IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 April 2023, accepted 5 June 2023, date of publication 13 June 2023, date of current version 21 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3285613

RESEARCH ARTICLE

HADAPS: Hierarchical Adaptive Multi-Asset Portfolio Selection

JINKYU KIM, DONGHEE CHOI, MOGAN GIM, AND JAEWOO KANG
Department of Computer Science and Engineering, Korea University, Seoul 02841, South Korea

Corresponding author: Jaewoo Kang (kangj@korea.ac.kr)

This work was supported in part by the National Research Foundation of Korea under Grant NRF-2023R1A2C3004176; and in part by the Ministry of Science and ICT (MSIT), South Korea, under the ICT Creative Consilience Program, supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2023-2020-0-01819.

ABSTRACT Multi-asset portfolio selection is an asset allocation strategy involving a variety of assets. Adaptive investment strategies which consider the dynamic market characteristics of individual assets and asset classes are vital for maximizing returns and minimizing risks. We introduce *HADAPS*, a novel computational method for multi-asset portfolio selection which utilizes the Soft-Actor-Critic (SAC) framework enhanced with Hierarchical Policy Network. Contrary to previous approaches that have relied on heuristics for constructing asset allocations, *HADAPS directly* outputs a continuous vector of action values depending on current market conditions. In addition, *HADAPS* performs multi-asset portfolio selection involving *multiple* asset classes. Experimental results show that *HADAPS* outperforms baseline approaches in not only cumulative returns but also risk-adjusted metrics. These results are based on market price data from sectors with various behavioral characteristics. Furthermore, qualitative analysis shows *HADAPS*' ability to adaptively shift portfolio selection strategies in dynamic market conditions where asset classes and different assets are uncorrelated to each other.

INDEX TERMS Portfolios, investment, stock markets, cryptocurrency, reinforcement learning.

I. INTRODUCTION

Portfolio selection is an investment strategy that seeks a combination of assets best satisfying an invester's needs under uncertain market circumstances [1], [2]. The goal of portfolio selection is maximizing returns while minimizing risks through asset diversification [3], [4], [5]. Previous researches in finance domain have attempted to construct diversified portfolios with uncorrelated assets considering their individual returns and volatilities (e.g., different national markets [6], [7], [8] or different asset classes [9]). To reduce the complexity of investing in various assets with multiple asset classes, [10], [11] suggested using hierarchical decision-making systems. However, such approaches still have not fully overcome burdens associated with multi-asset portfolio selection where it requires massive amount of time and discretion.

To solve the difficulties of portfolio selection, deep learning methods, especially reinforcement learning approaches, were proposed to deal with non-stationary and uncertain characteristics of market conditions [12]. Moreover, deep reinforcement learning has contributed to improving automatic portfolio selection tasks [12], [13], [14]. However, such previous methods rely on heuristic decision layers utilized in value-based networks [14], [15]. This raises a need for a reinforcement learning framework that learns to *directly* determine actions in multi-asset portfolio selection with the goal of better investment outcomes.

We introduce *HADAPS* which is a reinforcement learning model based on Soft Actor Critic (SAC) framework [16]. The SAC framework enables our model to directly determine the asset-wise proportions in a portfolio represented as continuous action values. To facilitate adaptive allocation strategies, we made the following modifications to the SAC framework. We replaced the policy network with our Hierarchical Policy Network consisting the Intra-Class Asset and Inter-Class Asset Layers. We also devised a novel parameter

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrahmane Lakas.

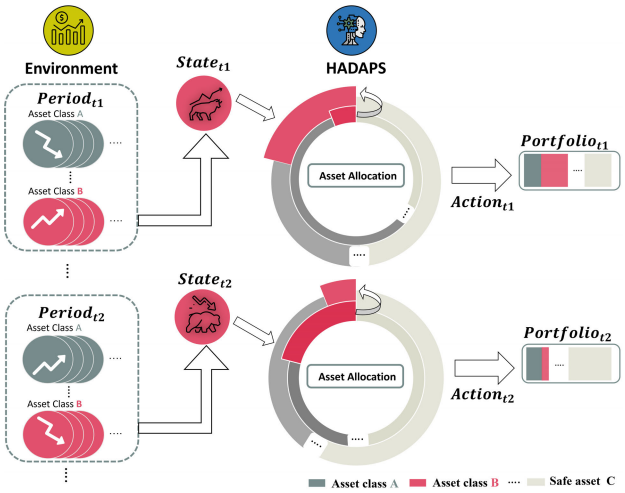


FIGURE 1. Overview of *HADAPS*. *HADAPS* make a decision of a multi-asset portfolio selection by capturing the states of markets consisting of multi-asset classes. According to the changes in Asset class B's prices illustrated on the left side, *HADAPS* increases the asset class B's allocation when the model classifies it as a bull market compared to other asset classes as shown in the upper part on the right side.

called Action Scale which is applied to the output values of policy networks.

In order to comprehensively evaluate the performance of *HADAPS* in the portfolio selection task, we conducted a series of experiments with three distinct groups, each comprising of a combination of assets. The assets in scenarios included a safe asset, such as the U.S. dollar, a moderate but important asset, represented by the U.S. stock market, and a more aggressive asset, exemplified by cryptocurrency, to provide a thorough assessment of *HADAPS*'s investment performance on assets with various volatilities. To the best of our knowledge, it is our first attempt to apply reinforcement learning techniques to portfolio investment involving heterogeneous asset classes.

Our key contributions are as follows:

- To the best of our knowledge, we utilized the Soft Actor Critic framework in the multi-asset portfolio selection task for the first time.
- We developed *HADAPS* by enhancing SAC framework with our novel Hierarchical Policy Network Layer and Action Scale parameter.
- We conducted experiments with three groups of scenarios including stocks and cryptos price data to compare *HADAPS*' investment performance with other multi-asset porfolio selection methods. Results show that *HADAPS* outperforms all baselines in not only cumulative returns but also risk-adjusted metrics.

II. RELATED WORK

A. ALGORITHMIC APPROACHES FOR PORTFOLIO SELECTION

Early Markowitz approach [1] suggested making decisions on portfolio selection by formulating a heuristic model through exploitation of expected return and risk indicators. Subsequently, other researches have attempted to compose different types of asset classes (e.g., different regional characteristics [6] and equity groups [9]) for better investment return. Furthermore, previous studies improved the Markowitz model by adopting other parametric mathematical methods such as correlation [17] and regime switching models [18].

TABLE 1. Table of notations.

Symbol	Description
$\mathcal{B}$	Assets
$\mathcal{C}$	A Group of asset classes
p	Price
v	Volatility of Price
k	Window size of state
T	Time frame.
$\mathbf{y}$	Portfolio selection vector.
m	The number of all assets
n	The number of all asset classes
$\mathbf{\Delta}$	Return
r	Reward
β	Action Scale
π_ϕ	Policy network.

B. DEEP REINFORCEMENT LEARNING FOR PORTFOLIO SELECTION

Despite many attempts based on parametric portfolio selection methods, uncertain characteristics of markets brought the necessity of creating generalized agents for automatic portfolio selection. Particularly in finance domain, deep learning methods have shown promising results in forecasting market states based past historical states [15], [19], [20], [21], [22], [23].

Prior automatic portfolio selection commonly relied on Deep Q Networks (DQN) [15] which is a reinforcement learning framework based on Value Network. The advent of SAC framework in deep reinforcement learning has brought improvement in other domain-specific downstream tasks which provides rationale for us to apply it in multi-asset portfolio selection.

Previous approaches have engaged in multi-asset portfolio selection involving *homogeneous* assets in the same asset class such as Stocks [22], [23], [24] or Cryptos [13], [14]. Distinguishable from other approaches, our work introduces a multi-asset portfolio selection method applied with the SAC framework applicable with *heterogeneous* asset classes.

III. PRELIMINARIES

The following details are the preliminaries for Multi-Asset Portfolio Selection task and its applied deep reinforcement learning framework.

Task 1: (**Multi-Asset Portfolio Selection**) Suppose a heterogeneous set $\mathcal{B}$, containing m Assets which can be categorized into n disjoint Asset Classes $\mathcal{C}$, where $\mathcal{B} = (\mathcal{B}_1 \cup \mathcal{B}_2 \cdots \cup \mathcal{B}_n)$, $|\mathcal{B}| = m$, $|\mathcal{C}| = n$ are given. Given the Assets

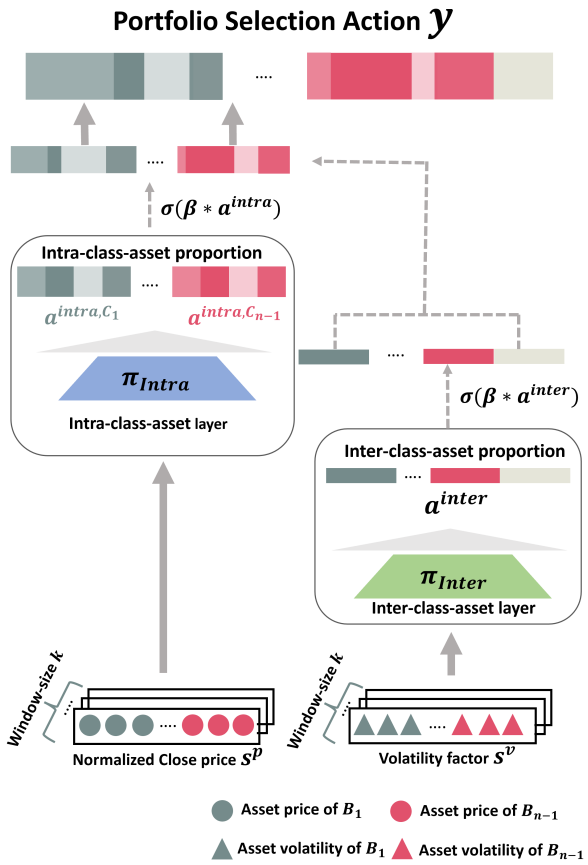


FIGURE 2. Model architecture of *HADAPS*. s^p and s^v are the input of *HADAPS*, which representation of a current market state. *HADAPS* make a decision of a portfolio selection action y for given a state (s^p, s^v).

above, we define **Multi-Asset Class Portfolio Selection** as constructing a portfolio selection action $\mathbf{y} \in \mathcal{R}^m$ of which objective is to maximize the Cumulative Reward in $\mathcal{R}$ over time frame T.

As our method involves the SAC enhanced with hierarchical policy network layer, we break down the **Multi-Asset Portfolio Selection** task into two sub-tasks.

Subtask 1: (**Inter-Class Portfolio Selection**) Given a set of Asset Classes $\mathcal{C}$, we define **Inter-Class Portfolio Selection** as constructing a portfolio selection action $\mathbf{y}^{\mathcal{C}} \in \mathcal{R}^n$ consisting a Class-wise allocation of n investments.

Subtask 2: (**Intra-Class Portfolio Selection**) Given a set of Assets within an Asset Class $\mathcal{B}_i$ where $i \in \{1, 2, \cdots, n\}$, we define **Intra-Class Portfolio Selection** as constructing a portfolio selection action $\mathbf{y}^i \in \mathcal{R}^{m_i}$ where $m_i = |\mathcal{B}_i|$. $\mathbf{y}^i$ consists of an Asset-wise allocation of m_i investments.

For applying reinforcement learning framework to our Multi-Asset Portfolio Selection Task, we define the environment and agent as follows:

- Environment: The environment produces a new state $(\mathbf{s}^p_{t+1}, \mathbf{s}^v_{t+1})$ and a reward r_t with given $\mathbf{y}_t$ for every time step t.

Algorithm 1 *HADAPS*Training Algorithm

Initialize θ, π
$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay buffer
for each iteration **do**
  for each environment step $t \in (k-1, T)$ **do**
    $\mathbf{a}^{inter}_t \sim \pi_{\phi_{inter}}(\mathbf{a}^{inter}_t | \mathbf{s}^v_t)$
    $\mathbf{a}^{inter}_t = \beta \times \mathbf{a}^{inter}_t$ ▷ Section IV-D
    $\mathbf{a}^{inter}_t = \sigma(\mathbf{a}^{inter}_t)$ ▷ σ is a softmax function.
    for each asset class $c_i \in \{c_1, \cdots, c_n\}$ **do**
      $\mathbf{a}^{intra,c_i}_t \sim \pi_{\phi_{intra,i}}(\mathbf{a}^{intra,c_i}_t | \mathbf{s}^p_t)$
      $\mathbf{a}^{intra,c_i}_t = \beta \times \mathbf{a}^{intra,c_i}_t$ ▷ Section IV-D
      $\mathbf{a}^{intra,c_i}_t = \sigma(\mathbf{a}^{intra,c_i}_t)$
      $\mathbf{a}^{c_i}_t = \mathbf{a}^{intra,c_i}_t \times \mathbf{a}^{inter,i}_t$
    end for
    $\mathbf{y}_t = Concat(\mathbf{a}^{c_1}_t, \cdots, \mathbf{a}^{c_n}_t)$ ▷ $\mathbf{y_t} \in \mathcal{R}^m$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{y}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}^p_t, \mathbf{s}^v_t, \mathbf{y}_t, r_t(s^p_t, y_t), \mathbf{s}^p_{t+1}, \mathbf{s}^v_{t+1})\}$
  end for
  Update θ using gradient decent using $\mathcal{D}$
end for

- Agent: The agent in portfolio selection task generates the portfolio selection action $\mathbf{y}_t$ for given state $(\mathbf{s}^p_t, \mathbf{s}^v_t)$ with the objective of maximizing the cumulative rewards.

To apply the deep reinforcement learning framework to our formulated Multi-Asset Portfolio Selection task, we define the following components $(\mathcal{S}^p, \mathcal{S}^v, \mathcal{Y}, \mathcal{R})$ in MDP which are Price State, Volatility State, Portfolio Selection Action, Reward respectively. The following definitions also include t and k which are the current timestamp and rolling window size.

- Price State $\mathbf{s}^p_t$: For each Asset, the Price State is defined as $\mathbf{s}^p_t = [p_{t-1}, \cdots, p_z, \cdots, p_{t-1-k}]$, where p_z is a price of an asset at time z and k is a window size. All prices p were normalized by the mean and standard deviation value calculated only within the training period.
- Volatility State $\mathbf{s}^v_t$: For each Asset, the Volatility State is defined as $\mathbf{s}^v_t = [v_{t-1}, \cdots, v_z, \cdots, v_{t-1-k}]$, where v_z denotes a standard deviation of previous prices from timestamp z to $z-k$ multiplied by ϵ where $\epsilon = 1$ if $(p_z - p_{z-1}) \geq 0$ else $\epsilon = -1$.
- Portfolio Selection Action $\mathbf{y}_t$: For all Assets at timestamp t, an Action is the agent's decision for the above-mentioned **Multi-Asset Class Portfolio Selection Action** $\mathbf{y_t}$ which is built based on $\mathbf{a}^{inter}_t \in \mathcal{R}^n$ in **Inter-Class Portfolio Selection** and $\mathbf{a}^{intra}_t \in \mathcal{R}^{|B_i|}$ in **Intra-Class Portfolio Selection** where $i \in \{1, 2, \cdots, n\}$.
- Reward $r_t(\mathbf{s}^p_t, \mathbf{s}^v_t, \mathbf{y}_t)$: For all m Assets ($m = |\mathcal{B}|$) at timestamp t, Reward is computed based on the agent's decision given the Price State $\mathbf{s}^p_t$ and Volatility State $\mathbf{s}^v_t$ and their corresponding $\mathbf{y}_t$. The reward is calculated based on the summation of Asset-wise products between returns and $\mathbf{y}_t$. The Asset-wise return is defined as

$\Delta \in \mathcal{R}^m$ since there are m Assets as defined above in our **Multi-Asset Class Portfolio Selection**. Each Asset-wise return at timestamp t is calculated based on the rate of price change from timestamp to t to timestamp $t-1$. The calculation of Asset-wise returns Δ_t timestamp t is mathematically expressed as follows:

$$\delta_{z,t} = \frac{p_{z,t} - p_{z,t-1}}{p_{z,t}}, \delta_{z,t} \in \Delta_t \tag{1}$$

where $p_{z,t}$ is the price at timestamp t, asset z.

Subsequently, the calculation of Reward r_t for all m Assets is mathematically expressed as follows:

$$r_t(\delta_t, y_t) = 100 \times \sum_{z=1}^{m} (\delta_{z,t} \cdot y_{z,t}) \tag{2}$$

where $y_{z,t}$ is the resultive portfolio selection action for the zth Asset at timestamp t.

Using the above definition of MDPs, we define a model's policy π_ϕ at given time t.

- Policy $\pi_\phi(\mathbf{y}_t|\mathbf{s}_t^p, \mathbf{s}_t^v)$: An agent's decision based on current state at time t of composing $\mathbf{y}_t$ among given assets. We set the goal of an agent as maximizing cumulative rewards.

IV. HADAPS

A. ARCHITECTURE

HADAPS is a hierarchical adaptive multi-asset portfolio selection system which is constructed based on the SAC [16] framework where the agent generates portfolio $\mathbf{y}_t$ at timestamp t given state $(\mathbf{s}_t^p, \mathbf{s}_t^v)$.

The SAC framework is an off-policy actor-critic method using the maximum entropy reinforcement learning framework. Within the framework, the agent is trained to directly learn to make decisions on multi-dimensional continuous action values with stability [16], [25]. The SAC framework consists two different networks which are Q Network and Hierarchical Policy Network. All layers used in both Q network and Policy Network use a five-layered MLP which is mathematically expressed as,

$$L = \delta(Dropout(BN(Linear(X))) \tag{3}$$

$$MLP(X) = L_5(L_4(L_3(L_2(L_1(X))))) \tag{4}$$

where $X \in \mathcal{R}^{z \times k}$, BN stands for batch normalization, z is a number of assets or asset classes, k is a window size. And the output dimensions for L_1, L_2, L_3, L_4, L_5 are 256, 128, 64, 32, 16, respectively.

The Q network in *HADAPS* evaluates the current state s_t and y_t estimating the expected reward to guide the decision of asset allocation layers. The mathematical formulation of the Q network is as follows,

$$Q_y(Y) = MLP_{q_y}(Y) \tag{5}$$

$$Q_s(S) = MLP_{q_s}(S) \tag{6}$$

$$O_q(Y, S) = Concat(Q_y(Y), Q_s(S)) \tag{7}$$

$$Q_\theta(Y, S) = Linear_q(ReLU(O_q(Y, S))) \tag{8}$$

B. HIERARCHICAL POLICY NETWORK

The Hierarchical Policy Network comprises Asset Allocation Layers (Inter-Class and Intra-Class Asset Layer) with an additional Action Scale parameter. Our intuition for this design approach aligns with a robust investment strategy that involves making investment decisions in a hierarchical manner. The Inter-Class layer was designed to help the model make allocations on asset classes based on its understanding in the overall market situation while the Intra-Class layer was designed to subsequently lead the model to make fine-grained decisions on individual assets within the same class based on their priority.

We set this layer to decide asset allocation via predicting proper Portfolio Selection Action $\mathbf{y}_t$ for each assets in $\mathcal{B}$. The Inter-Class Asset Layer is mathematically expressed as follows,

$$\pi_{\phi_{inter}}(s^v) = tanh(Linear_{\pi_{inter}}(MLP_{\pi_{inter}}(s^v))) \tag{9}$$

Meanwhile the Intra-Class Asset Layers for each $c_i \in (c_1, c_2, \cdots, c_n)$ is mathematically expressed as follows,

$$\pi_{\phi_{intra,c_i}}(s^p) = tanh(Linear_{\pi_{intra,c_i}}(MLP_{\pi_{intra,c_i}}(s^p))) \tag{10}$$

where the output dimension for $\pi_{\phi_{inter}}$, $\pi_{\phi_{intra,c_i}}$ are $|\mathcal{C}|$, $|\mathcal{B}_i|$, respectively. We remark that for an asset class z when $|\mathcal{B}^z| = 1$, we removed the corresponding Intra-Class Asset Layer and where the Inter-Class Portfolio Action value for c_z is directly used for computing the output. The hyperbolic tangent *tanh* was included in every last part of each layer for training stability [25].

C. TRAINING ALGORITHM

As shown in Algorithm 1, we initialize the replay buffer and the layer-wise parameters of Q network and policy networks in *HADAPS* (Section IV-A). For every environment step t, we let the agent explore the environment by allocating $\mathbf{y}_t$. While *HADAPS* consists Inter-Class and Intra-Class Asset Layers, *HADAPS* determines its action in two steps at each time t based on the outputs from the layers. Given volatility state $\mathbf{s}_t^v$, *HADAPS* sets the proportions of each asset class in $\mathcal{C}$ by using the policy network $\pi_{\phi_{inter}}$ in from line 5 to 7. Meanwhile, *HADAPS* also selects the proportions of each asset in the asset class in $\mathbf{a}_{intra,c_i}$ using the given price state $\mathbf{s}_t^p$ from line 9 to 11. The proportions of each asset are multiplied by the proportions of the asset classes belonging to in line 12.

In SAC, the training objective is based on Maximum Entropy objective which enables *HADAPS* to explore continuous action spaces with the stochastic policy. The objective function comprises two parts which are entropy and reward:

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D}\big[\mathbb{E}_{y_t \sim \pi_\phi}[\alpha \log(\pi_\phi(y_t|s_t^p, s_t^v)) - r(s_t^p, s_t^v, y_t)\big] \tag{11}$$

where $\log(\pi_\phi(y_t|s_t^p, s_t^v))$, $r(s_t^p, s_t^v, y_t)$, α describes entropy, reward and temperature parameter, respectively. The temperature parameter determines the relative importance between reward and entropy.

For the reward part, we use off-policy double Q function methods used in [16]. And for the entropy part, we use a tractable policy network π_ϕ. We sample y_t for time step t from Gaussian Distribution parameterized by μ_t, σ_t which are the outputs from the policy network π_ϕ. The implementation details for the Q and policy networks are described in IV-B.

D. ACTION SCALE PARAMETER

Note that the last parts of every policy network layers are hyperbolic tangent *tanh* which leads to the values in μ_t and δ_t range from -1 to 1. To make y_t as a proportion of assets, we applied softmax function σ [14], [23], [24].

Note that *tanh* and σ tend to shrink the magnitudes of exploring steps sampled from Gaussian Distributions parameterized by the outputs of policy networks. We empirically show that such characteristics hinder *HADAPS*'s adaptive training in Section VI-B. To circumvent such issues, we introduce a novel parameter called *Action Scale*. The Action Scale remedies the shrinking effects of *tanh* and σ by amplifying the stochastic exploring steps from *HADAPS*'s Hierarchical Policy Network π_ϕ. We refer this to hyper-parameter β described in Algorithm 1 line 6 and 10.

V. EXPERIMENTS SETTINGS

We use gradient decent algorithm for training *HADAPS* with RMSProp and MSELoss. Also we use 1e-5, 0.3, 300, 4000, 32 for the learning rate, dropout rate, maximum epoch, maximum buffer size, and batch size, respectively.

A. DATASET AND EVALUATION

In our experiments, we selected three asset classes which are stocks, cryptos and cash as safe asset. We formulated three types of experimental scenarios containing a pair of asset groups consisting of five assets with cash to show *HADAPS*' robust investment performance. The selection criteria aligns with our motivation to investigate/evaluate *HADAPS*' investment strategy in various market environments.

- **Crypto&Stock**: We selected the top five crypto assets except stable coins[1] by their market cap as of June 26, 2022.[2] All crypto price data were last gathered prices for each day from CoinMarketCap.[3] Also, we selected the top five U.S. stocks in Nasdaq by their market cap as of June 26, 2022. [4] All stock price data were closing price from Yahoo Finance.[5]
- **Stock**$^{v1\uparrow}$**&Stock**$^{v2\uparrow}$: We employed a selection criterion of high volatility during the training periods to identify the top two stock sectors, each comprising of five stocks. Top 1 and 2 sectors are the Consumer service[6] and Health service,[7] respectively. All of the sectors in U.S. stocks are from Tradingview.[8]
- **Stock**$^{v1\downarrow}$**&Stock**$^{v2\downarrow}$: We chose the bottom two sectors with the lowest volatility, selecting five stocks from each sector, based on training period data. Bottom 1 and 2 sectors are the Communications[9] and Non energy minerals,[10] respectively.

We set one of the asset classes as a safe asset such as the U.S. dollar for all three scenarios. *HADAPS* gets an opportunity to invest in safe assets to minimize the risks of a portfolio when non-safe asset classes are in a bear market.

For **Crypto&Stock** to simulate and evaluate *HADAPS*'s understanding of the dynamic behavior of stock and crypto prices and its robustness on unseen future circumstances, we adopted time series cross validation on a rolling basis [26], [27], [28]. Given four years of price data contained in our dataset, we used the first three years (2018, 2019, 2020) for searching the hyperparameters and model choices of *HADAPS* where years 2018 and 2019 are used as training period and 2020 are used for validating *HADAPS*' performance on maximizing Cumulative Return. After fixing the best hyperparameters, we used the last three years (2019, 2020, 2021) of price data by re-training *HADAPS* with years 2019 and 2020, and testing it on the remaining year 2021 price data. The same dataset split and model validation scheme was applied to the baseline models as well. For each validation and test year, we split the year into twelve periods to measure the robustness of models by averaging the metric values for all periods. Before every inference periods, *HADAPS* learns the investment strategy using prior two years of training period. Therefore, the time frame (**T**) for training and inference period are set to two years and one month respectively.

In order to supplement the limited amount of price data available for the cryptocurrency market, we conducted additional experimental scenarios (**Stock**$^{v1\uparrow}$**&Stock**$^{v2\uparrow}$, **Stock**$^{v1\downarrow}$**&Stock**$^{v2\downarrow}$) utilizing longer time frames. The stock price data utilized in these experiments was sourced from the U.S. stock market of which the time period is from April 1st, 2008 to December 31st, 2017.

We used the metrics including Cumulative Return (CR %), Sharpe (Sha), Sortino (Sor), and Omega (Ome) [29] to evaluate *HADAPS*'s test performance on the unseen price data for each settings. $\sqrt{252}$ is used as a multiplier for the Sharp and Sortino. The Sortino ratio exploits only the negative deviation of a portfolio's reward, it gives investors a better view of a portfolio's risk-adjusted performance since positive volatility leads to better rewards. For the Omega ratio, we set *threshold* $= 0$ as a risk-free asset. While the Omega ratio uses the exact values from gains and losses, it does not depend on estimators from specific distributions which can

[1] Tether, USD Coin and Binance USD.
[2] The selected cryptos are BTC (Bitcoin), ETH (Ethereum), BNB (Binance Coin), XRP (Ripple) and ADA (Cardano).
[3] https://coinmarketcap.com/
[4] The selected tickers of stocks are AAPL, MSFT, AMZN, META and GOOG.
[5] https://finance.yahoo.com/
[6] BKNG, CMCSA, DIS, MCD, and SBUX
[7] CI, CNC, ELV, HUM, and UNH
[8] https://www.tradingview.com/markets/stocks-usa/sectorandindustry-sector/
[9] AMOV, AMX, T, TMUS, and VZ
[10] BHP, FCX, RIo, SCCO, and VALE

TABLE 2. Evaluation results of baseline experiments. For the **Crypto&Stock**, every value is the average of each phase. Also, we experiment ten times for each model, and use the average values for each metric.

	Crypto & Stock				$\text{Stock}^{v1\uparrow}$ & $\text{Stock}^{v2\uparrow}$				$\text{Stock}^{v1\downarrow}$ & $\text{Stock}^{v2\downarrow}$			
	CR↑	Sha↑	Sor↑	Ome↑	CR↑	Sha↑	Sor↑	Ome↑	CR↑	Sha↑	Sor↑	Ome↑
Market	3.518	1.940	4.146	1.536	**52.669**	0.731	1.043	1.159	42.529	0.602	0.835	1.115
Momentum	3.013	**2.156**	4.301	1.586	47.235	0.680	0.975	1.143	39.033	0.590	0.839	1.112
MLP	3.196	1.945	4.135	1.534	32.513	0.731	1.044	1.160	26.990	0.606	0.848	1.117
CNN	3.507	1.939	4.135	1.535	36.458	0.727	1.038	1.159	30.814	0.619	0.866	1.120
MAPS [14]	3.451	1.935	4.092	1.535	52.620	0.730	1.043	1.162	42.216	0.603	0.843	1.119
HADAPS	**5.653**	1.969	**4.930**	**1.653**	50.735	**0.815**	**1.177**	**1.178**	**46.667**	**0.622**	**0.880**	**1.125**

TABLE 3. Results of ablation experiments on Hierarchical Portfolio Selection Layer on the **Crypto&Stock**. **SAC** is our base model without using hierarchical selection architecture. All models below **SAC** uses hierarchical selection architecture, but **SAC+Intra**, **SAC+Inter** use a fair distribution strategy for assets in a given class and asset classes, respectively. IV-B explains the asset allocation layers.

	CR↑	Sha↑	Sor↑	Ome↑
SAC	3.588	1.962	4.187	1.548
SAC+Intra	3.503	1.817	4.023	1.538
SAC+Inter	3.133	1.929	**5.577**	**1.771**
HADAPS	**5.653**	**1.969**	4.930	1.675

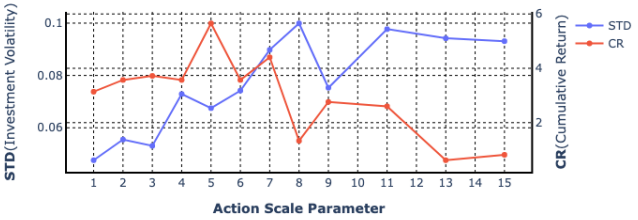


FIGURE 3. Results of ablation experiments for action Scale parameter on the **Crypto&Stock**. **STD** is a standard deviation value for proportion of asset classes. **CR** shows average cumulative return of each phase.

make disturbance for values with abnormal values or skewed distributions [30].

B. MODEL BASELINES

The baselines for comparatively evaluation of *HADAPS* are the following,

- **Market, Momentum**: Market strategy shows a result of equally distributed composition for given assets. Momentum strategy, which involves selecting the top performing assets within a training period and distributing them equally within a testing period, shows a result of following a simple traditional momentum strategy [31]. These strategies show the results of simply constructing a portfolio but not in an adaptive way.
- **MLP, CNN**: We adopted these models that were used for forecasting prices of individual assets [32]. We trained the models to predict three classes which are *Long*, *Hold*, and *Short*. We set the class of a day as *Short* if the return is below -3% or *Long* if return is above 3%.
- **MAPS**: As for our SOTA baseline, we trained a value network to measure a state with three dimensions including *Long*, *Hold*, and *Short* [14].

VI. RESULTS

A. MODEL COMPARISON WITH BASELINES

Table 2 and 3 show the results of our main and ablation experiments. We conducted experiments to compare *HADAPS*'s performance with its baselines using the four evaluation metrics. For each model, we repeated the experiments ten times using different random seeds and calculated the mean values of their test performance. For **Crypto&Stock** scenario, all four evaluation metrics for each experiment were averaged period-wise as our test year is split into twelve inference periods.

The first baseline Market in Table 2 are the representative assets when following the evenly distributed portfolio strategies. This baselines fell behind *HADAPS* in all four evaluation metrics with all of the scenarios except CR (52.669) in $\textbf{Stock}^{v1\uparrow}\textbf{\&Stock}^{v2\uparrow}$ where *HADAPS*'s CR (50.735). However even in this case, other risk adjusted metrics including *HADAPS*'s Sha (0.815), Sor (1.177), and Ome (1.178) are better than the market's. This demonstrates the effectiveness of constructing multi-asset portfolios instead of a market following heuristic investment approach.

As shown in Table 2, baselines using simple neural networks (MLP, CNN) did not show improvement compared to Fair Trading especially in risk-adjusted evaluation metrics. Even replacing these models with SOTA approach (MAPS) did not seem to grant remarkable improvement in all evaluation metrics as well except CR (52.620) in $\textbf{Stock}^{v1\uparrow}\textbf{\&Stock}^{v2\uparrow}$ than *HADAPS*' CR (50.735). We speculated that the performance of the MAPS is almost similar to that of the market. Therefore, the MAPS approach may lack adaptability in terms of asset allocation. Consequently, *HADAPS* yielded superior results in a majority of cases compared to baselines.

B. MODEL ABLATION ON ACTION SCALE PARAMETER

We performed ablation tests on *HADAPS* by modifying the action scale parameter on **Crypto&Stock**. We initially expected the action scale parameter to determine the overall adaptive investment behavior of *HADAPS*. To quantify its investment volatility we calculated the standard deviation value (STD) of the proportion of asset classes. Higher STD

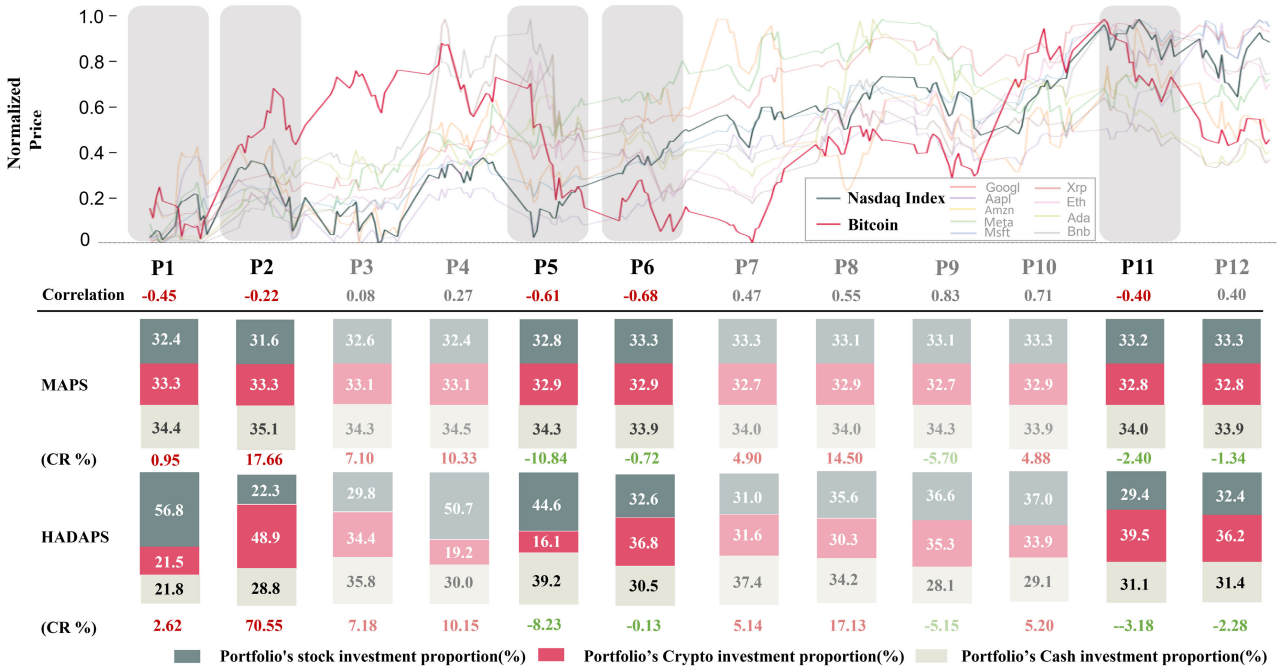


FIGURE 4. Monthly asset class-wise proportions of portfolios on **Crypto&Stock**. The upper part illustrates the normalized prices of each asset in the test period. Highlighted lines are nasdaq index and bitcoin which are the representative assets for stocks and cryptos respectively. The asset class-wise proportions of portfolios made by ***HADAPS*** and MAPS for each period are shown in the lower part. We calculated the asset class-wise proportions by summing the asset-wise proportions for each asset classes.

value means the model is sensitive to a market situation. According to Figure 3, the STD value was proportional to action scale parameters from 1 to 8 while the Cumulative Return has reached its peak when the action scale was set to 5. On the contrary, action scale parameters above 8 did not result in increased returns and instead consistently exhibited elevated levels of investment volatility. This demonstrates the importance of selecting the optimal action scale parameter as it may impact the overall investment results in terms of *HADAPS*' adaptive behavior.

C. MODEL ABLATION ON HIERARCHICAL POLICY NETWORK LAYERS

We performed ablation tests on *HADAPS* to probe the effects of utilizing Hierarchical Policy Network Layers of which the results are shown in Table 3 on **Crypto&Stock**. As remarked in the previous Section IV-D, the non-linear activation functions tend to restrict our model's *adaptive* asset selection behavior in various market circumstances. This further supports our choice of enhancing the SAC framework with the Hierarchical Policy Network and Action Scale parameter as empirically shown in the *HADAPS*' results. We replaced each of the Policy Network Layers (Inter-Class Asset Layer, Intra-Class Asset Layer) with Uniformly Distributed action values: $Even(X) = \frac{1}{k}$, where k is 3 for inter-class allocation layer, 5,5,1 for each assets in asset classes (stocks, cryptos, and safe asset), respectively.

As shown in Table 3, utilizing Inter-Class and Intra-Class Asset Layers have significantly contributed to earning more returns (CR 5.653) compared to other ablations (SAC+Intra CR 3.503, SAC+Inter CR 3.133, Without Both CR 3.588 from Table 2).

D. QUALITATIVE ANALYSIS ON MONTHLY RESULTS

Under assumption that uncorrelated assets may provide opportunities to secure more returns in portfolio selection [1], we classified all twelve test periods into correlated and uncorrelated ones by calculating the correlation between stock and crypto asset classes in the **Crypto&Stock**. The correlation metric is defined as follows:

$$Corr(X, Y) = \frac{\mathbb{E}[(X - \mu_x)(Y - \mu_y)]}{\sigma_x \sigma_y} \tag{12}$$

where X, Y are price data in test period averaged on five assets per asset class which are stocks and cryptos respectively. For each asset, all price data were normalized using min-max scaling. μ_x, μ_y, σ_x, σ_y are mean and standard deviation of X, Y.

Having calculated the correlation values as shown in Figure 4, we indicated the negative correlation periods with gray areas and examined the portfolio proportions of each asset class (P1, P2, P5, P6 and P11) where each period is annotated with total Cumulative Return.

For P1 and P2, our *HADAPS* earned positive returns (2.62, 70.55) compared to MAPS. During P1 where the stock market has a bull-run period compared to crypto's down-trends, *HADAPS* took advantage and gained better returns by acquiring more stocks. Moreover, making use of a transition

in market trends, *HADAPS* responded by shifting its portfolio proportions from stocks to cryptos.

For P5, as the crypto market was on its bear-run and the stock market rebounded from its lowest bear-run, both models suffered losses in cumulative return. However, *HADAPS* managed to minimize its casualties by reducing its investments in crypto assets. For P6, we speculated that *HADAPS* showed agile response to the fluctuating prices of cryptos and succeeded in mitigating its negative returns to almost zero (-0.13) while MAPS was inflicted with relatively more losses (-0.72). For P11, *HADAPS* did not provide favorable outcomes (-3.18) compared to MAPS (-2.40) when both market negatively correlated downtrends.

Overall, *HADAPS* showed adaptive portfolio selection patterns throughout the twelve periods in test data while MAPS maintained its position of not drastically allocating more on specific group of assets. This concludes that not only MAPS is similar to Fair Trading strategies but *HADAPS* has better potential in acquiring massive returns if it precisely captures the bull-runs among volatile assets.

VII. CONCLUSION

We devised *HADAPS*, a multi-asset portfolio selection SAC framework enhanced with the Hierarchical Policy Network and Action Scale parameter. The former helps *HADAPS* adjust the portfolio proportions of different asset classes with response to dynamic market situations while the latter inherently expands its exploration of action space in the SAC framework. Experiments on investment in stocks, cryptos and safe asset with three different scenarios demonstrated *HADAPS*'s ability to adaptively invest in volatile assets and show mostly better results on returns and risk-adjusted metrics than other baselines and ablations. Further investigation on *HADAPS*'s portfolio selection given twelve test periods on **Crypto&Stock** also show *HADAPS*'s adaptability to dynamic market trends.

Through the results of this paper, we confirmed that *HADAPS* can effectively adjust portfolios for various assets in response to dynamic markets. However, more in-depth research is needed on these points, and future work will proceed in the following directions. First, *HADAPS* has currently conducted experiments on stocks, cryptocurrencies, and safe assets, but research is needed to verify the versatility of the model, including various asset classes. Second, a more detailed analysis of the effects of hierarchical policy networks and behavioral scale parameters is needed. In particular, it is important to better understand how the two factors interact to affect overall performance.

Finally, *HADAPS* adapts well to dynamic market trends, but further research is needed on its performance under extreme market conditions. For example, testing *HADAPS*'s ability to respond to prolonged economic downturns or rapid market fluctuations will be one of the pillars of future research. Through this direction, *HADAPS* will show better performance and expand its applicability in real investment scenarios.

ACKNOWLEDGMENT

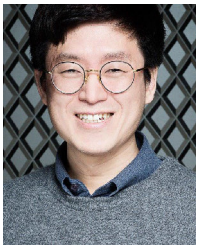
(Jinkyu Kim and Donghee Choi contributed equally to this work.)

REFERENCES

[1] H. M. Markowitz, "Portfolio selection: Efficient diversification of investments," in *Cowles Foundation Monograph*, vol. 16. New York, NY, USA: Wiley, 1959.
[2] A. Meucci, *Risk and Asset Allocation*, vol. 1. Cham, Switzerland: Springer, 2005.
[3] K. Adam, A. Marcet, and J. P. Nicolini, "Stock market volatility and learning," *J. Finance*, vol. 71, no. 1, pp. 33–82, Feb. 2016.
[4] Y. Choueifaty and Y. Coignard, "Toward maximum diversification," *J. Portfolio Manage.*, vol. 35, no. 1, pp. 40–51, Oct. 2008.
[5] A. Buraschi, P. Porchia, and F. Trojani, "Correlation risk and optimal portfolio choice," *J. Finance*, vol. 65, no. 1, pp. 393–420, Feb. 2010.
[6] H. Levy and M. Sarnat, "International diversification of investment portfolios," *Amer. Econ. Rev.*, vol. 60, no. 4, pp. 668–675, 1970.
[7] F. Longin and B. Solnik, "Is the correlation in international equity returns constant: 1960–1990?" *J. Int. Money Finance*, vol. 14, no. 1, pp. 3–26, Feb. 1995.
[8] B. H. Solnik, "Why not diversify internationally rather than domestically?" *Financial Analysts J.*, vol. 30, no. 4, pp. 48–54, Jul. 1974.
[9] P. Byrne and S. Lee, "Is there a place for property in the multi-asset portfolio?" *J. Property Finance*, vol. 6, no. 3, pp. 60–83, Sep. 1995.
[10] H. Ning, "Hierarchical portfolio management: Theory and applications," Erasmus Res. Inst. Manag., Erasmus Univ. Rotterdam, Tech. Rep. EPS-2007-118-F&A, 2007. [Online]. Available: http://hdl.handle.net/1765/10868
[11] T. Raffinot, "Hierarchical clustering-based asset allocation," *J. Portfolio Manage.*, vol. 44, no. 2, pp. 89–99, Dec. 2017.
[12] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," 2017, *arXiv:1706.10059*.
[13] G. Lucarelli and M. Borrotti, "A deep Q-learning portfolio management framework for the cryptocurrency market," *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17229–17244, Dec. 2020.
[14] J. Lee, R. Kim, S.-W. Yi, and J. Kang, "MAPS: Multi-agent reinforcement learning-based portfolio management system," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 4520–4526.
[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.
[16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[17] S. Pafka and I. Kondor, "Estimated correlation matrices and portfolio optimization," *Phys. A, Stat. Mech. Appl.*, vol. 343, pp. 623–634, Nov. 2004.
[18] X. Y. Zhou and G. Yin, "Markowitz's mean-variance portfolio selection with regime switching: A continuous-time model," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1466–1482, Jan. 2003.
[19] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1970–1979.
[20] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 261–269.
[21] J. Wang, Y. Zhang, K. Tang, J. Wu, and Z. Xiong, "AlphaStock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1900–1908.
[22] R. Wang, H. Wei, B. An, Z. Feng, and J. Yao, "Commission fee is not enough: A hierarchical reinforced framework for portfolio management," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 626–633.
[23] Z. Wang, B. Huang, S. Tu, K. Zhang, and L. Xu, "DeepTrader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 643–650.
[24] H. Park, M. K. Sim, and D. G. Choi, "An intelligent financial portfolio trading strategy using deep Q-learning," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113573.
[25] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

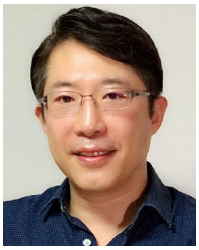
[26] A. P. Ratto, S. Merello, L. Oneto, Y. Ma, L. Malandri, and E. Cambria, "Ensemble of technical analysis and machine learning for market trend prediction," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 2090–2096.
[27] T. Guo, A. Bifet, and N. Antulov-Fantulin, "Bitcoin volatility forecasting with a glimpse into buy and sell orders," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 989–994.
[28] A. Picasso, S. Merello, Y. K. Ma, L. Oneto, and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction," *Expert Syst. Appl.*, vol. 135, pp. 60–70, Nov. 2019.
[29] C. K. William and F. Shadwick, "A universal performance measure," *J. Perform. Meas.*, vol. 6, no. 3, pp. 59–84, 2000.
[30] C. Keating and W. F. Shadwick, "An introduction to Omega," *AIMA Newslett.*, Apr. 2002.
[31] T. J. Moskowitz, Y. H. Ooi, and L. H. Pedersen, "Time series momentum," *J. Financial Econ.*, vol. 104, no. 2, pp. 228–250, 2012.
[32] L. Di Persio and O. Honchar, "Artificial neural networks architectures for stock price prediction: Comparisons and applications," *Int. J. Circuits, Syst. Signal Process.*, vol. 10, pp. 403–413, Jan. 2016.

JINKYU KIM received the B.S. degree in computer science from Korea University, South Korea, in 2019, where he is currently pursuing the Ph.D. degree in computer science. His current research interest includes developing effective investment methods and applying them to various assets, such as stocks, cryptocurrency, and bonds. He is specifically interested in employing reinforcement learning techniques to develop optimal portfolios that incorporate various assets.



DONGHEE CHOI received the B.S. degree in computer science from Korea University, South Korea, in 2012, and the M.S. degree from the Interdisciplinary Graduate Program in Bioinformatics, Korea University, in 2014, where he is currently pursuing the Ph.D. degree in computer science. His research interests include natural language processing and data mining. Specifically, he applies artificial intelligence techniques to mine and generalize domain-aware knowledge in areas, such as food, biomedical, and finance to aid users in their decision-making processes.

MOGAN GIM received the B.S. degree in computer science from Korea University, South Korea, in 2018, where he is currently pursuing the Ph.D. degree in computer science. His current research interest includes developing effective set-oriented data representation methods and applying them to various research domains, such as food science, material science, and drug discovery.



JAEWOO KANG received the B.S. degree in computer science from Korea University, Seoul, South Korea, in 1994, the M.S. degree in computer science from the University of Colorado at Boulder, Boulder, CO, USA, in 1996, and the Ph.D. degree in computer science from the University of Wisconsin–Madison, Madison, WI, USA, in 2003. From 1996 to 1997, he was a Technical Staff Member with AT&T Labs Research, Florham Park, NJ, USA. From 1997 to 1998, he was a Technical Staff Member of Savera Systems Inc., Murray Hill, NJ, USA. From 2000 to 2001, he was the CTO and a Co-Founder of WISEngine Inc., Santa Clara, CA, USA, and Seoul. From 2003 to 2006, he was an Assistant Professor with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA. Since 2006, he has been a Professor with the Department of Computer Science, Korea University, where he is also the Department Head of the Interdisciplinary Graduate Program in Bioinformatics. In 2021, he founded AIGEN Sciences Inc., a cutting-edge start-up in the field of AI-driven drug discovery, where he is currently the CEO.