Theoretically Motivated Data Augmentation and Regularization for Portfolio Construction

Liu Ziyin
The University of Tokyo
Japan

Kentaro Minami
Preferred Networks, Inc.
Japan

Kentaro Imajo
Preferred Networks, Inc.
Japan

ABSTRACT

The task we consider is portfolio construction in a speculative market, a fundamental problem in modern finance. While various empirical works now exist to explore deep learning in finance, the theory side is almost non-existent. In this work, we focus on developing a theoretical framework for understanding the use of data augmentation for deep-learning-based approaches to quantitative finance. The proposed theory clarifies the role and necessity of data augmentation for finance; moreover, our theory implies that a simple algorithm of injecting a random noise of strength $\sqrt{|r_{t-1}|}$ to the observed return r_t is better than not injecting any noise and a few other financially irrelevant data augmentation techniques.

CCS CONCEPTS

• **Theory and algorithms for application domains** → **Machine learning theory**; • **Applied computing** → *Economics.*

KEYWORDS

data augmentation, deep learning, portfolio selection

ACM Reference Format:
Liu Ziyin, Kentaro Minami, and Kentaro Imajo. 2022. Theoretically Motivated Data Augmentation and Regularization for Portfolio Construction. In *3rd ACM International Conference on AI in Finance (ICAIF '22), November 2–4, 2022, New York, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561720

1 INTRODUCTION

There is an increasing interest in applying machine learning methods to problems in the finance industry. This trend has been expected for almost forty years [11], when well-documented and fine-grained (minute-level) data of stock market prices became available. In fact, the essence of modern finance is fast and accurate large-scale data analysis [13, 26], and it is hard to imagine that machine learning should not play an increasingly crucial role in this field. In contemporary research, the central theme in machine-learning-based finance is to apply existing deep learning models to financial time-series prediction problems [4, 12, 16, 19, 20, 22, 34], which has demonstrated the hypothesized usefulness of deep learning for the financial industry.

ICAIF '22, November 2–4, 2022, New York, NY, USA

ACM ISBN 978-1-4503-9376-8/22/11…$15.00
https://doi.org/10.1145/3533271.3561720

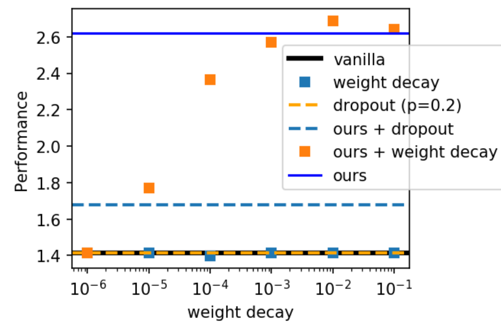

Figure 1: Performance (measured by the Sharpe ratio) of various algorithms on MSFT (Microsoft) from 2018-2020. Directly applying generic machine learning methods, such as weight decay, fails to improve the vanilla model. The proposed method shows significant improvement.

However, one major existing gap in this interdisciplinary field of deep-learning finance is the lack of a theory relevant to justify finance-oriented algorithm design. This work aims to propose a framework where machine learning practices are analyzed in a traditional financial-economic utility theory setting. Our theory implies that a simple theoretically motivated data augmentation technique is better for the task portfolio construction than not injecting any noise at all or some naive noise injection methods that have no theoretical justification. To summarize, our main contributions are (1) to demonstrate how we can use utility theory to analyze practices of deep-learning-based finance and (2) to theoretically study the role of data augmentation in the deep-learning-based portfolio construction problem.[1]

2 RELATED WORKS

Existing deep learning finance methods. In recent years, various empirical approaches to apply state-of-the-art deep learning methods to finance have been proposed [4, 12, 16–20]. However, one crucial gap is the complete lack of theoretical analysis or motivation in this interdisciplinary field of AI-finance. This work makes one initial step to bridge this gap. One theme of this work is that finance-oriented prior knowledge and inductive bias is required to design the relevant algorithms. For example, Ziyin et al. [36] shows that incorporating prior knowledge into architecture design is key to the success of neural networks and applied neural networks with periodic activation functions to the problem of financial index prediction.

In fact, most generic and popular machine learning techniques are proposed and have been tested for standard ML tasks such as

[1]We refer the readers to a longer version of this work online for more detailed discussion, extensive experiments, and a code for reproducing our results: :https://arxiv.org/abs/2106.04114.

image classification or language processing. Directly applying the ML methods that work for image tasks is unlikely to work well for financial tasks, where the nature of the data is different. See Figure 1, where we show the performance of a neural network directly trained to maximize wealth return on MSFT during 2019-2020. Using popular, generic deep learning techniques such as weight decay or dropout does not result in any improvement over the baseline. In contrast, our theoretically motivated method does. Combining the proposed method with weight decay has the potential to improve the performance a little further, but the improvement is much lesser than the improvement of using the proposed method over the baseline. This implies that a generic machine learning method is unlikely to capture well the inductive biases required to tackle a financial task. The present work proposes to fill this gap by showing how finance knowledge can be incorporated into algorithm design.

Data augmentation. Consider a training loss function of the additive form $L = \frac{1}{N}\sum_i \ell(x_i, y_i)$ for N pairs of training data points $\{(x_i, y_i)\}_{i=1}^N$. Data augmentation amounts to defining an underlying data-dependent distribution and generating new data points stochastically from this underlying distribution. A general way to define data augmentation is to start with a datum-level training loss and transform it to an expectation over an augmentation distribution $P(z|(x_i, y_i))$ [8], $\ell(x_i, y_i) \to \mathbb{E}_{(z_i,g_i)\sim P(z,g|(x_i,y_i))}[\ell(z_i, g_i)]$, and the total training loss function becomes

$$L_{\rm aug} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{(z_i,g_i)\sim P(z,g|(x_i,y_i))}[\ell(z_i, g_i)]. \tag{1}$$

One common example of data augmentation is injecting isotropic gaussian noise to the input [12, 31], which is equivalent to setting $P(z, g|(x_i, y_i)) \sim \delta(g-y_i)\exp\left[-(z-x_i)^{\rm T}(z-x_i)/(2\sigma^2)\right]$ for some specified strength σ^2. Despite the ubiquity of data augmentation in deep learning, existing works are often empirical in nature [1, 12, 31, 35]. For a relevant example, Fons et al. [12] empirically evaluates the effect of different types of data augmentation in a financial series prediction task. Dao et al. [8] is one major recent theoretical work that tries to understand modern data augmentation theoretically; it shows that data augmentation is approximately learning in a special kernel. He et al. [14] argues that data augmentation can be seen as an effective regularization. However, no theoretically motivated data augmentation method for finance exists yet. One major challenge and achievement of this work is to develop a theory that bridges the traditional finance theory and machine learning methods. Data augmentation methods have mainly been applied to neural networks. We thus do not discuss or compare with many previous works that apply non-deep-learning machine learning methods to finance. We restrict ourselves to deep learning in this work. In the next section, we introduce the portfolio theory.

3 BACKGROUND: MARKOWITZ PORTFOLIO THEORY

Portfolio theory's central concern is how to make the optimal investment in a financial market. One unfamiliar with the portfolio theory may easily confuse the task the portfolio construction with wealth maximization trading or future price prediction. Before we introduce the portfolio theory, we first stress that the task of portfolio construction is not equivalent to wealth maximization or accurate price prediction. One can construct an optimal portfolio without predicting the price or maximizing the wealth increase.

Consider a market with an equity (a stock) and a fixed-interest rate bond (a government bond). We denote the price of the equity at time step t as S_t, and the *price return* is defined as $r_t = \frac{S_{t+1}-S_t}{S_t}$, which is a random variable with variance C_t, and the expected return $g_t := \mathbb{E}[r_t]$. Our wealth at time step t is $W_t = M_t + n_t S_t$, where M_t is the amount of cash we hold, and n_i the shares of stock we hold for the i-th stock. As in the standard finance literature, we assume that the shares are infinitely divisible. Usually, a positive n denotes holding (long) and a negative n denotes borrowing (short). The wealth we hold initially is $W_0 > 0$, and we would like to invest our money on the equity. We denote the relative value of the stock we hold as $\pi_t = \frac{n_t S_t}{W_t}$. π is called a *portfolio*. The central challenge in portfolio theory is to find the best π. At time t, our wealth is W_t; after one time step, our wealth changes due to a change in the price of the stock (setting the interest rate to be 0): $\Delta W_t := W_{t+1} - W_t = W_t \pi_t r_t$. The goal is to maximize the wealth return $G_t := \pi_t \cdot r_t$ at every time step *while minimizing risk*[2]. The risk is defined as the variance of the wealth change:

$$R_t := R(\pi_t) := {\rm Var}_{r_t}[G_t] = \left(\mathbb{E}[r_t^2] - g_t^2\right)\pi_t^2 = \pi_t^2 C_t. \tag{2}$$

The standard way to control risk is to introduce a "risk regularizer" that punishes the portfolios with a large risk [25, 28].[3] Introducing a parameter λ for the strength of regularization (the factor of 1/2 appears for convention), we can now write down our objective:

$$\pi_t^* = \arg\max_\pi U(\pi) := \arg\max_\pi \left[\pi^{\rm T} G_t - \frac{\lambda}{2}R(\pi)\right]. \tag{3}$$

Here, U stands for the utility function; λ can be set to be the desired level of risk-aversion. When g_t and C_t is known, this problem can be explicitly solved. However, one main problem in finance is that its data is highly limited, and we only observe one particular *realized* data trajectory, and g_t and C_t are hard to estimate. This fact motivates the necessity of data augmentation and synthetic data generation in finance [2]. In this paper, we treat the case where there is only one asset to trade in the market, and the task of utility maximization amounts to finding the best balance between cash-holding and investment. The equity we are treating is allowed to be a weighted combination of multiple stocks (a portfolio of some public fund manager, for example), and so our formalism is not limited to single-stock situations.

4 PORTFOLIO CONSTRUCTION AS A TRAINING OBJECTIVE

Recent advances have shown that the financial objectives can be interpreted as training losses for an appropriately inserted neural-network model [4, 37]. It should come as no surprise that the utility function (3) can be interpreted as a loss function. When the goal is portfolio construction, we parametrize the portfolio $\pi_t = \pi_{\mathbf{w}}(x_t)$ by a neural network with weights $\mathbf{w}$, and the utility maximization problem becomes a maximization problem over the weights of the

[2] It is important to not to confuse the *price return* r_t with the wealth return G_t.

[3] In principle, any concave function in G_t can be a risk regularizer from classical economic theory [32]. One common alternative would be $R(G) = \log(G)$ [21], and our framework can be easily extended to such cases.

neural network. The time-dependence is modeled through the input to the network x_t, which possibly consists of the available information at time t for determining the future price.[4] The objective function (to be maximized) plus a pre-specified data augmentation transform $x_t \to z_t$ with underlying distribution $p(z|x_t)$ is then

$$\pi_t^* = \arg\max_{\mathbf{w}} \left\{ \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_t \left[G_t(\pi_{\mathbf{w}}(z_t)) \right] - \lambda \mathrm{Var}_t [G_t(\pi_{\mathbf{w}}(z_t))] \right\}, \tag{4}$$

where $\mathbb{E}_t := \mathbb{E}_{z_t \sim p(z|x_t)}$. In this work, we abstract away the details of the neural network to approximate π. We instead focus on studying the maximizers of this equation, which is a suitable choice when the underlying model is a neural network because one primary motivation for using neural networks in finance is that they are universal approximators and are often expected to find such maximizers [4, 17].

The ultimate financial goal is to construct π^* such that the utility function is maximized with respect to the true underlying distribution of S_t, which can be used as the generalization loss (to be maximized):

$$\pi_t^* = \arg\max_{\pi_t} \left\{ \mathbb{E}_{S_t} \left[G_t(\pi) \right] - \lambda \mathrm{Var}_{S_t} [G_t(\pi)] \right\}. \tag{5}$$

Note the difference in taking the expectation between Eq (4) and (5) is that $\mathbb{E}_t$ is computed with respect to the training set we hold, while $\mathbb{E}_{S_t} := \mathbb{E}_{S_t \sim p(S_t)}$ is computed with respect to the underlying distribution of S_t given its previous prices. We used the same shorthands for Var_t and Var_{S_t}. Technically, the true utility we defined is an *in-sample counterfactual* objective, which roughly evaluates the expected utility to be obtained if we restart from yesterday, which is a relevant measure for financial decision making.

4.1 Standard Models of Stock Prices

The expectations in the true objective Equation (5) need to be taken with respect to the true underlying distribution of the stock price generation process. In general, the price follows the following stochastic process $\Delta S_t = f(\{S_i\}_{i=1}^t) + g(\{S_i\}_{i=1}^t)\eta_t$ for a zero-mean and unit variance random noise η_t; the term f reflects the short-term predictability of the stock price based on past prices, and g reflects the extent of unpredictability in the price. A key observation in finance is that g is non-stationary (heteroskedastic) and price-dependent (multiplicative). One model is the geometric Brownian motion (GBM)

$$S_{t+1} = (1+r)S_t + \sigma_t S_t \eta_t, \tag{6}$$

which is taken as the minimal standard model of the motion of stock prices [3, 24]; this paper also assumes the GBM model as the underlying model. Here, we note that the theoretical problem we consider can be seen as a discrete-time version of the classical Merton's portfolio problem [27]. The more flexible Heston model [15] takes the form $dS_t = rS_t dt + \sqrt{v_t} S_t dW_t$, where v_t is the instantaneous volatility that follows its own random walk, and dW_t is drawn from a Gaussian distribution. Despite the simplicity of these models, the statistical properties of these models agree well with the known statistical properties of the real financial markets [10].

[4] It is helpful to imagine x_t as, for example, the prices of the stocks in the past 10 days.

4.2 No Data Augmentation

In practice, there is no way to observe more than one data point for a given stock at a given time t. This means that it can be very risky to directly train on the raw observed data since nothing prevents the model from overfitting to the data. Without additional assumptions, the risk is zero because there is no randomness in the training set conditioning on time t. To control this risk, we thus need data augmentation. One can formalize this intuition through the following proposition.

Proposition 4.1. (Utility of no-data-augmentation strategy.) *Let the price trajectory be generated with GBM in Eq.* (6) *with initial price S_0, then the true utility for the no-data-augmentation strategy is*

$$U_{\text{no-aug}} = [1 - 2\Phi(-r/\sigma)]r - \frac{\lambda}{2}\sigma^2, \tag{7}$$

where $U(\pi)$ is the utility function defined in Eq. (3)*; Φ is the c.d.f. of a standard normal distribution.*

Proof. When there is no data augmentation, $\mathbb{E}_t [G_t(\pi)] = b_t r_t$ and $\mathrm{Var}_t [G_t(\pi)] = 0$. The utility is thus maximized at

$$\pi_t^* = \begin{cases} 1, & \text{if } r_t \geq 0 \\ -1, & \text{if } r_t < 0. \end{cases} \tag{8}$$

For a time-dependent strategy π_t^*, the true utility is defined as[5]

$$U(\pi^*) = \mathbb{E}_{S'_0, S'_1, \dots, S'_T, S'_{T+1}} \left[\frac{1}{T} \sum_{t=1}^{T+1} \pi_t^* r'_t - \left(\frac{\lambda}{2T} \sum_{t=1}^{T} (\pi_t^* r'_t)^2 - \mathbb{E}_{S'_0, S'_1, \dots, S'_T, S'_{T+1}} [\pi_t^* r'_t]^2 \right) \right], \tag{9}$$

where $S'_1, \dots, S'_T, S'_{T+1}$ is an independently sampled distribution for testing, and $r'_t := \frac{S'_{t+1} - S'_t}{S'_t}$ are their respective returns. Now, we note that we can write the price dynamics in terms of the returns:

$$S_{t+1} = (1+r)S_t + \sigma_t S_t \eta_t \to r_t = r + \sigma \eta_t, \tag{10}$$

which means that $r_t \sim \mathcal{N}(r, \sigma^2)$ obeys a Gaussian distribution. Therefore,

$$U(\pi^*) = \frac{r}{T} \sum_{t=1}^{T} \pi_t^* - \frac{\lambda \sigma^2}{2T} \sum_{t=1}^{T} (\pi_t^*)^2. \tag{11}$$

Now we would like to average over π_t^*, because we also want to average over the realizations of the training set to make the true utility independent of the sampling of the training set.

Recall that the strategy is defined as

$$\pi_t^* = \begin{cases} 1, & \text{if } r_t \geq 0 \\ -1, & \text{if } r_t < 0. \end{cases} = \Theta(r_t \geq 1) - \Theta(r_t < 1) \tag{12}$$

for a training set $\{S_0, \dots, S_T\}$. We thus have that

$$\begin{cases} (\pi_t^*)^2 = 1; \\ \mathbb{E}_{S_1, \dots S_{T+1}} [\pi_t^*] = \mathbb{E}_{S_1, \dots S_{T+1}} [\Theta(r_t \geq 0) - \Theta(r_t < 0)] = 1 - 2\Phi(-r/\sigma), \end{cases}$$

[5] While we mainly use $\Theta(x)$ as the Heaviside step function, we overload this notation a little. When we write $\Theta(x > 0)$, Θ is defined as the indicator function. We think that this is harmless because the difference is clearly shown by the argument to the function.

where Φ is the Gaussian c.d.f. Noticing that the training set and the test set are independent, we obtain

$$U = \mathbb{E}_{S_1,...S_{T+1}}[U(\pi^*)] \tag{13}$$

$$= \frac{1}{T}\sum_{t=1}^{T}[1-2\Phi(-r/\sigma)]r - \frac{\lambda}{2}\frac{1}{T}\sum_{t=1}^{T}\sigma^2 \tag{14}$$

$$= [1-2\Phi(-r/\sigma)]r - \frac{\lambda}{2}\sigma^2. \tag{15}$$

This finishes the proof. □

This means that, the larger the volatility σ, the smaller is the utility of the no-data-augmentation strategy. This is because the model may easily overfit to the data when no data augmentation is used. In the next section, we discuss the case when a simple data augmentation is used.

4.3 Additive Gaussian Noise

While it is still far from clear how the stock price is correlated with the past prices, it is now well-recognized that $\text{Var}_{S_t}[S_t|S_{t-1}] \neq 0$ [5, 24]. This motivates a simple data augmentation technique to add some randomness to the financial sequence we observe, $\{S_1, ..., S_{T+1}\}$. This section analyzes a vanilla version of data augmentation of injecting simple Gaussian noise, which we will compare with a more sophisticated data augmentation method in the next section. Here, we inject random Gaussian noises $\epsilon_t \sim \mathcal{N}(0,\rho^2)$ to S_t during the training process such that $z_t = S_t + \epsilon$. Note that the noisified return needs to be carefully defined since noise might also appear in the denominator, which may cause divergence; to avoid this problem, we define the noisified return to be $\tilde{r}_t := \frac{z_{t+1}-z_t}{S_t}$, i.e., we do not add noise to the denominator. Theoretically, we can find the optimal strength ρ^* of the gaussian data augmentation to be such that the true utility function is maximized for a fixed training set. The result can be shown to be

$$(\rho^*)^2 = \frac{\sigma^2}{2r}\frac{\sum_t (r_t S_t^2)^2}{\sum_t r_t S_t^2}. \tag{16}$$

The fact the ρ^* depends on the prices of the whole trajectory reflects the fact that time-independent data augmentation is not suitable for a stock price dynamics prescribed by Eq. (6), whose inherent noise $\sigma S_t \eta_t$ is time-dependent through the dependence on S_t. Finally, we can plug in the optimal ρ^* to obtain the optimal achievable strategy for the additive Gaussian noise augmentation. As before, the above discussion can be formalized, with the true utility given in the next proposition. The proof is similar to the proof we provide in the next section, and is thus omitted for brevity.

Proposition 4.2. (Utility of additive Gaussian noise strategy.) *Under additive Gaussian noise strategy, and let other conditions the same as in Proposition 4.1, the true utility is*

$$U_{\text{Add}} = \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2}\Theta\left(\sum_t r_t S_t^2\right)\right], \tag{17}$$

where Θ is the Heaviside step function.

4.4 Multiplicative Gaussian Noise

In this section, we derive a general kind of data augmentation for the price trajectories specified by the GBM and the Heston model. From the previous discussions, one might expect that a better kind of augmentation should have $\rho = \rho_0 S_t$, i.e., the injected noise should be *multiplicative*; however, we do not start from imposing $\rho \to \rho S_t$; instead, we consider $\rho \to \rho_t$, i.e., a general time-dependent noise. In the derivation, one can find an interesting relation for the optimal augmentation strength:

$$(\rho_{t+1}^*)^2 + (\rho_t^*)^2 = \frac{\sigma^2}{2r}r_t S_t^2. \tag{18}$$

We first prove that the optimal strategy given by Eq. (19):

Lemma 4.3. *The maximizer of the utility function in Eq.* (4) *with multiplicative gaussian noise is*

$$\pi_t^*(\rho) = \begin{cases} \frac{r_t S_t^2}{\lambda(\rho_t^2+\rho_{t+1}^2)}, & \text{if } -1 < \frac{r_t S_t^2}{2\lambda(\rho_t^2+\rho_{t+1}^2)} < 1; \\ \text{sgn}(r_t), & \text{otherwise.} \end{cases} \tag{19}$$

Proof. With Gaussian noise, we have

$$\begin{cases} \mathbb{E}_t\left[G_t(\pi)\right] = \pi_t \mathbb{E}_t\left[\frac{S_{t+1}+\rho_{t+1}\epsilon_{t+1}-S_t-\rho_t\epsilon_t}{S_t}\right] == \pi_t r_t; \\ \text{Var}_t\left[G_t(\pi)\right] = \pi_t^2 \text{Var}_t\left[\frac{\rho_{t+1}\epsilon_{t+1}-\rho_t\epsilon_t}{S_t}\right] = \frac{2\gamma_t^2\pi_t^2}{S_t^2}, \end{cases} \tag{20}$$

where we have defined $2\gamma_t^2 := \rho_t^2 + \rho_{t+1}^2$. The training objective becomes

$$\pi_t^* = \arg\max_{\pi_t}\left\{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_t\left[G_t(\pi)\right] - \frac{\lambda}{2}\text{Var}_t\left[G_t(\pi)\right]\right\} \tag{21}$$

$$= \arg\max_{\pi_t}\left\{\frac{1}{T}\sum_{t=1}^{T}\pi_t r_t - \lambda\frac{\gamma_t^2\pi_t^2}{S_t^2}\right\}. \tag{22}$$

This maximization problem can be maximized for every t respectively. Taking derivative and set to 0, we find the condition that π_t^* satisfies

$$\frac{\partial}{\partial \pi_t}\left(\pi_t r_t - \lambda\frac{\gamma_t^2\pi_t^2}{S_t^2}\right) = 0 \tag{23}$$

$$\longrightarrow \quad \pi_t^*(\gamma_t) = \frac{r_t S_t^2}{2\lambda\gamma_t^2}. \tag{24}$$

By definition, we also have $|\pi_t| \le 1$, and so

$$\pi_t^*(\gamma_t) = \begin{cases} \frac{r_t S_t^2}{2\lambda\gamma_t^2}, & \text{if } -1 < \frac{r_t S_t^2}{2\lambda\gamma_t^2} < 1; \\ \text{sgn}(r_t), & \text{otherwise,} \end{cases} \tag{25}$$

which is the desired result. □

The following proposition gives the true utility of using this data augmentation.

Proposition 4.4. (Utility of general multiplicative Gaussian noise strategy.) *Under general multiplicative noise augmentation strategy, and let other conditions the same as in Proposition 4.1, then the true utility is*

$$U_{\text{mult}} = \frac{r^2}{2\lambda\sigma^2}[1-\Phi(-r/\sigma)]. \tag{26}$$

Proof. Most of the proof is similar to the Gaussian case by replaing ρ^2 with γ_t^2. Following the same procedure, We have:

$$U(\pi^*) = \frac{r}{T}\sum_{t=1}^{T}\pi_t^* - \frac{\lambda\sigma^2}{2T}\sum_{t=1}^{T}(\pi_t^*)^2. \tag{27}$$

Plug in the preceding lemma, we have

$$U(\pi^*) = \frac{r}{T}\sum_{t=1}^{T}\frac{r_t S_t^2}{2\lambda\gamma_t^2} - \frac{\lambda\sigma^2}{2T}\sum_{t=1}^{T}\left(\frac{r_t S_t^2}{2\lambda\gamma_t^2}\right)^2. \tag{28}$$

This utility is a function of of the data augmentation strength γ_t, and, unlike the additive Gaussian case, can be maximized term by term for different t. For a fixed training set, we would like to find the best γ_t that maximizes the above utility. Note that the maximizer of the utility is different depending on the sign of $r_t S_t^2$. Taking derivative and set to 0, we obtain that

$$(\gamma_t^*)^2 = \begin{cases} \frac{\sigma^2}{2r} r_t S_t^2, & \text{if } r_t S_t^2 > 0 \\ \infty, & \text{otherwise.} \end{cases} \tag{29}$$

Plug in to the previous lemma, we have

$$\pi_t^*(\gamma_t^*) = \frac{r_t S_t^2}{2\lambda(\gamma_t^*)^2} = \frac{r}{\lambda\sigma^2}\Theta(r_t). \tag{30}$$

One thing to notice that the optimal strength is independent of λ, which is an arbitrary value and dependent only on the psychology of the investor. Plug into the utility function and take expectation with respect to the training set, we obtain

$$U_{\text{mult}} = \mathbb{E}_{S_1,\dots,S_{T+1}}\left[U(\pi^*(\rho^*))\right] \tag{31}$$

$$= U(\pi^*) = \frac{r}{T}\sum_{t=1}^{T}\pi_t^*(\gamma_t^*) - \frac{\lambda\sigma^2}{2T}\sum_{t=1}^{T}[\pi_t^*(\gamma_t^*)]^2 \tag{32}$$

$$= \frac{r^2}{\lambda\sigma^2}[E]_t[\Theta(r_t)] - \frac{r^2}{2\lambda\sigma^2}\Theta(r_t)[E]_t[\Theta(r_t)] \tag{33}$$

$$= \frac{r^2}{2\lambda\sigma^2}[1-\Phi(-r/\sigma)] \tag{34}$$

$$= \frac{r^2}{2\lambda\sigma^2}\Phi(r/\sigma) \tag{35}$$

This finishes the proof. □

Combining the above propositions, we can prove the main theorem of this work, which shows that the mean-variance utility of the proposed augmentation is strictly higher than that of no data-augmention and that of additive Gaussian noise.

Theorem 4.5. *If $\sigma \neq 0$, then $U_{\text{mult}} > U_{\text{add}}$ and $U_{\text{mult}} > U_{\text{no-aug}}$ with probability* 1.

Proof. We first show that $U_{\text{mult}} > U_{\text{add}}$. Recall that

$$U_{\text{Add}} = \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t)^2}{\sum_t (r_t S_t)^2}\Theta\left(\sum_t r_t S_t^2\right)\right] \tag{36}$$

$$\leq \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t \Theta(r_t>0))^2}{\sum_t (r_t S_t)^2}\Theta\left(\sum_t r_t S_t^2\right)\right] \tag{37}$$

$$\leq \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\frac{(\sum_t r_t S_t \Theta(r_t>0))^2}{\sum_t (r_t S_t)^2}\right] \tag{38}$$

$$\leq_{\text{(Cauchy Inequality)}} \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\sum_t \frac{(r_t S_t \Theta(r_t>0))^2}{(r_t S_t)^2}\right] \tag{39}$$

$$= \frac{r^2}{2\lambda\sigma^2 T}\mathbb{E}_{S_t}\left[\sum_t \Theta(r_t>0)\right] \tag{40}$$

$$= \frac{r^2}{2\lambda\sigma^2}\Phi(r/t) = U_{\text{mult}}. \tag{41}$$

The Cauchy equality holds if and only if $S_1 = \dots = S_{T+1}$; this event has probability measure 0, and so, with probability 1, $U_{\text{add}} < U_{\text{mult}}$.

Now we prove the second inequality. Recall that

$$U_{\text{no-aug}} = [1-2\Phi(-r/\sigma)]r - \frac{\lambda}{2}\sigma^2. \tag{42}$$

We divide into 2 subcases. Case 1: $\lambda > \frac{r}{\sigma^2}$. We have

$$U_{\text{no-aug}} < -2r\Phi(-r/\sigma) < 0 < U_{\text{mult}}. \tag{43}$$

Case 2: $0 < \lambda \leq \frac{r}{\sigma^2}$. We have

$$U_{\text{no-aug}} < [1-2\Phi(-r/\sigma)]r \tag{44}$$

$$< \Phi(r/\sigma)r \tag{45}$$

$$\leq \frac{r^2}{2\lambda\sigma^2}\Phi(r/t) = U_{\text{mult}}. \tag{46}$$

This finishes the proof. □

Heston Model and Real Price Augmentation. We also consider the more general Heston model. The derivation proceeds similarly by replacing $\sigma^2 \to v_t^2$; one arrives at the relation for optimal augmentation: $(\rho_{t+1}^*)^2 + (\rho_t^*)^2 = \frac{1}{2r}v_t^2 r_t S_t^2$. One quantity we do not know is the volatility v_t, which has to be estimated by averaging over the neighboring price returns. One central message from the above results is that one should add noises with variance proportional to $r_t S_t^2$ to the observed prices for augmenting the training set.

Our setting can be seen as the discrete-time version of the famous Merton's portfolio problem [27], where the optimal stationary portfolio is also found to be $\frac{r}{\lambda\sigma^2}$. Our result thus agrees with the classical result when the model parametrizes a stationary portfolio. Stationary portfolios are important in financial theory and can be shown to be optimal even among all dynamic portfolios in some situations [7, 27]. In the experiment section, we also compare with the optimal stationary portfolio.

5 ALGORITHMS

Our results strongly motivate for a specially designed data augmentation for financial data. For a data point consisting purely of past prices $(S_t, \dots, S_{t+L}, S_{t+L+1})$ and the associated returns $(r_t, \dots, r_{t+L-1}, r_{t+L})$, we use $x = (S_t, \dots, S_{t+L})$ as the input for our model f, possibly a neural network, and use S_{t+L+1} as the unseen future price for computing the training loss. Our results suggests that we should

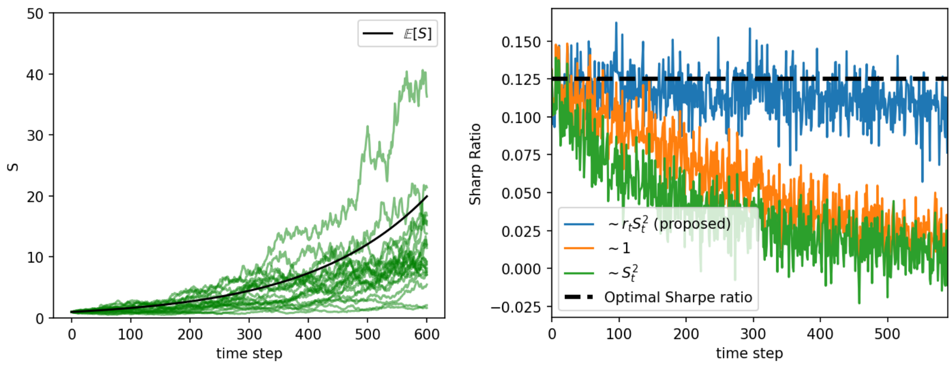

Figure 2: Experiment on geometric brownian motion; $S_0 = 1$, $r = 0.005$, $\sigma = 0.04$. Left: Examples of prices trajectories in green; the black line shows the expected value of the price. Right: Comparison with other related data augmentation techniques. The black dashed line shows the optimal achievable Sharpe ratio. We see that the proposed method stay close to the optimality across a 600-step trading period as the theory predicts.

randomly noisify both the input x and S_{t+L+1} at every training step by

$$\begin{cases} S_i \longrightarrow S_i + c\sqrt{\hat{\sigma}_i^2 |r_i| S_i^2}\epsilon_i & \text{for } S_i \in x; \\ S_{t+L+1} \longrightarrow S_{t+L+1} + c\sqrt{\hat{\sigma}_i^2 |r_{t+L}| S_{t+L}^2}\epsilon_{t+L+1}; \end{cases} \tag{47}$$

where ϵ_i are i.i.d. samples from $\mathcal{N}(0,1)$, and c is a hyperparameter to be tuned. While the theory suggests that c should be $1/2$, it is better to make it a tunable-parameter in algorithm design for better flexibility; $\hat{\sigma}_t$ is the instantaneous volatility, which can be estimated using standard methods in finance [9]. One might also assume $\hat{\sigma}$ into c.

5.1 Using return as inputs

Practically and theoretically, it is better and standard to use the returns $x = (r_t, ..., r_{t+L-1}, r_{t+L})$ as the input, and the algorithm can be applied in a simpler form:

$$\begin{cases} r_i \longrightarrow r_i + c\sqrt{\hat{\sigma}_i^2 |r_i|}\epsilon_i & \text{for } r_i \in x; \\ r_{t+L} \longrightarrow r_{t+L} + c\sqrt{\hat{\sigma}_i^2 |r_{t+L}|}\epsilon_{t+L+1}. \end{cases} \tag{48}$$

5.2 Equivalent Regularization on the output

One additional simplification can be made by noticing the effect of injecting noise to r_{t+L} on the training loss is equivalent to a regularization. Under the GBM model, the training objective can be written as

$$\arg\max_{b_t} \left\{ \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_z\left[G_t(\pi)\right] - \lambda c^2 \hat{\sigma}_t^2 |r_t| \pi_t^2 \right\}, \tag{49}$$

where the expectation over x is now only taken with respect to the input. This means that the noise injection on the r_{t+L} is equivalent to adding a L_2 regularization on the model output π_t. This completes the main proposed algorithm of this work. Also, it is well known that the magnitude of $|r_t|$ has strong time-correlation (i.e., a large $|r_t|$ suggests a large $|r_{t+1}|$) [6, 23], and this suggests that one can also use the average of the neighboring returns to smooth the $|r_t|$ factor in the last term for some time-window of width τ: $|r_t| \to |\hat{r}_t| = \frac{1}{\tau}\sum_0^\tau |r_{t-\tau}|$. In our S&P500 experiments, we use this smoothing technique with $\tau = 20$.

Derivation. We now derive Eq. (49). The original training loss is

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_t\left[G_t(\pi)\right] - \lambda \mathrm{Var}_t[G_t(\pi)]. \tag{50}$$

The last term can be written as

$$\begin{aligned} &\lambda \mathrm{Var}_t[G_t(\pi)] \\ &= \mathbb{E}_{z_1,...,z_t}[z_t^2\pi_t^2] - \mathbb{E}_{z_1,...,z_t}[z_t\pi_t]^2 \\ &= \mathbb{E}_{z_t}[z_t^2]\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t^2] - \mathbb{E}_{z_t}[z_t]^2\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t]^2 \\ &= \lambda r_t^2\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t^2] + \lambda c^2\hat{\sigma}_t^2|r_t|\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t^2] - \lambda r_t^2\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t]^2 \\ &= \lambda r_t^2 \mathrm{Var}_{z_1,...,z_{t-1}}[\pi_t] + \lambda c^2\hat{\sigma}_t^2|r_t|\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t^2] \end{aligned} \tag{51}$$

Plug in, this leads to the following maximization problem, which is the desired equation.

$$\arg\max_{b_t} \Bigg\{ \frac{1}{T}\sum_{t=1}^{T} \underbrace{\mathbb{E}_x\left[G_t(\pi)\right]}_{A\text{: wealth gain}} - \underbrace{\lambda r_t^2 \mathrm{Var}_{z_1,...,z_{t-1}}[\pi_t]}_{B\text{: Risk due to uncertainty in past price}} - \underbrace{\lambda c^2\hat{\sigma}_t^2|r_t|\mathbb{E}_{z_1,...,z_{t-1}}[\pi_t^2]}_{C\text{: Risk due to Future Price}} \Bigg\}, \tag{52}$$

where we have given each term a name for reference; the expectation is taken with respect to the augmented data points z_i:

$$r_i \longrightarrow z_i = r_i + c\sqrt{\hat{\sigma}_i^2|r_i|}\epsilon_i \quad \text{for } r_i \in x. \tag{53}$$

Under the GBM model (or when the optimal portfolio only weakly depends on $z_1, ..., z_{t-1}$), the optimal π_t does not depend on $z_1, ..., z_{t-1}$, and so the objective can be further simplified to be

$$\arg\max_{b_t} \left\{ \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_z\left[G_t(\pi)\right] - \lambda c^2 \hat{\sigma}_t^2 |r_t| \pi_t^2 \right\}; \tag{54}$$

the first term is the data augmentation for wealth gain, and the second term is a regularization for risk control. Most of the experiments in this paper use this equation for training. When it does not work well, the readers are encouraged to try the full training objective in Eq. (52).

Table 1: Sharpe ratio on S&P 500 by sectors; the larger the better. Best performances in Bold.

Industry Sectors	# Stock	Merton	no aug.	weight decay	additive aug.	naive mult.	proposed
Communication Services	9	$-0.06_{\pm 0.04}$	$-0.06_{\pm 0.04}$	$-0.06_{\pm 0.27}$	$\mathbf{0.22}_{\pm \mathbf{0.18}}$	$\mathbf{0.20}_{\pm \mathbf{0.21}}$	$\mathbf{0.33}_{\pm \mathbf{0.16}}$
Consumer Discretionary	39	$-0.01_{\pm 0.03}$	$-0.07_{\pm 0.03}$	$-0.06_{\pm 0.10}$	$0.48_{\pm 0.10}$	$0.41_{\pm 0.09}$	$\mathbf{0.64}_{\pm \mathbf{0.08}}$
Consumer Staples	27	$0.05_{\pm 0.03}$	$0.24_{\pm 0.03}$	$0.23_{\pm 0.11}$	$\mathbf{0.36}_{\pm \mathbf{0.08}}$	$\mathbf{0.34}_{\pm \mathbf{0.09}}$	$\mathbf{0.35}_{\pm \mathbf{0.07}}$
Energy	17	$0.07_{\pm 0.03}$	$0.03_{\pm 0.03}$	$-0.02_{\pm 0.12}$	$0.70_{\pm 0.09}$	$0.52_{\pm 0.10}$	$\mathbf{0.91}_{\pm \mathbf{0.10}}$
Financials	46	$-0.57_{\pm 0.04}$	$-0.61_{\pm 0.03}$	$-0.61_{\pm 0.09}$	$-0.06_{\pm 0.10}$	$-0.13_{\pm 0.09}$	$\mathbf{0.18}_{\pm \mathbf{0.08}}$
Health Care	44	$0.23_{\pm 0.04}$	$0.60_{\pm 0.04}$	$0.61_{\pm 0.11}$	$\mathbf{0.86}_{\pm \mathbf{0.09}}$	$\mathbf{0.81}_{\pm \mathbf{0.09}}$	$\mathbf{0.83}_{\pm \mathbf{0.07}}$
Industrials	44	$-0.09_{\pm 0.03}$	$-0.11_{\pm 0.03}$	$-0.11_{\pm 0.08}$	$0.36_{\pm 0.08}$	$0.28_{\pm 0.08}$	$\mathbf{0.48}_{\pm \mathbf{0.08}}$
Information Technology	41	$0.41_{\pm 0.04}$	$0.41_{\pm 0.04}$	$0.41_{\pm 0.11}$	$0.67_{\pm 0.10}$	$\mathbf{0.74}_{\pm \mathbf{0.11}}$	$\mathbf{0.79}_{\pm \mathbf{0.09}}$
Materials	19	$0.07_{\pm 0.03}$	$0.06_{\pm 0.03}$	$0.03_{\pm 0.14}$	$\mathbf{0.47}_{\pm \mathbf{0.13}}$	$\mathbf{0.43}_{\pm \mathbf{0.13}}$	$\mathbf{0.53}_{\pm \mathbf{0.10}}$
Real Estate	22	$-0.14_{\pm 0.04}$	$-0.39_{\pm 0.03}$	$-0.40_{\pm 0.12}$	$0.05_{\pm 0.10}$	$0.05_{\pm 0.09}$	$\mathbf{0.19}_{\pm \mathbf{0.07}}$
Utilities	24	$-0.29_{\pm 0.02}$	$-0.29_{\pm 0.02}$	$-0.28_{\pm 0.07}$	$-0.01_{\pm 0.06}$	$-0.00_{\pm 0.06}$	$\mathbf{0.15}_{\pm \mathbf{0.04}}$
S&P500 Avg.	365	$-0.02_{\pm 0.04}$	$-0.00_{\pm 0.04}$	$-0.01_{\pm 0.04}$	$0.39_{\pm 0.03}$	$0.35_{\pm 0.03}$	$\mathbf{0.51}_{\pm \mathbf{0.03}}$

6 EXPERIMENTS

We validate our theoretical claim that using multiplicative noise with strength $\sqrt{r}$ is better than not using any data augmentation or using a data augmentation that is not suitable for the nature of portfolio construction (such as an additive Gaussian noise). We emphasize that the purpose of this section is for demonstrating the relevance of our theory to real financial problems, not for establishing the proposed method as a strong competitive method in the industry. We start with a toy dataset that follows the theoretical assumptions and then move on to real data with S&P500 prices. For all the tasks, we observe a single trajectory of a single stock prices $S_1, ..., S_T$. For the toy tasks, $T = 400$; for the S&P500 task, $T = 800$. We then transform this into $T - L$ input-target pairs $\{(x_i, y_i)\}_{i=1}^{T-L}$, where

$$\begin{cases} x_i = (S_i, ..., S_{L-1}); \\ y_i = S_L. \end{cases} \tag{55}$$

x_i is used as the input to the model for training; y_i is used as the unseen future price for calculating the loss function. For the toy tasks, $L = 10$; for the S&P500 task, $L = 15$. In simple words, we use the most recent L prices for constructing the next-step portfolio. Unless otherwise specified, we use a feedforward neural network with the number of neurons $10 \to 64 \to 64 \to 1$ with ReLU activations. Training proceeds with the Adam optimizer with a minibatch size of 64 for 100 epochs with the default parameter settings.[6]

We use the Sharpe ratio as the performance metric (the larger the better). Sharpe ratio is defined as $SR_t = \frac{\mathbb{E}[\Delta W_t]}{\sqrt{\mathrm{Var}[\Delta W_t]}}$, which is a measure of the profitability per risk. We choose this metric because, in the framework of portfolio theory, it is the only theoretically motivated metric of success [30]. In particular, our theory is based on the maximization of the mean-variance utility in Eq. (3) and it is well-known that the maximization of the mean-variance utility is equivalent to the maximization of the Sharpe ratio. In fact, it is a classical result in classical financial research that all optimal strategies must have the same Sharpe ratio [29] (also called the efficient capital frontier). For the synthetic tasks, we can generate arbitrarily many test points to compare the Sharpe ratios unambiguously. We then move to experiments on real stock price series; the limitation is that the Sharpe ratio needs to be estimated and involves one additional source of uncertainty.

[6]In our initial experiments, we also experimented with different architectures (different depth or width of the FNN, RNN, and LSTM), and our conclusion that the proposed augmentation outperforms the specified baselines remain unchanged. We also tentatively compared with a QuantGAN-based approach [33], where we first train a QuantGAN on the raw data and generate data augmentations using the QuantGAN, and train a neural portfolio based on the augmented data. This approach achieves a Sharpe ratio of roughly 0.05 and does not seem to outperform the baseline significantly. We hypothesize that this is because the QuantGAN method is not designed for Sharpe ratio maximization.

6.1 Geometric Brownian Motion

We first start from experimenting with stock prices generated with a GBM, as specified in Eq. (6), and we generate a fixed price trajectory with length $T = 400$ for training; each training point consists of a sequence of past prices $(S_t, ..., S_{t+9}, S_{t+10})$ where the first ten prices are used as the input to the model, and S_{t+10} is used for computing the loss.

Results and discussion. See Figure 2. The proposed method is plotted in blue. The right figure compares the proposed method with the other two baseline data augmentations we studied in this work. As the theory shows, the proposed method is optimal for this problem, achieving the optimal Sharpe ratio across a 600-step trading period. This directly confirms our theory.

6.2 S&P 500 Prices

This section demonstrates the relevance of the proposed algorithm to real market data. In particular, we use the data from S&P500 from 2016 to 2020, with 1000 days in total. We test on the 365 stocks that existed on S&P500 from 2000 to 2020. We use the first 800 days as the training set and the last 200 days for testing. The model and training setting is similar to the previous experiment. We treat each stock as a single dataset and compare on all of the 365 stocks (namely, the evaluation is performed independently on 365 different datasets). Because the full result is too long, we report the average Sharpe ratio per industrial sector (categorized according to GISC) and the average Sharpe ratio of all 365 datasets.

Results and discussion. See Table 1. We see that, without data augmentation, the model works poorly due to its incapability of assessing the underlying risk. We also notice that weight decay does not improve the performance (if it is not deteriorating the performance). We hypothesize that this is because weight decay does not correctly capture the inductive bias that is required to deal with a financial series prediction task. Using any kind of data

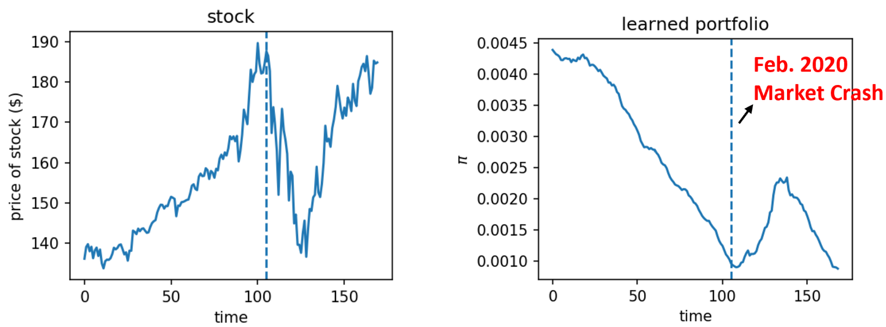

Figure 3: Case study of the performance of the model on MSFT from 2019 August to 2020 May. We see that the model learns to invest less and less as the price of the stock rises to an unreasonable level, thus avoiding the high risk of the market crash in February 2020.

augmentation seems to improve upon not using data augmentation. Among these, the proposed method works the best, possibly due to its better capability of risk control. In this experiment, we did not allow for short selling. We also compare with the Merton's portfolio [27], which is the classical optimal stationary portfolio constructed from the training data; this method does not perform well either. This is because the market during the time 2019−2020 is volatile and quite different from the previous years, and a stationary portfolio cannot capture the nuances in the change of the market condition. This shows that it is also important to leverage the flexibility and generalization property of the modern neural networks, along side the financial prior knowledge.

6.3 Case Study

In this section, we qualitatively study the behavior of the learned portfolio of the proposed method. The model is trained as in the other S&P500 experiments. See Figure 3. We see that the model learns to invest less and less as the stock price rises to an excessive level, thus avoiding the high risk of the market crash in February 2020. This avoidance demonstrates the effectiveness of the proposed method qualitatively.

7 OUTLOOK

In this work, we have presented a theoretical framework relevant to finance and machine learning to understand and analyze methods related to deep-learning-based finance. The limitation of the present work is obvious; we only considered the kinds of data augmentation that takes the form of noise injection. Other kinds of data augmentation may also be useful to the finance; for example, [12] empirically finds other helpful other data augmentation schemes besides what we have studied here, and it is interesting to apply our theoretical framework to analyze these methods as well; a correct theoretical analysis of these methods is likely to advance both the deep-learning based techniques for finance and our fundamental understanding of the underlying financial and economic mechanisms. Meanwhile, our understanding of the underlying financial dynamics is also rapidly advancing; we foresee better methods to be designed, and it is likely that the proposed method will be replaced by better algorithms soon. The cautionary note is that this work is only for the purpose of academic research, and should not be taken as an advice for monetary investment, and the readers should evaluate their own risk when applying the proposed method.

REFERENCES

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. 2017. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340* (2017).
[2] Samuel Assefa. 2020. Generating synthetic data in finance: opportunities, challenges and pitfalls. *Challenges and Pitfalls (June 23, 2020)* (2020).
[3] Fischer Black and Myron Scholes. 1973. The pricing of options and corporate liabilities. *Journal of political economy* 81, 3 (1973), 637–654.
[4] Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. 2019. Deep hedging. *Quantitative Finance* 19, 8 (2019), 1271–1291.
[5] Rama Cont. 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance* (2001).
[6] Rama Cont. 2007. Volatility clustering in financial markets: empirical facts and agent-based models. In *Long memory in economics*. Springer, 289–309.
[7] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA.
[8] Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. 2019. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*. PMLR, 1528–1537.
[9] Stavros Degiannakis and Christos Floros. 2015. Methods of volatility estimation and forecasting. *Modelling and Forecasting High Frequency Financial Data* (2015), 58–109.
[10] Adrian A Drăgulescu and Victor M Yakovenko. 2002. Probability distribution of returns in the Heston model with stochastic volatility. *Quantitative finance* 2, 6 (2002), 443–453.
[11] Eugene F Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The journal of Finance* 25, 2 (1970), 383–417.
[12] Elizabeth Fons, Paula Dawson, Xiao jun Zeng, John Keane, and Alexandros Iosifidis. 2020. Evaluating data augmentation for financial time series classification. arXiv:2010.15111 [q-fin.ST]
[13] Charles AE Goodhart and Maureen O'Hara. 1997. High frequency data in financial markets: Issues and applications. *Journal of Empirical Finance* 4, 2-3 (1997), 73–114.
[14] Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2019. Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data. arXiv:1909.09148 [cs.LG]
[15] Steven L Heston. 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies* 6, 2 (1993), 327–343.
[16] Kentaro Imajo, Kentaro Minami, Katsuya Ito, and Kei Nakagawa. 2020. Deep Portfolio Optimization via Distributional Prediction of Residual Factors. arXiv:2012.07245 [q-fin.PM]
[17] Shota Imaki, Kentaro Imajo, Katsuya Ito, Kentaro Minami, and Kei Nakagawa. 2021. No-Transaction Band Network: A Neural Network Architecture for Efficient Deep Hedging. *Available at SSRN 3797564* (2021).
[18] Katsuya Ito, Kentaro Minami, Kentaro Imajo, and Kei Nakagawa. 2020. Trader-Company Method: A Metaheuristic for Interpretable Stock Price Prediction. arXiv:2012.10215 [q-fin.TR]
[19] Patel Jay, Vasu Kalariya, Pushpendra Parmar, Sudeep Tanwar, Neeraj Kumar, and Mamoun Alazab. 2020. Stochastic neural networks for cryptocurrency price prediction. *IEEE Access* 8 (2020), 82804–82818.

[20] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059* (2017).

[21] John L Kelly Jr. 2011. A new interpretation of information rate. In *The Kelly capital growth investment criterion: theory and practice*. World Scientific, 25–34.

[22] Bryan Lim, Stefan Zohren, and Stephen Roberts. 2019. Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science* 1, 4 (2019), 19–38.

[23] Thomas Lux and Michele Marchesi. 2000. Volatility clustering in financial markets: a microsimulation of interacting agents. *International journal of theoretical and applied finance* 3, 04 (2000), 675–702.

[24] Benoit B Mandelbrot. 1997. The variation of certain speculative prices. In *Fractals and scaling in finance*. Springer, 371–418.

[25] Harry Markowitz. 1959. Portfolio selection.

[26] Ian WR Martin and Stefan Nagel. 2022. Market efficiency in the age of big data. *Journal of Financial Economics* 145, 1 (2022), 154–177.

[27] Robert C Merton. 1969. Lifetime portfolio selection under uncertainty: The continuous-time case. *The review of Economics and Statistics* (1969), 247–257.

[28] Mark Rubinstein. 2002. Markowitz's" portfolio selection": A fifty-year retrospective. *The Journal of finance* 57, 3 (2002), 1041–1045.

[29] William F Sharpe. 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance* 19, 3 (1964), 425–442.

[30] William F Sharpe. 1966. Mutual fund performance. *The Journal of business* 39, 1 (1966), 119–138.

[31] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.

[32] John Von Neumann and Oskar Morgenstern. 1947. Theory of games and economic behavior, 2nd rev. (1947).

[33] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. 2020. Quant gans: Deep generation of financial time series. *Quantitative Finance* 20, 9 (2020), 1419–1440.

[34] Zihao Zhang, Stefan Zohren, and Stephen Roberts. 2020. Deep Learning for Portfolio Optimization. *The Journal of Financial Data Science* 2, 4 (Aug 2020), 8–20. https://doi.org/10.3905/jfds.2020.1.042

[35] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13001–13008.

[36] Liu Ziyin, Tilman Hartwig, and Masahito Ueda. 2020. Neural networks fail to learn periodic functions and how to fix it. *arXiv preprint arXiv:2006.08195* (2020).

[37] L Ziyin, Z Wang, P Liang, R Salakhutdinov, L Morency, and M Ueda. 2019. Deep Gamblers: Learning to Abstain with Portfolio Theory.. In *Proceedings of the Neural Information Processing Systems Conference*.