Retrieval-Augmented Diffusion Modeling for Stochastic Discount Factor Portfolios

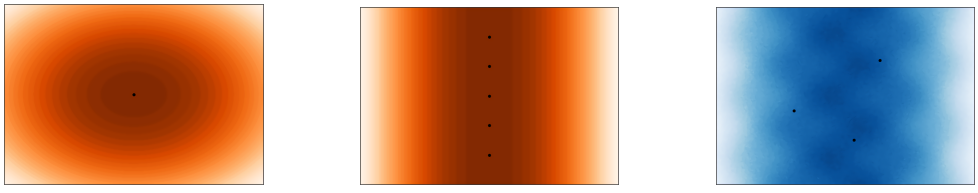
Kelvin J.L. Koa[*†] **Xinyang Li**[*] **Ke-Wei Huang**
National University of Singapore Asian Institute of Digital Finance
kelvin.koa@u.nus.edu, xinyangli5579@gmail.com, dishkw@nus.edu.sg

Abstract

In this work, we study portfolio optimization under the stochastic discount factor (SDF) framework by learning market state representations that capture the underlying risk structures of financial data. This is challenging due to several factors: financial markets exhibit non-stationary dynamics with shifting regimes, multimodal inputs such as price and news data often contain stochastic noise, and existing diffusion-based approaches, while effective for modeling stochastic dynamics, rely on assumptions such as isotropic Gaussian noise that fail to capture the state-dependent nature of financial uncertainty. To address these challenges, we introduce RADAR, a retrieval-augmented diffusion framework that learns market representations by conditioning on similar historical regimes. RADAR leverages retrieval to construct context-dependent noise distributions, applies conditional diffusion to denoise multimodal representations, and initializes the diffusion process using empirical statistics to reflect state-dependent uncertainty. Experiments show that RADAR achieves state-of-the-art performance on key risk-adjusted metrics while producing economically meaningful signals on asset returns and correlations. Our code is available at: https://github.com/xinyangli5579-star/RADAR.

1 Introduction

Portfolio optimization is a central problem in quantitative finance, where the objective is to allocate capital across assets to achieve an optimal trade-off between risk and return [38]. Classical approaches such as mean–variance optimization [37] formalize this trade-off by maximizing expected returns subject to a penalty on its variance, thereby optimizing for stable portfolio returns under uncertain conditions. In modern asset pricing theory, this objective can be equivalently expressed through the stochastic discount factor (SDF) [22, 23], which provides a unified framework for valuing uncertain future financial payoffs. The SDF characterizes asset values as expectations under a market state-dependent weighting of payoffs, thereby encoding both risk and return within a single object.



(a) The returns forecasting task optimizes for a single point optimum. (b) The SDF estimation task admits a manifold of equivalent solutions. (c) Noisy representations distort manifold, causing spurious minima.

Figure 1: Comparison of forecasting and SDF objectives, and the impact of noisy representations.

Unlike returns forecasting [9], which focuses on predicting conditional expectations, SDF estimation is inherently underdetermined and requires learning representations that are sufficient to satisfy

[*] Equal contribution. † Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

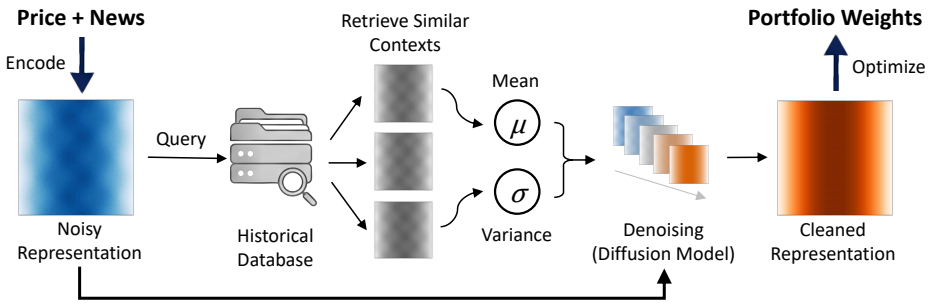


Figure 2: Overview of RADAR, which learns a clean market state representation for SDF estimation.

pricing constraints across all assets (see Figure 1a-b). This requires capturing the joint structure of market states and their associated risk, rather than extracting predictive signals for individual assets.

To obtain market representations, two fundamental limitations exist in current deep-learning methods. Firstly, financial markets are inherently non-stationary, often exhibiting regime shifts [16] that alter the structural relationships between features and returns. As a result, models can be biased by the relative coverage of different regimes in the dataset. Some methods attempt to address non-stationarity through temporal weighting [31] or trend decomposition [51], but they typically rely on heuristics and fail to capture contextual similarities across market states. Secondly, while textual data provides useful signals [20, 52], it is often noisy and contains irrelevant information with limited economic relevance [49]. Training on such embeddings makes it harder for the model to capture economically meaningful signals, leading to suboptimal pricing and asset allocation decisions (see Figure 1c).

Among existing modeling paradigms, diffusion-based methods have recently emerged as a promising approach for learning robust representations and capturing uncertainty in stochastic environments [24, 15]. However, they typically operate only on the time-series modality and rely on isotropic Gaussian noise during training, which does not capture the state-dependent and heteroskedastic nature of financial uncertainty. The closest alternative is Retrieval-Augmented Time-series Diffusion model (RATD) [30], which retrieves similar historical states as reference to a diffusion model. However, these retrieved states are used to condition the denoising process rather than guide the stochastic noise generation in the diffusion process. Furthermore, they also do not deal with multimodal inputs.

To address these, we propose **R**etrieval-**A**ugmented **D**iffusion-based **A**ssets **R**epresentation (RADAR), a framework for learning market representations for SDF estimation (see Figure 2). Firstly, to account for non-stationarity, we introduce a retrieval-augmented diffusion process that conditions on contextually similar historical regimes. This enables regime-aware representations that are grounded in relevant historical conditions rather than aggregated across unrelated periods. Secondly, to mitigate noise in unstructured data, we treat the latent representations of both price and news as corrupted observations and refine them via a conditional diffusion process, which iteratively denoises the representations. This allows the model to generalize across similar contexts while suppressing the idiosyncratic noise from individual observations. Lastly, to capture state-dependent uncertainty, we initialize the diffusion process using the mean and variance of the retrieved contexts. This grounds the denoising trajectory in empirical conditional distributions, enabling the learned representations to better reflect the risk–return characteristics of the current market state for downstream SDF estimation.

Experimental results show that RADAR consistently achieves state-of-the-art performance across key risk-adjusted metrics, including Sharpe, Sortino, and Calmar ratios. Additional analyses on the learned representations also show that the model produce economically meaningful signals on asset returns and correlations that are beneficial for portfolio outcomes. In summary, our contributions are:

- We introduce a state-dependent diffusion formulation where the noise distribution is conditioned on retrieved historical contexts, addressing a key mismatch between standard diffusion assumptions and financial data.
- We show that retrieval can be used to model conditional uncertainty, enabling regime-aware representation learning in non-stationary environments.
- We propose the RADAR framework for learning clean market representations for SDF-based portfolio optimization using multimodal inputs and context-aware denoising.
- We demonstrate strong empirical performance and provide analyses showing that the learned representations are economically meaningful and effective for portfolio construction.

2 Related Works

Financial Forecasting. The financial forecasting task is primarily concerned with extracting predictive signals from input data to estimate stock prices. While early approaches in financial forecasting explored the applicability of deep learning techniques on forecasting stock movements [9, 43], recent works have seen practitioners incorporating stylized financial characteristics into their model designs. For example, works have explored the use of textual information [20, 52] to capture market trends and volatility, following the concept of informationally efficient financial markets [11, 25]. Other works have also explored using diffusion noise to model the stochastic noise in financial prices [24, 15], following the idea of randomness in financial markets [36, 12]. These two perspectives are typically studied in isolation in existing works. In contrast, our approach integrates both information-driven signals and stochastic noise modeling, and applies this unified framework to portfolio applications.

Deep Learning for Portfolio Optimization. The portfolio optimization task aims to allocate capital across multiple assets to achieve optimal balance between risk and returns [38]. Beyond classical mean-variance approaches [37], recent works have also seen practitioners incorporate deep-learning methods into portfolio optimization applications. For example, Hierarchical Risk Parity (HRP) [34] leverages hierarchical clustering to construct diversified portfolios based on graph theory; reinforcement learning (RL) is used to learn portfolio weights that dynamically adjust over time [21, 50]. The most recent theory-based approaches adopt the stochastic discount factor (SDF) framework [22, 23], which provides a unified formulation for evaluating assets through risk-adjusted valuation. Rather than directly forecasting returns, this framework learns a pricing model that leverages shared structure across assets, implicitly capturing risk premia and state-dependent risk-return trade-offs. This aligns with a broader trend of learning structured representations in financial markets [47, 39, 14] which aim to encode cross-asset relationships for downstream portfolio decisions. However, learning such asset representations remains challenging in practice due to non-stationarity, noisy inputs, and the context-dependent nature of financial risk, which our work aims to tackle.

Non-Isotropic Diffusion Processes. Standard diffusion models typically adopt a fixed isotropic Gaussian noise process [18]. However, recent works have also explored the use of non-isotropic diffusion processes. For example, some approaches learn input-dependent multivariate noise schedules, allowing different data dimensions to receive noise at different rates [46]. Other works explored using flexible forward processes to make the reverse generation path easier to learn [4], or performing time-dependent non-linear transformations of the data during diffusion [3]. In time-series forecasting, works have introduced learned non-linear transformations and conditioning variables into the forward process [45], or adapted the endpoint distribution and noise schedule using forecasted conditional statistics [53, 28]. However, most of these works primarily learn how to modify the diffusion process, whereas our approach constructs the noise distribution directly from retrieved historical samples.

3 The RADAR Framework

In this section, we first study the portfolio optimization problem within the stochastic discount factor (SDF) framework from a deep-learning perspective. We then explain the proposed **R**etrieval-**A**ugmented **D**iffusion-based **A**ssets **R**epresentation (RADAR) framework, shown in Figure 3.

3.1 Problem Formulation

At each time step t, we observe two input modalities which are describing the market state: time-series asset prices $\mathbf{x}_t^{(p)}$ and textual news data $\mathbf{x}_t^{(n)}$. Our goal is to learn the joint representative embeddings:

$$\hat{\mathbf{z}}_t = f(\mathbf{x}_t^{(p)}, \mathbf{x}_t^{(n)}), \tag{1}$$

which should capture useful information about the current market state. This representation is used to learn the asset portfolio weights $\hat{\mathbf{w}}_t \in \mathbb{R}^N$, where N is the number of assets. Formally, we have:

$$\hat{\mathbf{w}}_t = g(\hat{\mathbf{z}}_t). \tag{2}$$

We adopt the stochastic discount factor (SDF) framework [23, 17] to define a principled training objective. Given the next-period asset return $\mathbf{r}_{t+1} \in \mathbb{R}^N$, the predicted weights induce a portfolio return $\hat{\mathbf{w}}_t^\top \mathbf{r}_{t+1}$, where a value of 0 corresponds to no change in wealth. The SDF is then defined as $1 - \hat{\mathbf{w}}_t^\top \mathbf{r}_{t+1}$, and minimizing its second moment makes it price every asset with zero error [23].

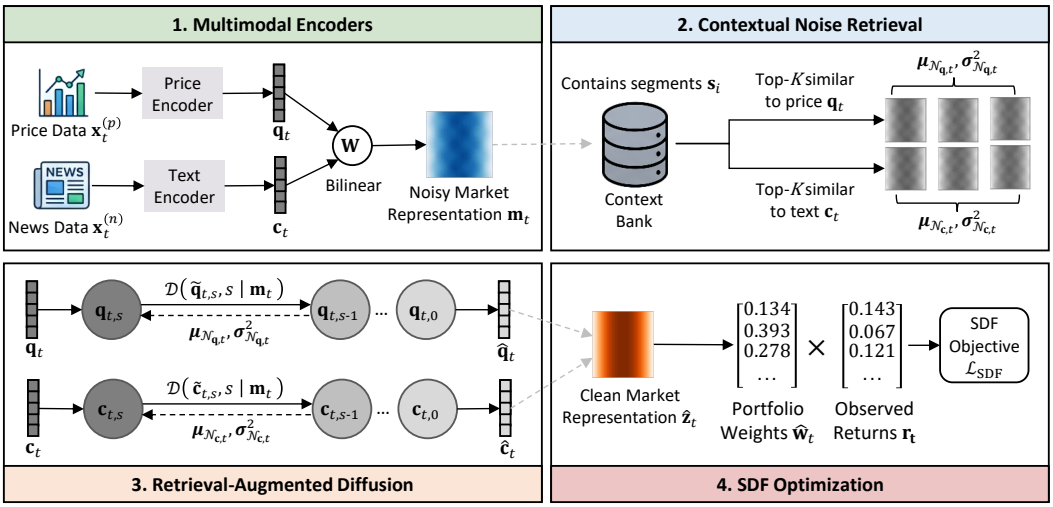


Figure 3: The RADAR framework. The price and text embeddings, noisy market representation and retrieved contextual noise are used in the diffusion process to obtain the clean market representation.

Intuitively, minimizing this pricing error encourages the model to ensure that portfolio outcomes are consistent with the information used to construct them. Both positive and negative pricing errors on an asset indicate a mismatch between the learned representation and future returns. In particular, systematic mispricing suggests that the representation $\hat{\mathbf{z}}_t$ fails to capture structure in the returns that could be explained by the input data. By penalizing such errors, the model is driven to learn representations that account for the predictable components of returns, rather than spurious patterns.

Following this, we then train the model by minimizing the corresponding SDF-based objective:

$$\mathcal{L}_{\text{SDF}} = \mathbb{E}_t \left[\left(1 - \hat{\mathbf{w}}_t^\top \mathbf{r}_{t+1}\right)^2 \right] + \lambda \|\hat{\mathbf{w}}_t\|_2^2, \tag{3}$$

where $\lambda \|\hat{\mathbf{w}}_t\|_2^2$ is a ℓ_2 regularization term to prevent any unstable extreme [27] portfolio weights.

3.2 Multimodal Encoders

To obtain the initial embeddings from the heterogeneous multi-modal inputs, we first employ modality-specific encoders [48] for the time-series asset prices and textual news data, and blend them.

For the price time-series data, we model the temporal dependencies using a gated recurrent unit (GRU) price encoder. Given the historical price sequence $\mathbf{x}_t^{(p)}$, we extract the hidden states, which are then aggregated via a temporal attention mechanism Attn_p to produce the price representation:

$$\mathbf{q}_t = \text{Attn}_p\big(\text{GRU}_p(\mathbf{x}_t^{(p)})\big), \tag{4}$$

For the text modality, we leverage pre-trained financial language models [2] to extract embeddings from the textual news data $\mathbf{x}_t^{(n)}$. We then model the sequential dependencies of these embeddings using a GRU news encoder and a sequential attention mechanism to derive the text representation:

$$\mathbf{c}_t = \text{Attn}_n\left(\text{GRU}_n\big(\text{Embed}(\mathbf{x}_t^{(n)})\big)\right), \tag{5}$$

Given the modality-specific representations $\mathbf{q}_t$ and $\mathbf{c}_t$, we blend them by applying a bilinear transformation which captures cross-modality dependencies. Specifically, we compute a joint representation:

$$\mathbf{m}_t = \text{ReLU}\left(\mathbf{q}_t^\top \mathbf{W} \mathbf{c}_t + \mathbf{b}\right), \tag{6}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the learnable parameters, and ReLU is the non-linear activation function.

3.3 Contextual Noise Retrieval

The learnt representation is unstable, as asset price data contains stochastic noise [24] while news data often contains chaotic irrelevant information [20], which requires denoising. On the other

hand, standard diffusion denoising techniques typically rely on isotropic Gaussian noise, which is independent of the input and may fail to preserve meaningful structure in the representation.

To address this issue, we introduce a retrieval-based mechanism to construct context-dependent noise distributions from historically similar market states. The data is first organized around news events. Let $\{\tau_i\}_{i=1}^{M}$ denote the time indices at which news events arrive, ordered chronologically. For each news event at time τ_i, we define an event segment consisting of the news observed at that time together with their subsequent asset price sequences up to, but excluding, the next news event:

$$\mathbf{s}_i = \left(\mathbf{x}_{\tau_i}^{(n)},\ \mathbf{x}_{\tau_i:\tau_{i+1}-1}^{(p)} \right). \tag{7}$$

Each segment $\mathbf{s}_i$ captures the impact of a news event on the market before new information arrives. We then encode each event segment using the same encoders and bilinear transformation used in Equations 4, 5 and 6, to obtain the segment representations $\tilde{\mathbf{s}}_i$, which are stored in a *context bank*.

At each time step t, we retrieve the most relevant historical segments in the current context bank across both modalities, calculated using the cosine similarity metric. For each modality, we have:

$$\mathcal{N}_{\mathbf{q},t} = \mathrm{TopK}_{i:\tau_i<t} \cos\left(\mathbf{q}_t, \tilde{\mathbf{s}}_i\right), \tag{8}$$

This captures similar historical price trajectories, and the news information that just preceded them.

$$\mathcal{N}_{\mathbf{c},t} = \mathrm{TopK}_{i:\tau_i<t} \cos\left(\mathbf{c}_t, \tilde{\mathbf{s}}_i\right), \tag{9}$$

This captures historical news with similar semantic content, and their subsequent price trajectories.

The retrieved sets represent the possible states that the market could be in, for each modality. We model their means and variances as noise distribution parameters. For the price modality, we have:

$$\boldsymbol{\mu}_{\mathcal{N}_{\mathbf{q},t}} = \frac{1}{K}\sum_{k=1}^{K} \tilde{\mathbf{s}}_k, \quad \boldsymbol{\sigma}^2_{\mathcal{N}_{\mathbf{q},t}} = \frac{1}{K}\sum_{k=1}^{K} \left(\tilde{\mathbf{s}}_k - \boldsymbol{\mu}_{\mathcal{N}_{\mathbf{q},t}}\right)^2, \qquad \tilde{\mathbf{s}}_k \in \mathcal{N}_{\mathbf{q},t}. \tag{10}$$

The noise distribution of the text modality $\left(\boldsymbol{\mu}_{\mathcal{N}_{\mathbf{c},t}}, \boldsymbol{\sigma}^2_{\mathcal{N}_{\mathbf{c},t}}\right)$ is then calculated in a similar manner.

3.4 Retrieval-Augmented Diffusion

In this step, we refine the learnt modality embeddings $\mathbf{q}_t$ and $\mathbf{c}_t$ through a diffusion process, guided by the retrieved contextual noise distribution and conditioned on the joint market representation.

Forward Process. We adopt a dual-path formulation that perturbs the price and text representations independently. At each diffusion step s, we inject the retrieved contextual noise into the embeddings:

$$\tilde{\mathbf{q}}_{t,s} = \sqrt{\bar{\alpha}_s}\mathbf{q}_t + \sqrt{1-\bar{\alpha}_s}\left(\boldsymbol{\mu}_{\mathcal{N}_{\mathbf{q},t}} + \boldsymbol{\sigma}_{\mathcal{N}_{\mathbf{q},t}} \odot \boldsymbol{\epsilon}_{\mathbf{q}}\right), \tag{11}$$

$$\tilde{\mathbf{c}}_{t,s} = \sqrt{\bar{\alpha}_s}\mathbf{c}_t + \sqrt{1-\bar{\alpha}_s}\left(\boldsymbol{\mu}_{\mathcal{N}_{\mathbf{c},t}} + \boldsymbol{\sigma}_{\mathcal{N}_{\mathbf{c},t}} \odot \boldsymbol{\epsilon}_{\mathbf{c}}\right), \tag{12}$$

where $\boldsymbol{\epsilon}_{\mathbf{q}}, \boldsymbol{\epsilon}_{\mathbf{c}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\bar{\alpha}_s$ controls the noise schedule. Unlike standard diffusion models that employ isotropic Gaussian noise, the perturbations are governed by context-specific statistics, allowing them to model potential unrealized paths using historically similar market states.

Reverse Process. To recover clean representations, we employ the denoising functions $\mathcal{D}_p$ and $\mathcal{D}_n$:

$$\hat{\mathbf{q}}_t = \mathcal{D}_p\left(\tilde{\mathbf{q}}_{t,s}, s \mid \mathbf{m}_t\right), \qquad \hat{\mathbf{c}}_t = \mathcal{D}_n\left(\tilde{\mathbf{c}}_{t,s}, s \mid \mathbf{m}_t\right). \tag{13}$$

Each denoiser conditions on the shared market representation $\mathbf{m}_t$, enabling cross-modal information exchange during reconstruction, while denoising towards a generalized state for each modality.

The denoiser contains a self-attention layer, applied to the two noisy embeddings $\mathbf{q}_t$ and $\mathbf{c}_t$, together with the market-state conditioning $\mathbf{m}_t$, and a sinusoidal embedding of the diffusion step s. The layer applies LayerNorm, followed by the query, key, value projections and scaled dot-product attention.

Denoising Objective. We train the model using a joint reconstruction objective over both modalities:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{t,s}\left[\|\mathbf{q}_t - \hat{\mathbf{q}}_t\|_2^2 + \|\mathbf{c}_t - \hat{\mathbf{c}}_t\|_2^2\right]. \tag{14}$$

This objective encourages the model to reconstruct modality-specific representations from their noisy counterparts, serving as refined features for subsequent market representation construction.

A derivation of the diffusion process formulation used in RADAR can be found in Appendix A.

Final Representation. Using the denoised modality-specific representations $\hat{\mathbf{q}}_t$ and $\hat{\mathbf{c}}_t$, we construct the final, refined market representation by aggregating across the refined cross-modal signals:

$$\hat{\mathbf{z}}_t = (1-\gamma)\,(\hat{\mathbf{q}}_t + \hat{\mathbf{c}}_t) + \gamma \mathbf{q}_t, \tag{15}$$

where γ is a hyperparameter that balances the contribution between the denoised cross-modal information and the original price representation. The price representation is used as an additional signal for portfolio optimization, with cross-modal information serving as the cleaned state representation.

3.5 Overall Training Objective

Finally, the model is trained by jointly optimizing the diffusion and SDF-based portfolio objective:

$$\mathcal{L} = \mathcal{L}_{\text{SDF}} + \rho \cdot \mathcal{L}_{\text{Diff}}, \tag{16}$$

where ρ balances the market representation learning and downstream portfolio optimization task.

4 Experiments

Dataset. We evaluate RADAR on the S&P 500 constituent stocks. We collect daily adjusted close prices from Yahoo Finance and financial news articles from FNSPID [10], and the final dataset includes 377 U.S. equities that appear in both the S&P 500 index and the news corpus. The duration spans from June 2011 to June 2020, covering approximately 2,270 trading days. To avoid look-ahead bias, we adopt a rolling-window protocol: each window uses 4 years of data for training (with a 90:10 train/validation split) and the subsequent 1 year for out-of-sample testing, rolling forward by 1 year.

Baselines. Following our motivation, we compare RADAR against two families of baselines.

Time-series forecasting models. We compare against StockNet [52], which incorporates both price and textual signals; HAN [20], which models hierarchical text representations; NGAT [41], which captures cross-asset relationships via graph attention; iTransformer [32], a recent transformer-based time-series model; and RATD [30], a retrieval-augmented diffusion model for time-series forecasting. RATD is the closest related work as it also incorporates context retrieval into the diffusion process; however, it uses retrieved sequences as conditional guidance over the diffusion process, rather than using them to construct context-dependent noise for the diffusion process, as was done in our work. These models were used to predict future stock returns. From these forecasts, we construct portfolios by ranking the predicted returns and equally weighting the top-50 stocks at each rebalancing date.

End-to-end portfolio optimization models. We also compare against the same baselines used in the original SDF work: BSV [5], a linear characteristic-based model; DKKM [7], a high-dimensional nonlinear model based on random features; MLP, a deep neural network using own-asset characteristics; Linear Attention (LinAttn), which introduces cross-asset interactions through attention; and SDF [23], a transformer-based model that incorporates cross-asset information sharing, which performed the best in the SDF work. These models directly learn portfolio weights from the input features.

We evaluate portfolio performance using returns-based, risk-based, and risk-adjusted portfolio metrics. The cumulative and annualized returns measure the overall profitability of the portfolio. However, profitability metrics alone may be insufficient, as high returns can usually be achieved by taking on excessive risk. To account for this, we include risk-based metrics such as maximum drawdown and volatility, which capture the downside risk and the variability of returns. Finally, we report risk-adjusted metrics including the Sharpe, Sortino, and Calmar ratios, which assess the efficiency of returns relative to different notions of risk, and serve as the main indicators of portfolio quality.

Implementation Details. RADAR uses GRU-based encoders with additive attention pooling for both price and text modalities, fused via a bilinear layer. The hidden dimension is set to 64. The diffusion module uses $T = 100$ timesteps with a linear noise schedule. We retrieve the top $K = 10$ nearest neighbors from the context bank, which is refreshed every 5 epochs. We use an input sequence length of 60 trading days and a rebalancing horizon of $H = 7$ days. The model is optimized with AdamW and cosine annealing over 50 epochs with early stopping (patience $= 10$). Gradients are clipped at 1.0. All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

5 Results

Table 1: Performance comparison. The best baselines are underlined, and the best results are bolded.

Model	Sharpe (↑)	Sortino (↑)	Calmar (↑)	CumRet (↑)	AnnRet (↑)	MaxDD (↓)	Vol (↓)
Benchmarks							
S&P 500	0.513	0.576	0.244	0.487	0.083	0.339	0.191
Equal Weight [6]	<u>0.795</u>	0.873	0.393	1.005	0.149	0.380	0.200
Forecasting							
HAN [20]	0.705	0.816	0.355	0.892	0.123	0.347	0.191
StockNet [52]	0.780	<u>0.899</u>	0.387	<u>1.218</u>	<u>0.156</u>	0.404	0.216
iTransformer [32]	0.758	0.877	0.393	0.997	0.134	0.342	0.190
NGAT [41]	0.710	0.817	0.334	0.893	0.123	0.369	0.189
RATD [30]	0.680	0.754	0.302	0.844	0.130	0.431	0.214
Portfolio Opt							
BSV [5]	0.361	0.481	0.164	0.252	0.046	0.281	0.160
DKKM [7]	0.367	0.489	0.177	0.229	0.042	0.238	0.139
LinAttn	0.407	0.558	0.326	0.195	0.036	<u>0.111</u>	0.100
MLP	0.502	0.710	<u>0.406</u>	0.203	0.038	**0.093**	**0.080**
SDF [23]	0.505	0.690	0.313	0.211	0.039	0.125	<u>0.083</u>
Our Model							
RADAR	**1.062**	**1.352**	**0.924**	**2.502**	**0.285**	0.308	0.271

Performance Comparison. Table 1 reports the forecasting results. We can observe the following:

- The Equal Weight portfolio obtained strong results across all metrics, and achieved the second-highest Sharpe ratio. Prior work [6] have shown that it is often difficult to outperform naive equal-weighted diversification in practice, as forecasted returns are often noisy, and portfolio optimization strategies amplifies any forecasting errors. Equal weighting avoids this issue by ignoring forecasts altogether and therefore remains highly robust on the out-of-sample test data.
- The forecasting models achieved close performance, with StockNet delivering the second-best results on some metrics. Here, the portfolio is constructed by buying the forecasted top-50 assets, which allows these models to perform better on the returns compared to the portfolio methods, but also realizing higher volatilities. However, we note that their results remain relatively consistent with each other and similar to equal-weighting regardless of model forecasting performance. By buying a larger number of assets, any forecasting errors for an asset get neutralized by the others, showing that forecasting ability might not matter much in a real-life multi-asset portfolio setting.
- The portfolio methods show results that are largely consistent with those reported in the original SDF work [23]. The Sharpe performance increases from BSV which is linear, across higher levels of model expressiveness in DKKM, LinAttn and MLP. The original SDF model introduces cross-asset information sharing, which outperforms the others. In that sense, our results are consistent with the reported takeaway that both model expressiveness and cross-asset structures matter for portfolio performance. We note that these models demonstrate lower drawdowns and volatility than the forecasting models due to not optimizing only for returns. However, their Sharpe ratios remain lower, which might be due to their lower expressiveness compared to the deep-learning models.
- RADAR attains the best performance on most metrics including the Sharpe, Sortino, and Calmar ratios, indicating the best overall risk-return trade-off among all methods. It also achieved lower drawdowns than the forecasting models. The non-ratios metric performances could be less crucial here, as the model seeks to optimize for the best trade-offs to get the best portfolio results.

We also provide statistical significance and robustness analyses in Appendix B.

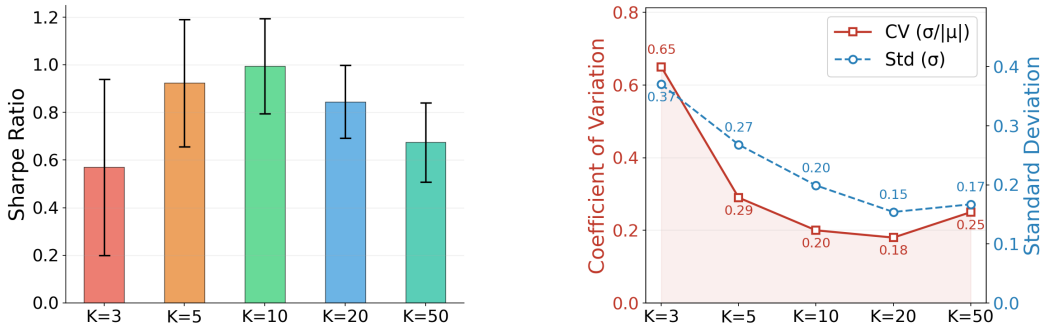
Table 2: Ablation study across different data and component variations of the RADAR framework.

Model	Sharpe (↑)	Sortino (↑)	Calmar (↑)	CumRet (↑)	AnnRet (↑)	MaxDD (↓)	Vol (↓)
Data							
w/o News	0.650	0.821	0.304	1.229	0.174	0.573	0.332
w/o Price	0.797	0.878	0.395	1.003	0.149	0.377	0.200
Components							
SDF (baseline)	0.505	0.690	0.313	0.211	0.039	**0.125**	**0.083**
+ *Embeddings*	0.859	0.959	0.429	1.138	0.164	0.383	0.201
+ *Diffusion*	0.411	0.466	0.108	0.501	0.085	0.784	0.525
Our Model							
RADAR	**1.062**	**1.352**	**0.924**	**2.502**	**0.285**	0.308	0.271

Ablation Study. We conduct an ablation study to demonstrate the effectiveness of the model design.

- From Table 2, removing either of the news or price inputs degrade performance, showing that both information sources are crucial for model performance. Among the two, the No Price variant shows relatively low volatility results, which suggests that much of the volatility is driven by the noisy price data. This is consistent with prior literature [6, 24] which show that price information often contain noise, leading to more volatile forecasts, which are further amplified in portfolio strategies.
- Moving from the SDF baseline to +*Embeddings* yields a clear improvement across most metrics. In the original SDF work [23], the portfolio learning component operates directly on the raw features, whereas we first learn a market representation before optimization. The improvement suggests that the representation learning helps to organize the market data into a cleaner latent space, separating the information from noise and making the downstream portfolio optimization more effective.
- Moving from +*Embeddings* to +*Diffusion* unexpectedly reduces performance. A plausible explanation is that the learnt embeddings already capture most of the useful information available in the data, so further corrupting them with unrelated Gaussian noise weakens the representation rather than improving it. This could be corroborated by the degraded performance in the max drawdown and volatility metrics. Here, diffusion noise appears to act more as distortion than regularization.
- RADAR achieves the best performance across most key metrics. Unlike +*Diffusion*, RADAR introduces a context bank that retrieves relevant historical states, hence the added noise is no longer arbitrary but grounded in meaningful past market conditions. This provides a set of plausible future states over which the portfolio can be optimized, resulting in a more robust risk-return trade-off.

Parameter Selection. For the context bank, we do a study over different values of top K, which determines the number of relevant historical segments used to determine the diffusion parameters.



(a) Sharpe Ratio (mean ± std)

(b) Cross-Seed Instability: CV and Std

Figure 4: Effect of different top-K values on performance and stability across random seeds.

From Figure 4, we see a clear trade-off between retrieval robustness and signal fidelity as K varies. When K is small, performance exhibits substantial cross-seed variability, indicating that the retrieval process is not stable. Intuitively, relying on only a few retrieved neighbors makes the estimated signal highly sensitive to small perturbations, which propagates into downstream portfolio outcomes.

On the other hand, at larger values of K, although variability remains low, the mean Sharpe performance begins to deteriorate. This indicates that incorporating too many retrieved samples also introduces less relevant information, effectively diluting the signal and reducing overall effectiveness.

We chose $K = 10$ as our setting, which shows strong performance while maintaining low variability.

Market Representation Study. Our work aims to learn market representations that improve portfolio outcomes. We study whether the denoised embeddings capture meaningful market structure.

In Figure 5, we study the asset correlation information (required for portfolio optimization) captured by the original noisy embeddings $\mathbf{m}_t$ and the cleaned RADAR embeddings $\hat{\mathbf{z}}_t$.

Here, all asset pairs are ranked by cosine similarity in the embedding space and grouped into deciles, from least (Decile 1) to most similar (Decile 10). For each decile, we compute the average realised return correlation between the corresponding asset pairs.

A meaningful representation should produce a monotonic relationship: higher embedding similarities should correspond to stronger correlations.

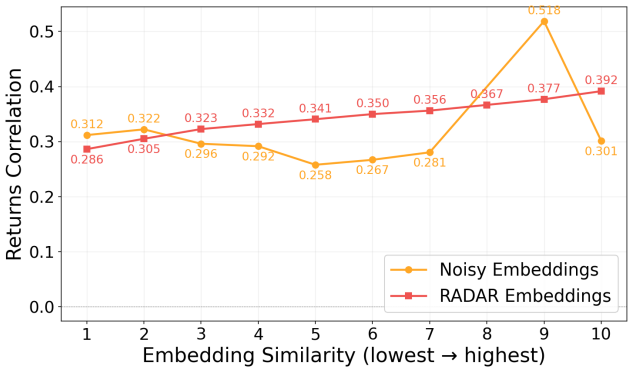


Figure 5: Embedding Similarity *vs* Returns Correlation. Reported values are aggregated over 5 rolling windows.

As shown, the RADAR embeddings yield a clearer and more monotonic increase in correlation across deciles compared to the noisy embeddings. This indicates that the context-aware denoising process helps to improve alignment between embedding similarity and true market co-movement.

In Figure 6, we evaluate whether the cleaned RADAR embeddings $\hat{\mathbf{z}}_t$ capture any useful information about the constituent asset returns performance, which are also required for portfolio optimization.

At each rebalancing date, the learnt asset embeddings are mapped to scalar scores via a MLP scoring network. Assets are then ranked by these scores and partitioned into quintiles, from highest (Q1) to lowest (Q5). Each quintile forms an equal-weighted portfolio, and we track cumulative NAV over time using forward returns.

A meaningful representation should induce a monotonic ordering: portfolios formed from the higher-scoring assets should outperform those formed from lower-scoring assets.

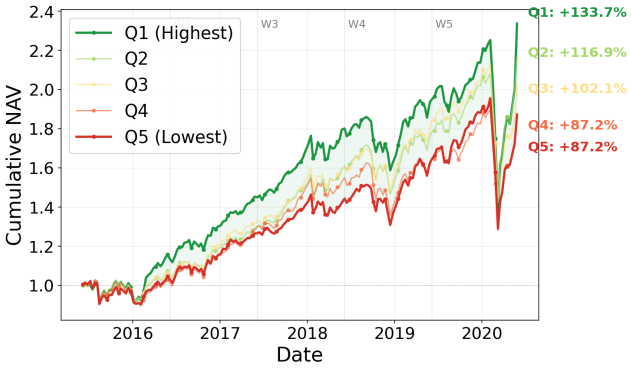


Figure 6: Quintile Portfolio NAV by Asset Scoring.

As shown, the NAV curves exhibit a clear separation, with Q1 consistently outperforming the lower quintiles. This indicates that the RADAR embeddings capture economically meaningful cross-sectional asset returns signals that translate into improved portfolio performance.

Robustness to Context and Target Lengths. We further evaluate the robustness of RADAR across different input and output lengths by varying the input length $L \in \{60, 120\}$ and rebalancing horizon $H \in \{1, 7, 20\}$. As shown in Figure 7, RADAR consistently achieves the highest Sharpe ratio and cumulative return across all six configurations. Performance generally decreases as the rebalancing horizon increases as longer horizons introduce greater uncertainty in future price movements, but RADAR remains consistently above the forecasting baselines. These results demonstrate that the performance gains of RADAR are not specific to any particular choice of input or output lengths.

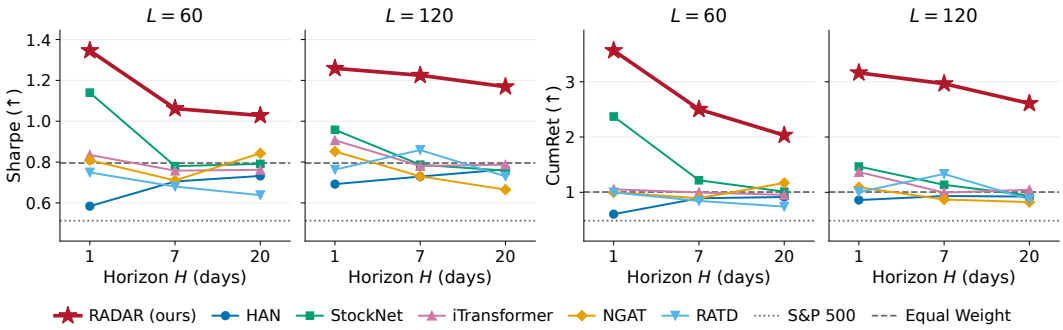


Figure 7: Performance across different input length L and rebalancing horizon H. Each pair shares its y-axis. S&P 500 and Equal Weight do not depend on L and H, and are drawn as reference lines.

Generalizability to Other Domains. For generalizability, we further evaluate RADAR on Time-MMD [29], a multi-modal (text + numerical) time-series benchmark spanning multiple domains.

Overall, we find that RADAR is largely generalizable across multiple domains. In some cases such as Security and Traffic, *No Diffusion* performs the best, which might be attributed to the lack of noise in the data. In other cases, injecting *Gaussian* noise does not appear to degrade performance, unlike the observations in our financial domain. This might indicate that the noise distribution of data in these domains is closer to Gaussian. These observations could be studied further in future work, based on the specific noise characteristics of each domain.

Table 3: Generalizability on different domains.

Domain	No Diff.	Gaussian	RADAR
Agriculture	2.328	2.375	**2.240**
Climate	0.484	**0.464**	**0.464**
Energy	1.753	0.540	**0.517**
Environment	0.931	0.946	**0.928**
Health (AFR)	2.413	4.203	**1.635**
Health (US)	1.007	0.956	**0.821**
Security	**1.526**	1.584	1.557
SocialGood	0.953	0.547	**0.512**
Traffic	**1.154**	1.350	1.290

Additional Results. More detailed performance analyses on the RADAR framework are provided in the appendix. In particular, Appendix C reports cross-sectional analyses across volatility groups and industry sectors, while Appendix D provides financial factor analyses and portfolio alpha studies.

6 Conclusion

In this work, we studied the portfolio task under the stochastic discount factor (SDF) framework. We explore learning clean market representations to better capture the underlying structure of financial data to perform portfolio optimization. This problem is challenging due to several factors: financial markets exhibit time-varying dynamics with shifting regimes, the multimodal inputs of price and news data often contain noise, and isotropic noise do not capture the state-dependent and heteroskedastic nature of financial uncertainty. To address these issues, we introduced RADAR, a retrieval-augmented diffusion framework that learns context-aware market representations by conditioning on similar historical regimes to denoise multimodal features. We conducted extensive experiments on forecasting and portfolio models, and showed that RADAR achieves strong performance on key risk-adjusted metrics, while producing representations that align with assets co-movement and returns structure.

Limitations. Our experiments rely on the FNSPID news dataset [10], which is relatively sparse in coverage. Since our retrieval mechanism depends on the availability of relevant historical contexts, limited news coverage would affect the quality of the learnt representations. However, we also note that the FNSPID dataset is widely used in literature [33, 29, 1], whereas more comprehensive but proprietary sources (*e.g.,* Bloomberg, Reuters) could reduce the overall replicability of the work.

Compared to classical SDF approaches [22, 23] that operate directly on raw characteristics, our use of embeddings to optimize portfolio weights may reduce interpretability, as the resulting representations are less directly tied to economically meaningful factor inputs. This limits the ability of the framework to capture feature importance. However, this limitation could be mitigated by applying post-hoc explanation techniques such as SHAP [35] or LIME [44], which we will leave for future work.

References

[1] Divyansh Agarwal, Alexander Richard Fabbri, Ben Risher, Philippe Laban, Shafiq Joty, and Chien-Sheng Wu. Prompt leakage effect and mitigation strategies for multi-turn llm applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1255–1275, 2024.

[2] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[3] Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural diffusion models. *arXiv preprint arXiv:2310.08337*, 2023.

[4] Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *Advances in Neural Information Processing Systems*, 37:73952–73985, 2024.

[5] Michael W Brandt, Pedro Santa-Clara, and Rossen Valkanov. Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. *The Review of Financial Studies*, 22(9):3411–3447, 2009.

[6] Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The review of Financial studies*, 22(5):1915–1953, 2009.

[7] Antoine Didisheim, Shikun Barry Ke, Bryan T Kelly, and Semyon Malamud. Apt or "aipt"? the surprising dominance of large factor models. Technical report, National Bureau of Economic Research, 2024.

[8] Francis X Diebold and Robert S Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002.

[9] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *IJCAI*, volume 15, pages 2327–2333, 2015.

[10] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. Fnspid: A comprehensive financial news dataset in time series. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4918–4927, 2024.

[11] Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

[12] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. Enhancing stock movement prediction with adversarial training. *arXiv preprint arXiv:1810.09936*, 2018.

[13] Ronald Aylmer Fisher. Statistical methods for research workers. 1934.

[14] Xavier Gabaix, Ralph SJ Koijen, Robert J Richmond, and Motohiro Yogo. Asset embeddings. Technical report, National Bureau of Economic Research, 2025.

[15] Yuan Gao, Haokun Chen, Xiang Wang, Zhicai Wang, Xue Wang, Jinyang Gao, and Bolin Ding. Diffsformer: A diffusion transformer on stock factor augmentation. *arXiv preprint arXiv:2402.06656*, 2024.

[16] James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the econometric society*, pages 357–384, 1989.

[17] Lars Peter Hansen and Scott F Richard. The role of conditioning information in deducing testable restrictions implied by dynamic asset pricing models. *Econometrica: Journal of the Econometric Society*, pages 587–613, 1987.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[19] Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 261–269, 2018.

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

[22] Bryan Kelly, Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu. Large (and deep) factor models. *arXiv preprint arXiv:2402.06635*, 2024.

[23] Bryan T Kelly, Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu. Artificial intelligence asset pricing models. Technical report, National Bureau of Economic Research, 2025.

[24] Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. Diffusion variational autoencoder for tackling stochasticity in multi-step regression stock price prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1087–1096, 2023.

[25] Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM web conference 2024*, pages 4304–4315, 2024.

[26] Oliver Ledoit and Michael Wolf. Robust performance hypothesis testing with the sharpe ratio. *Journal of Empirical Finance*, 15(5):850–859, 2008.

[27] Olivier Ledoit and Michael Wolf. Honey, i shrunk the sample covariance matrix. *UPF economics and business working paper*, (691), 2003.

[28] Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Mingyuan Zhou, et al. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *International Conference on Learning Representations*, volume 2024, pages 18604–18622, 2024.

[29] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933, 2024.

[30] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. *Advances in Neural Information Processing Systems*, 37:2766–2786, 2024.

[31] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35:9881–9893, 2022.

[32] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[33] Zhaowei Liu, Xin Guo, Zhi Yang, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Mengping Li, Qi Qi, Zhiqiang Liu, Yiyang Han, et al. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*, 2025.

[34] Marcos Lopez de Prado. Building diversified portfolios that outperform out-of-sample. *Journal of Portfolio Management*, 2016.

[35] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[36] Burton G. Malkiel. *A Random Walk Down Wall Street*. W. W. Norton & Company, 1973.

[37] Harry M. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952. doi: 10.2307/2975974.

[38] Harry M. Markowitz. The utility of wealth. *The Journal of Political Economy*, 60(2):151–158, April 1952. doi: 10.1086/257177.

[39] Dhagash Mehta, John RJ Thompson, Hoyoung Lee, and Yongjae Lee. Clustering and similarity learning in financial markets: A tutorial for the practitioners. *Available at SSRN 5587353*, 2025.

[40] Whitney K Newey and Kenneth D West. A simple, positive semi-definite, heteroskedasticity and autocorrelationconsistent covariance matrix. 1986.

[41] Yingjie Niu, Mingchuan Zhao, Valerio Poti, and Ruihai Dong. Ngat: A node-level graph attention network for long-term stock prediction. In *International Conference on Artificial Neural Networks*, pages 204–215. Springer, 2025.

[42] Dimitris N Politis and Joseph P Romano. The stationary bootstrap. *Journal of the American Statistical association*, 89(428):1303–1313, 1994.

[43] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

[44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[45] J Rishi, GVS Mothish, and Deepak Subramani. Conditional diffusion model with nonlinear data transformation for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

[46] Subham S Sahoo, Aaron Gokaslan, Chris De, and Volodymyr Kuleshov. Diffusion models with learned adaptive noise. *Advances in Neural Information Processing Systems*, 37:105730–105779, 2024.

[47] Bhaskarjit Sarmah, Nayana Nair, Riya Jain, Dhagash Mehta, and Stefano Pasquali. Learning embedded representation of the stock correlation matrix using graph machine learning. In *2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, pages 1–9. IEEE, 2024.

[48] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8415–8426, 2020.

[49] Paul C Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168, 2007.

[50] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1900–1908, 2019.

[51] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[52] Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, 2018.

[53] Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

A Diffusion Process Formulation

The standard diffusion process corrupts data with Gaussian noise of $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In Equations (11) and (12), we instead corrupt the representations with Gaussian noise parameterized by $(\boldsymbol{\mu}_{\mathcal{N}_{q,t}}, \boldsymbol{\sigma}_{\mathcal{N}_{q,t}})$, which are the the retrieved statistics. A shifted, rescaled Gaussian remains Gaussian, so the process is still a standard diffusion process expressed in new coordinates. We can derive the formulation:

Forward Process. Equation (11) gives the marginal distribution at an arbitrary diffusion step s:

$$\tilde{\mathbf{q}}_{t,s} = \sqrt{\bar{\alpha}_s}\,\mathbf{q}_t + \sqrt{1-\bar{\alpha}_s}\left(\boldsymbol{\mu}_{\mathcal{N}_{q,t}} + \boldsymbol{\sigma}_{\mathcal{N}_{q,t}} \odot \boldsymbol{\epsilon}\right).$$

For compactness, we define:

$$r_s = \sqrt{1-\bar{\alpha}_s}, \qquad \boldsymbol{\Sigma}_{q,t} = \operatorname{diag}\left(\boldsymbol{\sigma}^2_{\mathcal{N}_{q,t}}\right),$$

and let $\alpha_s = 1 - \beta_s$. A stepwise forward process consistent with Equation (11) is:

$$q\left(\tilde{\mathbf{q}}_{t,s} \mid \tilde{\mathbf{q}}_{t,s-1}, \mathcal{N}_{q,t}\right) = \mathcal{N}\left(\sqrt{\alpha_s}\,\tilde{\mathbf{q}}_{t,s-1} + \delta_s \boldsymbol{\mu}_{\mathcal{N}_{q,t}},\ \beta_s \boldsymbol{\Sigma}_{q,t}\right),$$

where:

$$\delta_s = r_s - \sqrt{\alpha_s}\, r_{s-1}.$$

If we define a mean-centered variable:

$$\mathbf{y}^q_{t,s} = \tilde{\mathbf{q}}_{t,s} - r_s \boldsymbol{\mu}_{\mathcal{N}_{q,t}},$$

then:

$$\mathbf{y}^q_{t,s} = \sqrt{\alpha_s}\,\mathbf{y}^q_{t,s-1} + \sqrt{\beta_s}\,\boldsymbol{\sigma}_{\mathcal{N}_{q,t}} \odot \boldsymbol{\epsilon}_s.$$

This is the standard diffusion formulation, after subtracting the context-dependent mean and scaling the noise by the context-dependent standard deviation.

Reverse Posterior. Given the clean representation $\mathbf{q}_t$, the exact posterior is:

$$q\left(\tilde{\mathbf{q}}_{t,s-1} \mid \tilde{\mathbf{q}}_{t,s}, \mathbf{q}_t, \mathcal{N}_{q,t}\right) = \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_{q,s},\ \tilde{\beta}_s \boldsymbol{\Sigma}_{q,t}\right),$$

where:

$$\tilde{\beta}_s = \frac{\beta_s(1-\bar{\alpha}_{s-1})}{1-\bar{\alpha}_s},$$

and:

$$\tilde{\boldsymbol{\mu}}_{q,s} = A_s \mathbf{q}_t + B_s\left(\tilde{\mathbf{q}}_{t,s} - r_s \boldsymbol{\mu}_{\mathcal{N}_{q,t}}\right) + r_{s-1}\boldsymbol{\mu}_{\mathcal{N}_{q,t}},$$

with:

$$A_s = \frac{\beta_s\sqrt{\bar{\alpha}_{s-1}}}{1-\bar{\alpha}_s}, \qquad B_s = \frac{\sqrt{\alpha_s}(1-\bar{\alpha}_{s-1})}{1-\bar{\alpha}_s}.$$

Link to Equation (13)**.** The true $\mathbf{q}_t$ is unavailable during reverse sampling. The denoising function predicts it as:

$$\hat{\mathbf{q}}_t = D_p\left(\tilde{\mathbf{q}}_{t,s}, s \mid \mathbf{m}_t\right).$$

We define the learned reverse process by replacing the unknown $\mathbf{q}_t$ in the exact posterior with $\hat{\mathbf{q}}_t$:

$$p_\theta\left(\tilde{\mathbf{q}}_{t,s-1} \mid \tilde{\mathbf{q}}_{t,s}, \mathbf{m}_t, \mathcal{N}_{q,t}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{\theta,q,s},\ \tilde{\beta}_s \boldsymbol{\Sigma}_{q,t}\right),$$

where:

$$\boldsymbol{\mu}_{\theta,q,s} = A_s D_p\left(\tilde{\mathbf{q}}_{t,s}, s \mid \mathbf{m}_t\right) + B_s\left(\tilde{\mathbf{q}}_{t,s} - r_s \boldsymbol{\mu}_{\mathcal{N}_{q,t}}\right) + r_{s-1}\boldsymbol{\mu}_{\mathcal{N}_{q,t}}.$$

Sampling one reverse step is then:

$$\tilde{\mathbf{q}}_{t,s-1} = \boldsymbol{\mu}_{\theta,q,s} + \sqrt{\tilde{\beta}_s}\,\boldsymbol{\sigma}_{\mathcal{N}_{q,t}} \odot \mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

At the final step, $s = 1$,

$$\tilde{\mathbf{q}}_{t,0} = D_p\left(\tilde{\mathbf{q}}_{t,1}, 1 \mid \mathbf{m}_t\right).$$

The same formulation applies to the text path.

Deriving Equation (14)**.** The denoising network is defined as a x_0-prediction model:

$$D_p\left(\tilde{\mathbf{q}}_{t,s}, s \mid \mathbf{m}_t\right) \approx \mathbf{q}_t.$$

Under the reverse formulation above, the exact variational term for $s > 1$ is proportional to:

$$\lambda_s \left\| \boldsymbol{\Sigma}_{q,t}^{-1/2} \left(\mathbf{q}_t - \hat{\mathbf{q}}_t\right) \right\|_2^2,$$

where:

$$\lambda_s = \frac{A_s^2}{2\tilde{\beta}_s}.$$

Equation (14) is the simplified x_0-prediction objective obtained by dropping the timestep-dependent and covariance-dependent weighting λ_s and $\boldsymbol{\Sigma}_{q,t}^{-1/2}$:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{t,s}\left[\|\mathbf{q}_t - \hat{\mathbf{q}}_t\|_2^2 + \|\mathbf{c}_t - \hat{\mathbf{c}}_t\|_2^2\right].$$

This follows established diffusion practices: the classic denoising diffusion probabilistic model [18] discards the timestep-dependent variational weighting in its objective, while other works [19] also adopt an unweighted squared-error objective for a generalized non-isotropic diffusion process.

B Statistical Significance and Robustness

We evaluate if the performance improvements of RADAR over baselines are statistically significant.

Difference series. For each baseline, we construct the out-of-sample returns difference series:

$$\text{Difference} = r_t^{\text{RADAR}} - r_t^{\text{baseline}},$$

and test the one-sided hypothesis that RADAR portfolio returns outperforms the baselines.

Pooled time-series tests. We first perform significance tests on the pooled return differences:

- **Monthly t-test:** Aggregates daily returns to the monthly level before testing. This reduces the microstructure noise and short-term dependence, providing a lower-frequency validation to show that the performance gains persist beyond daily fluctuations and transient market effects.
- **Newey–West HAC t-test** [40]: Adjusts for autocorrelation and heteroskedasticity in the return difference series. This is important because daily financial returns violate the i.i.d. assumption, and naive t-tests would otherwise systematically overstate statistical significance.
- **Diebold–Mariano (DM) test** [8]: Evaluates differences in predictive accuracy over time by comparing the loss (return differences) sequence directly. Unlike standard t-tests, it is specifically designed for time-series forecast comparisons and remains statistically valid under serial dependence.
- **Ledoit–Wolf (LW) test** [26]: Tests for differences in Sharpe ratios while accounting for estimation error in the mean and covariance of returns. This is particularly important because Sharpe ratios are non-linear functionals of returns and cannot be reliably compared using standard t-tests.

Robustness checks. To further validate the reliability of the results, we perform robustness checks:

- **Block bootstrap** [42]: Resamples the return series using block-based sampling to preserve temporal dependence, providing a non-parametric validation that does not rely on asymptotic assumptions.
- **Bootstrap Sharpe test:** Evaluates the significance of Sharpe ratio improvements under resampling. This complements parametric tests by directly assessing the variability of the performance metric.
- **Fisher aggregation** [13]: Combines p-values across seeds, verifying that significance holds consistently across different random initializations rather than being driven by a subset of runs.

Table 4: Pooled significance tests comparing RADAR against baselines. The reported results are taken from observations across five random seeds. Significance levels: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$.

Baseline	Monthly p	HAC p	DM p	LW p
Equal Weight	0.0016***	0.0015***	0.0015***	0.0079***
HAN	0.0015***	0.0016***	0.0016***	0.0105**
StockNet	0.0021***	0.0030***	0.0030***	0.0137**
iTransformer	0.0041***	0.0042***	0.0041***	0.0402**
NGAT	0.0011***	0.0013***	0.0012***	0.0081***
RATD	0.0003***	0.0005***	0.0005***	0.0018***
BSV	0.0026***	0.0050***	0.0050***	0.0086***
DKKM	0.0017***	0.0014***	0.0014***	0.0071***
MLP	0.0029***	0.0024***	0.0024***	0.0140**
LinAttn	0.0045***	0.0052***	0.0052***	0.0253**
SDF	0.0046***	0.0036***	0.0035***	0.0189**

Table 5: Robustness statistical significance. Significance levels: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$.

Baseline	Bootstrap p	Boot ΔSharpe p	Fisher HAC p	Fisher Monthly p
Equal Weight	0.0023***	0.0153**	0.0012***	0.0020***
HAN	0.0019***	0.0129**	0.0008***	0.0013***
StockNet	0.0045***	0.0236**	0.0041***	0.0064***
iTransformer	0.0044***	0.0314**	0.0024***	0.0036***
NGAT	0.0020***	0.0126**	0.0007***	0.0011***
RATD	0.0012***	0.0044***	0.0005***	0.0010***
BSV	0.0043***	0.0134**	0.0086***	0.0139**
DKKM	0.0012***	0.0132**	0.0010***	0.0016***
MLP	0.0019***	0.0203**	0.0007***	0.0011***
LinAttn	0.0038***	0.0307**	0.0048***	0.0072***
SDF	0.0032***	0.0437**	0.0013***	0.0021***

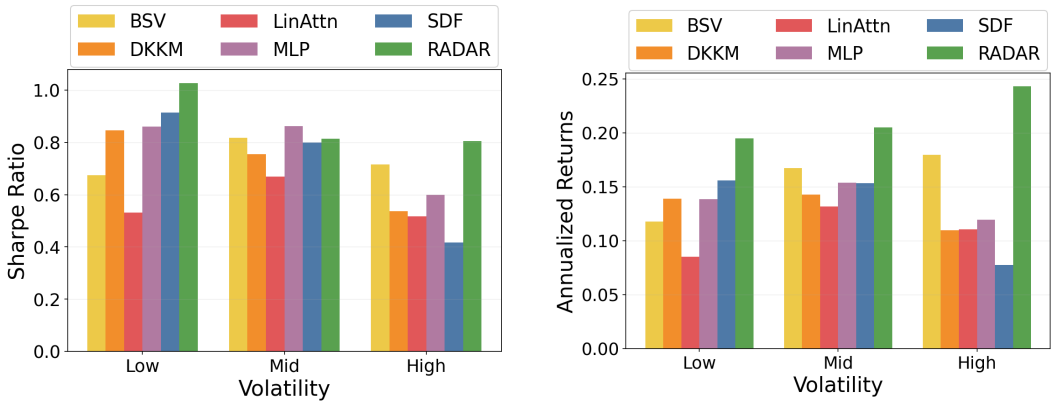
From Table 4, we observe that RADAR consistently achieves statistically significant improvements across all baselines and testing procedures. The significance holds under monthly aggregation, HAC-adjusted inference, and Sharpe-based evaluation, indicating that the results are not driven by any particular modeling assumption. From Table 5, the results remain robust under block bootstrap resampling and cross-seed aggregation, indicating that the findings are stable with respect to temporal dependence and random initialization. Furthermore, the bootstrap Sharpe and Ledoit–Wolf tests confirm that the improvements in Sharpe ratio (our primary evaluation metric) are also statistically significant, which reinforces the practical relevance of the observed performance gains.

C Cross-Sectional Performance Analysis

We conduct experiments to analyze cross-sectional performance across volatility regimes and market sectors, providing further insights into model behavior under heterogeneous market conditions.

Figure 8 reports the results across assets sorted by volatility groups. We can observe the following:

- RADAR achieves the highest annualized returns across all volatility groups, with the improvement gap widening as volatility increases. Typically, high-volatility environments are more challenging for the baseline models, as the the signal-to-noise ratio is lower, making it harder for them to extract useful information. The denoising process in RADAR helps to refine latent representations and mitigate the impact of noise, allowing the model to generalize effectively under noisy conditions.
- The Sharpe Ratio performance trend do not strictly follow those of the annualized returns, as market volatility would cause some degradation in Sharpe (Sharpe Ratio is a ratio of returns over portfolio volatility). In terms of risk-adjusted performance, RADAR attains the highest Sharpe ratios in both the low- and high-volatility groups, and remains competitive in the mid-volatility group. This indicates that the returns improvements are not driven solely by higher risk-taking, but rather by more efficient extraction of market representative signals within each volatility regime.



(a) Sharpe Ratio across volatility groups.

(b) Annualized Returns across volatility groups.

Figure 8: Performance comparison across different volatility groups. RADAR consistently achieves higher risk-adjusted performance while maintaining competitive returns across all volatility segments.

- These results are consistent with the framework design of RADAR. The retrieval-augmented component enables the model to condition on historically similar regimes, which becomes especially important in high-volatility periods where market dynamics deviate from average conditions. The diffusion-based denoising further refines the latent representations by suppressing idiosyncratic noise, allowing the model to generalize more effectively across similar contexts. Together, these mechanisms help improve stability and model performance, even in high-volatility environments.

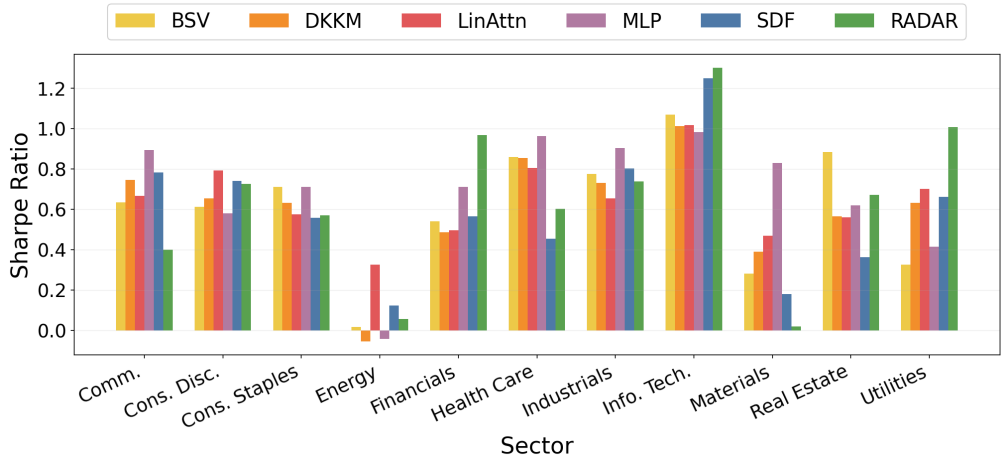


Figure 9: Sharpe ratio across sector groups.

Figures 9 and 10 present the results across assets sorted by sectors. We can observe the following:

- RADAR achieves the strongest performance in most sectors, with particularly large gains in Information Technology and Financials. These sectors are typically characterized by higher cross-sectional heterogeneity and stronger interdependencies across firms. The improvements suggest that RADAR is effective at capturing complex interactions and non-linear relationships within these sectors.
- In more stable sectors such as Utilities and Consumer Staples, RADAR remains competitive and achieves strong Sharpe performance. This suggests that the model does not rely solely on high-volatility opportunities, but is also able to extract signals within lower-variance environments.
- We note that the gains are more limited in sectors such as Energy and Real Estate, where returns are often driven by exogenous macro factors (*e.g.,* commodity prices or interest rates) which may not be fully captured by firm-level representations that was learnt in our work. This indicates that while RADAR improves modeling of cross-sectional structure, its benefits are less pronounced in sectors dominated by external shocks. We believe that the RADAR framework could be further improved with the addition of macro-economic data, which we seek to incorporate in future work.

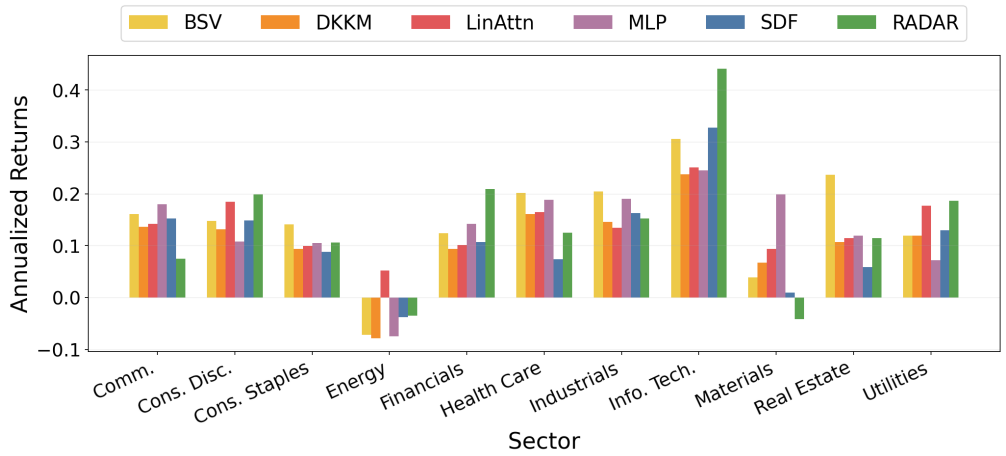


Figure 10: Annualized returns across sector groups.

Overall, the results suggest that RADAR improves portfolio performance by better modeling state-dependent and noisy market environments. The retrieval-augmented mechanism enables the model to adapt to non-stationarity by conditioning on historically similar regimes, while the diffusion-based denoising refines latent representations by reducing noise in the multimodal inputs. This appears especially beneficial in high-volatility assets and sectors with complex interactions, where baselines such as MLP and transformers (SDF) show degraded performance. The initialization of the diffusion process using empirical conditional statistics further grounds the learnt representations, allowing the model to more accurately capture the underlying risk–return trade-offs in different market states.

D Financial Analysis of Portfolios

To understand the sources of portfolio performance, we conduct standard financial analysis using the Capital Asset Pricing Model (CAPM) and the Fama–French factor model. These models decompose returns into components explained by systematic risk factors and residual alpha, allowing us to assess whether the performance arises from true signal extraction or exposure to common risk premia.

CAPM regression. We first estimate the CAPM model on daily returns:

$$R_{\text{RADAR},t} - R_{f,t} = \alpha + \beta\big(R_{\text{market},t} - R_{f,t}\big) + \epsilon_t,$$

where:

- $(R_{\text{RADAR},t} - R_{f,t})$: The excess return of the RADAR portfolio on day t.
- $(R_{\text{market},t} - R_{f,t})$: The excess return of the market benchmark on day t.
- **Alpha (α):** The intercept, representing the portion of returns not explained by market exposure.
- **Beta (β):** The slope, which measures the RADAR portfolio's systematic risk relative to market.

A strong strategy should exhibit a positive and statistically significant α, indicating that it generates excess returns beyond what can be explained by market exposure. An α close to zero would suggest performance comparable to the CAPM benchmark. In addition, we report the R^2, which measures the proportion of return variation explained by the model. A lower R^2 implies that the strategy is not simply replicating common risk exposures, and is instead capturing useful idiosyncratic signals.

The CAPM results reveal several key findings. Firstly, RADAR achieves the highest annualized alpha (18.71%), which is statistically significant at the 5% level. This indicates that its performance cannot be explained solely by market exposure. In contrast, baseline models exhibit substantially lower alpha, even when statistically significant.

Secondly, the estimated beta for RADAR is close to one (0.977), suggesting moderate market exposure rather than a market-neutral strategy. Importantly, the relatively low R^2 (0.476) indicates that a large portion of return variation is not explained by market movements, pointing to meaningful

Table 6: CAPM regression results. Significance levels: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$.

Model	α	t-stat	p-value	β	R^2
Equal Weight	4.78%***	2.712	0.003	1.030	0.957
BSV	6.60%*	1.614	0.053	1.248	0.849
DKKM	4.10%***	2.403	0.008	1.019	0.960
MLP	4.86%**	1.864	0.031	0.893	0.897
LinAttn	7.13%**	2.208	0.014	1.051	0.859
SDF	5.34%*	1.352	0.088	0.983	0.816
RADAR (Ours)	**18.71%****	2.159	0.016	0.977	0.476

idiosyncratic signal extraction. By comparison, traditional models such as Equal Weight exhibit much higher R^2 values (above 0.95), implying that their returns are largely driven by market factors.

Fama–French factor analysis. We further analyze performance using a three-factor model:

$$R_{p,t} - R_{f,t} = \alpha + \beta_m(R_m - R_f) + \beta_s \text{SMB} + \beta_h \text{HML} + \epsilon_t,$$

where:

- SMB: Size factor, defined as the return difference between small-cap and large-cap stocks.
- HML: Value factor, defined as the return difference between high and low book-to-market stocks.
- $\beta_m, \beta_s, \beta_h$: Factor loadings measuring exposure to the market, size, and value factors, respectively.

Similar to the CAPM setting, we look for a positive and statistically significant value α, which indicates excess returns beyond common risk factors. The R^2 now reports the proportion of return variation explained jointly by all factors. A lower R^2 suggests that the strategy captures useful idiosyncratic signals. The β coefficients capture the model's exposure to the respective factors.

Table 7: Fama–French regression results. Significance levels: $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$.

Model	α	t-stat	p-value	β_m	β_{SMB}	β_{HML}	R^2
Equal Weight	4.63%***	3.233	0.001	0.999	0.079	0.159	0.975
BSV	8.73%***	2.833	0.002	1.192	0.402	0.327	0.918
DKKM	3.88%***	2.735	0.003	0.988	0.042	0.156	0.974
MLP	2.30%	0.870	0.192	0.892	-0.079	-0.064	0.896
LinAttn	8.05%***	2.448	0.007	1.007	0.134	0.245	0.889
SDF	7.42%**	2.002	0.023	0.927	0.108	0.347	0.863
RADAR (Ours)	**15.47%****	1.814	0.035	0.988	0.064	-0.173	0.488

Under the multi-factor specification, RADAR continues to exhibit strong and statistically significant alpha (15.47%), confirming that its performance cannot be attributed to the other common risk factors such as size or value. Notably, RADAR shows relatively small exposure to SMB and a negative loading on HML, suggesting that its returns are not driven by standard factor tilts such as small-cap or value strategies. It also achieves the lowest average R^2 among all baseline models, indicating that a much smaller fraction of its overall returns variation can be explained by the compared factors.

Overall, these results demonstrate that RADAR captures economically meaningful alpha beyond market explanations. The combination of high alpha, moderate beta, and low R^2 supports the conclusion that the model extracts genuine predictive signals rather than relying on factor exposures.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and scope are clearly stated in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed under *Limitations* in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The reproducibility information is under *Implementation Details* in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides open access to the code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting/details can be found in Section 4 *Experiments*.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experiment statistical significance can be found in Appendix Section B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: A single GPU was used for the experiments, as was discussed in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is used in financial applications and has no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The dataset is cited within *Dataset* in Section 4. For the FNSPID dataset, the original license (CC BY-NC 4.0) appears to be superseded on their GitHub page.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.