents the number of independent replications, and  $q$  denotes the polynomial degree.

#### A. European and American Put Options

The payoff function of a European put option is

$$g(S_T) = (K - S_T)^+ = \max(K - S_T, 0). \quad (1)$$

Under the risk-neutral pricing framework, the value of a European put option at the initial time is

$$P_0^E = \mathbb{E}^Q[e^{-rT} g(S_T)]. \quad (2)$$

Therefore, the European put option price can be written as

$$P_0^E = \mathbb{E}^Q[e^{-rT} (K - S_T)^+]. \quad (3)$$

Under the Black–Scholes framework, the price of a European put option has the closed-form:

$$P_0^E = K e^{-rT} \Phi(-d_2) - S_0 \Phi(-d_1), \quad (4)$$

where

$$d_1 = \frac{\ln(S_0/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}. \quad (5)$$

Here,  $P_0^E$  represents the European put option price,  $S_0$  is the initial asset price,  $K$  is the strike price,  $r$  is the risk-free interest rate,  $T$  is the time to maturity,  $\sigma$  is the volatility, and  $\Phi(\cdot)$  is the cumulative distribution function of the standard normal distribution.

Unlike a European put option, which can only be exercised at maturity, an American put option can be exercised at any permitted date before or on the expiration date. That is why the American put option generally does not have a closed-form Black-Scholes formula.

#### B. Underlying Asset-Price Models

This paper works on two models for the underlying asset price: the Geometric Brownian Motion model (GBM) and the Ornstein–Uhlenbeck model (OU).

##### 1) Geometric Brownian motion

Under the risk-neutral probability measure  $\mathbb{Q}$ , the Geometric Brownian Motion model is

$$dS_t = rS_t dt + \sigma S_t dW_t^Q, \quad (6)$$

where  $r$  is the risk-free interest rate,  $\sigma$  is the volatility, and  $W_t^Q$  is a standard Brownian motion under the risk-neutral probability measure.

On one time interval, the asset price is simulated using

$$S_{t_{n+1}} = S_{t_n} \exp\left[\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} Z_{n+1}\right], \quad (7)$$

where  $Z_{n+1} \sim N(0,1)$ .

##### 2) Ornstein–Uhlenbeck model

The Ornstein–Uhlenbeck model describes a mean-reverting asset-price process. Under the pricing measure used in this paper, the process is

$$dS_t = \kappa(\mu - S_t) dt + \sigma dW_t^Q, \quad (8)$$

where  $\mu$  is the long-run mean,  $\kappa$  is the speed of mean reversion,  $\sigma$  is the volatility parameter, and  $W_t^Q$  is a standard Brownian motion.

The exact discrete-time transition used to generate the simulated paths is

$$S_{t_{n+1}} = \mu + (S_{t_n} - \mu)e^{-\kappa\Delta t} + \sigma\sqrt{\frac{1 - e^{-2\kappa\Delta t}}{2\kappa}} Z_{n+1}, \quad (9)$$

where  $Z_{n+1} \sim N(0,1)$ .

#### C. American Put Option and Optimal Stopping

For American option, the set of possible exercise dates is

$$0 = t_0 < t_1 < \dots < t_N = T, \quad t_n = n\Delta t, \quad \Delta t = \frac{T}{N}. \quad (10)$$

That means the length is  $\Delta t$  for each time interval, which is the total time to maturity  $T$  is divided into  $N$  intervals.

The payoff of an American put option at exercise date  $t_n$  is

$$g(S_{t_n}) = (K - S_{t_n})^+ = \max(K - S_{t_n}, 0), \quad (11)$$

where  $S_{t_n}$  is the underlying asset price and  $K$  is the strike price.

The value of the American put option at the initial time is

$$V_0^A = \sup_{\tau \in \mathcal{T}} \mathbb{E}^Q[e^{-r\tau} g(S_{t_\tau})], \quad (12)$$

where  $\tau$  is a stopping-time index and  $\mathcal{T} \subseteq \{0, 1, \dots, N\}$  is the set of possible stopping-time point.

At exercise date  $t_n$ , the conditional expected continuation value is

$$C_n(S_{t_n}) = \mathbb{E}^Q[e^{-r\Delta t} V_{n+1}(S_{t_{n+1}}) | S_{t_n}]. \quad (13)$$

Therefore, the option value satisfies the dynamic programming equation

$$V_n(S_{t_n}) = \max\{g(S_{t_n}), C_n(S_{t_n})\}. \quad (14)$$

The corresponding optimal stopping rule is

$$\text{Exercise at } t_n \Leftrightarrow g(S_{t_n}) \geq C_n(S_{t_n}). \quad (15)$$

Thus, the main task in the traditional American option pricing problem is to estimate the conditional expected continuation value accurately.

#### D. Least Squares Monte Carlo Estimation

The LSM formulation introduced in this section provides the basis for constructing the High-Precision LSM Teacher in Section 4

Because the continuation value is generally unknown, the LSM method approximates it using polynomial regression:

$$\hat{C}_n(S_{t_n}) = \sum_{j=0}^q \hat{\beta}_{j,n} \phi_j(S_{t_n}), \quad (16)$$

where  $q$  is the polynomial degree,  $\phi_j(\cdot)$  is the  $j$ -th basis function, and  $\hat{\beta}_{j,n}$  is the estimated regression coefficient at exercise date  $t_n$ .

The estimation is processing on backward in time. At each exercise date, the discounted future cash flows of the in-the-money paths are regressed on functions of the current state variable. Then compared the estimated continuation value with the immediate exercise payoff to determine whether each path should be exercised or not.

#### E. Performance Measures

Suppose that the complete experiment is independently repeated  $R$  times. For the  $r$ -th replication, the pricing difference between the Student and Teacher is

$$D_r = \hat{V}_{0,r}^S - \hat{V}_{0,r}^T, \quad r = 1, \dots, R. \quad (17)$$

A negative value of  $D_r$  means that the Student value is underprices of the value of Teacher, and a positive value indicates overpricing.

To determine whether the Student in repeated simulations systematically overestimates or underestimates the option price, this paper also calculates the pricing bias. A single

pricing difference may be influenced by Monte Carlo simulation. Moreover, the average of the signed pricing errors across multiple simulations reflects the

overall direction of the error. A bias approaching zero indicates that the Student does not have an obvious systematic pricing difference. A positive bias means overestimation, while a negative bias means underestimation. Since the absolute error can only measure the size of the pricing error, not its direction, combining bias and absolute error can provide a more complete analysis of the Student's pricing performance.

The estimated pricing bias is

$$\text{Bias} = \frac{1}{R} \sum_{r=1}^R D_r = \frac{1}{R} \sum_{r=1}^R (\hat{V}_{0,r}^S - \hat{V}_{0,r}^T). \quad (18)$$

The relative bias is

$$\text{Relative Bias}(\%) = \frac{1}{R} \sum_{r=1}^R \frac{\hat{V}_{0,r}^S - \hat{V}_{0,r}^T}{\hat{V}_{0,r}^T} \times 100. \quad (19)$$

The mean absolute error is

$$\text{MAE} = \frac{1}{R} \sum_{r=1}^R |\hat{V}_{0,r}^S - \hat{V}_{0,r}^T|. \quad (20)$$

For a sequence of price estimates  $\hat{V}_{0,1}, \dots, \hat{V}_{0,R}$ , the standard deviation is

$$\text{SD} = \sqrt{\frac{1}{R-1} \sum_{r=1}^R (\hat{V}_{0,r} - \bar{V}_0)^2}, \quad (21)$$

where

$$\bar{V}_0 = \frac{1}{R} \sum_{r=1}^R \hat{V}_{0,r}. \quad (22)$$

The standard deviation measures the variability of the estimated option prices across the  $R$  independent replications. A smaller SD indicates that the pricing results are more stable across repeated simulations. The standard error measures the uncertainty of the average estimated option price, and it decreases as the number of replications increases.

The corresponding standard error is

$$\text{SE} = \frac{\text{SD}}{\sqrt{R}} \quad (23)$$

The relative error for the  $r$ -th replication is defined as

$$\text{Relative Error}_r = \frac{|\hat{V}_{0,r}^S - \hat{V}_{0,r}^T|}{\hat{V}_{0,r}^T}, \quad r = 1, \dots, R. \quad (24)$$

The mean relative error over  $R$  independent replications is

$$\text{Mean Relative Error} = \frac{1}{R} \sum_{r=1}^R \frac{|\hat{V}_{0,r}^S - \hat{V}_{0,r}^T|}{\hat{V}_{0,r}^T}. \quad (25)$$

## IV. TEACHER-STUDENT FRAMEWORK

This section introduces the process of how to construct the original pathwise labels, explains their main limitation, and presents the Teacher-Student framework. The framework uses a High-Precision LSM Teacher to generate stopping-decision labels and a quadratic Logistic Regression Student to learn the Teacher's stopping policy.

#### A. The Naive Label Design

The initial method transforms the optimal stopping problem into a binary classification problem. For each exercise date  $t_n$ , a label is assigned to each simulated path. If exercising immediately is better than continuing, the label is 1; otherwise, the label is 0.

For the  $i$ -th simulated path, let

$$\tau_i = \min\{m > n: L_{i,m} = 1\} \quad (26)$$

denote the future exercise-time index, where  $L_{i,m} = 1$  indicates that the option is exercised at time  $m$ . The future payoff realized along this individual path and discounted back to exercise date  $t_n$  is

$$Y_{\text{cont},n}^{(i)} = e^{-r\Delta t(\tau_i - n)} g(S_{t_{\tau_i}}^{(i)}). \quad (27)$$

The pathwise label is defined as

$$Y_n^{\text{old},(i)} = \mathbf{1}\{g(S_{t_n}^{(i)}) \geq Y_{\text{cont},n}^{(i)}\}, \quad (28)$$

where label 1 represents immediate exercise and label 0 represents continuation.

In other words, if the payoff from exercising immediately is greater than the payoff obtained by continuing along this particular path, the observation is labeled as exercise. Otherwise, it is labeled as continuation.

This method is easy to understand, but it has an important problem. It only uses the future payoff from a single path, but theoretically, the optimal decision should depend on the conditional expectation continuation value.

The theoretically relevant continuation value is

$$C_n(S_{t_n}) = \mathbb{E}^Q[Y_{\text{cont},n} | S_{t_n}]. \quad (29)$$

Thus, the original label compares the immediate exercise payoff with a future payoff along a single simulated path rather than with the conditional expected continuation value. As a result, observations with similar current states may receive different labels because of future path-level randomness. Therefore, the original labels contain a large amount of Monte Carlo noise and may not accurately represent the optimal stopping decision.

#### B. High-Precision LSM Teacher

To reduce the Monte Carlo randomness in the original labels, this paper constructs a High-Precision Least Squares Monte Carlo model as the Teacher. The Teacher uses a large number of independently simulated paths and a flexible polynomial basis to estimate the conditional expected continuation value.

The state variable is normalized as

$$z_n = \frac{S_{t_n} - K}{K}. \quad (30)$$

The estimation is performed backward in time. At each exercise date, the discounted future cash flows of the in-the-money paths are regressed on polynomial functions of the normalized current state variable.

The continuation value estimated by the Teacher is

$$\hat{C}_n^{\text{HP}}(S_{t_n}) = \sum_{j=0}^{q_T} \hat{\beta}_{j,n}^T z_n^j, \quad q_T = 8, \quad (31)$$

where  $q_T$  denotes the polynomial degree used by the Teacher. Equivalently,

$$\hat{C}_n^{\text{HP}}(S_{t_n}) = \hat{\beta}_{0,n}^T + \hat{\beta}_{1,n}^T z_n + \hat{\beta}_{2,n}^T z_n^2 + \dots + \hat{\beta}_{8,n}^T z_n^8. \quad (32)$$

In this paper, the High-Precision LSM Teacher is constructed using 100,000 simulated paths and an eighth-degree polynomial basis. Compared with the future payoff along a single simulated path, the continuation value estimated by the Teacher provides a more stable approximation of the conditional expected continuation value.

#### C. Teacher Label Generation

After construct the High-Precision LSM Teacher, it is applied to an independent set of training paths to generate new stopping-decision labels. The continuation-value function is estimated only once using the Teacher paths and is then applied to independent training observations to generate the Teacher labels.

The Teacher label for the  $i$ -th training path at exercise date  $t_n$  is

$$Y_n^{\text{T},(i)} = \mathbf{1}\{g(S_{t_n}^{(i)}) \geq \hat{C}_n^{\text{HP}}(S_{t_n}^{(i)})\}. \quad (33)$$

Equivalently,

$$Y_n^{\text{T},(i)} = \begin{cases} 1, & g(S_{t_n}^{(i)}) \geq \hat{C}_n^{\text{HP}}(S_{t_n}^{(i)}), \\ 0, & g(S_{t_n}^{(i)}) < \hat{C}_n^{\text{HP}}(S_{t_n}^{(i)}). \end{cases} \quad (34)$$

A label of 1 represents immediate exercise, whereas a label of 0 represents continuation. Unlike the original pathwise label, the Teacher label does not depend on the future payoff based on one particular simulated path. Instead, it is based on the conditional expected continuation value estimated by the High-Precision LSM Teacher.

Therefore, the Teacher-generated label design reduces the effect of Monte Carlo noise in the classification problem and provides more consistent labels for training the Student classifier.

#### D. Logistic Regression Student

After generating the Teacher labels have, the optimal stopping problem becomes a binary classification problem. The Student no longer estimates the continuation value. Instead, it directly predicts whether the option should be exercised or not based on the current state.

At each exercise date  $t_n$ , the Student uses quadratic Logistic Regression to estimate the conditional probability of exercise:

$$\begin{aligned} \hat{p}_n(S_{t_n}) &= \mathbb{P}(Y_n^{\text{T}} = 1 | z_n) \\ &= \frac{1}{1 + \exp[-(\beta_{0,n}^S + \beta_{1,n}^S z_n + \beta_{2,n}^S z_n^2)]}. \end{aligned} \quad (35)$$

The Student coefficients are estimated by maximizing the binary log-likelihood function

$$\begin{aligned} \ell_n(\beta_n^S) &= \sum_{i=1}^{M_{\text{train}}} [Y_n^{T,(i)} \log(\hat{p}_{i,n}) \\ &+ (1 - Y_n^{T,(i)}) \log(1 - \hat{p}_{i,n})], \end{aligned} \quad (36)$$

where

$$\hat{p}_{i,n} = \hat{p}_n(S_{t_n}^{(i)}). \quad (37)$$

The Student stopping decision is

$$\hat{Y}_n^{S,(i)} = \begin{cases} 1, & \hat{p}_n(S_{t_n}^{(i)}) \geq 0.5, \\ 0, & \hat{p}_n(S_{t_n}^{(i)}) < 0.5. \end{cases} \quad (38)$$

Thus, the Student directly learns the exercise or continue decisions generated by the Teacher. The threshold of 0.5 converts the estimated probability into the final binary stopping decision.

#### E. Out-of-Sample Pricing

To evaluate the pricing performance of the new method, this paper uses three independent sets of simulated paths: Teacher paths, training paths, and testing paths. Each dataset is generated independently and is used for a different purpose.

First, the Teacher paths are used to construct the High-Precision LSM Teacher. The Teacher estimates the conditional expected continuation value and determines the stopping decision at each exercise date.

Second, an independent set of training paths, also called Student training paths, is simulated. The trained Teacher is applied to these paths to generate the Teacher labels. These labeled observations are then used to train the Student classifier to learn the Teacher's stopping policy.

Finally, another independent set of testing paths is generated for out-of-sample pricing. The trained Teacher and Student are separately applied to the same previously unseen testing paths. This allows the pricing difference to reflect differences between their stopping policies instead of differences between independently simulated test samples.

The three datasets can be summarized as follows:

$$\begin{aligned} &\text{Teacher Paths} \\ &\rightarrow \text{High-Precision LSM Teacher,} \end{aligned} \quad (39)$$

$$\begin{aligned} &\text{Training Paths} \rightarrow \text{Teacher Labels} \\ &\rightarrow \text{Student Training,} \end{aligned} \quad (40)$$

and

$$\begin{aligned} &\text{Testing Paths} \\ &\rightarrow \text{Teacher and Student Out-of-Sample Pricing.} \end{aligned} \quad (41)$$

For the  $i$ -th testing path, the Student exercise-time index is

$$\hat{t}_i^S = \inf\{n \in \{0, \dots, N\} : \hat{Y}_n^{S,(i)} = 1\}. \quad (42)$$

If the Student does not select early exercise before maturity, then the exercise-time index is set equal to  $N$ . Using  $M_{\text{test}}$  independent testing paths, the Student price is estimated by

$$\hat{V}_0^S = \frac{1}{M_{\text{test}}} \sum_{i=1}^{M_{\text{test}}} e^{-r\hat{t}_i^S \Delta t} g\left(S_{\hat{t}_i^S}^{(i)}\right). \quad (43)$$

Similarly, let  $\hat{t}_i^T$  denote the first exercise-time index selected by the Teacher on testing path  $i$ . The Teacher price

is

$$\hat{V}_0^T = \frac{1}{M_{\text{test}}} \sum_{i=1}^{M_{\text{test}}} e^{-r\hat{t}_i^T \Delta t} g\left(S_{\hat{t}_i^T}^{(i)}\right). \quad (44)$$

Since the Teacher, training, and testing paths are simulated independently, the pricing results are not evaluated using the same paths employed to construct the Teacher or train the Student. This sample-splitting design reduces the risk of overfitting and provides a more reliable evaluation of the Student's out-of-sample pricing performance.

#### F. Complexity analysis

This subsection distinguishes the computational cost of constructing the Teacher and training the Student from the cost of applying the fitted stopping policies to independent testing paths. This distinction corresponds to the three-sample design introduced in the preceding subsection: Teacher paths are used to estimate the continuation-value functions, Student-training paths are used for label generation and classifier fitting, and testing paths are used for out-of-sample pricing.

Let  $N$  denote the number of time intervals, so that the number of backward regression dates is of order  $N$ . Let  $M_T$ ,  $M_S$ , and  $M_{\text{test}}$  denote the numbers of Teacher, Student-training, and testing paths, respectively. At date  $t_n$ , let  $m_{T,n} \leq M_T$ ,  $m_{S,n} \leq M_S$ , and  $m_{\text{test},n} \leq M_{\text{test}}$  denote the numbers of observations that are actually processed. These date-specific counts allow for the fact that the LSM regression uses only eligible, typically in-the-money and alive, paths and that some testing paths may have stopped at earlier dates.

The Teacher uses a polynomial of degree  $q_T$  and therefore has  $p_T = q_T + 1$  regression coefficients in the one-dimensional setting of this paper. The quadratic Logistic Regression Student uses  $\phi(z) = (1, z, z^2)'$  and hence has  $p_S = 3$  explicit features, including the intercept. For the optimization-based classifiers, let  $I_{\text{LR}}$  and  $I_{\text{SVM}}$  denote the respective iteration counts. These counts depend on the solver, regularization, stopping tolerance, and conditioning of the problem and therefore need not be equal.

##### 1) High-precision LSM teacher

At exercise date  $t_n$ , the Teacher solves a least-squares problem with design matrix  $X_n \in \mathbb{R}^{m_{T,n} \times p_T}$ . Its  $i$ th row is  $\phi_T(z_n^{(i)})' = (1, z_n^{(i)}, \dots, (z_n^{(i)})^{q_T})$ , and the response vector contains the discounted future cash flows. The individual operations at date  $t_n$  can be counted as

$$\begin{aligned} \text{constructing } X_n &: \mathcal{O}(m_{T,n} p_T), \\ \text{forming } X_n' X_n &: \mathcal{O}(m_{T,n} p_T^2), \\ \text{forming } X_n' Y_n &: \mathcal{O}(m_{T,n} p_T), \\ \text{solving the } p_T \times p_T \text{ linear system} &: \mathcal{O}(p_T^3). \end{aligned} \quad (45)$$

The quadratic term arises because each of the  $p_T^2$  entries of  $X_n' X_n$  is an inner product involving  $m_{T,n}$  observations. The lower-order  $\mathcal{O}(m_{T,n} p_T)$  terms are dominated when  $p_T > 1$ . Hence, a dense least-squares calculation at one date has cost

$$\mathcal{O}(m_{T,n} p_T^2 + p_T^3). \quad (46)$$

The preceding decomposition uses the normal equations only to count the operations. A numerically stable

implementation should solve the least-squares problem by QR factorization, SVD, or an equivalent routine rather than by explicitly computing  $(X_n'X_n)^{-1}$ . For a tall dense matrix, these routines have the same leading order  $\mathcal{O}(m_{T,n}p_T^2)$  when  $m_{T,n} \gg p_T$ .

Consequently, the total Teacher regression cost is

$$\mathcal{O}\left(\sum_n [m_{T,n}p_T^2 + p_T^3]\right) \subseteq \mathcal{O}(NM_T p_T^2 + Np_T^3). \quad (47)$$

Because the empirical specification has  $q_T = 8$  and  $M_T$  is much larger than  $p_T$ , this upper bound is commonly summarized as  $\mathcal{O}(NM_T p_T^2)$ . Simulating the Teacher paths and updating their discounted cash flows add  $\mathcal{O}(NM_T)$  operations. If all simulated states are retained for backward induction, path storage is  $\mathcal{O}(NM_T)$ ; the date-specific design matrix requires at most  $\mathcal{O}(M_T p_T)$  additional working memory.

## 2) Teacher label generation

After the Teacher has been fitted, its continuation polynomial is evaluated on the independent Student-training observations. Directly computing all powers of  $z$  is unnecessary. Horner's rule rewrites the continuation estimate as

$$\hat{C}_n^{HP}(z) = \hat{\beta}_{0,n} + z(\hat{\beta}_{1,n} + z[\hat{\beta}_{2,n} + \dots + z\hat{\beta}_{q_T,n}]). \quad (48)$$

This evaluation uses  $q_T$  multiplications and approximately  $q_T$  additions, and therefore costs  $\mathcal{O}(p_T)$  per observation. Comparing the resulting continuation estimate with the immediate exercise payoff takes constant time. The work at date  $t_n$  is consequently  $\mathcal{O}(m_{S,n}p_T)$ , and summing over all dates gives

$$\mathcal{O}\left(p_T \sum_n m_{S,n}\right) \subseteq \mathcal{O}(NM_S p_T). \quad (49)$$

Simulation of the Student-training paths contributes an additional  $\mathcal{O}(NM_S)$  term. Together with Teacher construction and Student fitting, these operations constitute the model-construction and training cost. They must be repeated if a change in the contract or state-process parameters requires the stopping policy to be re-estimated.

## 3) Logistic regression student

At each exercise date, the quadratic Logistic Regression Student estimates a three-dimensional coefficient vector. Let  $\Phi_n \in \mathbb{R}^{m_{S,n} \times p_S}$  denote the Student feature matrix and let  $Y_n^T$  collect the Teacher labels. One first-order iteration performs the following main operations:

$$\begin{aligned} \text{linear scores } \eta_n = \Phi_n \beta_n &: \mathcal{O}(m_{S,n} p_S), \\ \text{probabilities } p_n = \sigma(\eta_n) &: \mathcal{O}(m_{S,n}), \\ \text{log-likelihood evaluation} &: \mathcal{O}(m_{S,n}), \\ \text{gradient } \Phi_n'(p_n - Y_n^T) &: \mathcal{O}(m_{S,n} p_S). \end{aligned} \quad (50)$$

Thus, one objective-and-gradient evaluation costs  $\mathcal{O}(m_{S,n} p_S)$ . Under a first-order or quasi-Newton implementation, the total training cost is approximately

$$\mathcal{O}\left(I_{LR} p_S \sum_n m_{S,n}\right) \subseteq \mathcal{O}(NI_{LR} M_S p_S). \quad (51)$$

This expression is solver-specific. For example, a full

Newton or IRLS step also constructs the Hessian

$$H_n = \Phi_n' W_n \Phi_n, \quad (52)$$

which costs  $\mathcal{O}(m_{S,n} p_S^2)$ , and solves a  $p_S \times p_S$  system at cost  $\mathcal{O}(p_S^3)$ . Its per-iteration cost is therefore  $\mathcal{O}(m_{S,n} p_S^2 + p_S^3)$ . The implementation should report the solver, regularization, convergence tolerance, and maximum iteration count. Regularization is also relevant because the Teacher-generated labels can be nearly separable, in which case an unregularized Logistic Regression maximum likelihood estimate may fail to be finite.

## 4) Support vector machine comparisons

For a Linear SVM trained by a first-order primal method, one pass over the data requires computing  $m_{S,n}$  inner products of dimension  $p_S$ , identifying the margin violations, and updating the coefficients. The inner products and updates cost  $\mathcal{O}(m_{S,n} p_S)$ , while checking the hinge losses costs  $\mathcal{O}(m_{S,n})$ . Under this solver assumption, the overall training cost is approximately

$$\mathcal{O}\left(I_{SVM} p_S \sum_n m_{S,n}\right). \quad (53)$$

For a kernel SVM, including the polynomial-kernel and RBF specifications used in the numerical comparison, the symmetric kernel matrix at date  $t_n$  requires

$$\frac{m_{S,n}(m_{S,n} - 1)}{2} + m_{S,n} \quad (54)$$

pairwise evaluations when its diagonal is included. If one kernel evaluation costs  $\kappa(d)$  for a  $d$ -dimensional raw state, the construction cost at that date is therefore  $\mathcal{O}(m_{S,n}^2 \kappa(d))$ .

Across all exercise dates, this becomes

$$\mathcal{O}\left(\kappa(d) \sum_n m_{S,n}^2\right) \quad (55)$$

Storing the matrix requires  $m_{S,n}^2$  numerical entries and hence up to  $\mathcal{O}(\max_n m_{S,n}^2)$  working memory. This is the cost of kernel-matrix construction, not the full training complexity. The subsequent quadratic-programming cost is solver-dependent and can scale between roughly quadratic and cubic in the number of training observations in unfavorable cases. For this reason, runtime, peak memory, the solver tolerance, and the number of support vectors provide more informative empirical comparisons than a single solver-independent Big-O expression.

## 5) Out-of-sample policy evaluation

After the models have been fitted, the Teacher and Student stopping policies are separately applied to the same independent testing paths, as described in the out-of-sample pricing procedure. For one state, Horner's rule evaluates the Teacher's polynomial using  $q_T$  multiplications and approximately  $q_T$  additions, giving  $\mathcal{O}(p_T)$  operations. On the testing paths, its total evaluation cost is

$$\mathcal{O}\left(p_T \sum_n m_{\text{test},n}\right) \subseteq \mathcal{O}(NM_{\text{test}} p_T). \quad (56)$$

For the Logistic Regression Student, computing the

sigmoid is unnecessary for the binary stopping decision because

$$\hat{p}_n(S) \geq 1/2 \Leftrightarrow \hat{\beta}_{0,n} + \hat{\beta}_{1,n}z + \hat{\beta}_{2,n}z^2 \geq 0. \quad (57)$$

The Student can therefore apply its policy by evaluating a quadratic score, which requires only a constant number of multiplications and additions because  $p_S = 3$ . More generally, it costs  $\mathcal{O}(p_S)$  per queried state and

$$\mathcal{O}\left(p_S \sum_n m_{\text{test},n}\right) \subseteq \mathcal{O}(NM_{\text{test}}p_S). \quad (58)$$

A Linear SVM likewise evaluates one  $p_S$ -dimensional inner product per state. In contrast, if the kernel SVM at date  $t_n$  retains  $s_n$  support vectors, it must compute  $s_n$  kernel values and form their weighted sum. Its evaluation cost is therefore

$$\mathcal{O}\left(\kappa(d) \sum_n m_{\text{test},n} s_n\right), \quad (59)$$

which can be substantially larger when many support vectors are retained.

#### 6) Summary and interpretation

Table 1 summarizes the principal costs. The expressions use the date-specific sample sizes; replacing every  $m_{\cdot,n}$  by its corresponding maximum gives the simpler worst-case bounds stated above.

The proposed framework therefore adds model-construction and training costs through the High-Precision LSM Teacher, Teacher-label generation, and Student fitting. These costs should not be interpreted as evidence that the Student is computationally preferable to the Teacher in every application. In the present one-dimensional setting, both the eighth-degree Teacher and the quadratic Logistic Student have out-of-sample evaluation costs that are linear in the number of processed testing observations. The Student therefore does not improve the asymptotic order relative to the Teacher. Its computational benefits are a simpler decision rule, interpretability, and a smaller evaluation constant.

Table 1. Computational costs of the Teacher-Student framework

| Component                   | Time complexity                                                              | Principal memory cost                        |
|-----------------------------|------------------------------------------------------------------------------|----------------------------------------------|
| Teacher regression          | $\mathcal{O}\left(\sum_n [m_{T,n}p_T^2 + p_T^3]\right)$                      | $\mathcal{O}(NM_T + M_T p_T)$                |
| Teacher label generation    | $\mathcal{O}\left(p_T \sum_n m_{S,n}\right)$                                 | $\mathcal{O}(NM_S)$ if all labels are stored |
| Logistic Student training   | $\mathcal{O}(I_{LR} p_S \sum_n m_{S,n})$ for a first-order method            | $\mathcal{O}(M_S p_S)$                       |
| Linear SVM training         | $\mathcal{O}(I_{SVM} p_S \sum_n m_{S,n})$ under the stated solver assumption | Solver-dependent                             |
| Kernel-matrix construction  | $\mathcal{O}\left(\kappa(d) \sum_n m_{S,n}^2\right)$                         | $\mathcal{O}(\max_n m_{S,n}^2)$              |
| Teacher policy evaluation   | $\mathcal{O}\left(p_T \sum_n m_{\text{test},n}\right)$                       | $\mathcal{O}(p_T)$ beyond the paths          |
| Logistic Student evaluation | $\mathcal{O}\left(p_S \sum_n m_{\text{test},n}\right)$                       | $\mathcal{O}(p_S)$ beyond the paths          |
| Kernel SVM evaluation       | $\mathcal{O}\left(\kappa(d) \sum_n m_{\text{test},n} s_n\right)$             | $\mathcal{O}(\max_n s_n d)$                  |

The difference is more substantial relative to kernel classifiers. Their kernel-matrix memory requirement is quadratic in the number of training observations, their optimization cost is solver-dependent, and their out-of-sample evaluation cost grows with the number of support vectors. These comparisons support quadratic Logistic Regression as a parsimonious Student for the numerical analysis, while avoiding an overstatement of its computational advantage over the fitted Teacher.

#### G. Extensions

Several ways of extending the proposed Teacher-Student framework come to mind. For instance, it would be quite natural to extend the framework to an optimal stopping problem with multidimensional states. At the moment, the stopping rule depends solely on the normalized underlying state  $z_n$ . However, one could more generally consider a  $d$ -dimensional state vector

$$\mathbf{X}_n = (X_{1,n}, \dots, X_{d,n})', \quad (60)$$

containing the underlying price, volatility, interest rate or any path-dependent variable. The continuation value can be estimated by the High-Precision LSM Teacher as

$$\hat{C}_n^{HP}(\mathbf{X}_n) = \sum_{j=1}^{p_T} \hat{\beta}_{j,n}^T \phi_j(\mathbf{X}_n), \quad (61)$$

where  $\phi_j(\mathbf{X}_n)$  stands for some basis functions. The Teacher label in this case would take the form

$$Y_n^T = \mathbf{1}\{g(\mathbf{X}_n) \geq \hat{C}_n^{HP}(\mathbf{X}_n)\}. \quad (62)$$

The Stopping Decision can thus be learned using the functions that use the same state variables. Such an extension retains the core elements of the proposed architecture, except that the number of basis functions and the computational expense can grow very quickly with the state space dimension.

The second extension focuses on observations near the exercise boundary. The Teacher decision margin is defined as

$$D_n(\mathbf{X}_n) = g(\mathbf{X}_n) - \hat{C}_n^{HP}(\mathbf{X}_n). \quad (63)$$

If  $|D_n(\mathbf{X}_n)|$  is small, the exercise payoff is nearly equal to the continuation value. The corresponding label is more sensitive to simulation and regression errors. To account for this uncertainty, observations near the exercise boundary can be given smaller weights when training the Student, while observations farther from the boundary receive larger weights. For example, a weighted Logistic Regression Student can be trained by maximizing the following objective function:

$$\begin{aligned} & \ell_n(\boldsymbol{\beta}_n^S) \\ &= \sum_{i=1}^{M_{\text{train}}} w_{i,n} [Y_n^{T,(i)} \log(\hat{p}_{i,n}) \\ & \quad + (1 - Y_n^{T,(i)}) \log(1 - \hat{p}_{i,n})], \end{aligned} \quad (64)$$

where  $w_{i,n}$  is an increasing function of  $|D_n(\mathbf{X}_n^{(i)})|$ . Thus, the Student can take into account the heterogeneity of the Teacher-label reliability rather than treating all binary labels equally.

These extensions maintain the separation between Teacher construction, label generation, Student training, and out-of-sample pricing described in this section. Further numerical experiments are needed to determine whether the resulting improvement in the stopping rule is sufficient to justify the additional model complexity.

## V. NUMERICAL EXPERIMENTS

This section evaluates the Teacher–Student framework through several numerical experiments. It first checks the pricing implementation under the GBM model using the Binomial Tree method as a benchmark. It then illustrates the noise in pathwise labels and compares the two label-generation methods. The remaining experiments examine the robustness of the results across different OU parameter settings, Teacher simulation sizes, and repeated Monte Carlo samples. Finally, Logistic Regression is compared with several more flexible classifiers.

### A. Implementation Validation under GBM

Before investigating the “Teacher–Student” framework based on the OU model, this paper first checks the implementation of the logistic regression LSM method under the GBM model, using the binomial tree method as a standard numerical benchmark.



Fig. 1. Pricing bias of the logistic regression LSM method under different volatility levels.

Fig. 1 presents the signed pricing difference between the Logistic Regression LSM price and the Binomial Tree price. For each volatility level, the pricing difference is calculated as

$$\text{Difference}(\sigma) = \hat{V}^{\text{LSM}}(\sigma) - V^{\text{B}}(\sigma), \quad (65)$$

where uses the same value of  $\sigma$  in both methods. The horizontal red line represents zero pricing difference. A positive value indicates overpricing relative to the Binomial Tree benchmark, and a negative value indicates underpricing. Then consider the pricing difference varies across the volatility levels, with the largest negative difference occurring at approximately  $\sigma = 0.25$ , where the difference is close to  $-0.1$ .

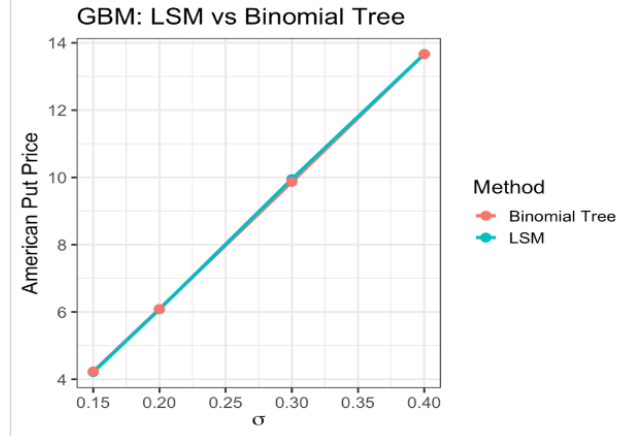


Fig. 2. Comparison of American put option prices obtained from the LSM and Binomial Tree methods under the GBM model.

Fig. 2 compares the corresponding American put option prices. The red line represents the Binomial Tree price, while the blue line represents the Logistic Regression LSM price. These two curves remain close across various volatility levels, indicating that the LSM prices align closely with the binomial tree benchmark prices. Furthermore, both methods show that the price of an American put option increases as volatility rises. This pattern is consistent with option pricing theory because higher volatility increases the possibility of a substantial decline in the underlying asset price and therefore increases the potential payoff of the put option.

### Repeated GBM simulations

Fig. 3 presents the distributions of the signed pricing differences using 100, 500, and 1000 Monte Carlo replications. Almost all observations lie below zero, that means the Logistic Regression LSM method generally underprices the Binomial Tree benchmark in this experiment.

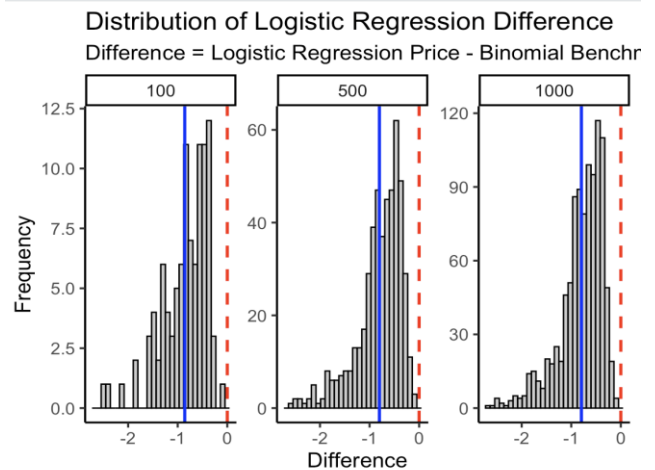


Fig. 3. Distribution of GBM pricing differences using the first 100, 500, and 1000 Monte Carlo replications.

Fig. 4 presents the corresponding kernel density estimates. In all three cases, most of the density lies below zero. The blue vertical line represents mean pricing difference which remains similar as the number of replications increases. Thus, using more replications produces a more stable description of the pricing-difference distribution, but does not by itself eliminate the observed underpricing.

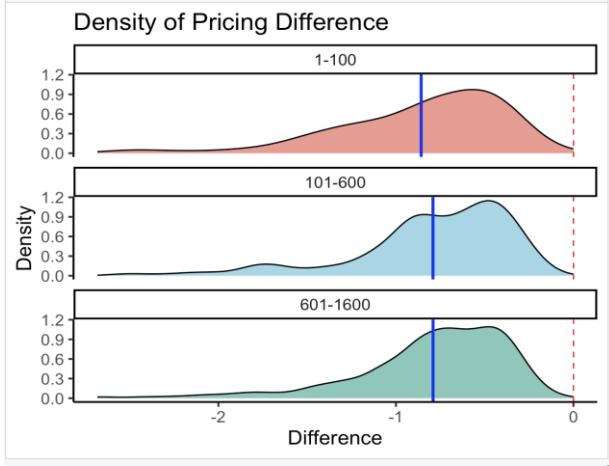


Fig. 4. Kernel density estimates of GBM pricing differences using the first 100, 500, and 1000 Monte Carlo replications.

### B. Why Are Pathwise Labels Noisy?

The pathwise label noise discussed in the previous section is provided with a numerical illustration by selecting two observations from the simulated Student training data. They occur at the same exercise date and have the same current state to the reported precision,

$$S_{t_n} = 99.79416, \quad (66)$$

which produces the same immediate exercise payoff,

$$g(S_{t_n}) = 0.2058401. \quad (67)$$

For the first observation, the realized continuation value is

$$Y_{\text{cont},n}^{(1)} = 0. \quad (68)$$

Since  $g(S_{t_n}) > Y_{\text{cont},n}^{(1)}$ , its original pathwise label is

$$Y_n^{\text{old},(1)} = 1. \quad (69)$$

For the second observation, the continuation value is

$$Y_{\text{cont},n}^{(2)} = 1.996869. \quad (70)$$

Since  $g(S_{t_n}) < Y_{\text{cont},n}^{(2)}$ , its original pathwise label is

$$Y_n^{\text{old},(2)} = 0. \quad (71)$$

Thus, observations with essentially identical current states receive different original labels because their future cash flows are different. By contrast, the High-Precision LSM Teacher estimates the same conditional continuation value for both observations:

$$\hat{C}_n^{\text{HP}}(S_{t_n}) = 3.383795. \quad (72)$$

Because

$$g(S_{t_n}) < \hat{C}_n^{\text{HP}}(S_{t_n}), \quad (73)$$

both observations receive the same Teacher label,

$$Y_n^{\text{T},(1)} = Y_n^{\text{T},(2)} = 0. \quad (74)$$

This example illustrates the central difference between the two label designs. The original label depends on one future payoff an individual path, where the Teacher-generated label is based on an estimate of the conditional expected continuation value. The Teacher produces more consistent stopping decisions for observations with the same current state.

### C. Effect of Teacher-Generated Labels

#### 1) Baseline student–teacher comparison

Table 2 presents the baseline pricing comparison between the High-Precision LSM Teacher and the Logistic Regression Student. Both methods are evaluated using the same independent testing paths.

Table 2. Baseline pricing comparison between the high-precision LSM teacher and the logistic regression student

| Method                      | Price    | Standard Error | Difference from Teacher | Relative Difference (%) |
|-----------------------------|----------|----------------|-------------------------|-------------------------|
| High-Precision LSM Teacher  | 5.026379 | 0.1136384      | 0.000000                | 0.0000                  |
| Logistic Regression Student | 5.056910 | 0.1166861      | 0.030530                | 0.6074                  |

Table 2 compares the baseline pricing results of the High-Precision LSM Teacher and the Logistic Regression Student. The Teacher produces an estimated option price of 5.026379, while the Student produces an estimated price of 5.056910. Therefore, the Student–Teacher pricing difference is 0.030530, corresponding to a signed relative difference of 0.6074%. The positive sign indicates that the Student slightly overprices the option relative to the Teacher in this baseline experiment. However, the relative difference remains below 1%, showing that the Logistic Regression Student closely approximates the pricing outcome generated by the High-Precision LSM Teacher. The reported standard errors are also similar for the two methods, suggesting comparable levels of Monte Carlo variability.

$$\hat{V}_0^{\text{T}} = 5.026379, \quad (75)$$

while the Logistic Regression Student produces

$$\hat{V}_0^{\text{S}} = 5.056910. \quad (76)$$

The absolute pricing difference is therefore

$$|\hat{V}_0^{\text{S}} - \hat{V}_0^{\text{T}}| = 0.030531. \quad (77)$$

The signed relative difference is defined as

$$\text{Relative Difference} = \frac{\hat{V}_0^{\text{S}} - \hat{V}_0^{\text{T}}}{\hat{V}_0^{\text{T}}}. \quad (78)$$

For the baseline experiment,

$$\begin{aligned} & \text{Relative Difference} \\ &= \frac{5.056910 - 5.026379}{5.026379} = 0.006074 \quad (79) \\ & \approx 0.607\%. \end{aligned}$$

The positive sign indicates that the Student slightly overprices the option relative to the Teacher in this experiment. Moreover, the relative difference remains below 1%, indicating that the quadratic Logistic Regression Student produces a pricing result which close to the more computationally intensive Teacher policy.

The Teacher has a standard error of 0.1136, while the Student has a standard error of 0.1167. Their similar magnitudes indicate that the two stopping policies generate discounted cash flows with comparable Monte Carlo variation in the baseline experiment.

## 2) Direct comparison of label designs

The next experiment isolates the effect of label construction. The classification model is quadratic Logistic Regression under both label designs; only changes the training labels. The following methods are compared:

- 1) Logistic Regression trained using original pathwise labels;
- 2) Logistic Regression trained using Teacher-generated labels; and
- 3) the High-Precision LSM Teacher used as the pseudo-benchmark.

For this comparison, the OU parameters are

$$\mu = 110, \quad \kappa = 1, \quad \sigma = 20. \quad (80)$$

Table 3. Comparison of original and teacher-generated label designs

| Method                     | Price  | Std. Error | Rel. Diff. | Rel. Diff. (%) |
|----------------------------|--------|------------|------------|----------------|
| Original Pathwise Labels   | 4.1196 | 0.0462     | -0.1609    | -16.0904       |
| Teacher-Generated Labels   | 4.8846 | 0.0233     | -0.0051    | -0.5088        |
| High-Precision LSM Teacher | 4.9095 | 0.0165     | 0.0000     | 0.0000         |

For a general method, this paper calculates the signed relative difference from the Teacher as:

$$\text{Relative Difference} = \frac{\hat{V}_0^{\text{Method}} - \hat{V}_0^T}{\hat{V}_0^T}. \quad (81)$$

As shown in Table 3, the Student trained using original pathwise labels produces a price of 4.1196, compared with the Teacher price of 4.9095. Its relative difference is

$$\frac{4.1196 - 4.9095}{4.9095} \times 100\% = -16.09\%. \quad (82)$$

The negative relative difference indicates that the Student trained using original pathwise labels underestimates the Teacher price by approximately 16.09%.

$$\frac{4.1196 - 4.9095}{4.9095} = -0.1609 \approx -16.09\%. \quad (83)$$

After replacing the original labels with Teacher-generated labels, the Student price increases to 4.8846, and the relative difference becomes

$$\frac{4.8846 - 4.9095}{4.9095} = -0.0051 \approx -0.51\%. \quad (84)$$

Therefore, the magnitude of the relative pricing difference decreases from approximately 16.09% to 0.51%. The standard error also decreases from 0.0462 to 0.0233. Because both designs use the same quadratic Logistic Regression specification, the change in label construction, rather than a change in the classification algorithm, leads to this improvement. This result supports the argument that replacing path-specific realized continuation outcomes with Teacher-generated stopping decisions produces a more reliable training target.

## D. Robustness and Stability

The baseline results establish that the Student can approximate the Teacher under one specification. This subsection examines whether that conclusion remains valid when the OU parameters change, when the number of paths used to construct the Teacher changes, and across independent Monte Carlo replications. In all comparisons, the Teacher and Student policies are evaluated on the same testing paths, and the signed relative pricing difference is calculated as in Eq. (78).

### 1) Robustness across OU parameter settings

The first robustness test evaluates the proposed Teacher-Student framework under different parameter values of the Ornstein-Uhlenbeck process,

$$dS_t = \kappa(\mu - S_t) dt + \sigma dW_t^Q, \quad (85)$$

where  $\mu$  is the long-run mean,  $\kappa$  is the speed of mean reversion, and  $\sigma$  is the volatility parameter. Changes in these parameters affect the distribution of the underlying state variable and may also change the location and shape of the optimal exercise boundary.

The robustness experiment considers the following parameter combinations:

$$\mu \in \{100, 110\}, \quad \kappa \in \{0.5, 1, 2, 3\}, \quad \sigma \in \{5, 10, 20, 30\}. \quad (86)$$

For each parameter combination, a separate High-Precision LSM Teacher is estimated using an independent set of simulated paths. The fitted Teacher then assigns stopping labels to a new training sample, which is used to fit a quadratic Logistic Regression Student. Finally, both the Teacher and Student policies are evaluated on the same independent test paths.

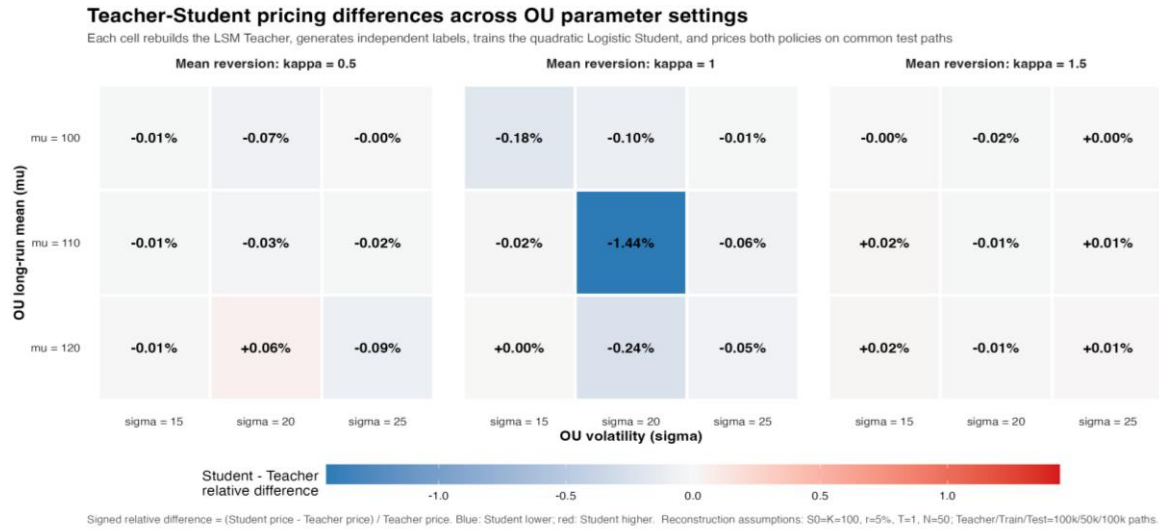


Fig. 5. Signed relative pricing differences between the logistic regression student and the high-precision LSM teacher across OU parameter settings.

Figure 5 presents the signed relative pricing differences for the OU parameter configurations. The difference is close to zero in most cases, showing that the Student generally produces prices similar to those of the Teacher. One configuration has a larger negative difference of approximately  $-1.44\%$ . This result suggests that the accuracy of the quadratic Student may vary across parameter settings, possibly because the stopping boundary changes with the OU parameters.

The Teacher prices also have the expected qualitative interpretation. Higher volatility increases the American put price, whereas a higher long-run mean generally reduces the price by pulling the state process toward a higher level. The effect of the mean-reversion speed depends on the relationship among the initial state, the long-run mean, and the exercise boundary.

## 2) Robustness to teacher simulation size

The preceding comparison shows that Teacher-generated labels improve the Student's pricing performance. This subsection examines whether the quality of those labels depends on the number of paths used to construct the Teacher. The Teacher is fitted using 1,000, 5,000, 10,000, 50,000, and 100,000 paths. For each Teacher simulation size, the Student is independently trained and evaluated over 10 replications, with all other settings held fixed.

Table 4. Sensitivity of student pricing performance to teacher simulation size (10 independent replications)

| Teacher Paths | Teacher Price | Student Price | Mean SRD (%) | Mean ARD (%) |
|---------------|---------------|---------------|--------------|--------------|
| 1,000         | 4.7548        | 4.7615        | 0.1407       | 0.3746       |
| 5,000         | 4.8329        | 4.8342        | 0.0253       | 0.0710       |
| 10,000        | 4.8488        | 4.8453        | -0.0711      | 0.0807       |
| 50,000        | 4.8600        | 4.8575        | -0.0514      | 0.0514       |
| 100,000       | 4.8620        | 4.8607        | -0.0261      | 0.0329       |

Notes: Teacher Price and Student Price are averages across 10 independent replications. SRD denotes the mean signed relative difference, and ARD denotes the mean absolute relative difference. Both are calculated replication by replication relative to the corresponding Teacher price.

Table 4 summarizes the Teacher and Student prices and their signed and absolute relative differences for each simulation size. As the number of Teacher paths increases from 1000 to 100,000, the mean Teacher price changes from

4.7548 to 4.8620, while the mean Student price changes from 4.7615 to 4.8607.

The highest mean absolute relative difference, 0.3746%, is obtained when the Teacher is built based on just 1000 paths. This number drops significantly to 0.0710% for 5000 paths. Despite the slight increase to 0.0807% for 10,000 paths, the mean absolute relative difference continues to drop to 0.0514% for 50,000 paths and 0.0329% for 100,000 paths. Thus, the mean absolute relative difference tends to decrease as the number of Teacher paths increases, although this trend is not monotonic.

The Student slightly overprices the option when the Teacher is constructed with 1000 or 5000 paths. For the larger simulation sizes, it slightly underprices the option instead. In all cases, however, the mean signed relative difference is less than 0.15% in absolute value.

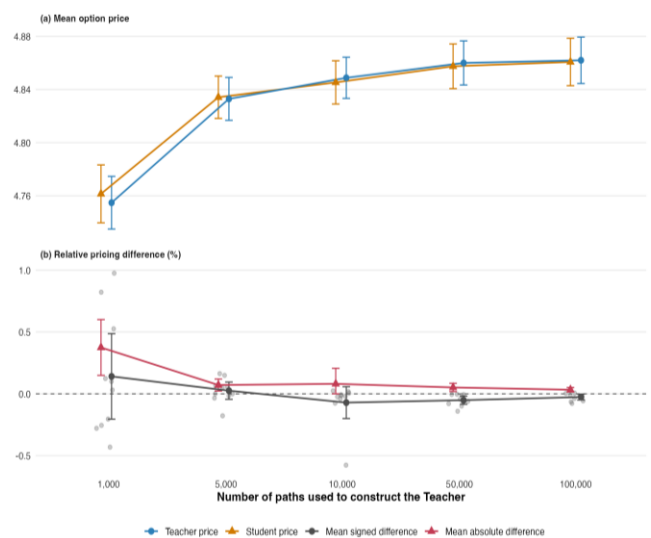


Fig. 6. Sensitivity of Teacher and Student pricing results to the number of paths used to construct the Teacher. Panel (a) reports the mean Teacher and Student prices, with error bars showing variation across independent replications. Panel (b) reports the mean signed and mean absolute relative Student-Teacher pricing differences. The gray points represent the signed relative differences from individual replications.

Fig. 6 summarizes the Teacher-size sensitivity results. Panel (a) shows that the mean Teacher and Student prices are similar across all simulation sizes. Both prices rise as more Teacher paths are used and begin to stabilize at the larger

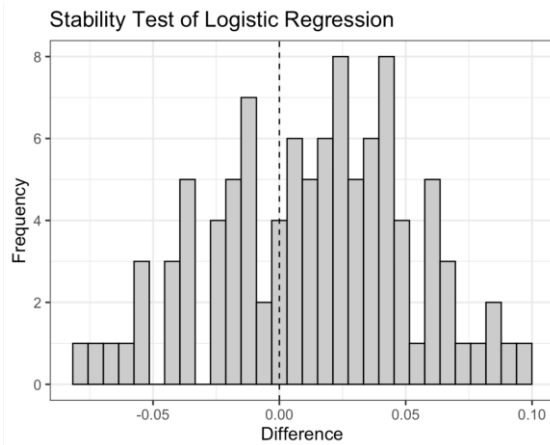
sample sizes. The overlapping error bars also suggest that the difference between the two price estimates is small.

Panel (b) presents the relative pricing differences. The gray points show the signed Student–Teacher differences for the individual replications, and the black line shows the average of these differences. The results are more widely dispersed when the Teacher uses only 1000 paths. This dispersion becomes smaller as the number of Teacher paths increases. The red line represents the mean absolute relative difference. It falls from 0.3746% with 1,000 paths to 0.0329% with 100,000 paths. Although the decline is not consistent at every simulation size, the general pattern shows that a larger

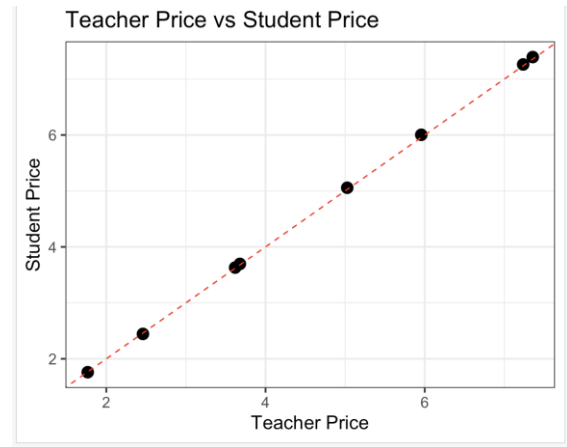
Teacher sample leads to closer agreement between the Student and Teacher policies.

In general, the Student remains close to the corresponding Teacher across the different simulation sizes. The mean absolute relative difference generally decreases as the number of Teacher paths increases and reaches its lowest value, 0.0329%, with 100,000 Teacher paths. However, since each case is based on only 10 independent replications, these results provide evidence of a general stabilization pattern rather than a definitive conclusion.

### 3) Repeated-simulation stability



(a) Distribution of Student–Teacher pricing differences.



(b) Teacher versus Student prices.

Fig. 7. Stability of the Student pricing results relative to the High-Precision LSM Teacher. Panel (a) presents the distribution of Student–Teacher pricing differences across repeated simulations. Panel (b) plots the Student prices against the Teacher prices. The 45-degree line represents equal prices.

Across the repeated Teacher–Student experiments, the standard deviation of the Student price estimates is 0.1148. Relative to the Teacher price computed on the same testing paths, the mean absolute pricing difference is 0.0133, and the mean absolute relative difference is 0.27%. These values show that the Student prices remain close to the corresponding Teacher prices across replications.

The signed pricing difference shown in Fig. 7(a) is defined as

$$\text{Difference} = \hat{V}_0^S - \hat{V}_0^T. \quad (87)$$

The differences are concentrated around zero, with both positive and negative values. This means that the Student may produce a slightly higher or lower price than the Teacher in any individual replication. The small mean absolute difference, low mean absolute relative difference, and narrow distribution around zero show that the Student prices are generally close to the corresponding Teacher prices across replications.

Fig. 7(b) compares the paired Teacher and Student prices. The red 45-degree line represents equality between the two prices. Most points lie close to this line, confirming that the Student price closely tracks the Teacher price across repeated experiments.

#### E. Comparison with More Flexible Classifiers

The proposed Teacher–Student framework uses quadratic Logistic Regression as the Student classifier. To examine whether a more flexible classifier improves pricing performance, this paper compares Logistic Regression with

linear, quadratic polynomial, and Radial Basis Function (RBF) Support Vector Machines.

#### 1) Experimental setup

All classifiers use the same Teacher-generated labels for training. This paper uses the same Teacher, training paths, testing paths, and simulation settings in all experiments, and changes only the Student classifier. This paper repeats the pricing experiment 100 times using independent training and testing paths.

#### 2) Classification performance

Table 5. Classification performance of the SVM student models

| Model         | Accuracy | Sensitivity | Specificity |
|---------------|----------|-------------|-------------|
| Linear SVM    | 0.9964   | 0.9890      | 0.9999      |
| Quadratic SVM | 0.9974   | 0.9922      | 0.9999      |
| RBF SVM       | 0.9974   | 0.9922      | 0.9999      |

All three SVM classifiers achieve an accuracy above 99.6%. The quadratic and RBF SVMs achieve the highest accuracy, while the linear SVM performs slightly worse. All three models also achieve sensitivity and specificity close to one, indicating that they correctly classify both exercise and continuation decisions.

#### 3) Pricing performance

Table 6. Pricing performance over 100 independent replications

| Model                         | Mean Price | SD Price | Mean Difference | Mean Relative Difference (%) |
|-------------------------------|------------|----------|-----------------|------------------------------|
| Linear SVM                    | 4.90       | 0.0224   | 0.0056          | 0.114                        |
| Quadratic Logistic Regression | 4.90       | 0.0222   | 0.0068          | 0.138                        |
| Quadratic SVM                 | 4.90       | 0.0225   | 0.0065          | 0.132                        |
| RBF SVM                       | 4.90       | 0.0225   | 0.0065          | 0.132                        |

Table 6 reports the pricing performance over 100 independent replications. All four classifiers produce almost identical option prices, with mean relative differences below 0.15%. The linear SVM achieves the smallest mean relative difference of 0.114%, while quadratic Logistic Regression achieves a comparable value of 0.138%. The quadratic and RBF SVMs both achieve a mean relative difference of 0.132%. These results show that increasing classifier complexity provides only a very limited improvement once the Student learns the Teacher-generated labels. Therefore, quadratic Logistic Regression provides a good balance between pricing accuracy, interpretability, and computational efficiency, making it an appropriate Student classifier for the proposed Teacher–Student framework.

#### F. Main Findings

The numerical experiments produce five main findings. First, the GBM benchmark comparison shows that the Logistic Regression LSM implementation produces prices close to the Binomial Tree benchmark, although repeated simulations show a small underpricing tendency. Second, the example of two nearly identical states shows that original pathwise labels can differ because of path-level randomness, while the Teacher-generated labels remain consistent.

Third, the Teacher-generated labels substantially improve the Student's pricing performance. In the direct label comparison, the magnitude of the relative difference decreases from approximately 16.09% under the original labels to approximately 0.51% under the Teacher-generated labels. The baseline and stability experiments also show that the Student price remains close to the Teacher price across repeated simulations.

Fourth, the robustness experiments show that the Student generally remains close to the Teacher across the examined OU parameter settings and Monte Carlo replications. Increasing the number of Teacher paths generally reduces the Student–Teacher pricing difference, with the smallest mean absolute relative difference observed at 100,000 paths. This pattern is consistent with more stable continuation-value estimation as the Teacher simulation size increases, although the 10-replication experiment does not establish a formal convergence threshold. These findings establish robustness of the Student's approximation to the estimated Teacher policy; they do not establish convergence to the true American option value.

Finally, the SVM comparison shows that more flexible classifiers provide only limited pricing improvement once the Student learns the Teacher-generated labels, suggesting that quadratic Logistic Regression is sufficient for learning the Teacher's stopping policy in the proposed framework.

## VI. CONCLUSION

This paper proposes a Teacher–Student framework for learning the optimal stopping policy of an American option, especially paying attention to the role played by the quality of the training labels on the pricing performance of the classifier. With the standard label design based on the pathwise comparison approach, the immediate exercise payoff is compared with the future payoff realized along one simulated path. Since the future payoff contains much Monte Carlo randomness, observations with almost the same current

states may be labeled as different due to different future path realizations. To deal with this problem, this paper first builds a High-Precision LSM Teacher based on a large number of simulated paths. The Teacher estimates the conditional expected continuation value and generates more stable exercise-or-continue labels. Then a quadratic Logistic Regression Student learns the optimal stopping policy of the Teacher based on these labels, thereby reformulating the optimal stopping problem into a binary classification problem.

These numerical results confirm that label formation has a significant impact on pricing performance of the Student. Given the Logistic Regression model specification, the Student with the original pathwise labels performs the pricing at a level of about  $-16.09\%$  compared to the Teacher. Once the original labels are substituted for the labels generated by the Teacher, the difference reduces in size to about  $0.51\%$ . In the basic setup, the difference between the pricing of the Student and the Teacher is  $0.6074\%$ . Furthermore, the repeated-simulation analysis shows that the mean absolute pricing difference between the Teacher and the Student is  $0.0133$ , while the mean absolute relative difference is  $0.27\%$ . The pricing differences are centered around zero. These numerical results confirm that Teacher-generated labels help to achieve a good and stable approximation of the stopping rule with a simple Logistic Regression model of the Student.

The robustness and sensitivity analyses provide further support for these findings. Across the examined values of the OU long-run mean, mean-reversion speed, and volatility, the Student prices generally remain close to the corresponding Teacher prices. One parameter configuration produces a larger negative relative difference of approximately  $1.44\%$ , indicating that the quality of the quadratic Student approximation may depend on the shape of the stopping boundary.

The Teacher-size sensitivity experiment shows that the Student–Teacher pricing difference generally becomes smaller as the number of paths used to construct the Teacher increases. The mean absolute relative difference decreases overall from  $0.3746\%$  with 1000 Teacher paths to  $0.0329\%$  with 100,000 Teacher paths, although the pattern is not strictly monotonic. Because each Teacher-size specification is based on only 10 independent replications, this result should be interpreted as evidence of an overall stabilization pattern rather than as a formal convergence result.

In terms of comparisons between different classifiers, it can be said that quadratic Logistic Regression and linear, quadratic polynomial, and RBF Support Vector Machines provide extremely similar price estimations in cases where they have been trained on the same Teacher-generated labels. The mean relative difference in prices of all classifiers is under  $0.15\%$ . This means that, once good labels are generated, adding complexity to the classifier will provide only marginal benefits. Thus, the main lesson of this paper is not the fact that a more complex model leads to better option pricing in America, but the fact that label generation may matter more than adding complexity.

Despite these achievements, there still have been some limitations. First of all, the High-Precision LSM Teacher is still dependent on a finite simulation sample, chosen

regression basis functions, and model parameters, and therefore it is not possible to consider it as the true optimal stopping policy. The major experiments were conducted with a single-asset American put option in the framework of one-dimensional OU process, while the GBM experiment does not directly apply the whole Teacher–Student framework. It might be interesting for future research to adapt the introduced method to the case of multidimensional state variables, path-dependent options, and other stochastic processes. Moreover, more precise numerical benchmarks or even theoretical error bounds could be applied to assess the quality of the Teacher. Finally, future research could focus on the behavior of the Teacher–Student framework in the areas close to the exercise boundary when the value of the immediate exercise payoff and the continuation value become close. This paper showed that improvements in constructing the training labels allow a simple and understandable classifier to learn the stopping policy created by the High-Precision Teacher efficiently.

#### CONFLICT OF INTEREST

The authors declare no conflict of interest.

#### REFERENCES

- Anderson, D. and Ulrych, U. (2023). Accelerated american option pricing with deep neural networks. *Quantitative Finance and Economics*, 7, (2), 207–228. doi: 10.3934/QFE.2023011.
- Azze, A., D’Auria, B., and García-Portugués, E. (2024). Optimal exercise of american options under time-dependent ornstein–uhlenbeck processes. *Stochastics*,. doi: 10.1080/17442508.2024.2325402.
- Becker, S., Cheridito, P., and Jentzen, A. (2019). Deep optimal stopping. *Journal of Machine Learning Research*, 20, (74), 1–25.
- Broadie, M. and Glasserman, P. (1997). Pricing american-style securities using simulation. *Journal of Economic Dynamics and Control*, 21, (8–9), 1323–1352. doi: 10.1016/S0165-1889(97)00029-8.
- Carr, P. and Itkin, A. (2021). Semi-analytical solutions for barrier and american options written on a time-dependent ornstein–uhlenbeck process. *The Journal of Derivatives*, 29, (1), 9–26. doi: 10.3905/jod.2021.1.133.
- Carriere, J. F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19, (1), 19–30. doi: 10.1016/S0167-6687(96)00004-2.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, (3), 273–297. doi: 10.1007/BF00994018.
- Cox, J. C., Ross, S. A., and Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of Financial Economics*, 7, (3), 229–263. doi: 10.1016/0304-405X(79)90015-1.
- Djagba, P. and Ndizihiwe, C. (2024). Pricing american options using machine learning algorithms. *arXiv preprint arXiv:2409.03204*,. doi: 10.48550/arXiv.2409.03204.
- Gaspar, R. M., Lopes, S. D., and Sequeira, B. (2020). Neural network pricing of american put options. *Risks*, 8, (3), 73. doi: 10.3390/risks8030073.
- Gramacy, R. B. and Ludkovski, M. (2015). Sequential design for optimal stopping problems. *SIAM Journal on Financial Mathematics*, 6, (1), 748–775. doi: 10.1137/140980089.
- Jang, H. and Lee, J. (2019). Generative bayesian neural network model for risk-neutral pricing of american index options. *Quantitative Finance*, 19, (4), 587–603. doi: 10.1080/14697688.2018.1490807.
- Kohler, M., Krzyżak, A., and Todorovic, N. (2010). Pricing of high-dimensional american options by neural networks. *Mathematical Finance*, 20, (3), 383–410.
- Lapeyre, B. and Lelong, J. (2021). Neural network regression for bermudan option pricing. *Monte Carlo Methods and Applications*, 27, (3), 227–247.
- Longstaff, F. A. and Schwartz, E. S. (2001). Valuing american options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14, (1), 113–147, Oxford University Press. doi: 10.1093/rfs/14.1.113.
- Tsitsiklis, J. N. and Van Roy, B. (1999). Optimal stopping of markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, 44, (10), 1840–1851. doi: 10.1109/9.793723.
- Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5, (2), 177–188. doi: 10.1016/0304-405X(77)90016-2.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).