[Doctoral Consortium]

Temporal representation learning for stock similarities and its applications on investment management

2024.07.12

황윤태 (Hwang yoontae)
yoontae@unist.ac.kr
Financial Engineering Lab
Department of Industrial Engineering

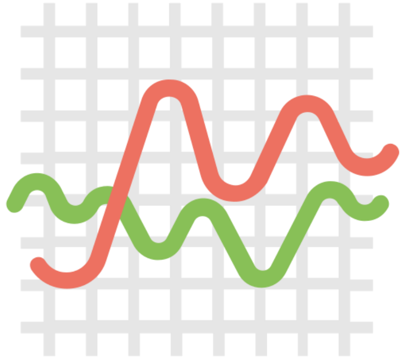
Motivation

Accurate estimation of financial parameters is crucial

Example : **Pair trading** : How do I find similar stocks to pair trade? → Cointegration test

What is cointergration?

Two time series are cointegrated if a linear combination has constant mean and standard deviation. In other words, the two series never stray too far from one another ***in the historical period****.*



Finding Similar Stocks
using the Cointergration Test

Historical period

?

Future period

Motivation

Accurate estimation of financial parameters is crucial

Example : **Portfolio optimization** : How do I estimate μ and Σ ? ⇨ using the sample means of its historical returns given a **lookback window**.

The long-only Mean-Variance Optimization problem is here:

Transaction cost

$$\text{Maximize: } w^T\mu - \psi 1^T|w - w_0|$$
$$\text{Subject to: } w^T\Sigma w \leq \sigma_{\text{target}}^2 \quad w^T 1 = 0, 0 \leq w_k \leq \quad \text{for all } k = 1,2,\ldots,N.$$

$$u_{ti} = \frac{1}{T}\sum_{d=t-1}^{t-T} r_{di}$$

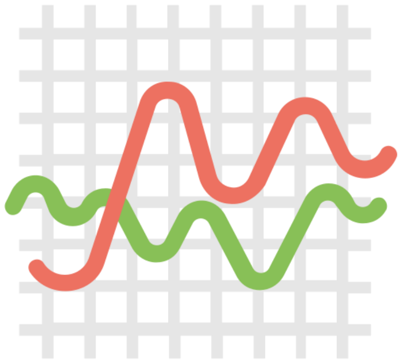
The expected return at time t for asset i is estimated using the sample means of its historical returns given a lookback window of T-months

Historical period

(μ, σ)

Future period

Motivation



Finding Similar Stocks
using the Cointergration Test

Historical period

?

Future period

$$u_{ti} = \frac{1}{T}\sum_{d=t-1}^{t-T} r_{di}$$

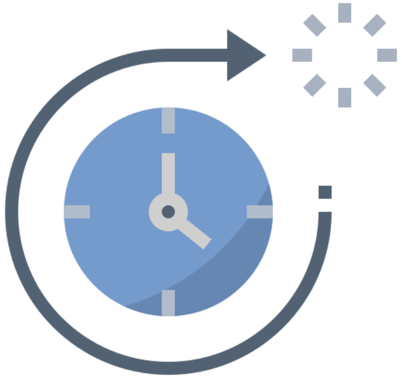
The expected return at time t for asset i is estimated using the sample means of its historical returns given a lookback window of T-months

Historical period

(μ, σ)

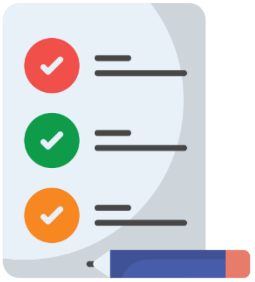
Future period

Main challenge



Temporal domain shift

Caused by the non-stationarity of financial markets



Static data

3-Statement, Firm description and etc.

Ambiguity & Lack of labels

Due to rapid globalization and digitalization

Main observation

- **Temporal domain shift**: The movement of stocks continuously changes over time. This is mainly due to the unique characteristics of individual stocks as well as interactions between different stocks and various factors that can lead to domain shifts.
- **Static data**: Stocks are characterized not only by price data but also by a variety of static information.
- **Ambiguity**: Ambiguity in conventional regional and sector classifications due to rapid globalization and digitalization.
- **Lack of labels**: There is no appropriate label for identifying similar stocks.

Related work

Selected related work : Self-supervised learning & Temporal domain generalization

- Self-supervised learning has primarily evolved within the field of computer vision.
- **Most existing works in SSL have focused on invariance**[6][7]. That is, they rely on simple inductive biased that two similar observations should yield similar outputs, and there have proven to be effective when augmenting data (mostly for images)[8][9].
  - ➢ For non-stationary data, such as stocks, it is quite challenging to incorporate these distribution shift into the SSL framework.

- Domain generalization refers to the learning of general model representation, and various methods have been proposed for this purpose[9][10][11].
  - ➢ **Existing studies** assume that the domain index set spans time and **cannot adaptively learn temporal shift over time**.
  - ➢ Fortunately, **DRAIN**[12] is the first temporal domain generalization method to address this limitation by adaptively learning temporal drifts across multiple source domains at **supervised learning task**.

[5] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 15750–15758.
[6] Jean-Bastien Grill et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems 33 (2020), 21271–21284.
[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning. PMLR, 1597–1607
[8] Longlong Jing and Yingli Tian. 2020. Self-supervised visual feature learning with deep neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence 43, 11 (2020), 4037–4058.
[9] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.
[10] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 23–30
[11] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. 2019. Dlow: Domain flow for adaptation and generalization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2477–2486.
[12] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. 2017. Domain generalization and adaptation using low rank exemplar SVMs. IEEE transactions on pattern analysis and machine intelligence 40, 5 (2017), 1114–1127.
[13] Bai, G., Ling, C., & Zhao, L. (2022). Temporal Domain Generalization with Drift-Aware Dynamic Neural Networks. ICLR2023, Spotlight

Related work

Selected related work : Self-supervised learning & Temporal domain generalization

- **Most existing works in SSL have focused on invariance**[6][7]. That is, they rely on simple inductive biased that two similar observations should yield similar outputs, and there have proven to be effective when augmenting data (mostly for images)[8][9].
  - For non-stationary data, such as stocks, it is quite challenging to incorporate these distribution shift into the SSL framework.

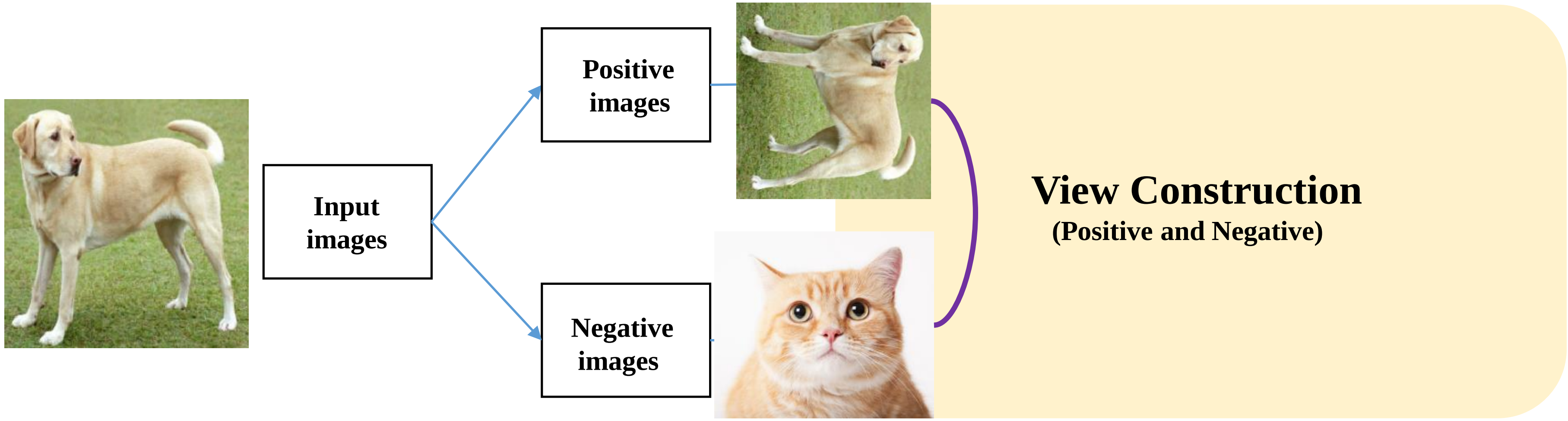


Figure 1. A simple framework for contrastive learning of visual representations

Related work

Selected related work : Self-supervised learning & Temporal domain generalization

- **Most existing works in SSL have focused on invariance**[6][7]. That is, they rely on simple inductive biased that two similar observations should yield similar outputs, and there have proven to be effective when augmenting data (mostly for images)[8][9].
  - ➢ For non-stationary data, such as stocks, it is quite challenging to incorporate these distribution shift into the SSL framework.

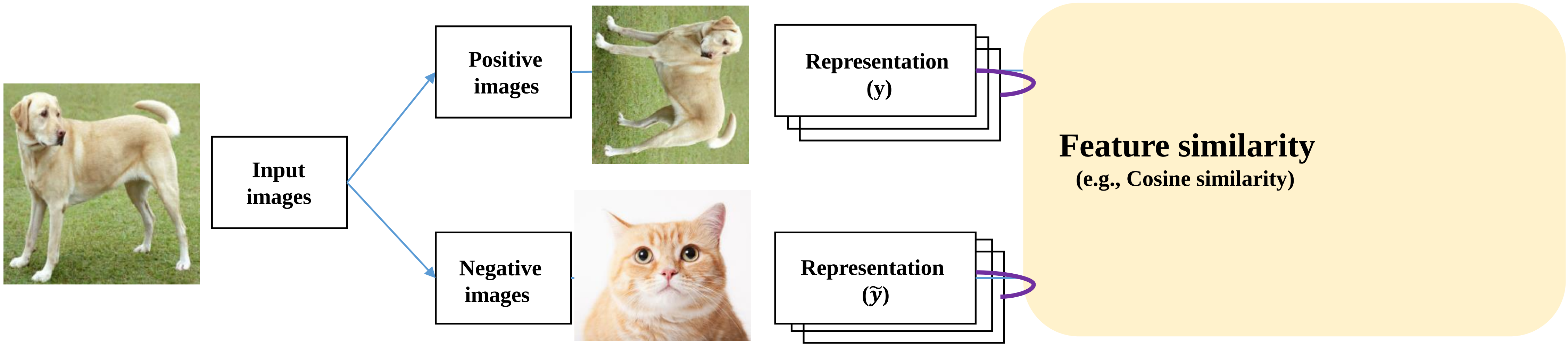


Figure 1. A simple framework for contrastive learning of visual representations

Related work

Selected related work : Self-supervised learning & Temporal domain generalization

- **Most existing works in SSL have focused on invariance**[6][7]. That is, they rely on simple inductive biased that two similar observations should yield similar outputs, and there have proven to be effective when augmenting data (mostly for images)[8][9].
  - For non-stationary data, such as stocks, it is quite challenging to incorporate these distribution shift into the SSL framework.

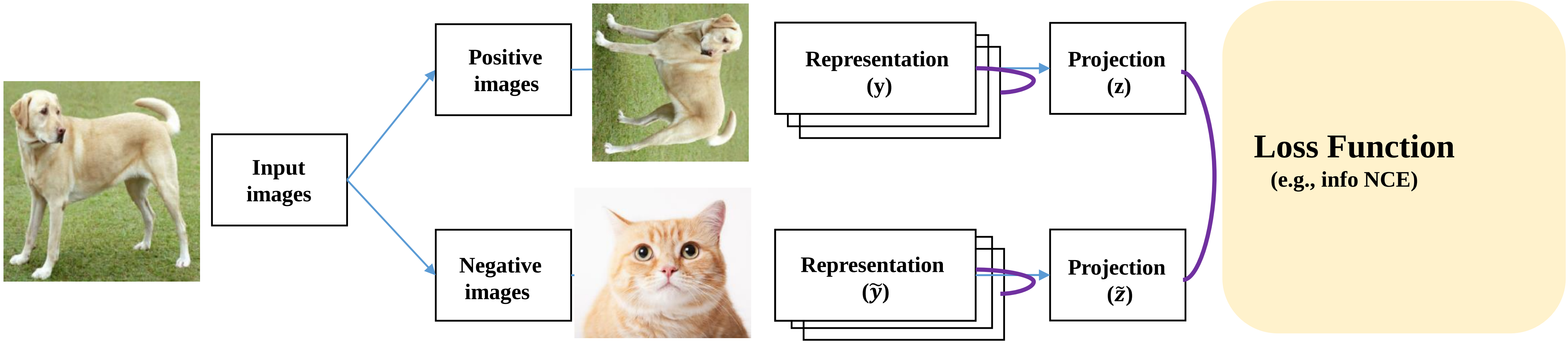


Figure 1. A simple framework for contrastive learning of visual representations

Related work

Selected related work : Self-supervised learning & Temporal domain generalization

- **Most existing works in SSL have focused on invariance**[6][7]. That is, they rely on simple inductive biased that two similar observations should yield similar outputs, and there have proven to be effective when augmenting data (mostly for images)[8][9].
  - For non-stationary data, such as stocks, it is quite challenging to incorporate these distribution shift into the SSL framework.

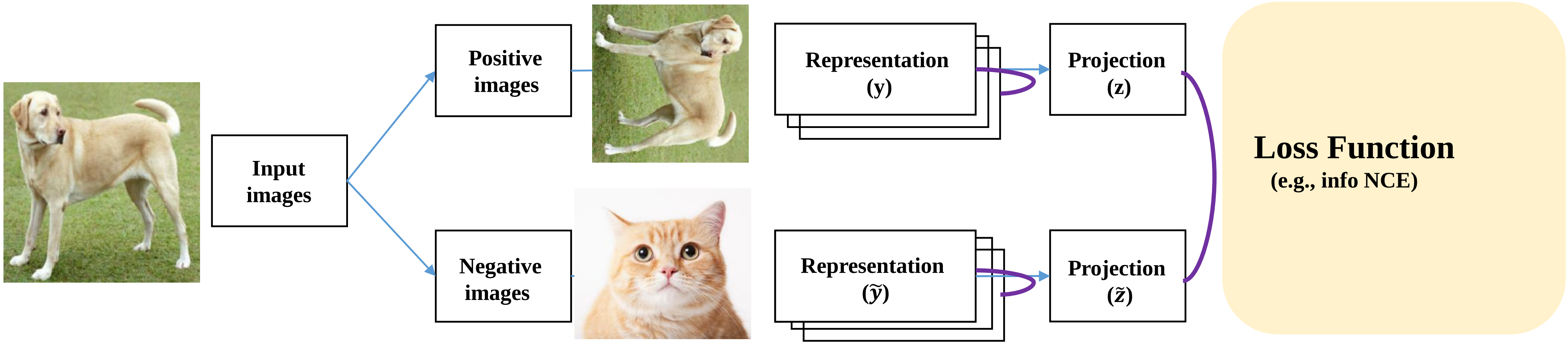
- What about **temporal data settings**(e.g., time-series)?

How to define *view* in the time-series data?

How to exploit the *inductive bias* for time-series data?

How to learn *temporal context*?

Related work



How to define *view* in the time-series data?

How to exploit the *inductive bias* for time-series data?

How to *learn* temporal context?

=> Temporal Dimension corruption

=> Temporal Domain Generalization

⇒ Triplet loss

Overview of SimStock

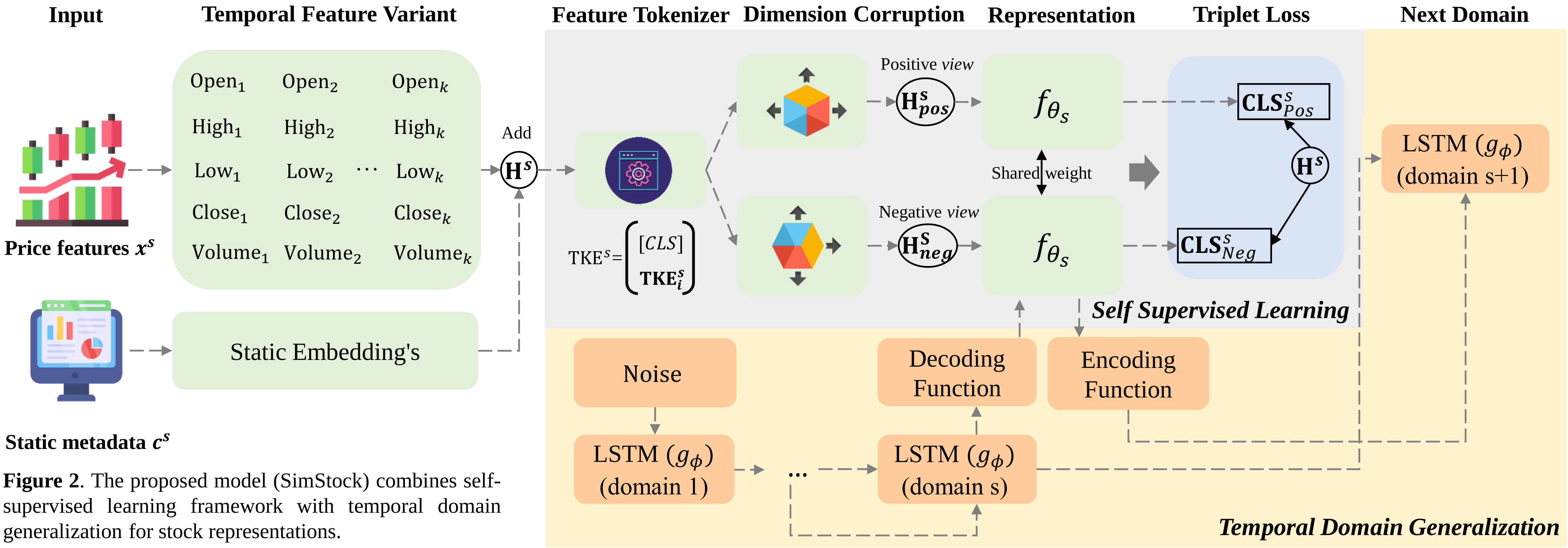


Figure 2. The proposed model (SimStock) combines self-supervised learning framework with temporal domain generalization for stock representations.

SimStock

- We propose **SimStock** to effectively extract stock representations.
- The keys of this research lies in using elaborately designed Temporal domain generalization and self-supervised learning to address the challenges previously mentioned.

Experiment settings

Dataset

Exchange	Stocks
NYSE NASDAQ	4,231 stocks, we refer to them as the US exchanges
SSE(Shanghai Stock exchange)	1,408 stocks
SZSE(Shenzhen Stock Exchange)	1,696 stocks
TSE(Tokyo Stock Exchange)	3,882 stocks

Time period

Training period : Jan 01, 2018 to Dec 31, 2021
Reference period : Jan 01, 2022 to Dec 31, 2022
Test period : Jan 01, 2023 to Dec 31, 2023

Baseline models

Corr1 : past one-year returns correlation
Corr2 : training period returns correlation
Peer : list of similar stocks provided by Google, Yahoo Finance, and Financial Modeling Prep
TS2VEC : Deep learning based state-of-the-art method

Can SimStock find similar stocks?

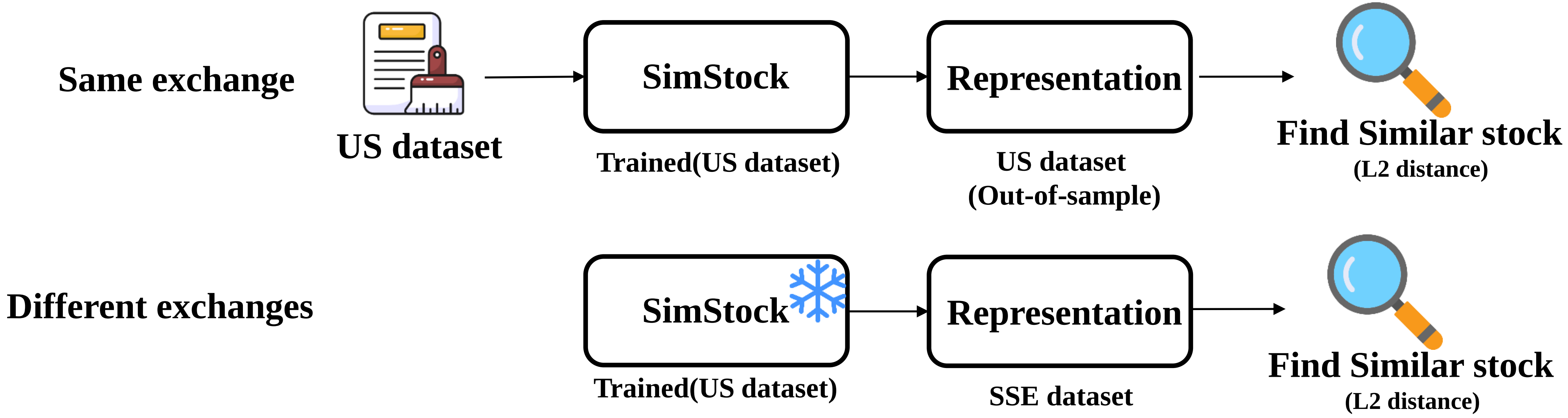


Figure 4. High-level overview of evaluation scenarios

Evaluation scenarios

- In different exchanges scenario, We apply the weights of a model trained on a specific exchange to a different exchange.
- For example, models trained on the US exchange can be used to find similar stocks in the SSE, SZSE, or TSE exchanges.

How to find similar stocks?

- If the query stock is JP Morgan, we can find the *K* stocks that are most similar to JP Morgan with L2 distance in embedding space among all stocks on the exchange.

Can SimStock find similar stocks? (Same exchange scenario)

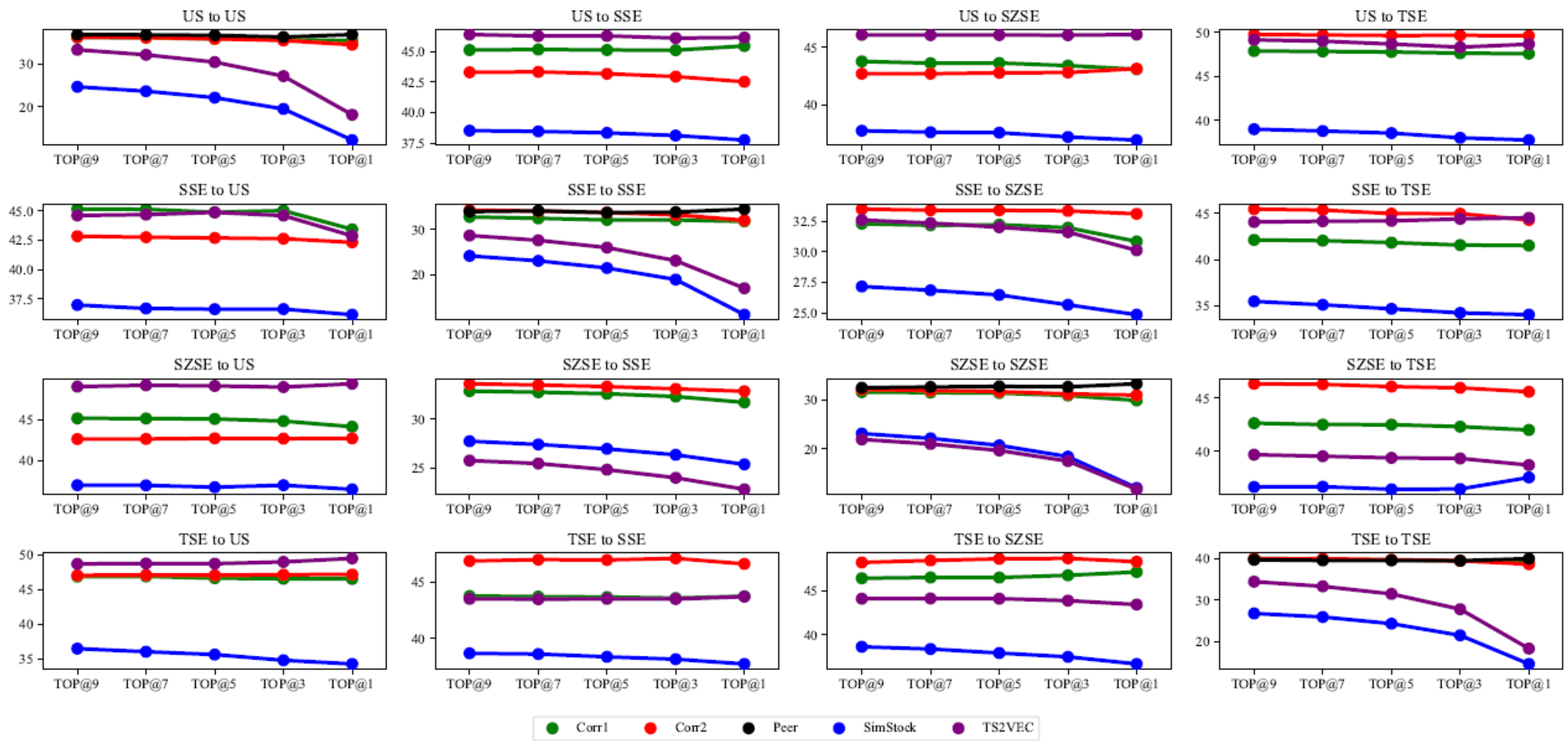


Figure 5. Performance of models in same exchange and different exchanges scenarios for finding similar stocks.

One-to-one : Given a query stock, we find similar stocks within the same exchange.

- The **diagonal plots** in this figure illustrate the performance(DTW) of different models in same exchange scenario.
- DTW measure by selecting the top TOP@9, TOP@7, TOP@5, TOP@3 and TOP@1 similar stocks.
- It is clear that **SimStock** stands out as the best performer in the same exchange scenario compared to all other baseline models except SZSE to SZSE.

Can SimStock find similar stocks? (Same exchange scenario)

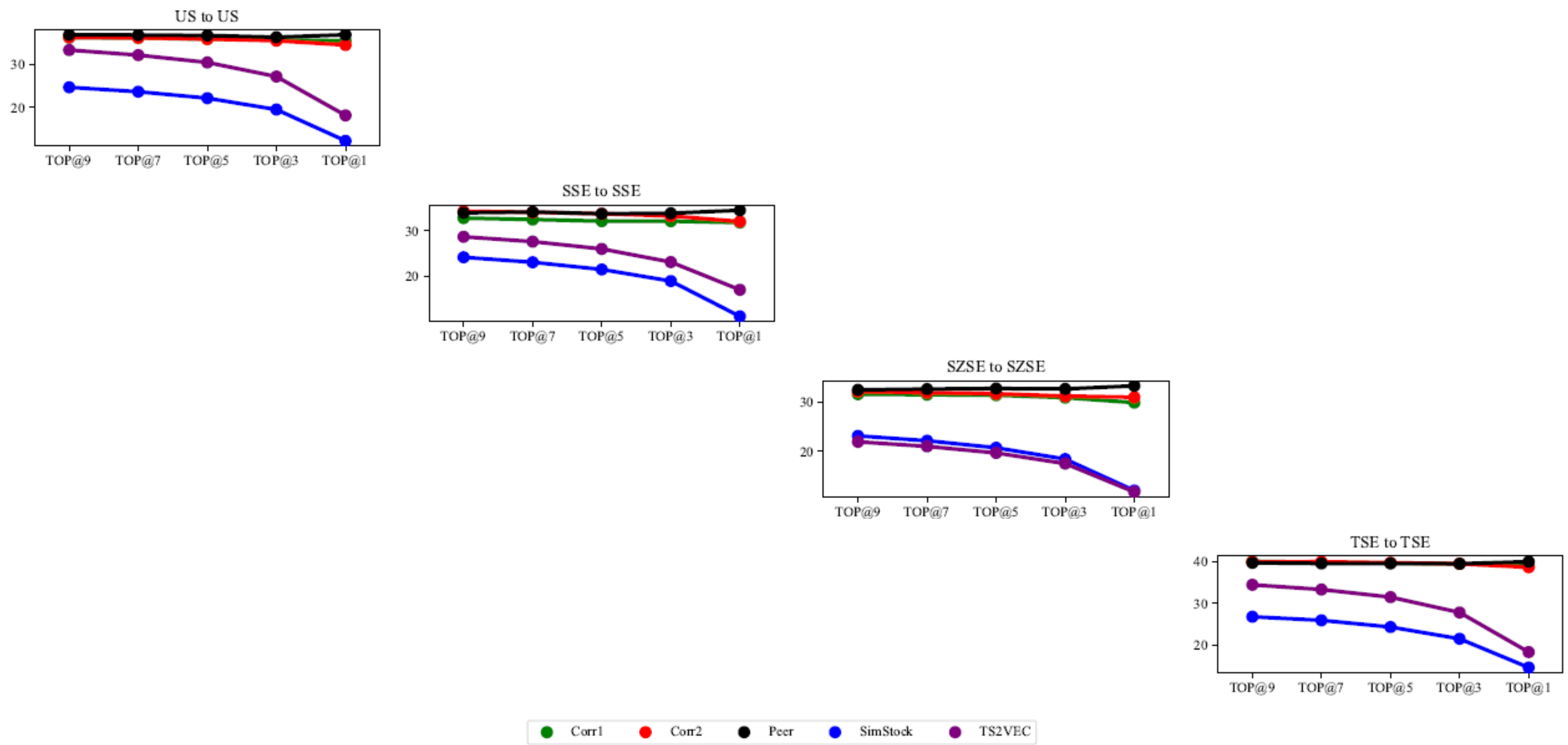


Figure 5. Performance of models in same exchange and different exchanges scenarios for finding similar stocks.

One-to-one : Given a query stock, we find similar stocks within the same exchange.

- The **diagonal plots** in this figure illustrate the performance(DTW) of different models in same exchange scenario.
- DTW measure by selecting the top TOP@9, TOP@7, TOP@5, TOP@3 and TOP@1 similar stocks.
- It is clear that **SimStock** stands out as the best performer in the one-to-one scenario compared to all other baseline models except SZSE to SZSE.

Can SimStock find similar stocks? (Different exchanges)

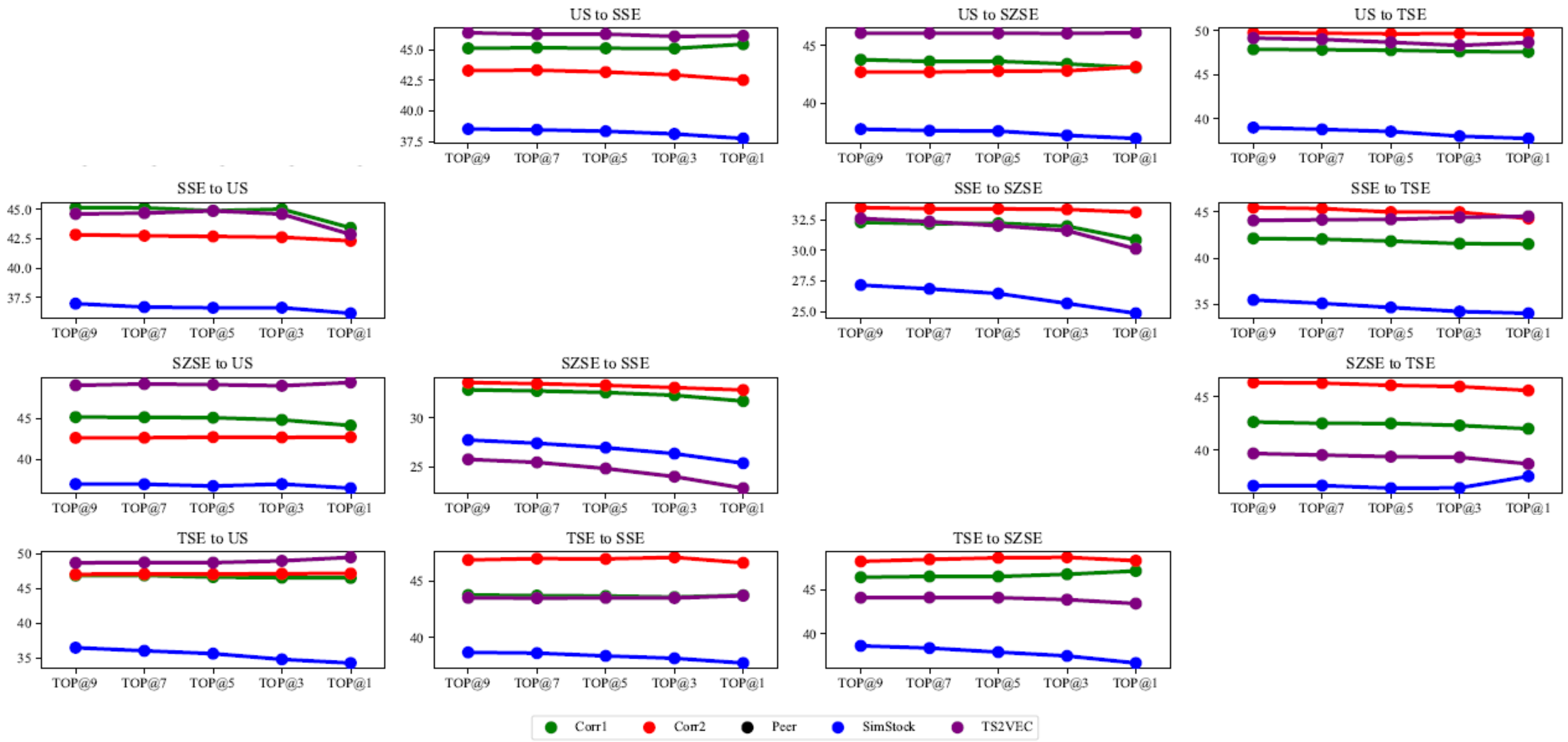


Figure 5. Performance of models in same exchange and different exchanges scenarios for finding similar stocks.

Different exchanges : Given a query stock, we find similar stocks within another exchange

- The **off-diagonal** plots in this figure illustrate the performance(DTW) of different models in different exchanges scenario.
- Peer is not available for this scenario, because most trading platforms do not provide information on similar stocks in other exchanges.
- SimStock performed exceptionally well in all one-to-many scenarios except for one case. (SZSE to SSE)

Application to Pairs trading (Result) (Motivation skip)

Query Stock	SimStock	TS2VEC	Wealth Corr1	Corr2	Coint
AAPL	**961.04 ±474.43**	NaN**	234.69 ±1165.48	916.07 ±338.95	NaN**
CMG	**546.95 ±724.24**	282.38 ±714.16	-1070.09 ±722.33	-1098.98 ±893.35	-857.35 ±2967.72
MSFT	**754.12 ±69.73**	498.11 ±785.76	474.85 ±1651.85	-306.61 ±2114.7	257.29 ±637.56
WFC	**562.95 ±173.07**	-780.57 ±1118.2	389.75 ±737.45	NaN**	-478.31 ±2011.8
V	353.09 ±117.31	23.7 ±222.98	329.12 ±99.36	**406.14 ±165.45**	241.9 ±1070.96
XOM	389.74 ±266.41	356.79 ±252.04	-2.86 ±164.5	103.69 ±1097.92	**2131.95 ±1424.47**
PFE	-411.46 ±677.54	114.38 ±2267.64	192.45 ±1656.14	-163.88 ±1545.7	**419.19 ±211.39**
AMZN	121.02 ±244.41	386.65 ±1305.54	-597.9 ±1003.48	**2047.26 ±2090.29**	-1184.76 ±3526.17
BA	**572.82 ±2258.07**	16.24 ±853.6	-1211.11 ±658.4	-653.36 ±1339.05	143.52 ±725.58
META	1344.86 ± NaN*	-589.62 ±1253.38	**1820.84 ±1903.45**	-1695.45 ±3261.33	325.51 ±1269.18
MA	122.96 ±65.76	-99.61 ±266.94	**246.72 ±69.7**	-247.12 ±954.71	-577.71 ±893.93
CVS	**1092.8 ±528.37**	-634.47 ±774.1	989.94 ±761.13	795.5 ±367.36	-213.04 ±1318.44

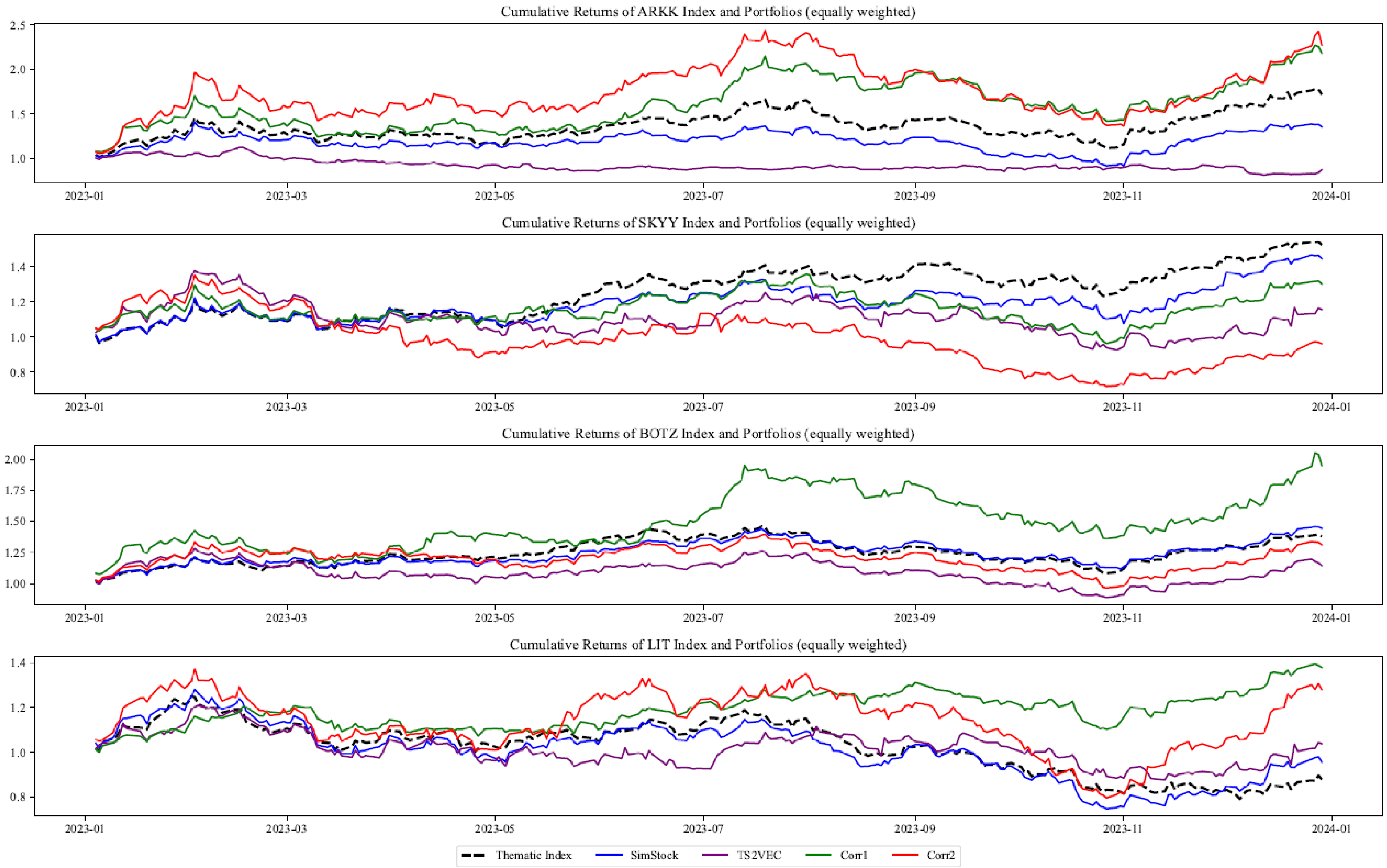
Query Stock	SimStock	Maximum Drawdown (%) TS2VEC	Corr1	Corr2	Coint
AAPL	**-1.91 ±0.67**	NaN**	-6.06 ±3.47	-3.4 ±2.01	NaN**
CMG	**-2.99 ±2.54**	-6.38 ±2.22	-14.98 ±6.37	-15.55 ±10.38	-20.05 ±15.48
MSFT	**-2.82 ±1.88**	-5.67 ±0.24	-7.16 ±4.7	-12.92 ±12.56	-13.57 ±2.58
WFC	**-2.2 ±3.78**	-9.2 ±8.24	-5.76 ±4.54	NaN**	-16.07 ±13.41
V	-2.22 ±2.55	-2.39 ±2.78	**-0.43 ±0.75**	-3.17 ±4.48	-7.48 ±5.89
XOM	-3.4 ±0.41	**-3.03 ±2.97**	-4.72 ±1.43	-6.55 ±6.9	-5.78 ±4.3
PFE	**-7.84 ±7.41**	-10.03 ±13.5	-9.99 ±9.42	-10.45 ±5.08	-9.05 ±5.56
AMZN	-5.06 ±4.99	**-4.68 ±2.33**	-15.14 ±9.38	-8.97 ±4.15	-31.92 ±13.14
BA	-12.1 ±11.58	**-5.86 ±2.15**	-15.48 ±8.63	-12.87 ±3.26	-14.44 ±7.33
META	**-4.92 ±NaN***	-8.77 ±8.96	-11.59 ±4.95	-34.15 ±26.03	-25.43 ±11.24
MA	**-2.18 ±2.51**	-5.01 ±5.2	-2.58 ±2.72	-6.85 ±7.24	-11.51 ±4.68
CVS	-3.02 ±1.88	-9.2 ±8.57	-4.57 ±4.6	**-2.92 ±1.36**	-19.87 ±17.05

Table 1 : Average terminal wealth (first row) and maximum drawdown (MDD) (second row) achieved by applying pairs trading to the top@3 similar stocks identified by SimStock, TS2VEC, Corr1, Corr2, coint for each query stock. NaN** values in both the terminal wealth and MDD indicate that the method failed to generate buy/sell signals for all three stocks in the pair. NaN* values only in the standard deviation indicate that the method failed to generate buy/sell signals for two out of the three stocks in the pair. For all other values, all method generated buy/sell signals for all three stocks in the pair.

- **We employ price ratio approach for pairs trading.**

Settings
- Initial trading capital : 10,000 USD
- Predetermined threshold (Stop loss) : 500 USD
- Z-score threshold : ±1.25 (Buy & Sell)
- Z-score threshold : ±0.5 (Position closed)
- Finding top 3 similar stocks

Application to index tracking of thematic ETFs (Results) (Motivation Skip)



Index tracking of thematic ETFs

Settings

- Equal-weighted portfolio
- Tracking portfolios constructed using the top 10 similar stocks

Figure 2 Cumulative return curves of the four **thematic ETFs** (ARKK, SKYY, BOTZ, and LIT) and their corresponding tracking portfolios constructed using the top 10 similar stocks identified by **SimStock** and the baseline methods (**TS2VEC**, **Corr1**, and **Corr2**) from the US exchange. The closer a portfolio's curve follows the respective ETF curve (dotted black line), the better the tracking performance.

Application to Portfolio optimization

Previous portfolio weights

Maximize : $w^T\mu - \psi 1^T|w - w_0|$

Portfolio's expected return - Transaction costs

Subject to : $w^T\Sigma w \leq \sigma_{\text{target}}^2$,

$w^T 1 = 0$,

$0 \leq w_k \leq 1$ for all $k = 1,2,..,N$

Portfolio variance must not exceed predetermined risk target

Introduction

We investigate whether **SimStock** embeddings can enhance portfolio optimization. Specifically, we construct the correlation matrix using the SimStock embedding as a similarity measure, and use it as an input for portfolio optimization. We compare the portfolio performance using the **SimStock** embedding with other covariance estimators.

Application to Portfolio optimization

The Gerber Statistic: a Robust Co-movement Measure for Portfolio Optimization

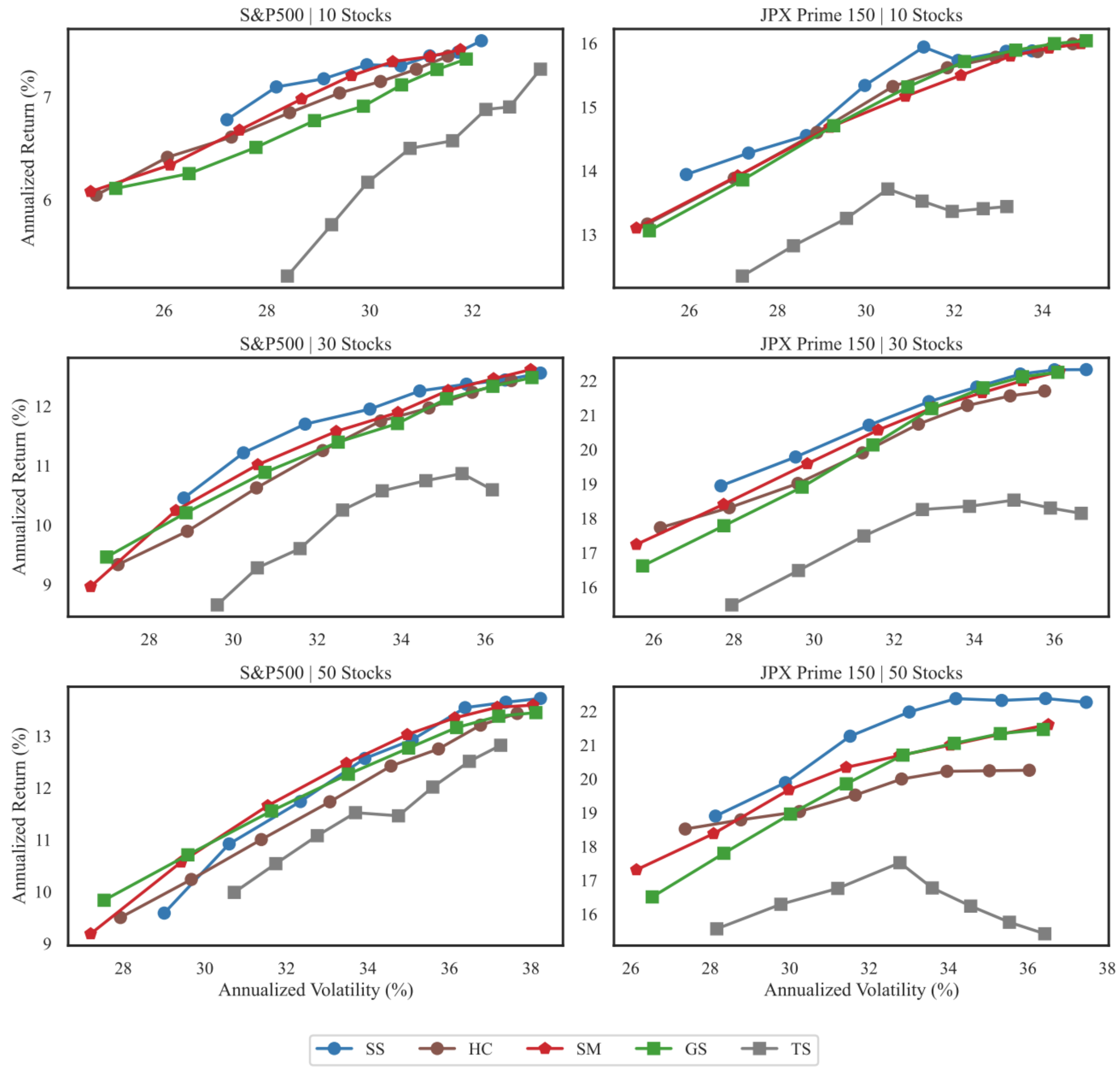
Philip Ernst, Ph.D. (Chair in Statistics)
Department of Mathematics, Imperial College London

Joint work with Sander Gerber (Hudson Bay Capital Management, LP), Harry Markowitz (Rady School of Management, UCSD), Yinsen Miao (Fidelity Investments), Babak Javid (Hudson Bay Capital Management, LP), and Paul Sargen (Hudson Bay Capital Management, LP)

The Journal of Portfolio Management, 48(3): 87-102, 2022.

September 20, 2022

Application to Portfolio optimization



Maximize : $w^T\mu - \psi 1^T |w - w_0|$

Subject to : $w^T \Sigma w \leq \sigma^2_{\text{target}}$,

$w^T 1 = 0$,

$0 \leq w_k \leq 1$ for all $k = 1,2,..,N$

Result

- The results demonstrate that the proposed SimStock embedding outperforms other methods. However, this performance is achieved by taking on slightly more risk compared to other models, leading to better returns.
- On the other hand, TS2VEC, represented by the gray line, shows very poor performance.
- This suggests that even with a data-driven approach, whether or not the temporal domain is taken into account can be a crucial factor in portfolio performance.

Figure 3 Ex-post efficient frontiers displaying annualized return and volatility of portfolios optimized for different risk targets. The black vertical dotted lines represent the average volatility of the S&P500 and JPX Prime 150, respectively

Application to Portfolio optimization

S&P500 — 30 Stocks	Target Volatility (24%)					Target Volatility (27%)					Target Volatility (30%)					Target Volatility (33%)				
Covariance Method	SS	HC	SM	GS	TS	SS	HC	SM	GS	TS	SS	HC	SM	GS	TS	SS	HC	SM	GS	TS
Arithmetic Return (%)	11.31	10.04	9.61	10.14	9.40	12.22	10.75	11.02	11.05	10.51	12.84	11.66	12.03	11.97	10.86	13.33	12.50	12.85	12.69	11.49
Geometric Return (%)	10.39	9.30	8.97	9.43	8.19	11.158	9.86	10.21	10.17	9.43	11.60	10.57	10.967	10.83	9.56	11.88	11.18	11.51	11.32	10.37
Cumulative Return (%)	37.36	33.07	31.60	33.40	26.64	40.54	35.40	36.63	36.49	31.04	42.64	38.44	39.930	39.61	31.53	44.18	41.27	42.59	42.05	34.46
Annualized SD (%)	28.82	27.27	26.61	27.00	29.54	30.24	28.92	28.66	28.90	30.24	31.72	30.58	30.623	30.79	31.37	33.25	32.17	32.49	32.54	32.26
Annualized Skewness	-0.12	-0.15	-0.12	-0.122	-0.17	-0.14	-0.17	-0.16	-0.16	-0.21	-0.18	-0.20	-0.191	-0.20	-0.21	-0.22	-0.22	-0.22	-0.23	-0.24
Annualized Kurtosis	3.17	3.22	3.23	3.161	2.81	3.22	3.28	3.29	3.24	2.86	3.27	3.33	3.343	3.28	2.88	3.27	3.33	3.33	3.28	2.93
Maximum Drawdown (%)	-24.90	-23.96	-23.47	-23.59	-25.59	-25.65	-25.36	-24.57	-25.04	-25.93	-26.69	-26.44	-25.971	-26.57	-26.64	-28.20	-27.63	-27.57	-28.08	-26.47
Monthly 95% VaR (%)	-10.44	-10.22	-9.99	-10.05	-11.11	-10.77	-10.63	-10.53	-10.52	-11.38	-11.19	-10.95	-10.921	-11.01	-11.69	-11.72	-11.40	-11.48	-11.56	-12.2
Sharpe Ratio	0.44	0.40	0.39	0.42	0.31	0.46	0.41	0.43	0.43	0.31	0.46	0.42	0.447	0.43	0.35	0.45	0.43	0.44	0.43	0.36
Annualized Turnover	8.68	8.39	8.49	8.36	7.92	8.69	8.48	8.56	8.49	8.04	8.73	8.54	8.592	8.56	7.99	8.67	8.57	8.51	8.54	7.86

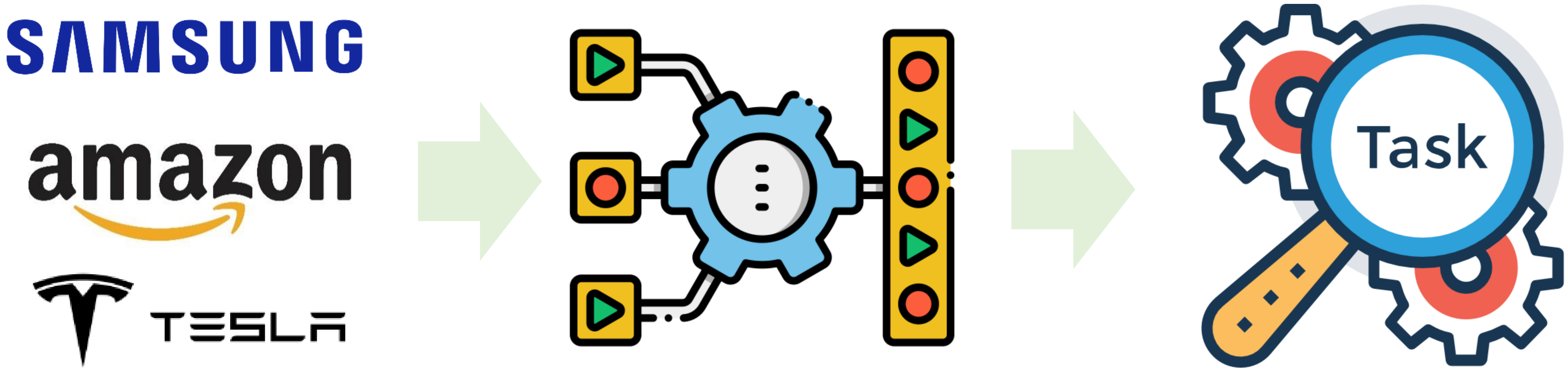
Table 5. This table presents the performance metrics for four portfolio construction methods in the S&P500: Simstock embedding (SS), historical covariance (HC), shrinkage method (SM), Gerber statistic (GS) and TS2VEC embedding (TS). The portfolios were optimized for four different risk target levels: 24%, 27%, 30%, and 33%. The performance was evaluated over the full testing period from January 2022 to February 2024. The 3-month U.S. Treasury Bill rate was used as the risk-free rate for performance calculations. Transaction costs were modeled as 10 basis points of the traded volume for each rebalancing event.

JPX Prime 150 — 30 Stocks	Target Volatility (24%)					Target Volatility (27%)					Target Volatility (30%)					Target Volatility (33%)				
Covariance Method	SS	HC	SM	GS	TS	SS	HC	SM	GS	TS	SS	HC	SM	GS	TS	SS	HC	SM	GS	TS
Arithmetic Return (%)	20.12	19.02	18.61	17.80	14.60	20.85	19.78	20.02	19.20	15.67	21.70	20.65	21.40	20.51	16.83	22.47	21.70	22.50	21.96	18.45
Geometric Return (%)	18.63	17.72	17.24	16.62	12.96	19.24	18.31	18.40	17.78	14.25	19.96	19.01	19.58	18.90	15.19	20.62	19.90	20.57	20.14	16.79
Cumulative Return (%)	71.15	66.53	64.60	61.55	44.16	74.32	69.56	70.25	67.23	49.12	77.95	73.23	76.27	72.90	52.85	81.67	77.87	81.37	79.30	59.32
Annualized SD (%)	26.83	26.16	25.56	25.71	27.19	28.51	27.88	27.74	27.74	28.39	29.99	29.59	29.83	29.69	29.62	31.33	31.20	31.59	31.46	30.66
Annualized Skewness	0.28	0.16	0.17	0.13	0.11	0.32	0.17	0.21	0.18	0.15	0.34	0.19	0.23	0.21	0.16	0.32	0.19	0.23	0.20	0.13
Annualized Kurtosis	3.37	2.94	2.99	2.92	2.69	3.43	3.02	3.08	3.01	2.67	3.49	3.06	3.17	3.09	2.73	3.51	3.12	3.22	3.16	2.72
Maximum Drawdown (%)	-19.17	-19.63	-19.52	-19.44	-21.87	-20.37	-20.89	-20.85	-20.62	-22.89	-21.16	-22.09	-22.15	-21.96	-22.57	-22.19	-23.28	-23.29	-23.14	-24.30
Monthly 95% VaR (%)	-8.52	-8.89	-8.49	-8.82	-9.86	-9.00	-9.46	-9.20	-9.36	-9.92	-9.41	-9.95	-9.79	-9.90	-10.29	-9.79	-10.46	-10.31	-10.40	-10.75
Sharpe Ratio	0.96	0.92	0.91	0.86	0.63	0.93	0.89	0.90	0.86	0.65	0.92	0.87	0.90	0.87	0.67	0.92	0.87	0.90	0.88	0.71
Annualized Turnover	8.81	8.38	8.50	8.54	8.21	8.83	8.52	8.57	8.62	8.15	8.81	8.59	8.56	8.59	8.29	8.85	8.58	8.54	8.54	8.19

Table 6. This table presents the performance metrics for four portfolio construction methods in the JPX Prime 150: Simstock embedding(SS), historical covariance (HC), shrinkage method (SM), Gerber statistic (GS) and TS2VEC embedding (TS). The portfolios were optimized for four different risk target levels: 24%, 27%, 30%, and 33%. The performance was evaluated over the full testing period from January 2022 to February 2024. The 3-month U.S. Treasury Bill rate was used as the risk-free rate for performance calculations. Transaction costs were modeled as 10 basis points of the traded volume for each rebalancing event.

Conclusion

- How can we find Stock representations to identify similar stocks? → Use SimStock
- If we can identify similar stocks, what are the applications? → Pair trading, Direct indexing, Portfolio optimization,..etc



- SimStock demonstrates that temporal self-supervised learning can effectively identify similar stocks, offering practical benefits for investment strategies.

Appendix
(Model Architecture)

SimStock
(Model Architecture)

What is Temporal domain generalization?

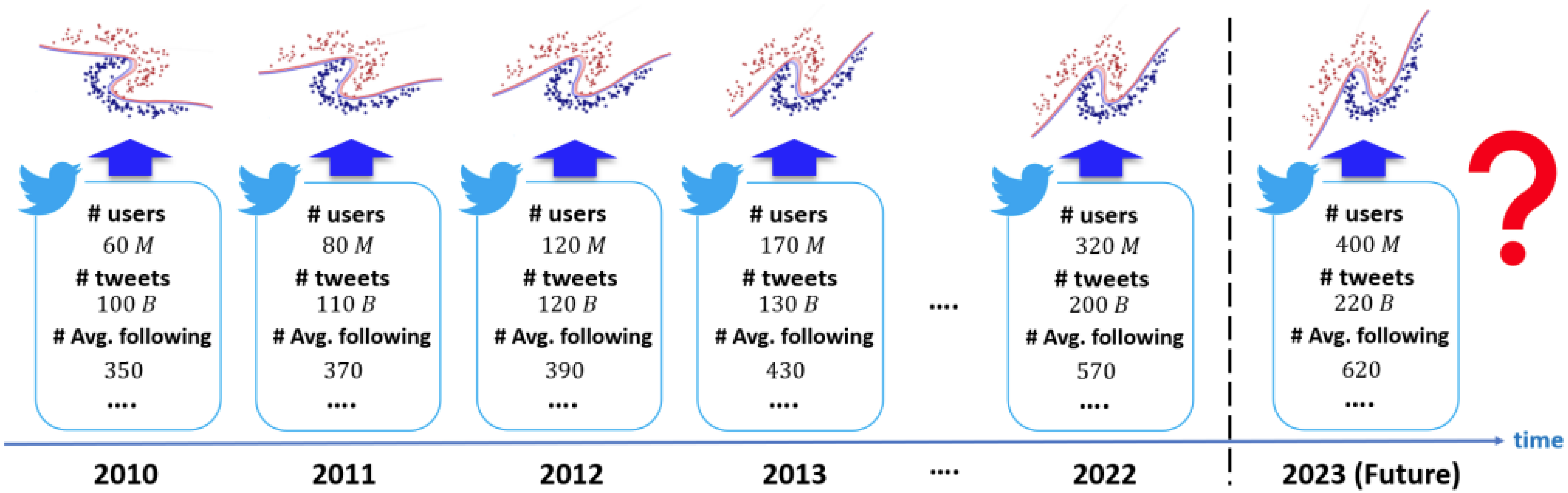


Figure 3. An illustrative example of temporal domain generalization

Temporal domain generalization(TDG)

- Consider training a model for some classification tasks based on the **annual twitter dataset** such that the trained model can generalize to the future domains (e.g., 2023). **The temporal drift of data distribution can influence the prediction model** such as the rotation of the decision boundary in this case.

What is Temporal domain generalization?

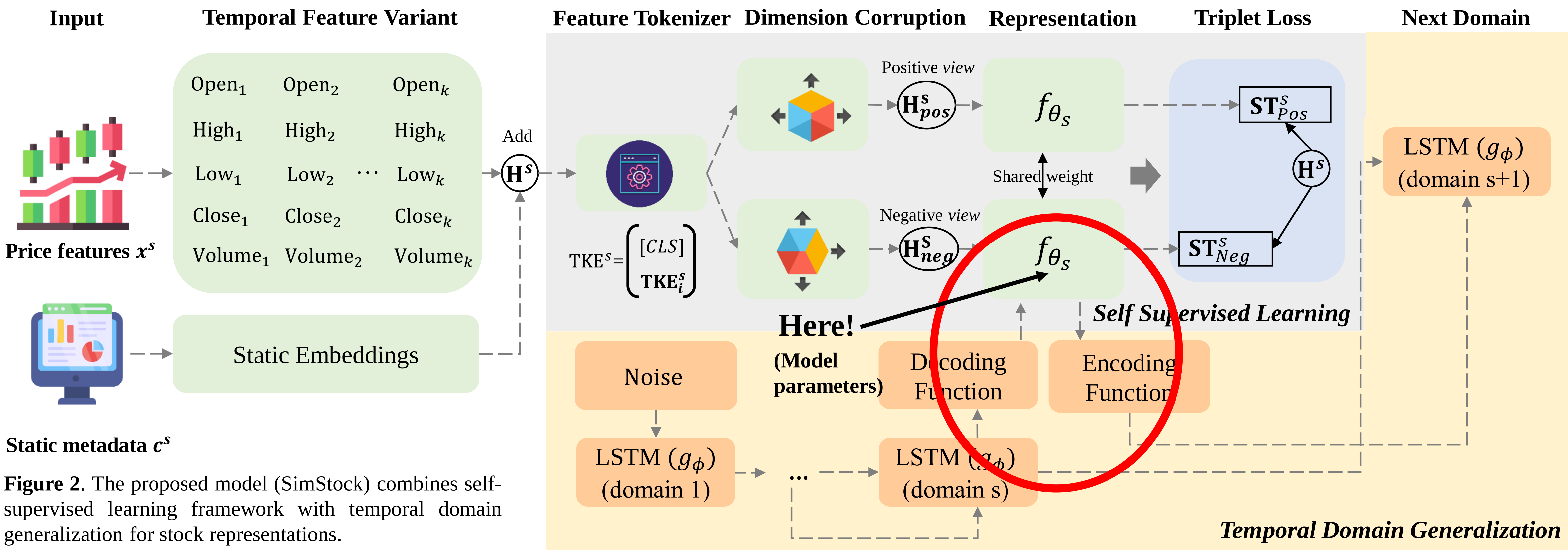


Figure 2. The proposed model (SimStock) combines self-supervised learning framework with temporal domain generalization for stock representations.

Temporal domain generalization(TDG)

- In each domain D_s, the representation f_{θ_s} can be trained by maximizing the conditional probability $\mathbb{P}(\theta_s|D_s)$.
- Here, θ_s signifies the state of the model parameters at timestamp t_s.
- Given the dynamic nature of D_s, the conditional probability $\mathbb{P}(\theta_s|D_s)$ will also change over time.

What is Temporal domain generalization?

Temporal domain generalization(TDG)

- The objective of temporal domain generalization is to estimate θ_{T+1} utilizing all the training data from $D_{1:T}$.
- From a probabilistic perspective, we can express this as:

$$\mathbb{P}(\theta_{T+1}|D_{1:T}) = \int_{\Omega} \underbrace{\mathbb{P}(\theta_{T+1}|\theta_{1:T}, D_{1:T})}_{\text{Inference}} \underbrace{\mathbb{P}(\theta_{1:T}|D_{1:T})}_{\text{Training}} \, d\theta_{1:T} \quad (1)$$

Where Ω denotes the space for model parameters $\theta_{1:T}$. In Eq. 1, the first term inside the $\mathbb{P}(\theta_{T+1}|\theta_{1:T}, D_{1:T})$ represents the inference phase, which is the process of predicting the future state of the target representation network (i.e., θ_{T+1}) given all historical state (i.e., $\theta_{1:T}, D_{1:T}$). The second term $\mathbb{P}(\theta_{1:T}|D_{1:T})$ signifies the training phase, which involves leveraging all training data $D_{1:T}$ to ascertain the state of the model on each source domain.

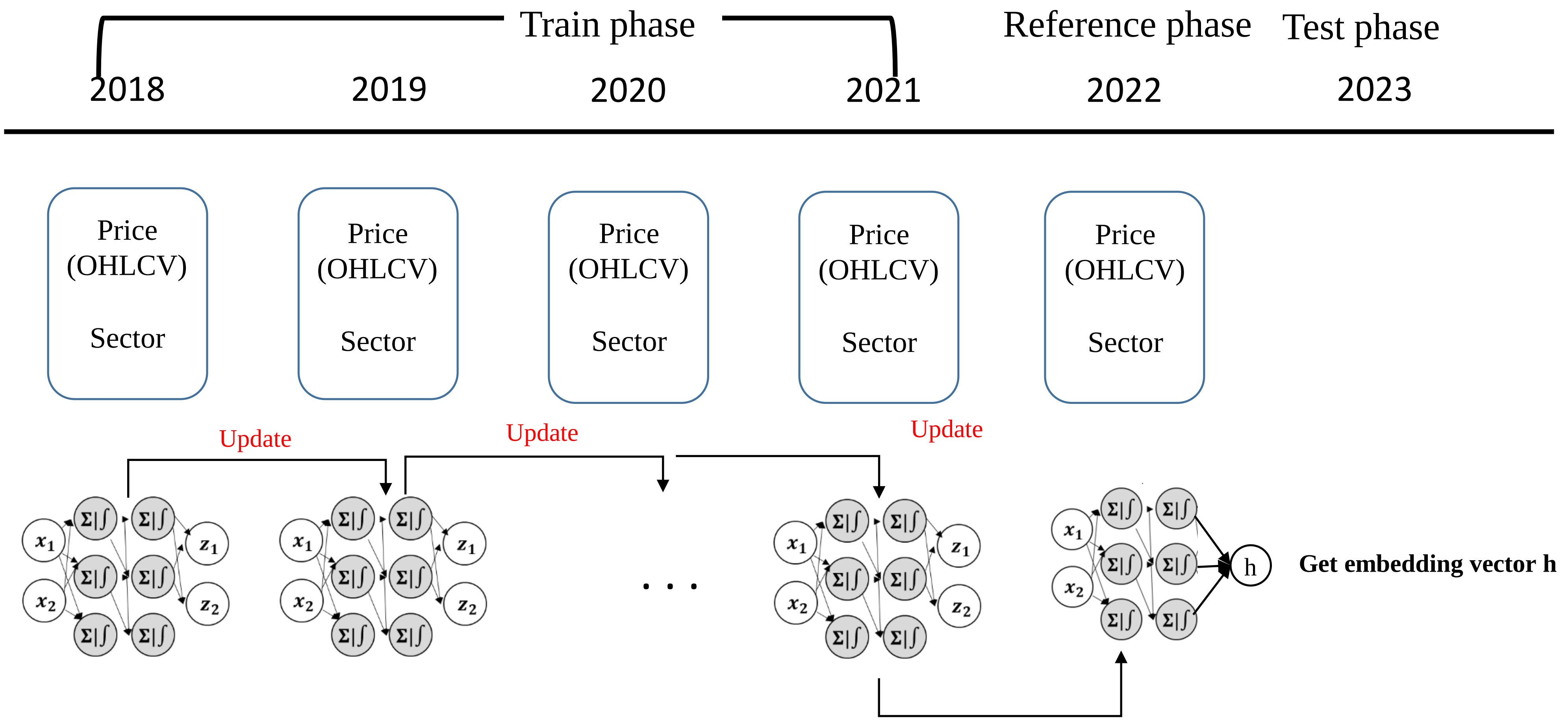
Training phase

- By chain rule, we can further decompose the training phases as follows:

$$\begin{aligned}\mathbb{P}(\theta_{1:T}|D_{1:T}) &= \prod_{s=1}^{T} \mathbb{P}(\theta_s|\theta_{1:s-1}, D_{1:T}) \\ &= \mathbb{P}(\theta_1|D_1)\mathbb{P}(\theta_2|\theta_1, D_{1:2}) \ldots \mathbb{P}(\theta_T|\theta_{1:T-1}, D_{1:T})\end{aligned}$$

Here, we assume for each time domain t_s, and the model parameters θ_s only depends on the current and previous domain, and there is no access to future data.

What is Temporal domain generalization?



: Representation layer

Q2. How to exploit the inductive bias for time-series data?
A2. Temporal domain generalization !

What is Self-Supervised learning?

- Self-supervised learning defines a pretext task based on **unlabeled inputs** to produce **representations**.
- Our goal is to learn a representation model $\boldsymbol{f_{\theta_s}}$, which captures the **stock data** that evolves over time.

To get a representation that reflects the characteristics of the stocks,

Q1. How can we create a **positive and negative *view***?

Q3. How do we *learn* temporal context?

To create a view for stocks, we propose the following method. (**A1 & A3**)

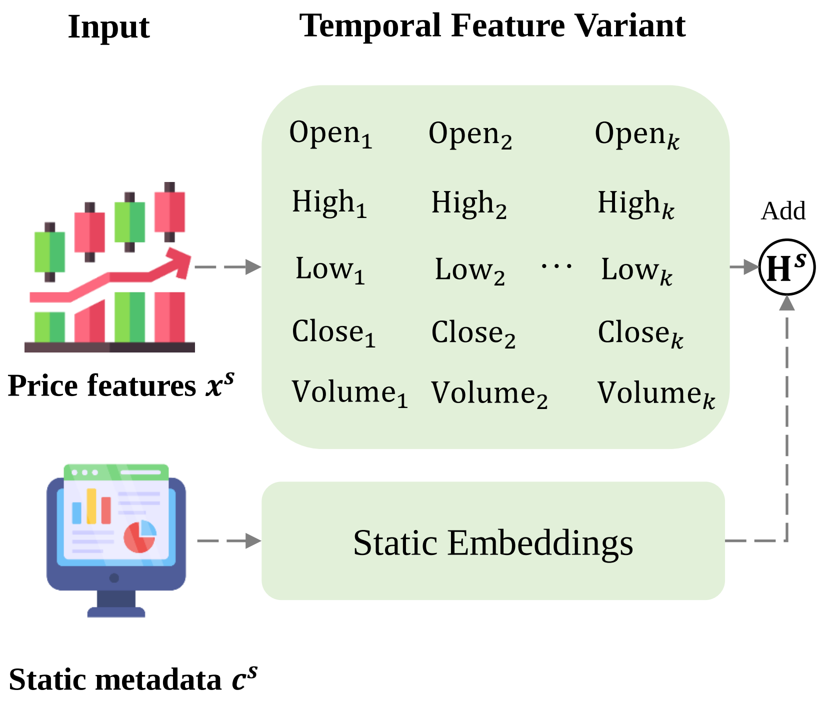
Temporal Feature Variant

Feature Tokenizer

(Temporal) Dimension Corruption

Triplet loss

Temporal Representation Learning (Step 1)



$\mathbf{H}^s$ is combined embedding that incorporates both temporal feature variant $\mu(x^s)$ and the embedded static meta data $\text{Embed}(c^s)$.

$$\mathbf{H}^s = \mu(x^s) + \text{Embed}(c^s) \in \mathbb{R}^{d_{mk}}$$

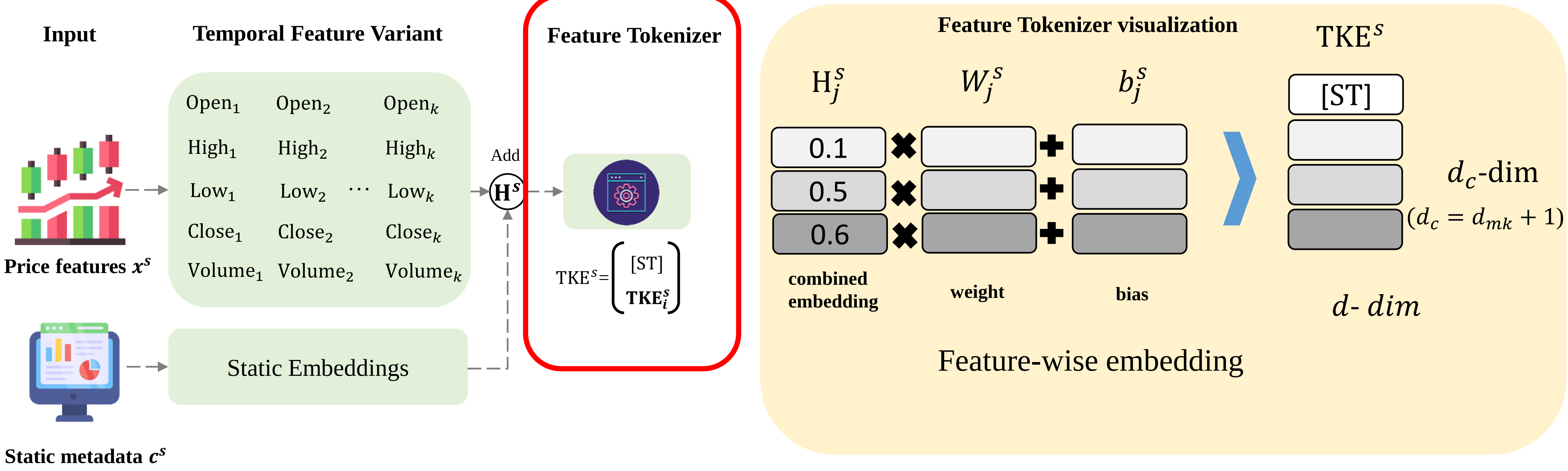
Where $\mu(x^s) = \text{CONCAT}(\mu_1(x^s), \mu_2(x^s), \dots, \mu_k(x^s)) \in \mathbb{R}^{d_{mk}}$.

Temporal Feature Variant (For make combined embedding)

- The time-varying patterns of stock prices are essential for identifying short- and long-term characteristics of stocks.
- To learn more rich representations, a price feature x^s is processed by a temporal transformation module μ.
- The price feature x^s is provided with k variations, denoted as $\mu(x^s) = \text{CONCAT}(\mu_1(x^s), \mu_2(x^s), \dots, \mu_k(x^s)) \in \mathbb{R}^{d_{mk}}$.

Here, $d_{mk} = d_m \times k$, and each $\mu_1, \mu_2 \dots, \mu_k \in U$, where U denotes the collection of temporal transformations.

Temporal Representation Learning (Step 2)



Feature Tokenizer

- The feature-wise token embedding $\mathbf{TKE}_j^s$ for given feature index j are computed as $\mathbf{TKE}_j^s = \mathrm{b}_j^s + \mathrm{H}_j^s \mathrm{W}_j^s$. Where $\mathrm{b}_j^s \in \mathbb{R}^d$ is the j-th feature bias term and $\mathrm{W}_j^s \in \mathbb{R}^d$ is the weight vector for j-th feature. Through this process, we can create efficient embeddings for various time-related features.
- The token embedding $\mathbf{TKE}^s \in \mathbb{R}^{d_c \times d}$ can be obtained by stacking all of the feature embedding $\mathbf{TKE}_j^s$ and adding a special [ST] token, which is known to process the essence of information after training.

Temporal Representation Learning (Step 3)

Temporal Dimension Corruption (*View* construction)

- We create positive and negative *views*, $\mathbf{H}^s_{pos}$ and $\mathbf{H}^s_{neg}$, by <u>randomly shuffling the dimension</u> within the $\mathbf{TKE}^s$. Here, we define two permutation matrices, $\mathbf{P}^s_{pos}$ and $\mathbf{P}^s_{neg}$ both size $d \times d$[1].

$$\mathbf{H}^s_{pos} = \lambda\mathbf{TKE}^s + (1-\lambda)\mathbf{TKE}^s\mathbf{P}^s_{pos} \tag{5}$$
$$\mathbf{H}^s_{neg} = (1-\lambda)\mathbf{TKE}^s + \lambda\mathbf{TKE}^s\mathbf{P}^s_{neg} \tag{6}$$

- In this case, the formulas (5) and (6) generate positive and negative *views* for SSL. The degree of this <u>perturbation</u> in both views is determined by <u>the mixing parameter λ.</u>
- The positive view $\mathbf{H}^s_{pos}$ has minor perturbations, maintaining much of the original token embedding ($\mathbf{TKE}^s$).
- The negative view $\mathbf{H}^s_{neg}$ is more altered, with greater dimension shuffling, deviation more from the original ($\mathbf{TKE}^s$).

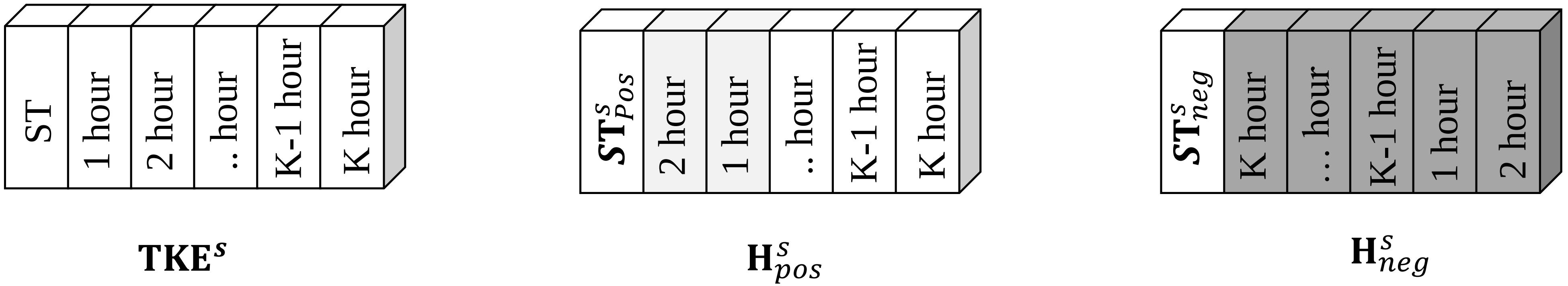


Fig 3. High-level overview of our dimension corruption method

1] A permutation matrix is a square 0-1 matrix that has exactly one enty of 1 in each row and each column and 0s elsewhere.

Temporal Representation Learning (Step 4)

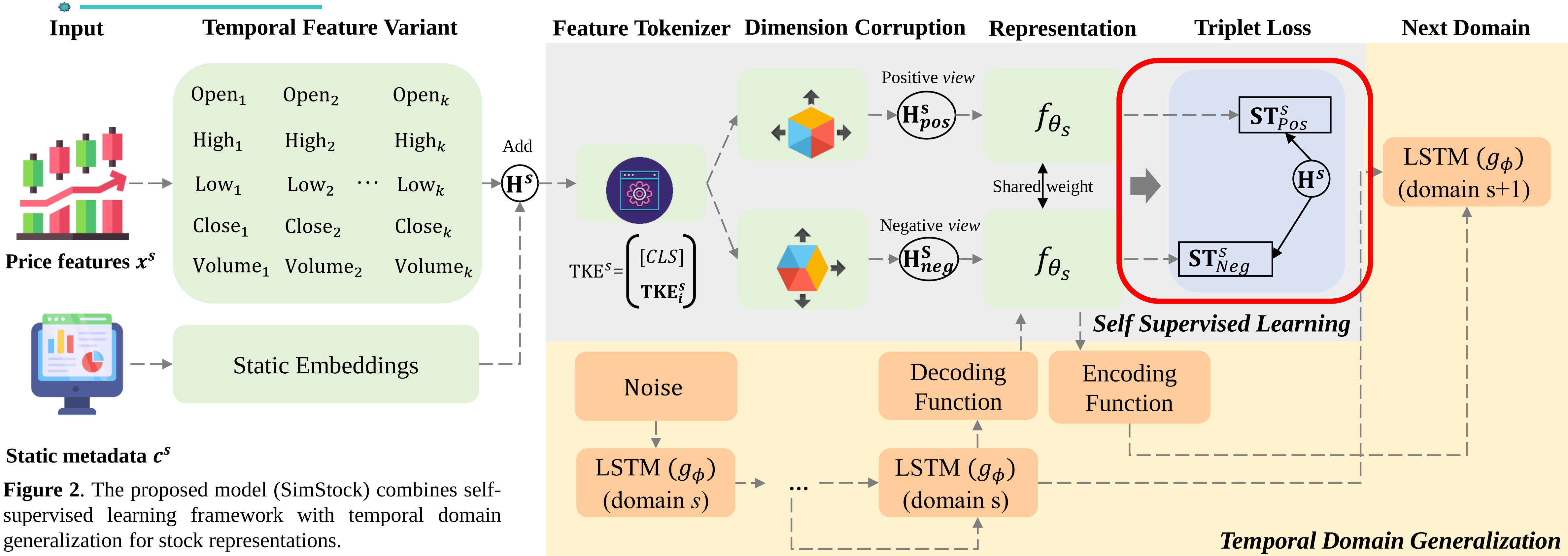


Figure 2. The proposed model (SimStock) combines self-supervised learning framework with temporal domain generalization for stock representations.

Triplet loss

- We train it to minimize a triplet loss, which is a popular choice in SSL.
- For the triplet ($\mathrm{ST}_{pos}^s, \mathrm{ST}_{neg}^s, \mathbf{H}^s$), where CLS_{pos}^s is the positive view, ST_{neg}^s is negative view, and $\mathbf{H}^s$ is the combined embedding's(anchor), the triplet loss is defined as follows:

$$\mathcal{L}_{\text{triplet}} = \mathrm{RELU}\big(\mathrm{sim}(\mathbf{H}^s, \mathrm{ST}_{pos}^s) - \mathrm{sim}(\mathbf{H}^s, \mathrm{ST}_{neg}^s) + \alpha\big), \alpha > 0$$

Pairs Trading

Application to Pairs trading (Motivation)

Example : **Pair trading** : How do I find similar stocks to pair trade? ⇨ Cointegration test

Cointegration is a very interesting property that can be exploited in finance for trading.

Predicting individual stocks can be difficult, but predicting the relative movements between stocks may be easier.

Illustrative example: A drunk man is walking a dog around the street (random walk). The paths of both the man and the dog are unpredictable and not fixed, but the distance between them tends to revert to the mean and remains relatively stable. Is it TRUE?

Historical period

?

Future period

Application to Pairs trading

If we find similar stocks, we can calculate the price ratio of the two stocks for pair trading as follows.

Apple price data

The spread (price ratio) $PR_t = \frac{y_{1t}}{y_{2t}}$ is mean reverting.

Microsoft price data

Finding similar stocks based on historical data.

- This mean-reverting property of the spread can be exploited for trading and it is commonly referred to as “pairs trading” or “statistical arbitrage”

Procedure

- To perform pairs trading, we need to identify "similar stocks”.
- Once these similar stocks are found, the spread between the two stocks is calculated.
- Since these two stocks are similar, they are expected to follow a mean-reverting property.

Application to Pairs trading

- Illustration on how to trade the price ratio $PR_t = \frac{y_{1t}}{y_{2t}}$.

The idea behind *pairs trading* is to

- short-sell the relatively overvalued stocks and buy the relatively undervalued stocks
- unwind the position when they are relatively fairly valued.

Application to Pairs trading (Result)

Query Stock	Method	TOP@3 similar stocks			Query Stock	Method	TOP@3 similar stocks		
		First	Second	Third			First	Second	Third
	SimStock	MSFT	TYL	INTU		SimStock	BNTX	MRNA	JNJ
	TS2VEC	AMZN	WTM	AMD		TS2VEC	NKNG	DJCO	ESGR
AAPL	Corr1	TMO	SNPS	CNDS	PFE	Corr1	ICL	MCBS	PTSI
	Corr2	GLOB	AMZN	TYL		Corr2	RMR	NVS	BUD
	Peer	MSFT	NVDA	ASML		Peer	LLY	ABBV	NVO
	SimStock	AMZN	MANH	MSFT		SimStock	CMG	INTU	MANH
	TS2VEC	NVR	USLM	NEU		TS2VEC	AAPL	F	PLPC
CMG	Corr1	DHR	DSGX	PCTY	AMZN	Corr1	ACMR	ADBE	FIVN
	Corr2	PAYC	PCTY	MANH		Corr2	TEAM	LYV	GENE
	Peer	HLT	RACE	AZO		Peer	TSLA	BKNG	SBUX
	SimStock	CDNS	MANH	TYL		SimStock	IVZ	SPR	UAA
	TS2VEC	GOOG	GOOGL	MA		TS2VEC	LPL	NNI	FCX
MSFT	Corr1	DHR	TMO	FAST	BA	Corr1	NCLH	SPR	SOHO
	Corr2	GOOGL	GOOGL	DAVA		Corr2	RVSB	ENVA	STT
	Peer	AAPL	NVDA	ASML		Peer	NOC	CNI	WM
	SimStock	BAC	FITB	FNB		SimStock	SPOT	PYPL	FORM
	TS2VEC	JPM	C	MA		TS2VEC	UHAL	MAR	MSFT
WFC	Corr1	BHLH	WNEB	RVSB	META	Corr1	CVNA	GREE	INDP
	Corr2	BAC	WBS	CFG		Corr2	SKYW	JAGX	CTHR
	Peer	CHTR	NTES	ATVI		Peer	MCD	LOW	TM
	SimStock	MA	SF	IHG		SimStock	V	BKNG	IHG
	TS2VEC	MA	MSFT	KO		TS2VEC	V	KO	NUE
V	Corr1	MA	TDY	ROP	MA	Corr1	V	TDY	FICO
	Corr2	MA	PLNT	RELX		Corr2	V	GES	RTO
	Peer	MA	ADBE	CSCO		Peer	JPM	BAC	V
	SimStock	MRO	CVE	HES		SimStock	CNC	BMO	MS
	TS2VEC	MRO	CVX	NUE		TS2VEC	HUM	VNR	BKNG
XOM	Corr1	MUR	MRO	EOG	CVS	Corr1	CCB	CHRD	RJF
	Corr2	MPC	HES	ERF		Corr2	SRCL	MLM	NMFC
	Peer	CVX	SHEL	TTE		Peer	ANTM	MDT	GSK

Table 2. Top@3 similar stocks identified by SimStock and baseline methods (TS2VEC, Corr1, Corr2, and Peer) for a diverse set of query stocks from the technology, healthcare, energy, and financial sectors.

Index tracking

Application to index tracking of thematic ETFs (Motivation)

An index is essentially a proxy for the entire universe of investments.

Characteristic	Passive Funds	Active Funds
Management Style	Passively tracks a specific index (e.g., S&P500)	Actively selected holdings based on fund manager's discretion
Costs	Very low	Relatively high
Investment Scope	Holdings within the tracked index	Varies based on fund manager's strategy
Diversification	Automatic diversification based on index composition	Depends on fund manager's strategy
Expected Returns	Average returns of the index	Potential to outperform the index, depending on fund manager's skill
Risks	Volatility of the index	Risks associated with fund manager's ability and strategy

← Even if the costs are low, these expenses typically burden individual investors.

Motivation

- Passive funds typically track an index itself, while active funds manage assets to maximize returns.
- However, individual investors might want to create a portfolio that suits their personal preferences, independent of the portfolio manager's discretion.
- We examine whether the proposed methodology enables individual investors to effectively track a specific index by selecting only a small number of assets.

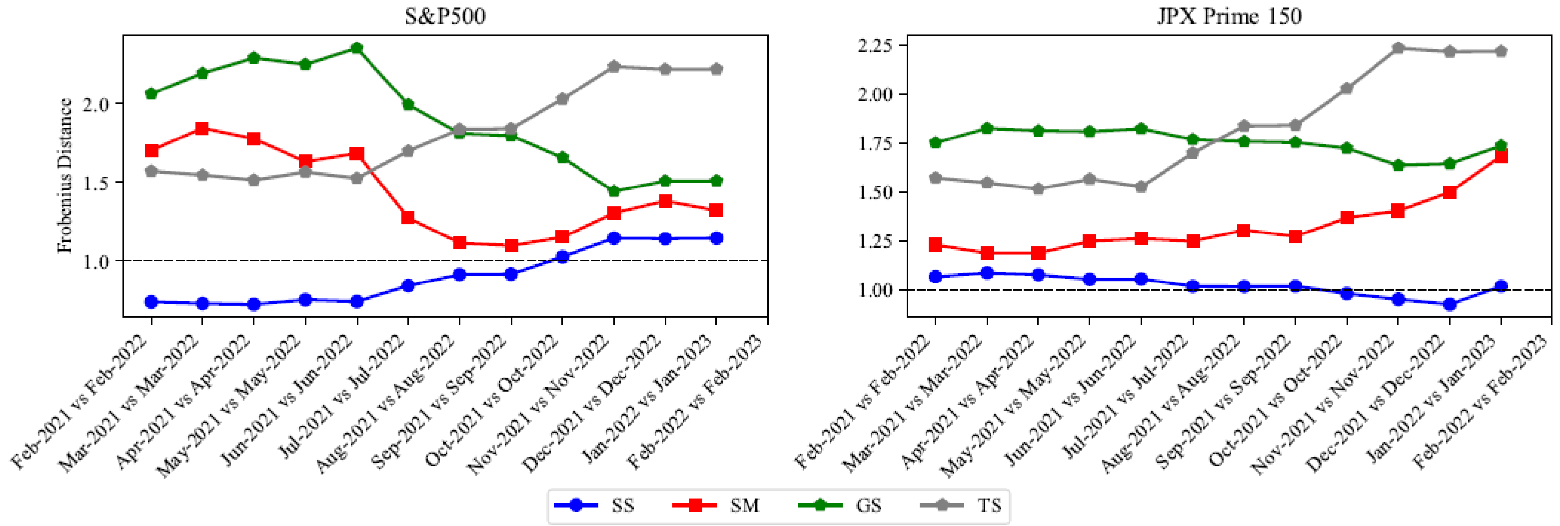
Portfolio Optimizations

Application to Portfolio optimization

What distinguishes from other portfolio optimization methods

$$\frac{\|MD - RC^{future}\|_F}{\|MD - RC^{past}\|_F} \leq 1$$

- Here, MD refers to the correlation matrix obtained using a specific methodology (e.g., SS, SM, GS, and TS), While RC^{future} and RC^{past} represent the realized correlation matrices for the future and past period, respectively.



Application to Portfolio optimization

Benchmark models

- **SimStock Embedding (SS) (ours)**
- **Historical Covariance (HC)**
- **The Shrinkage Method (SM) (Ledoit et al., 2003)**
- **The Gerber Statistic (GS) (Gerber et al., 2021)**
- **TS2VEC (TS) (Yue et al., 2022)**

Introduction

We estimate the expected return μ_{ti} for asset i at time t using the sample mean of its historical returns over a T-month lookback window. We set the T equal to 12 months. This setting same to Gerber et al., 2021