Action-Conditioned JEPA Representations for RL-driven Market Decision-Making in Low-Dimensional Trading Environments

Gurmanjot Singh
UserID: g395sing
Student Number: 21217394
Due: 5 Dec 2025
g395sing@uwaterloo.ca

Abstract

Self-supervised predictive learning has recently emerged as a promising alternative to contrastive training for learning useful representations from high-dimensional temporal data. In this work, we investigate whether a Joint-Embedding Predictive Architecture (JEPA) can extract informative latent structure from noisy financial time series and support downstream reinforcement learning in a trading environment. We construct a lightweight multi-asset environment derived from real market data, incorporating TARDIS-style randomized initialization and drawdown-aware termination conditions to produce a diverse but tractable learning setting. A transformer-based JEPA encoder is trained to predict the latent embedding of the next market window, and its representations are used by an actor-critic policy for trading. We evaluate whether JEPA improves stability, generalization, and reward quality compared to policies learned directly from raw price features. Our experiments provide preliminary evidence that predictive self-supervision can extract temporal structure even in low-dimensional financial series, though translating these representations into stable trading policies remains a significant challenge.

1 Introduction

1.1 Goal and Motivation

Financial markets exhibit complex temporal dependencies, non-stationarity, and high levels of noise. Most reinforcement learning (RL) approaches for trading rely on raw price features or handcrafted indicators, but these signals often fail to capture latent structure that governs price movements. Recent advances in self-supervised predictive learning—particularly the Joint-Embedding Predictive Architecture (JEPA)—suggest that models can learn robust, task-agnostic representations by predicting future latent states rather than reconstructing input data.

The goal of this work is to examine whether JEPA can serve as an effective representation learner for financial time series in an RL trading pipeline. Specifically, we investigate whether JEPA's next-state latent prediction objective can provide richer temporal abstractions than raw features, enabling policies to stabilize faster and generalize better across market regimes. We design a simple but expressive environment based on historical equity data and test whether JEPA-trained embeddings improve trading performance compared to learning directly from market observations.

1.2 Related Work

Agent-based and simulation-driven studies of financial markets have explored how simple interacting behaviors can produce realistic price dynamics, liquidity cycles, and volatility clustering (Wheeler and Varner, 2023; Byrd et al., 2019). While reinforcement learning has been applied to algorithmic trading, it often struggles with instability, overfitting, and sensitivity to noise.

Self-supervised learning has recently shifted from contrastive objectives toward predictive architectures. I-JEPA (Assran et al., 2023) demonstrated strong performance on image understanding by learning representations that predict missing high-level embeddings. Extensions to sequential data and control tasks (Kenneweg et al., 2025; Assran et al., 2025) suggest that future-state latent prediction can provide structured abstractions useful for planning and policy learning.

Despite rapid advances, JEPA has not been systematically evaluated in the context of financial markets, where the signal-to-noise ratio is low and predictive structure is subtle. This work provides an initial exploration of how JEPA behaves when applied to market-based RL tasks.

1.3 Significance of the Approach

Financial time series present an unusually challenging learning environment due to their high noise levels, non-stationarity, and weak short-term predictability. Traditional reinforcement learning methods struggle in such settings because they operate directly on raw observations, often overfitting to spurious short-lived trends.

The JEPA framework offers a compelling alternative: instead of reconstructing inputs or predicting precise future prices, the model predicts *latent* future representations. This shifts the learning signal away from pixel- or feature-level noise and toward higher-level temporal structure. If effective, this could provide a pathway toward more stable, data-efficient, and generalizable financial RL systems. Demonstrating that JEPA can extract useful temporal abstractions even in low-dimensional noisy environments would extend its applicability beyond vision and robotics to economic and agent-based domains.

2 Method

2.1 Data Splitting and Experimental Protocol

We use a multi-asset dataset consisting of daily market features for several U.S. equities and index funds. To avoid temporal leakage and preserve the causal structure of financial data, we apply a chronological split. The training period contains several heterogeneous regimes, while the validation period consists of the most recent segment of the dataset. This allows us to evaluate generalization to market conditions that differ from those seen during training, including sharp trend reversals and volatility shifts.

During training, environment start indices are sampled randomly using a TARDIS-style scheme. Formally, if T is the window length and N is the total number of timesteps, the initial step is drawn as

$$i_0 \sim \mathcal{U}(T,\ N-5),$$

ensuring that each episode begins with a full observation window and avoids trivial short sequences near the dataset boundary. In contrast, validation episodes start deterministically at the beginning of the held-out split for reproducibility.

2.2 Multi-Asset Trading Environment

We implement a lightweight environment, `MultiAssetEnv`, which produces a state observation consisting of the most recent $T = 30$ timesteps of market features. Each observation has shape $(30, F)$, where F is the feature dimension after preprocessing and normalization.

The agent interacts with a simplified trading mechanism with discrete actions:

$$a_t \in \{0\text{:hold},\ 1\text{:buy},\ 2\text{:sell}\}.$$

Only a single asset (SPY) is traded, although the observation includes features from multiple correlated assets.

The environment maintains a cash balance B_t, a share count S_t, and a net worth

$$W_t = B_t + S_t \cdot P_t,$$

where P_t is the current SPY price. The reward is defined as the increment in net worth:

$$r_t = W_t - W_{t-1}.$$

Drawdown-Aware Termination. To discourage catastrophic risk-taking, we terminate an episode early if the net worth falls below a fixed drawdown threshold relative to the initial balance:

$$W_t < \alpha \cdot W_0,$$

where $\alpha = 0.7$ in the main experiments. A penalty of $-W_0$ is issued upon termination. While this threshold is intentionally conservative, it prevents the learning process from repeatedly exploring unrealistically poor trajectories. In future experiments, we could evaluate the effects of higher thresholds (e.g., $\alpha = 0.6$ or $\alpha = 0.5$), which reduce strictness and allow longer exploratory episodes, or have other stopping conditions.

2.3 JEPA Representation Learning

2.3.1 Online and Target Encoders

We adopt a transformer-based Joint-Embedding Predictive Architecture (JEPA) similar to I-JEPA but adapted for temporal financial data. Given an input window $x_t \in \mathbb{R}^{30\times F}$, the online encoder produces a latent embedding:

$$z_t = f_\theta(x_t),$$

where f_θ is a single-layer transformer encoder with dimensionality $d = 64$.

A momentum target encoder

$$z_t^{\text{target}} = f_\xi(x_t),$$

is maintained via exponential moving average (EMA):

$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

with τ close to 1 (e.g., 0.995), allowing the target network to evolve smoothly and stabilize training.

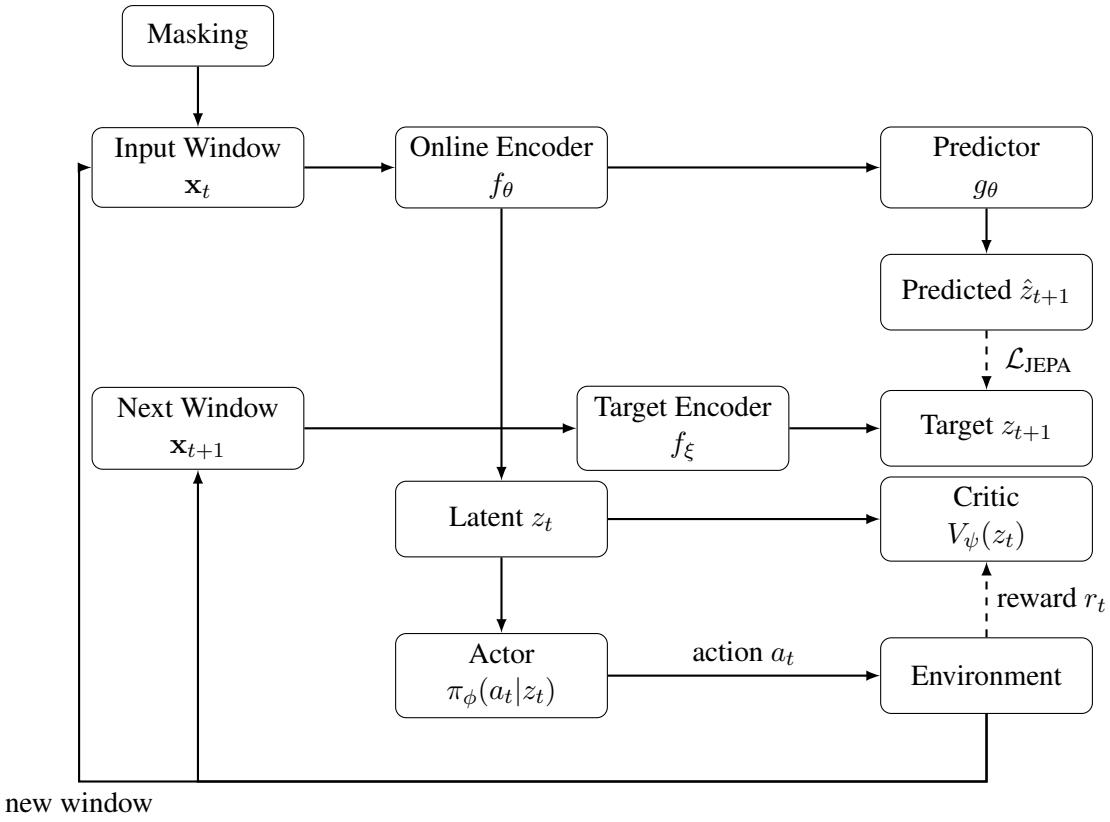

Figure 1: Overview of the JEPA–RL architecture. The JEPA module learns a predictive latent representation by matching the online encoder's predicted future embedding to the target encoder's embedding of the next window. The RL agent receives only the latent z_t and acts within the trading environment.

2.3.2 Predictor Network

A small MLP predictor g_θ maps the online latent z_t into a prediction of the next-state latent:

$$\hat{z}_{t+1} = g_\theta(z_t).$$

The JEPA loss encourages the online model to approximate the target representation of the next window:

$$\mathcal{L}_{\text{JEPA}} = \|\hat{z}_{t+1} - z_{t+1}^{\text{target}}\|_2^2.$$

2.3.3 Temporal Masking

To prevent shortcut solutions and encourage global temporal reasoning, we apply probabilistic masking to the online encoder's input. For each timestep index i in the window, we drop the entire feature vector with probability p:

$$x_t(i,:) = 0 \quad \text{if } m_i = 1, \quad m_i \sim \text{Bernoulli}(p).$$

This is analogous to patch masking in visual JEPA models but adapted for financial sequences.

2.4 Actor–Critic Reinforcement Learning Using JEPA Latents

Rather than operating directly on raw market windows, the policy receives the JEPA latent representation:

$$s_t = z_t.$$

We employ a lightweight actor–critic structure that only needs a single linear layer in each network. The actor outputs a categorical distribution over the three discrete actions:

$$\pi_\phi(a_t \mid s_t),$$

while the critic estimates the state value:

$$V_\psi(s_t).$$

The policy gradient loss is computed using variance normalized advantage estimates, and the total loss for one episode is

$$\mathcal{L} = \lambda_{\text{policy}}\mathcal{L}_{\text{policy}} + \mathcal{L}_{\text{critic}} + \beta\mathcal{L}_{\text{JEPA}}.$$

Here, β controls the influence of the JEPA objective, enabling representation learning and policy learning to co-evolve. We set $\lambda_{\text{policy}} = 50$ to ensure the weak policy signal isn't drowned out.

2.5 Training Procedure

Training alternates between:

1. sampling an episode using TARDIS init.
2. computing JEPA latents at each timestep
3. collecting rewards and log-probabilities
4. computing the actor–critic updates
5. updating both the JEPA online encoder and the policy network.

The target encoder is updated only through EMA.

This joint objective encourages the system to learn predictive temporal structure while simultaneously learning actions that exploit those representations for trading.

2.6 Risks and Challenges

Applying JEPA to financial markets in this way presents several inherent risks. First, market regimes can shift abruptly, and representations learned in one period may fail to generalize to future conditions. Second, JEPA relies on the assumption that future latent states are predictable from current ones, an assumption that may not always hold in markets influenced by exogenous events. Third, the joint optimization of representation learning and RL can introduce instability: if the JEPA latent space drifts too quickly, the policy may struggle to adapt.

From an evaluation perspective, positive results may overstate practical utility if the learned policy exploits artifacts of the historical dataset rather than persistent economic structure. Finally, while our environment includes mechanisms such as drawdown termination to limit unrealistic risk-taking, these constraints may also bias training dynamics. Taken together, these considerations highlight both the potential and the fragility of JEPA-based approaches in financial domains.

3 Results

We evaluate whether JEPA-based representations (i) capture predictive structure in financial time series and (ii) support reinforcement learning in a noisy trading environment.

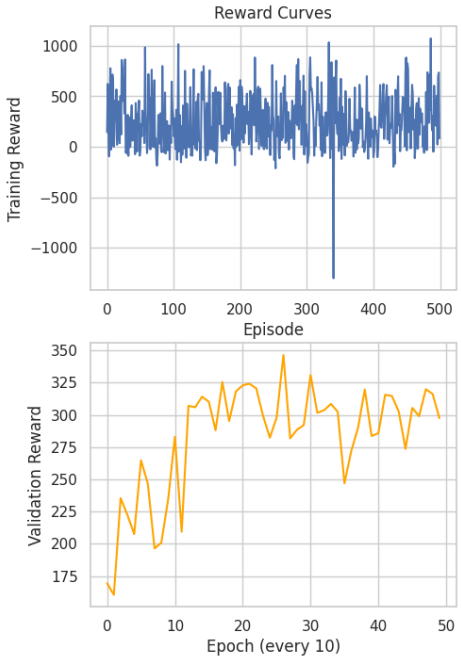

Figure 2: Training and validation net worth change (10-episode rolling mean) for the JEPA+RL agent.

3.1 Training and Validation Performance

Figure 2 shows the smoothed training and validation rewards of the JEPA+RL agent over epochs. Although returns are noisy, the validation performance remains consistently above zero for much of training, suggesting that the policy extracts some exploitable structure from the latent space rather than collapsing to purely random behavior.

To provide a more risk-sensitive view, we compute Sharpe-like statistics and maximum drawdown over validation rollouts. Let R_e denote the total return per episode. We report the mean and standard deviation of $\{R_e\}$, as well as

$$\text{Sharpe} = \frac{\mathbb{E}[R_e]}{\sqrt{\text{Var}[R_e] + 10^{-8}}},$$

and the maximum drawdown computed from the net worth trajectory within each episode. Table 1 summarizes these quantities.

Model	Mean R_e	Sharpe	MaxDD
JEPA+RL	251	0.96	-1325
Random (val)	200	2.16	0

Table 1: Validation episode statistics. Random policy is a sanity-check baselines.

On average, the JEPA+RL policy achieves higher raw returns than the random baseline, but

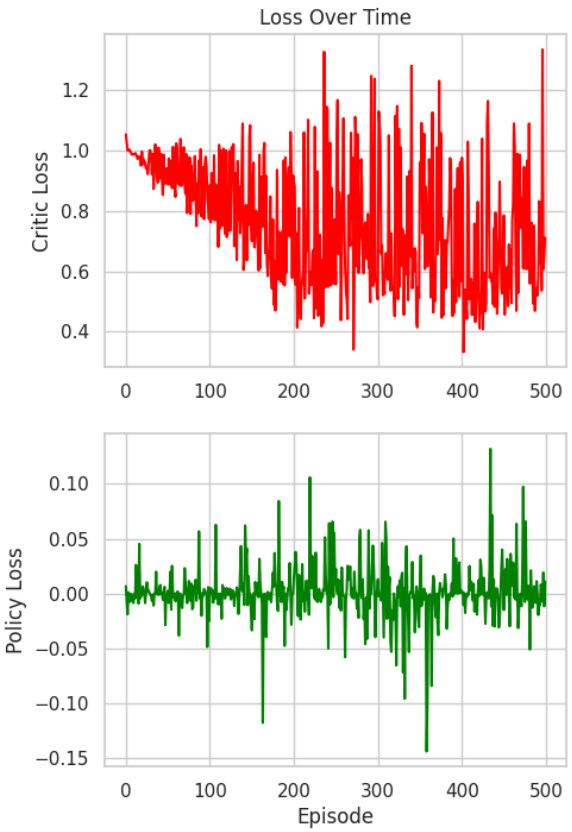

Figure 3: Policy and critic loss over training.

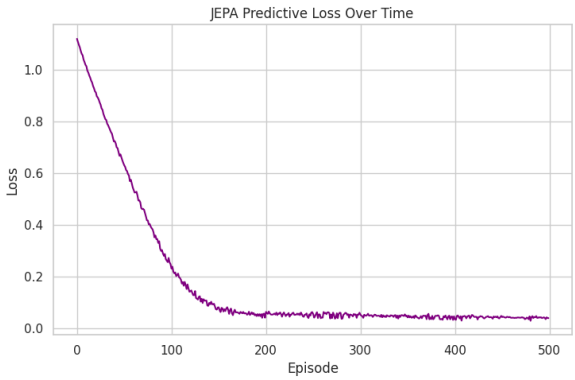

Figure 4: JEPA latent prediction loss over training.

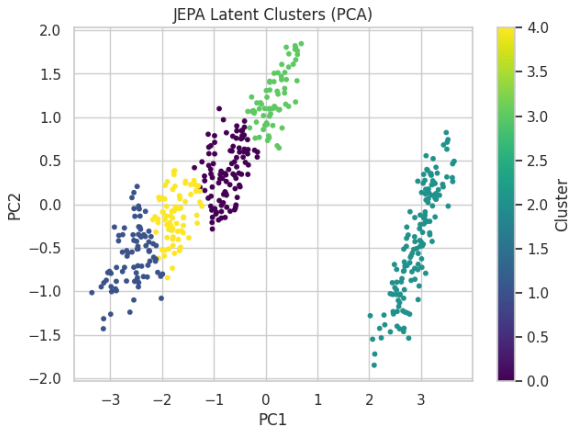

Figure 5: 2D projection of JEPA latents for validation windows, colored by $k = 5$-means cluster or next-day return sign.

does so by taking substantially larger risks, which manifest as frequent large drawdowns and a Sharpe ratio below 1. In contrast, the random policy rarely accumulates enough position to trigger drawdown penalties, causing its episode returns to remain nearly flat. This artificially reduces its return variance and inflates its Sharpe statistic. Thus, while the JEPA agent extracts exploitable structure, it does so in an unstable manner that sacrifices risk-adjusted return. As shown later, this suggests an unstable actor-critic setup for our policy.

3.2 Convergence of Actor-Critic Policy

We observe the loss contribution from the policy and critic in Figure 3. The critic seems to drop consistently until 0.8, by when it starts becoming more erratic and jumping between extremely high and low losses, which suggests our agent's learning was not very stable.

We also see that the policy's advantage loss estimate oscillates around 0 very stochastically, with no indication that the spikes are dropping, confirming that the reinforcement learning does not provide a very strong signal, and so does not converge to a stable solution.

This could be due to having an actor-critic that is not expressive enough or that our dataset is not sufficiently large and is too noisy. This also explains the risk taking behaviour that leads to poor Sharpe and drawdown as we saw before.

3.3 JEPA Latent Prediction Quality

Figure 4 plots the JEPA latent prediction loss over training epochs. The mean squared error between predicted and target next-state embeddings decreases steadily to a very low stable value, indicating that the model learns a representation of market windows that is at least locally predictive under the chosen window length and masking scheme.

3.4 Structure in the Latent Space

To probe what representations JEPA learns, we embed validation windows into the latent space and apply PCA-2 dimensionality reduction. We then cluster the latent vectors using $k = 5$-means and color them using the resulting cluster assignments.
.

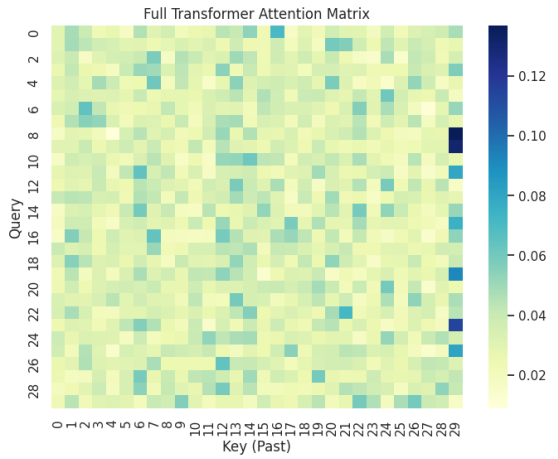

Figure 6: Full multi-head attention matrix for the first validation window. Concentration on the last key suggests that the model relies mostly on interactions between the most recent observation and a few select previous days.

Figure 5 shows an example projection. Points in the validation set organize into at least two distinct regions: a compact cluster and a more extended, curved manifold. K-means clustering over this space assigns multiple cluster labels along the extended manifold, which appears as a "rainbow" streak in the 2D projection. This suggests that the encoder is capturing a smooth, low-dimensional progression of market states (e.g., varying volatility or trend strength) rather than collapsing all windows into a single undifferentiated region. Hence, JEPA is capturing coarse market regimes, even though it is trained only with a self-supervised predictive objective.

3.5 Temporal Patterns in Encoder Attention

Because the latent space exhibited some meaningful structure, we next investigate how the encoder distributes attention within each window to produce these embeddings. We particularly look at how the transformer encoder distributes self-attention over the 30-step input window in Figure 6. The attention appears to be mostly noisy and unstructured, with some concentration on the last key.

From Figure 7, the model tends to allocate more attention to time steps which are associated with spikes in normalized closing prices, although the difference from how it treats other time steps is marginal.

To get a better idea of how the attention changes, for a sample of validation windows, we record the attention weights from the final self-attention layer

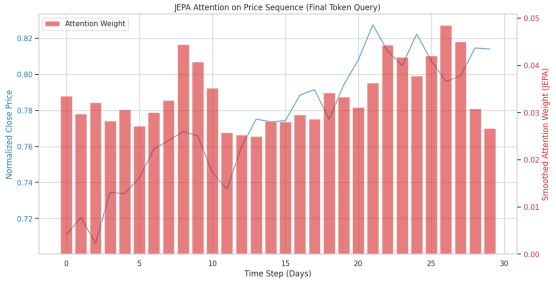

Figure 7: Final-token attention weights overlaid on the normalized close price for the first validation window. The encoder emphasizes days that are closer to spikes in the closing price, although it is overall inflated.

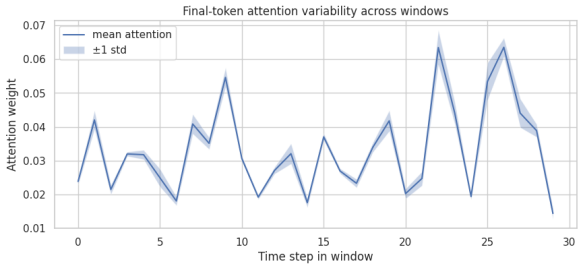

Figure 8: Mean attention weight by position in the 30-step window with a ± 1 std. interval.

and average over heads and examples. Figure 8 shows the mean attention per position and their variances. Its clear that the attention distribution stays nearly the same over the sampled windows, suggesting that the encoder relies on a relatively fixed temporal weighting pattern rather than dynamically adapting attention to the specific market conditions of each window.

4 Conclusion

This work explored whether a Joint-Embedding Predictive Architecture can supply stable and informative state representations for reinforcement learning in a noisy trading environment. Our results show that while the JEPA encoder successfully learns structured, predictive latent representations—evidenced by its low predictive loss, coherent latent-space organization, and repeatable attention patterns—the downstream actor–critic policy remains volatile and risk-seeking. This suggests that representation learning is not the primary bottleneck: rather, stable policy optimization in financial domains remains challenging even when high-quality latent states are available.

4.1 Limitations

Our study has several important limitations. First, the original goal of using a fully agent-based market simulator (ABM) had to be abandoned due to deprecation and instability issues in existing frameworks such as ABIDES, and the difficulty of implementing a custom simulator under time constraints.

As a result, we rely on a simplified environment built directly from historical price and indicator features, which lacks many aspects of real-world market microstructure.

Second, the model capacity and training budget are intentionally modest: we use a single-layer transformer encoder with a relatively low-dimensional latent space, trained on a limited time horizon. The positive results should therefore be interpreted as a proof of concept rather than a competitive trading system, and may not generalize to more complex regimes or longer-horizon forecasting tasks without some extra work.

Third, we focus on a single JEPA variant and a single reward definition (net worth change with an early drawdown termination rule). Alternative reward formulations, such as explicit risk-adjusted objectives or transaction-cost-aware returns, might substantially change the learned behavior. Finally, we evaluate on one asset and a fixed train/validation split; additional assets and rolling-window evaluations would be necessary to understand robustness to regime shifts.

4.2 Extensions and Future Work

There are several promising directions for extending this work. On the environment side, reconnecting with agent-based modeling is a natural next step: scalable ABM frameworks for financial markets (Wheeler and Varner, 2023; Byrd et al., 2019) could be combined with JEPA-style representation learning to study how predictive latents behave in multi-agent settings with explicit order flow and heterogeneous trader types. Large language model-based market simulators (Gao et al., 2024) also suggest an avenue for exploring JEPA in richer, narrative-driven financial environments.

On the modeling side, increasing the depth and width of the encoder, incorporating cross-asset attention, or stacking hierarchical JEPA modules over multiple time scales may yield more structured representations of market regimes.

Another direction is to use JEPA as a backbone for generating synthetic financial series: once a predictive latent is learned, it could be coupled with generative models to produce realistic but controllable market scenarios for stress testing and training.

Finally, JEPA could be integrated into multi-agent RL settings where each agent receives its own local view encoded by a shared or agent-specific JEPA module.

This would extend the current single-agent setup toward the multi-agent financial scenarios that we originally planned on working with, and could help clarify how predictive representation learning interacts with strategic behavior in complex economic systems.

5 Acknowledgments

The implementation for this project, including training and analysis, is available in the following Colab notebook:
https://colab.research.google.com/drive/1vA-epRqUB3AfY-aS1afoJfCuIB2Gm0NT?usp=sharing

Note that this code contains many experimental artifacts from rapid prototyping.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, and Yann LeCun. 2025. V-JEPA 2: Self-supervised video models enable understanding, prediction, and planning. *arXiv preprint*, arXiv:2506.09985.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture (I-JEPA). *arXiv preprint*, arXiv:2301.08243.

Daniel Byrd, Maria Hybinette, Tucker Balch, et al. 2019. Abides: Towards high-fidelity market simulation for AI research. *arXiv preprint*, arXiv:1904.12066.

Shuang Gao, Yuxin Wen, Mengdi Zhu, Jinglong Wei, Yuxuan Cheng, Qian Zhang, and Shu Shang. 2024. Simulating financial markets via large language model based agents (ASFM). *arXiv preprint*, arXiv:2406.19966.

Tristan Kenneweg, Sander Martin, Julien Gorbli, Mahmoud Assran, and Yann LeCun. 2025. JEPA for reinforcement learning. *arXiv preprint*, arXiv:2504.16591.

Andrew Wheeler and Jeffrey D. Varner. 2023. Scalable agent-based modeling for complex financial market simulations. *arXiv preprint*, arXiv:2312.14903.