Exploring Generative AI for Modeling the Dynamics of Asset Price Process

Jinseong Park[a], Hyungjin Ko[a,b], Jaewook Lee[a,*]

[a] *Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, South Korea*
[b] *Institute of Engineering Research, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, South Korea*

Abstract

Artificial Intelligence (AI) models have been recently studied in finance to discover data patterns. However, generative AI, particularly in image synthesis, remains relatively unexplored. In this paper, we investigate the potential of generative AI to model unknown dynamics using stock chart images. By mimicking the underlying movements of stock prices, we can generate synthetic images without predetermined assumptions about the underlying processes. The experimental results demonstrate that the proposed method successfully replicates well-known asset price processes and accurately estimates option pricing on the S&P 500. We conclude that financial simulation with AI can be a novel approach to financial decision-making.

Keywords: Generative AI, Image synthesis, Asset price process, Financial simulation, Stock chart images

*Corresponding author
Email addresses: jinseong@snu.ac.kr (Jinseong Park), hyungjinko@snu.ac.kr (Hyungjin Ko), jaewook@snu.ac.kr (Jaewook Lee)

Exploring Generative AI for Modeling the Dynamics of Asset Price Process

Anonymous

Abstract

Artificial Intelligence (AI) models have been recently studied in finance to discover data patterns. However, generative AI, particularly in image synthesis, remains relatively unexplored. In this paper, we investigate the potential of generative AI to model unknown dynamics using stock chart images. By mimicking the underlying movements of stock prices, we can generate synthetic images without predetermined assumptions about the underlying processes. The experimental results demonstrate that the proposed method successfully replicates well-known asset price processes and accurately estimates option pricing on the S&P 500. We conclude that financial simulation with AI can be a novel approach to financial decision-making.

Keywords: Generative AI, Image synthesis, Asset price process, Financial simulation, Stock chart images

1. Introduction

The third AI boom in 2010 brought considerable attention to AI models using their ability to identify hidden patterns. Recently, generative AI models such as DALLE-2[1] for image generation (Ramesh et al., 2022) and ChatGPT[2] for natural language processing have expanded the application of deep learning to real-world scenarios. These models can generate new samples by modeling the underlying probability distribution of training data, allowing them to create unseen data. However, the application of generative AI has been less explored in financial research compared to its importance (Assefa et al., 2020; Ko & Lee, 2023). Motivated by this, in this paper, we investigate the potential of generative AI for modeling the dynamics of the asset price process.

Traditional asset price dynamics rely on stochastic processes like the Itô or Jump processes (Taylor, 2011). However, real-world pricing may not follow these diffusion price processes (Hendershott et al., 2022). To address this limitation, we propose training AI models to learn the underlying process of stock price movements, allowing us to generate new data samples that capture the

[1] https://openai.com/product/dall-e-2
[2] https://openai.com/blog/chatgpt

underlying patterns. This approach resolves the discordance between modeling and the underlying data. We then leverage these synthetic paths in financial simulations, such as Monte Carlo simulation, to provide a flexible solution where analytical formulas are not feasible (Clewlow & Strickland, 1998).

Furthermore, we focus to generate opening, high, low, and closing (OHLC) stock chart images. As deep learning models are well-performing in image processing, previous studies have demonstrated the advantages of treating stock price charts as images of OHLC charts (Jiang et al., 2020) and candle charts (Kim & Kim, 2019; Hung & Chen, 2021; Chen & Tsai, 2020). The fundamental idea is that AI can estimate human decision-making based on visual representations rather than numerical time series data. Furthermore, deep learning models can identify trends and fluctuations within 2D images, capturing valuable information such as momentum, reversal, or volatility (Jiang et al., 2020).

For empirical settings, we train Score-based Generative diffusion Models (SGMs) to synthesize OHLC chart images without relying on any prior assumptions about the asset price process. Empirical findings demonstrate that the synthetic images can mimic the well-known asset price processes and accurate estimation of option pricing for the real-world market S&P 500.

Our study contributes to understanding asset price processes by examining the potential of generative AI to learn unknown processes. Our approach offers financial profit as a viable alternative for financial simulation, eliminating the need for any preexisting knowledge regarding underlying processes.

The rest of this study is structured as follows: Section 2 details our methodology for image generation; Section 3 discusses our data and experimental design; Section 4 provides the empirical results of generating stochastic processes and applying our model to the real-world S&P 500 option pricing; and Section 5 provides our concluding remarks.

2. Methodology

2.1. Generative AI Model for Image Synthesis

Recently, Score-based Generative diffusion Models (SGMs) have emerged as the new state-of-art image generative models (Song et al., 2021; Ho et al., 2020).[3] The basic idea of SGMs is to find patterns in images by noising and denoising a stochastic differential equation (SDE). Specifically, SGMs first add random normal noise to the original image x based on Itô SDE:

$$dx = f(x,t)dt + g(t)d\omega, \tag{1}$$

with drift coefficient $f(\cdot,t)$ and diffusion coefficient $g(t)$ for time t, where $d\omega$ is the standard Wiener process. As the mathematical reverse process of Itô SDE is

[3]Note that there are various generative networks such as autoencoders (AE) (McClelland et al., 1987) and generative adversarial networks (GAN) (Goodfellow et al., 2020), but SGMs are widely known to generate better images than other existing models (Song et al., 2021).

known (Anderson, 1982), the SGM is updated to learn how to reverse Equation (1) as follows:

$$dx = (f(x,t) - g^2(t)\nabla_x \log p_t(x))dt + g(t)d\omega, \tag{2}$$

where $\nabla_x \log p_t(x)$ is called *score* with unknown marginal probability densities $p_t(x)$. Thus, SGMs learn the unknown score $\nabla_x \log p_t(x)$ of training data, enabling the generation of synthetic data that exhibits similar characteristics using this score function. In the context of image generation, SGMs leverage the U-Net (Ronneberger et al., 2015), a Convolutional Neural Network (CNN) with a U-shaped architecture, to effectively capture patterns in images. Note that convolutional filters allow for the detection and quantification of changes in stock prices (Jiang et al., 2020).[4]

2.2. Proposed framework

Our framework is designed to train generative AI models to capture the dynamics of asset price processes within the training data. The first step is to pre-process the data by converting stock price charts into image representations. Afterward, we train generative AI models on these chart images to learn patterns. Lastly, the trained models can be used to produce synthetic price chart images, where each image corresponds to one simulation. The whole procedure is illustrated in Figure 1, and detailed descriptions of each step are provided in the following subsections.

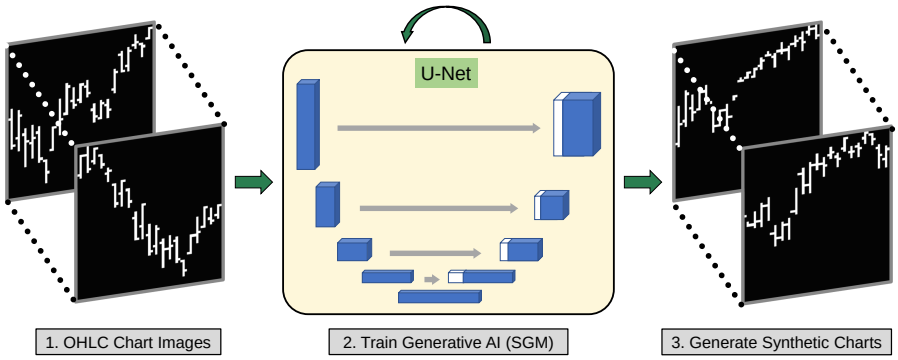


Figure 1: Illustration of the stock image generation framework. The framework includes pre-processing the OHLC stock charts and training a SGM to capture patterns in the charts. After training, we can generate synthetic OHLC chart images.

[4]Examples of 3×2 convolutional 2D filters to detect price change. Filters demonstrating no change: $\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$, small increase: $\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$, and large increase: $\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$.

2.2.1. Data Pre-processing: Imaging Price Charts

To represent stock price charts as images, we adopt the OHLC chart representation inspired by Jiang et al. (2020). This representation converts the chart into pixel images, where the chart pixels are white and the background is black. This pre-processing is suitable for image generation since it eliminates the need to generate all RGB colors. Figure 2a illustrates an example of an OHLC image of stock prices. Each time period is depicted as three pixels wide: the open price on the left, the high and low prices at the top and bottom of the middle bar, and the closing price on the right.

In pre-processing, the stock prices $[\mathcal{S}_0, \ldots, \mathcal{S}_P]$ are transformed into images with corresponding heights $[\mathcal{H}_0, \ldots, \mathcal{H}_P]$, covering a total period of P. To represent the OHLC chart, we denote the stock prices at period p as $\mathcal{S}_p = \{S_p^{open}, S_p^{low}, S_p^{high}, S_p^{close}\}$, where $S_p^{(\cdot)} \in \mathcal{S}_p$, and the image heights as $\mathcal{H}_p = \{h_p^{open}, h_p^{low}, h_p^{high}, h_p^{close}\}$, where $h_p^{(\cdot)} \in \mathcal{H}_p$. The key concept here is to ensure reversibility in the pre-processing step, enabling the recovery of $\mathcal{S}_p$ from $\mathcal{H}_p$ within synthetic images. This is crucial for calculating returns or derivatives by estimating the proportion of price $\mathcal{S}_p$ at period p relative to the initial price $\mathcal{S}_0$, or correspondingly any price $\mathcal{S}_t$ at time t. To achieve this, we fix the minimum value of the image to a fraction ρ of the initial open price S_0^{open}. With the image height $h_p^{(\cdot)}$, we can determine the synthetic stock price ratio of $S_p^{(\cdot)}$ to the initial price S_0^{open} as follows:

$$\frac{S_p^{(\cdot)}}{S_0^{open}} = \frac{h_p^{(\cdot)} + \rho h_0^{open}}{h_0^{open} + \rho h_0^{open}}. \tag{3}$$

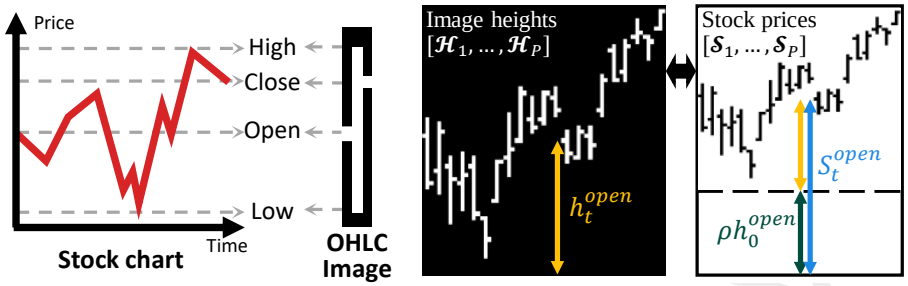
Note that the ratio $h_p^{(\cdot)}/h_0^{open}$ differs from $S_p^{(\cdot)}/S_0^{open}$. For the upper bound, we define it as the maximum value within the total periods in the image.[5] We visualized an image of pre-processed OHLC values for 21 consecutive days into a single image and its conversion to stock prices in Figure 2b. This pre-processing mechanism can facilitate further analysis of synthetic images.

2.2.2. Model Training and Image Generation

With the pre-processed OHLC chart images, we now train SGMs to synthesize image simulations that mimic the intrinsic stock patterns in training data. By noising images with Equation (1) and updating model parameters to denoise the images similar to the original one with Equation (2), we can obtain the trained SGM for generating OHLC chart images. The trained model can efficiently capture the intrinsic patterns in the training dataset and generate synthetic images mimicking the asset price dynamics.

However, some generated images do not meet the conditions of OHLC stock

[5] Classification models trained on stock images often perform well when the upper and lower bounds are defined as the maximum and minimum values of the total periods in the image (Jiang et al., 2020). However, this approach does not allow for reversing the prices from images, as the bottom values in synthetic images are unknown. This results in a disparity between the image heights and the actual stock prices.



(a) Stock chart to OHLC image. (b) 21-day OHLC image to stock price.

Figure 2: Conversion of stock charts and OHLC images.

chart images. To address this issue, we employ the Accept-Reject method (Chib & Greenberg, 1995) with a threshold to filter out awkward simulation paths for improved decision-making in finance (Goutte et al., 2023). Specifically, we drop the images by the following conditions: (i) any column containing more than two values or empty values, (ii) open or close values falling outside the range of high and low values, and (iii) the low value exceeding the high value. The generated images are shown in Figure 3, demonstrating near-perfect image reconstruction without any prior knowledge of chart patterns during training.[6]

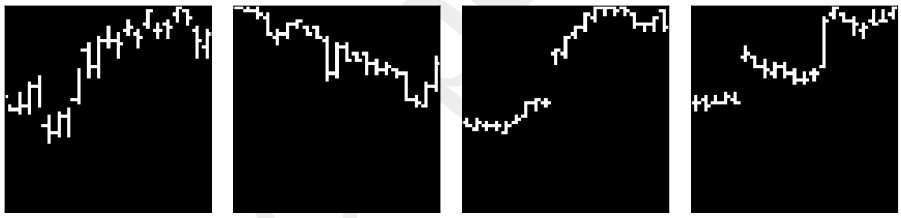


Figure 3: Generated images of 21-day OHLC chart.

3. Data and Experimental design

3.1. Data

For the continuous-time stock price modeling, we used two stock price processes: Geometric Brownian Motion (GBM) and Merton jump process (Merton, 1976). For GBM examples, we generated stock price S with the following GBM process:

$$dS = \mu dt + \sigma d\omega. \tag{4}$$

where μ and σ are drift and volatility. dS represents the changes of asset price over a small interval of time dt and $d\omega$ is normally distributed with mean zero

[6]The generated images before post-processing are presented in Figure A1.

and variance dt. Moreover, the jump-diffusion with additional jump term can be formulated as follows:

$$dS = \mu dt + \sigma d\omega + d\left[\sum_{i=1}^{n_t}(J_i - 1)\right], \tag{5}$$

where n_t is a Poisson process with rate λ and J_i is a log-normally distributed random variable with variance v^2. For experiments, we generated various price dynamics with constants $t = 1$, $\mu = 0.1$, and $\sigma \in \{0.1, 0.2\}$[7] for GBM and $t = 1$, $\mu = 0.1$, $\sigma = 0.2$, $v = 0.3$, and $\lambda \in \{0.1, 0.2\}$ for jump processes. For OHLC modeling, we divided each GBM time step into 5 smaller steps to determine the highest and lowest prices of a period.

For our real-world application, we tested S&P 500 index European call options during two distinct periods: pre-Covid19 (July 2016 to July 2019) and post-Covid19 (January 2020 to January 2023). During the training phase, we utilized three years of data, while the option pricing was tested on the final month of each period: July 2019 and January 2023. The corresponding risk-free interest rates r were set at 0.025 and 0.045, respectively. The data contained all daily call option data within these periods, and we extracted the last traded call option price with a minimum trading volume of one. Table 1 provides the descriptive statistics of S&P 500 index call options for both periods, categorized by moneyness[8] and maturity.

Table 1: Summary statistics of S&P500 index European call options by moneyness and maturity.

	Maturity								
	< 7			7 - 13			14 - 20		
Moneyness	Mean	Std.	Quantity	Mean	Std.	Quantity	Mean	Std.	Quantity
Panel A: Pre-Covid19									
0.96	114.19	21.68	505	115.63	21.68	249	122.24	22.56	134
0.98	59.42	16.98	890	62.44	15.88	613	71.13	14.96	452
1	18.07	10.37	684	24.67	11.03	977	31.68	10.98	842
1.02	0.73	1.32	796	3.93	3.63	976	8.61	5.38	912
1.04	0.11	0.28	320	0.49	0.51	489	1.46	0.96	559
Panel B: Post-Covid19									
0.96	139.71	42.58	1790	143.34	40.87	1522	164.33	38.84	942
0.98	87.04	28.24	1975	101.35	29.89	1873	119.56	28.75	1292
1	31.73	20.16	1999	60.66	18.87	1964	78.24	19.32	1417
1.02	7.15	8.22	1821	28.70	14.13	1495	44.47	15.57	1058
1.04	1.40	2.45	1208	10.01	7.19	874	20.71	9.78	715

Notes. Table 1 presents the summary statistics for the S&P500 index European call options by moneyness and maturity. Moneyness refers to the intrinsic value of an option's premium and maturity denotes the expiration of options. In each setting, we report the mean, standard deviation, and quantity of corresponding options.

[7] We set the volatility σ following the S&P 500 VIX, which generally shows between 0.1 (10%) and 0.2 (20%).

[8] The moneyness represents a range of ± 0.01, where 0.96 indicates the range (0.95, 0.97).

3.2. Experimental Design

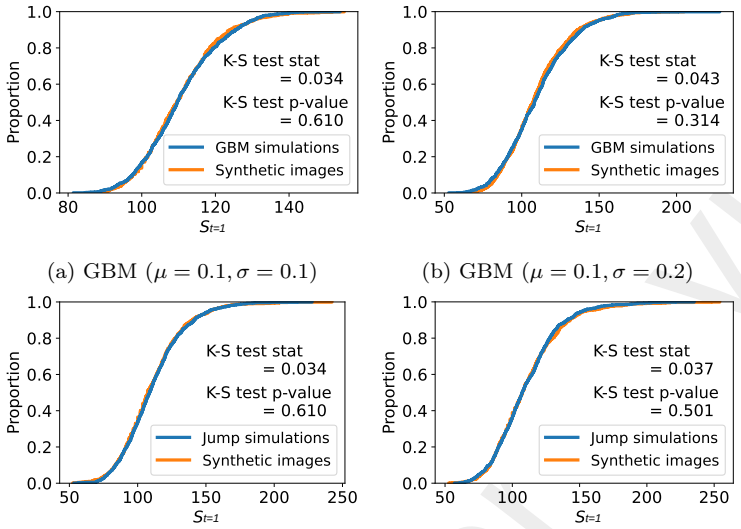
First, we evaluated the ability of the generated images to capture the underlying asset dynamics present in the training OHLC chart images. We assessed the homogeneity between the distributions of the generated images and those of both GBM and jump processes. Additionally, we compared the true expectation and variance of the asset dynamics with those obtained from the generated images, varying the number of simulations and time steps for estimation.

As determining the true asset dynamics is unattainable for the real-world market, we tested the effectiveness of synthetic images through option pricing. Specifically, we examined whether the model trained with the S&P 500 price data could effectively replicate the unknown asset dynamics observed in the market. Our focus was on European options, characterized by the payoff function $C_T = \max(0, S_T - K)$, where C_T represents the payoff at maturity date T, S_T denotes the stock price at time T, and K is the strike price. By using the generated images as multiple simulated paths, we employed Monte Carlo simulations to numerically estimate the option price. The estimation of the option value, denoted as $\hat{c}$, is computed as follows:

$$\hat{c} = \exp(-rT)\frac{1}{M}\sum_{j=1}^{M}\max(0, S_{T,j} - K), \tag{6}$$

where $j \in \{1, \ldots, M\}$ with M generated paths and $S_{T,j}$ is the price at T for simulation j. For measuring the error of S&P 500 option pricing, we used four widely used error metrics[9]: the mean absolute percentage error (MAPE), root mean squared percentage error (RMSPE), mean absolute error (MAE), and the root mean squared error (RMSE).

During training, we used 10,000 samples of OHLC images for each GBM and jump process and all the historical S&P index data in the training period without any historical option pricing data. For the fraction value ρ in Equation (3), we set $\rho = 0.5$ for GBMs, $\rho = 0$ for jump processes, and $\rho = 0.66$ for S&P 500.[10]



(a) GBM ($\mu = 0.1, \sigma = 0.1$) (b) GBM ($\mu = 0.1, \sigma = 0.2$)

(c) Jump ($\mu = 0.1, \sigma = 0.2, \lambda = 0.1$) (d) Jump ($\mu = 0.1, \sigma = 0.2, \lambda = 0.2$)

Figure 4: Cumulative distribution and results of Kolmogorov-Smirnov test between generated images and stochastic processes.

4. Empirical findings

4.1. Testing the Homogeneity

To validate the hypothesis that the synthetic images mimic the financial movement of the training data, we first conducted a two-sample Kolmogorov-Smirnov (K-S) test. The null hypothesis of this test is that the two testing distributions are identical. We compared 1,000 randomly selected examples of S_t when $t = 1$ with the simulations of stochastic processes and synthetic images generated from GBM and Jump processes. The results are shown in Figure 4 by varying volatility σ and jump intensity λ. In all settings, we can show that the cumulative distributions are almost identical. The p-values from the K-S test are 0.610, 0.314, 0.610, and 0.501, respectively. As all the p-values exceeded the threshold of 0.05 by large margins, the null hypothesis of both samples being drawn from the same distribution cannot be rejected.

[9] Let the market price c_i^{market}, predicted call option price $\hat{c}_i$, and corresponding errors $e_i = \hat{c}_i - c_i^{market}$ for i-th option. Then, each metric is calculated with the following equations. MAPE: $\frac{1}{N}\sum_{i=1}^{N}(|e_i|/c_i^{market})$, RMSPE: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(|e_i|/c_i^{market})^2}$, MAE: $\frac{1}{N}\sum_{i=1}^{N}|e_i|$, and RMSE: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}e_i^2}$.

[10] For training, we minimized the loss of the SGM and updated the U-Net architecture's weights for 100 epochs with a batch size of 32 and a learning rate of 0.0001 using the AdamW optimizer. We conducted all experiments using PyTorch and Python on four NVIDIA GeForce RTX 3090 GPUs.

To push further, we explored the predictive capabilities of these simulations in estimating the expectation $E[S_t]$ and variance $Var[S_t]$ across different internal time steps $t \in \{0.2, 0.4, 0.6, 0.8, 1\}$ with generated sample size of $M = 5000$.[11] Table 2 showed that the estimated expectation and variance at these internal steps are closely aligned with the true values.[12] This implies that synthetic images can be used to estimate the price movement in any steps within images, which can be used for financial applications such as pricing options of different maturities.

Table 2: Imaged-based estimation of the expectation $E[S_t]$ and variance $Var[S_t]$ for different time steps.

	Internal Time Steps									
	$t = 0.2$		$t = 0.4$		$t = 0.6$		$t = 0.8$		$t = 1$	
Results	Predict	True	Predict	True	Predict	True	Predict	True	Predict	True
Panel A: GBM process ($\mu = 0.1, \sigma = 0.1$)										
$E[S_t]$	101.93	102.02	104.03	104.08	106.03	106.18	108.25	108.33	110.47	110.52
$Var[S_t]$	4.63	4.56	6.60	6.59	8.20	8.24	9.54	9.71	11.07	11.08
Panel B: GBM process ($\mu = 0.1, \sigma = 0.2$)										
$E[S_t]$	101.98	102.02	104.06	104.08	106.05	106.18	108.29	108.33	110.09	110.52
$Var[S_t]$	9.16	9.14	13.23	13.22	16.42	16.55	19.56	19.53	22.27	22.23
Panel C: Jump process ($\mu = 0.1, \sigma = 0.2, \lambda = 0.1$)										
$E[S_t]$	101.8	101.92	103.71	103.87	106.19	106.04	108.29	108.35	110.59	110.51
$Var[S_t]$	9.74	9.72	13.99	13.94	17.70	17.52	20.75	20.79	22.27	22.23
Panel D: Jump process ($\mu = 0.1, \sigma = 0.2, \lambda = 0.2$)										
$E[S_t]$	101.80	102.09	104.14	104.32	106.64	106.42	108.45	108.51	111.15	111.09
$Var[S_t]$	10.08	10.15	14.99	15.05	18.94	18.91	22.29	22.34	25.56	25.55

Notes. Table 2 demonstrates the estimation performance of synthetic OHLC images for each panel. Each time step corresponds to an internal part of the generated processes with $t = 1$. We provide the predicted values and true values for the expectation $E[S_t]$ and variance $Var[S_t]$ of each process at time t. Since the predicted values closely match the true values, we can conclude that the generated images effectively mimic the entire process.

4.2. S&P 500 Option Pricing

We now present real-world applications of using generated images to calculate call option prices for the S&P 500.[13] As the Black-Scholes formula holds fundamental importance in option pricing, we present a comparison of errors between the Black-Scholes formula and the image-based method for two dissimilar periods: pre-Covid19 and post-Covid19. It is known that the post-Covid19 period exhibits higher volatility compared to the post-Covid19 period (Baek et al., 2020).

[11] We performed 5 times of experiments and stated their mean value as prediction, where $M = 5000$ showed the small enough variance. We additionally explored the predictive capabilities of these simulations in estimating the expectation $E[S_t]$ and variance $Var[S_t]$ at $t = 1$ across different sample sizes M of generated images in Table A1.

[12] The true mean and variance of GBM are calculated through closed-form solutions and those of jump models are estimated with 10,000 training OHLC image samples.

[13] The results of estimating call option prices in the Black-Scholes world are presented in Table A2.

The experimental results are shown in Table 3. In both panels, the proposed approach exhibits superior prediction performance regardless of moneyness and maturity. Specifically, the proposed methods significantly decrease prediction errors for longer prediction maturities and larger moneyness values in both panels. Thus, experimental results of both pre-Covid19 and post-Covid19 periods confirmed that image-based option pricing predicts adequately regardless of the underlying market condition. Therefore, trained SGMs can be used to mimic unidentifiable underlying processes in real markets.

Table 3: Prediction error for S&P 500 index call option.

		Moneyness									
		Black-Scholes Formula					Proposed				
Maturity	Errors	0.96	0.98	1	1.02	1.04	0.96	0.98	1	1.02	1.04
Panel A: Pre-Covid19											
< 7	MAPE	0.10	0.11	0.87	12.76	12.21	0.10	0.08	0.50	0.95	0.97
	RMSPE	0.45	0.15	1.39	16.90	16.94	0.45	0.12	0.61	0.95	0.97
	MAE	7.40	5.77	8.03	3.94	0.92	7.65	4.66	6.44	0.69	0.11
	RMSE	12.97	8.06	9.03	5.11	1.30	13.12	7.45	7.89	1.45	0.30
7 - 13	MAPE	0.13	0.21	0.79	6.16	16.60	0.12	0.11	0.58	0.93	0.96
	RMSPE	0.49	0.39	0.97	7.88	18.56	0.48	0.33	0.64	0.93	0.96
	MAE	9.63	11.11	14.22	12.07	5.38	8.77	6.11	12.27	3.70	0.47
	RMSE	15.57	13.15	14.97	12.58	5.93	15.17	9.46	13.55	5.06	0.69
14 - 20	MAPE	0.13	0.20	0.69	3.06	10.42	0.11	0.12	0.62	0.94	0.97
	RMSPE	0.38	0.23	0.78	3.63	11.57	0.37	0.15	0.66	0.94	0.97
	MAE	10.90	13.06	18.28	17.96	11.44	8.33	8.11	17.83	8.07	1.41
	RMSE	19.72	14.06	18.86	18.29	11.77	18.40	10.17	18.66	9.51	1.70
Panel B: Post-Covid19											
< 7	MAPE	0.33	0.27	0.69	3.24	5.32	0.34	0.24	0.42	0.79	1.60
	RMSPE	0.52	0.93	2.62	6.10	8.83	0.53	0.91	2.54	1.31	2.30
	MAE	35.22	17.26	10.78	7.74	3.18	35.70	15.52	6.96	3.24	0.97
	RMSE	46.89	24.30	13.55	10.25	4.90	47.45	23.15	10.97	5.76	1.91
7 - 13	MAPE	0.41	0.38	0.43	0.99	2.81	0.38	0.25	0.20	0.34	0.45
	RMSPE	0.54	0.51	0.54	1.29	4.53	0.51	0.37	0.25	0.39	0.71
	MAE	47.13	31.06	21.77	20.01	15.60	43.66	22.13	12.44	10.92	4.48
	RMSE	56.15	37.20	24.37	21.86	16.68	53.07	29.23	16.90	14.90	7.04
14 - 20	MAPE	0.32	0.33	0.41	0.76	1.57	0.28	0.19	0.17	0.29	0.35
	RMSPE	0.41	0.41	0.51	0.93	1.93	0.37	0.26	0.22	0.33	0.40
	MAE	44.73	34.08	28.16	26.89	24.35	39.06	20.87	13.40	13.86	8.10
	RMSE	52.72	39.25	31.12	29.42	25.76	47.72	27.69	17.86	17.97	11.04

Notes. Table 3 presents the prediction performance of option pricing for S&P 500 index call option. We report the MAPE, RMSPE, MAE, and RMSE for the results obtained using the Black-Scholes formula and the proposed synthetic image-based option pricing. Each error is calculated based on the corresponding maturity and moneyness expectations, as described in Table 1. The proposed framework demonstrates lower errors in both panels, indicating its robustness regardless of the underlying market conditions.

5. Concluding remarks

In conclusion, our empirical analysis confirms that the use of generative AI with OHLC images successfully imitates the dynamics of asset price pro-

cesses, including distribution homogeneity and statistical variables. Moreover, we demonstrate the ability to predict option prices for S&P 500 call options through simulations, by capturing the underlying market price process.

From an academic perspective, our study demonstrates the usage of generative AI in mimicking price trends for unknown market trends. This will help the researchers to find better modeling of the asset price processes. From an investment perspective, our framework enables us to leverage generative AI for forecasting stock prices and predicting options, futures, and other derivatives.

As the primary focus is to investigate the dynamics of asset price processes, we have a limitation of lacking different option pricing methodologies as baselines. Thus, our next goal is to build a practical framework designed for derivative markets and conduct a comparative analysis with recent models. Furthermore, as our framework is based on images, we believe that incorporating NLP-based generative AI models, such as ChatGPT, may be beneficial.

References

Anderson, B. D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, *12*, 313–326.

Assefa, S. A., Dervovic, D., Mahfouz, M., Tillman, R. E., Reddy, P., & Veloso, M. (2020). Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance* (pp. 1–8).

Baek, S., Mohanty, S. K., & Glambosky, M. (2020). Covid-19 and stock market volatility: An industry level analysis. *Finance research letters*, *37*, 101748.

Chen, J.-H., & Tsai, Y.-C. (2020). Encoding candlesticks as images for pattern classification using convolutional neural networks. *Financial Innovation*, *6*, 1–19.

Chib, S., & Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, *49*, 327–335.

Clewlow, L., & Strickland, C. (1998). *Implementing derivative models*. Wiley.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*, 139–144.

Goutte, S., Le, H.-V., Liu, F., & Von Mettenheim, H.-J. (2023). Deep learning and technical analysis in cryptocurrency market. *Finance Research Letters*, *54*, 103809.

Hendershott, T., Menkveld, A. J., Praz, R., & Seasholes, M. (2022). Asset price dynamics with limited attention. *The Review of Financial Studies*, *35*, 962–1008.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

Hung, C.-C., & Chen, Y.-J. (2021). Dpp: Deep predictor for price movement from candlestick charts. *Plos one*, *16*, e0252404.

Jiang, J., Kelly, B. T., & Xiu, D. (2020). (re-) imag (in) ing price trends. *Chicago Booth Research Paper (forthcoming in the Journal of Finance)*, .

Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. *PloS one*, *14*, e0212320.

Ko, H., & Lee, J. (2023). Can chatgpt improve investment decision? from a portfolio management perspective. *Available at SSRN*, (p. 4390529).

McClelland, J. L., Rumelhart, D. E., Group, P. R. et al. (1987). *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models* volume 2. MIT press.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, *3*, 125–144.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, .

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234–241). Springer.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, *2021*.

Taylor, S. J. (2011). *Asset price dynamics, volatility, and prediction*. Princeton university press.

Appendix

We provided additional experimental results. Table A1 illustrated the additional results of Section 4.1 by varying the number of generated OHLC simulations with $M = 100, 1000$, and 5000. Table A2 illustrated that the results of calculating call option prices in the Black-Scholes world, similar to S&P 500 in Section 4.2. Figure A1 illustrated the generated images before employing the post-processing by the Accept-Reject method.

Table A1: Expectation of generated OHLC simulations by varying the number of image samples.

	Number of Image Samples						
	$M = 100$		$M = 1000$		$M = 5000$		True
Results	Mean	Std.	Mean	Std.	Mean	Std.	
Panel A: GBM process ($\mu = 0.1, \sigma = 0.1$)							
$E[S_{t=1}]$	110.55	1.32	110.41	0.36	110.47	0.05	110.52
$Var[S_{t=1}]$	10.59	0.64	10.83	0.23	11.07	0.09	11.08
Panel B: GBM process ($\mu = 0.1, \sigma = 0.2$)							
$E[S_{t=1}]$	109.17	1.59	109.00	0.50	110.09	0.06	110.52
$Var[S_{t=1}]$	21.34	1.70	21.84	0.63	22.27	0.21	22.33
Panel C: Jump process ($\mu = 0.1, \sigma = 0.2, \lambda = 0.1$)							
$E[S_{t=1}]$	112.66	2.21	110.48	0.66	110.59	0.09	110.51
$Var[S_{t=1}]$	23.22	1.33	23.29	0.61	23.55	0.15	23.57
Panel D: Jump process ($\mu = 0.1, \sigma = 0.2, \lambda = 0.2$)							
$E[S_{t=1}]$	112.49	2.25	111.31	1.15	111.15	0.17	111.09
$Var[S_{t=1}]$	25.04	1.55	26.01	1.09	25.56	0.26	25.55

Notes. Table A1 offers the predictive capabilities of image-based simulations in estimating the true mean and variance. For each panel, we averaged 5 times of simulations and reported their mean and variance across different sample sizes: 100, 1000, and 5000. When $M = 100$, the expected mean is distinct from the true mean and the standard deviation is large. As the sample size increased, such as $M = 5000$, we observed a convergence of the expectation $E[S_{t=1}]$ and variance $Var[S_{t=1}]$ toward the true statistics of the stochastic processes. This convergence suggests that the generated images possess a similar distribution to the training set.

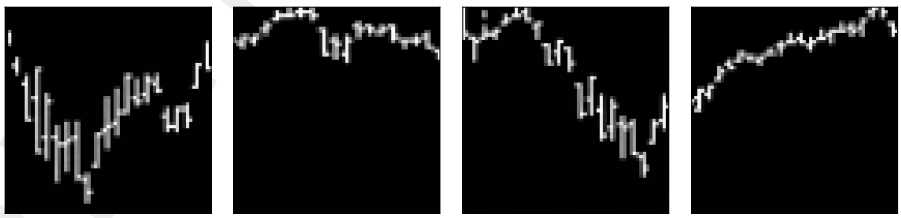


Figure A1: Generated images before post-processing of 21-day chart images.

Table A2: Call Option Pricing with GBM in Black-Scholes World.

		Number of Image Samples						True
		$M = 100$		$M = 1000$		$M = 5000$		
Interest Rates	Strike	Mean	Std.	Mean	Std.	Mean	Std.	
Panel A: GBM ($\mu = 0.1, \sigma = 0.1$)								
$r = 0.025$	K=90	12.94	0.89	12.67	0.29	12.69	0.06	12.65
	K=100	5.15	0.73	5.16	0.16	5.24	0.07	5.30
	K=110	1.52	0.44	1.47	0.12	1.46	0.03	1.47
$r = 0.05$	K=90	14.49	1.11	14.70	0.15	14.78	0.08	14.63
	K=100	6.82	0.57	6.64	0.20	6.79	0.08	6.80
	K=110	2.15	0.51	2.16	0.13	2.18	0.05	2.17
$r = 0.1$	K=90	18.95	0.84	18.76	0.39	18.83	0.07	18.63
	K=100	10.07	0.96	10.39	0.34	10.33	0.07	10.31
	K=110	3.76	0.64	4.13	0.21	4.19	0.07	4.22
Panel B: GBM ($\mu = 0.1, \sigma = 0.2$)								
$r = 0.025$	K=90	15.33	1.53	15.17	0.52	15.15	0.02	15.12
	K=100	8.66	1.01	9.05	0.34	9.08	0.01	9.16
	K=110	4.95	0.98	5.13	0.44	5.05	0.02	5.12
$r = 0.05$	K=90	16.58	1.71	16.95	0.54	16.80	0.02	16.70
	K=100	10.28	1.84	10.29	0.37	10.37	0.02	10.45
	K=110	6.25	1.11	6.00	0.34	5.97	0.01	6.04
$r = 0.1$	K=90	19.84	1.43	20.05	0.30	20.18	0.03	19.99
	K=100	13.75	1.57	13.35	0.37	13.25	0.02	13.27
	K=110	9.08	0.96	7.83	0.57	8.10	0.02	8.18

Notes. Table A2 shows the results of estimating the option pricing in the Black-Scholes world. The objective is to estimate the closed-form solution of call option price in the Black-Scholes formula with GBM models. We trained SGMs with OHLC images of GBM with $\mu = 0.1$ and $\sigma \in \{0.1, 0.2\}$. We reported the results of different simulation sizes M, as in Table A1. We calculated option prices for various interest rates r and strike prices K given an initial stock price of $S_0 = 100$. We first calculated call option prices based on Equation (6) and then multiplied $e^{r+\sigma^2/2-\mu}$ to convert the simulations generated from σ to the risk-free rate r. The results demonstrated that synthetic images accurately approximate true call option prices in all experimental settings. Panel A shows more accurate performance than Panel B, indicating that small variance is easier to estimate because the average value is highly affected by a few extreme values in the large variance case.