Paper Replication Presentation

Time-driven feature-aware Jointly Deep Reinforcement Learning (TFJ-DRL) for Financial Signal Representation and Algorithmic Trading

Lingfeng Li, March 7

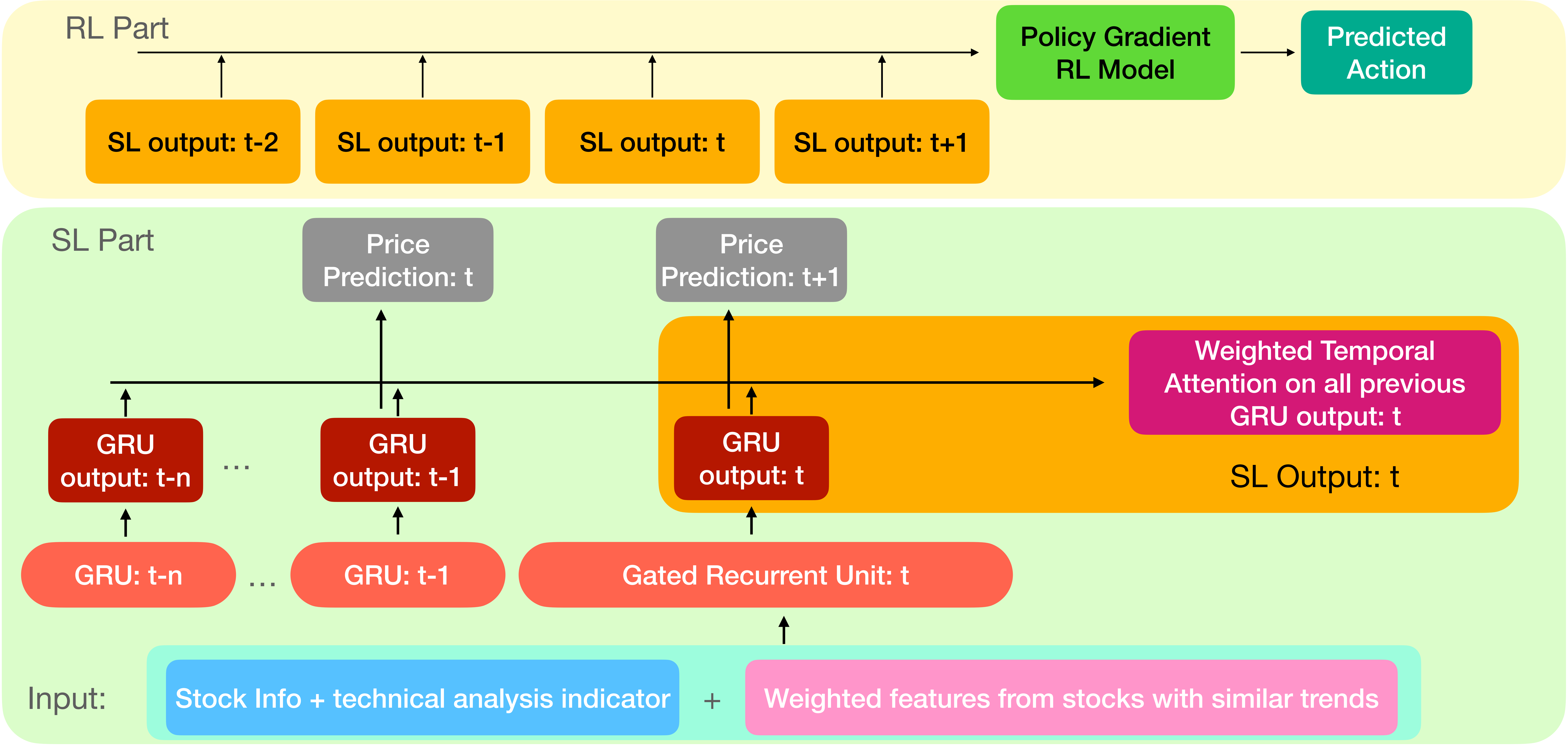
Problem & Flaws of Previous Attempts

- Design and train a ML algorithm that can:
  - Accurately and continuously perceive financial environment
  - Make profitable decisions in an online manner
- Flaws of Previous DL methods:
  - Costly to train
  - Mapping from price prediction to action cause second error propagation
- Flaws of Previous RL methods:
  - Cannot effectively utilize available environmental features

TFJ-DRL Model and Key Contribution

- TFJ-DRL model consists of two parts:
  - Supervised Learning (SL) for summarizing environmental features
  - Reinforcement Learning (RL) for making trading actions
- Advantage of the model:
  - SL model encodes environment instead of making price prediction: prevent second error propagation
  - SL model summarizes features for RL model: better understanding towards environment
  - Combination of SL and RL speeds up training

High Level Overview of TFJ-DRL



Data & Data Acquisition

- Source: Yahoo Finance
- Time Frame: Jan 2013 - Dec 2018
- Scale: Daily [Open, Close, High, Low, Volume]

Date	Open	High	Low	Close	Volume
2017-01-03	11.42	11.65	11.02	11.43	55182000
2017-01-04	11.45	11.52	11.24	11.43	40781200
2017-01-05	11.43	11.69	11.23	11.24	38855200
2017-01-06	11.29	11.49	11.11	11.32	34453500
2017-01-09	11.37	11.64	11.31	11.49	37304800

Data Preprocessing

Categories and numbers of technical analysis indicators.

Indicator category	Indicator name
Overlap studies	BBANDS, DEMA, EMA, SAREXT, SMA, TEMA, WMA
Momentum indicators	ADXR, APO, AROON, CCI, CMO, MFI, MACD, MOM, PLUS_DI, PPO, ROC, ROCP, RSI, STOCH, STOCHF, TRIX, ULTOSC, WILLR
Volume indicators	AD, OBV
Volatility indicators	ATR, NATR
Cycle indicators	HT_DCPERIOD, HT_SINE, HT_DCPHASE, HT_PHASOR

- Get all stock data given stock ticker list
- Calculate technical analysis indicators for all
- Remove first 90 entries of data (some indicators are NaN)
- Perform Cointegration test for **stock of interest** against all others
    - Cointegration tests if two series have correlation
    - Normalize and append data from stocks with high correlation
    - Fill with 0's if no stock meets requirement
- Normalize and convert data into shorter sequences (24 days) with overlap (12)

Experiment Design

- Ideally, if we want to predict actions for 30 days starting at day t, we could train the model with some historical data immediately before day t to adapt to the market as closely as possible

- For the convenience of evaluation, given a stock ticker, the model is fit with data Mar 2013 - Oct 2017, validated with data from Oct 2017 - Mar 2018, and tested on data Apr 2018 - Dec 2018

- In real life, all historical data are fed into model, and we try to get current predicted action from the model.

- To mimic real world use, to predict action for day t, 24-day data from t-25 to t-1 is inputted to the model.

- I.e. for 30 day action prediction, 30 * 24-day data are inputted to the model, each offset by 1 day.

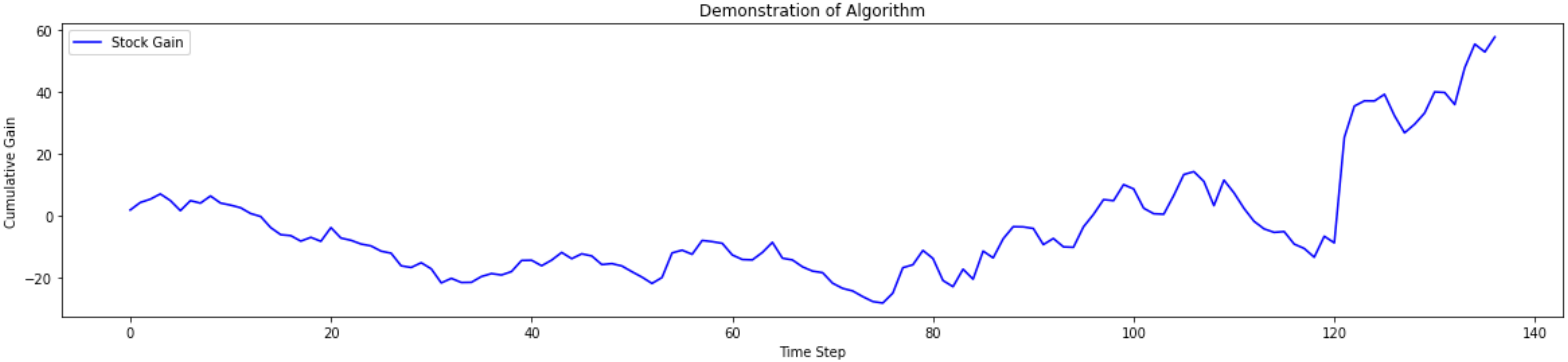
Experiment Result 1

- Stocks considered in Cointegration (correlation) test:

```
['COO','COF','ABBV','CCL','AMD','GOOG','ABMD','ABT','ACN','ADBE',
 'AES','NVDA','AIG','ALL', 'AMG','AMZN','APA','AAPL', 'ATVI','AXP',
 'BA','BBY','CAT','GE','CSCO', 'DRE','EA','EQR','FCX','FE','HST',
 'IBM','INTC','JCI','MMM','MO','ORCL','PPL','T','EXPD','VMC','VNO']
```

- Experiment on COO:

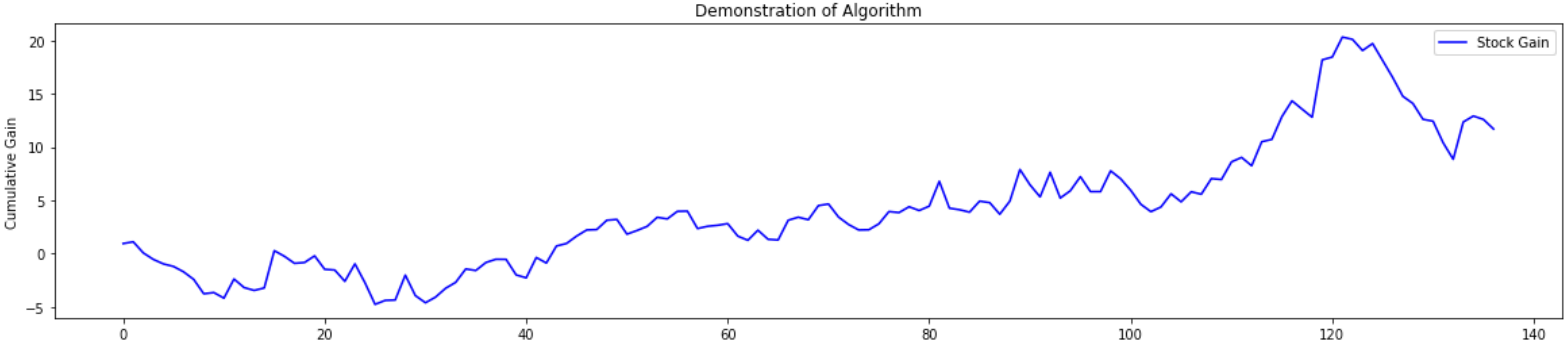
```
Demo Stock ticker: COO, change in closing price during testing period: $19.35
```



Experiment Result 2

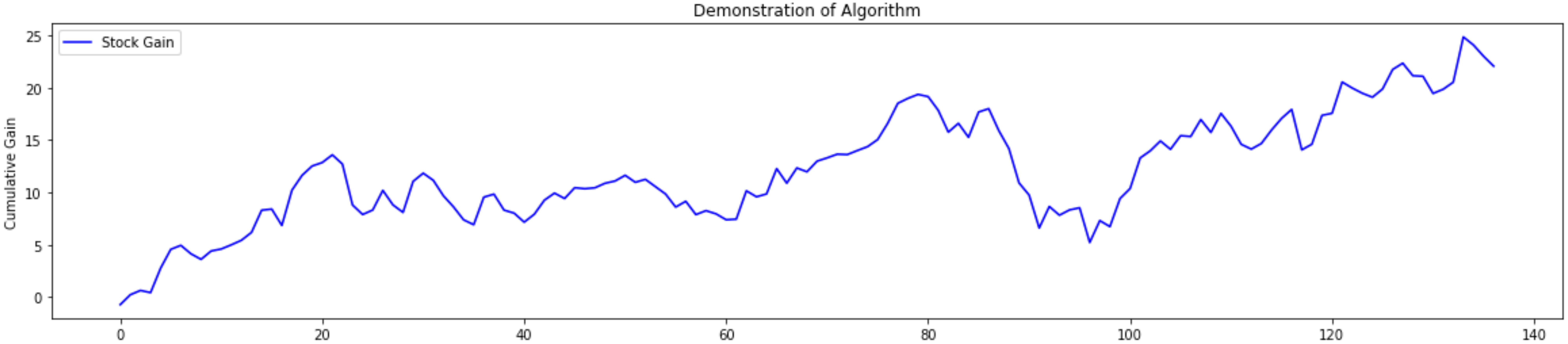
- Experiment on COF:

```
Demo Stock ticker: COF, change in closing price during testing period: $-16.23
```



- Experiment on ABBV:

```
Demo Stock ticker: ABBV, change in closing price during testing period: $-10.21
```



Loss Function Design

- Original Loss Function: MSE(t+1 price, predicted price) +

  Term 1: quality of environment encoding

  - -log(probability of taking same action as t-1)* (cumulative reward until t)

    Term 2a: minimize frequency of changing actions

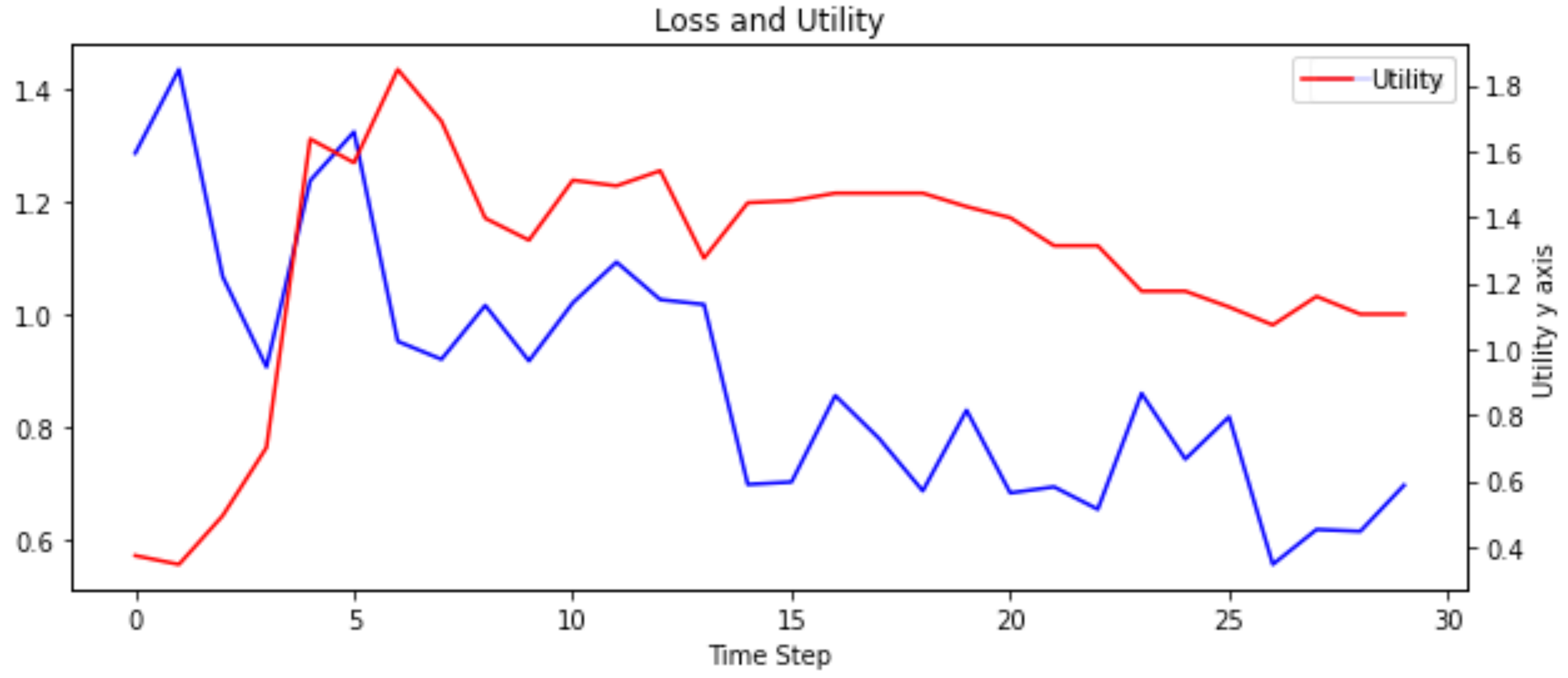
    Term 2b: encourage action to make profit

- Why **Term 2** works: penalize when probability is low but reward is high
- Problem: Penalize bad action choice, but no direct incentive for good action
- Solution: Add **Term 3** to encourage high reward:
  - CrossEntropy(predicted action, greedy action)

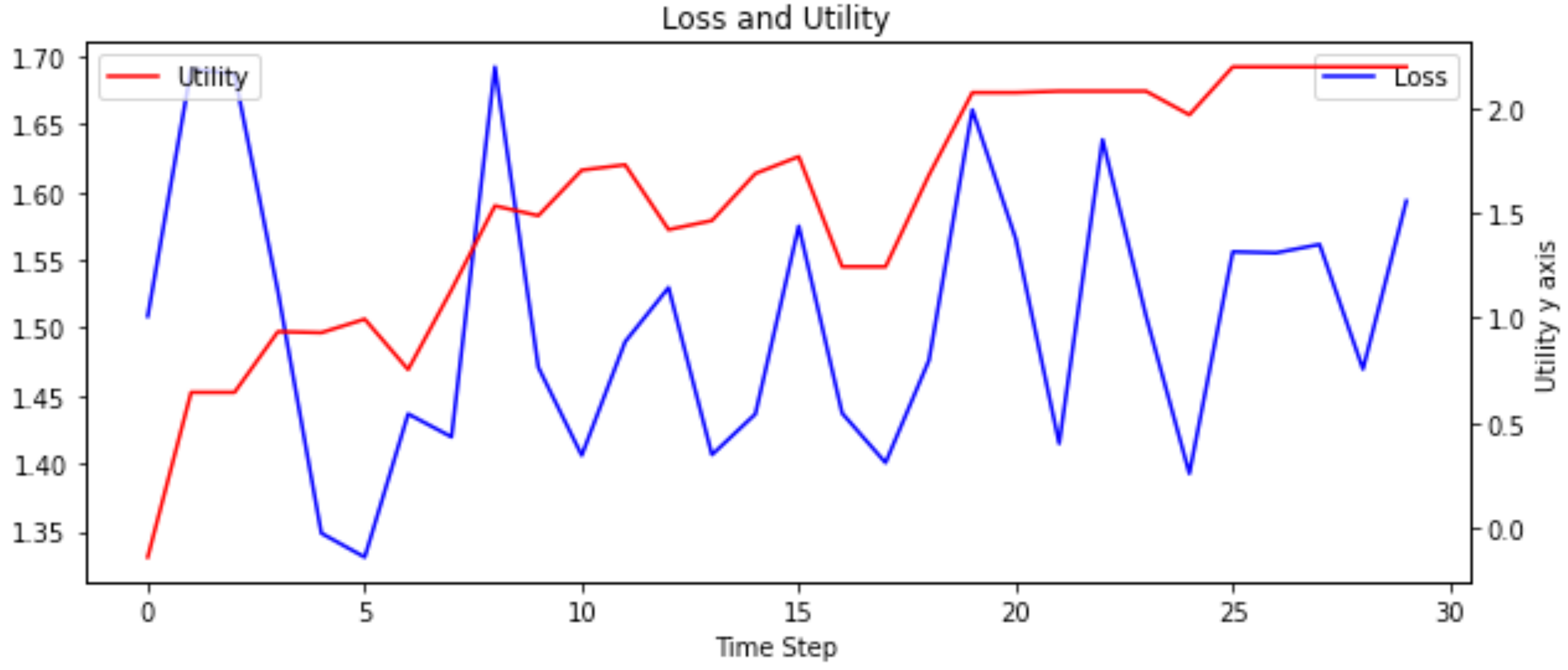
    Greedy Action: the action that maximizes profit at t

Training Comparison for Loss Function

Original Loss



Loss with Term 3

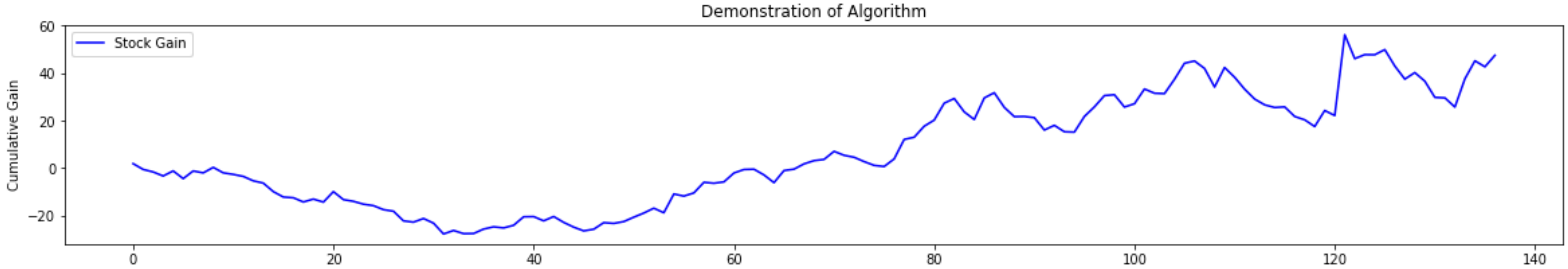


Choice Of Model Output

- Original model uses 3 logits: {-1, 0, 1} for {Short, Neutral, Long}
- Advantage: Neutral position for hard-to-decide environments
- Simplified output: Tanh: (-1, 1) for {Short, Long}
- Disadvantage: loses neutral position
- Advantage:
    - Simpler is better: slightly simplify model and code
    - Easier math with new loss function (MSE instead of CrossEntropy)
    - Generally achieve better results with new loss function

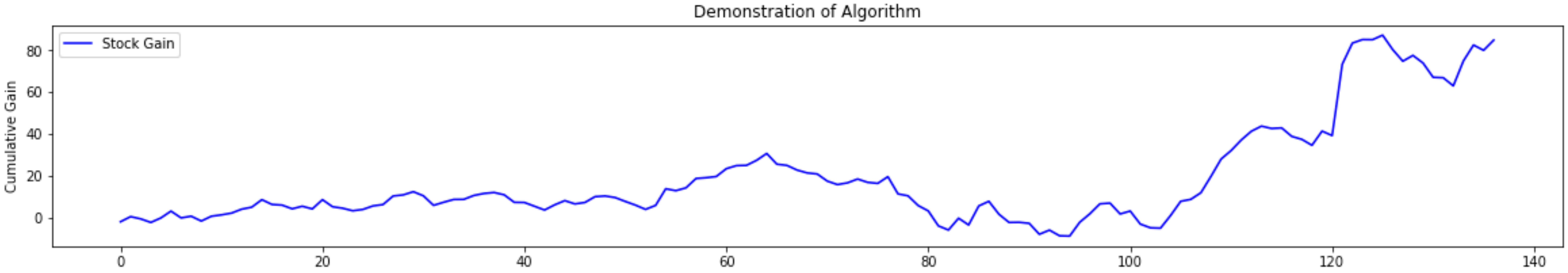
COO: One Hot

Demo Stock ticker: COO, change in closing price during testing period: $19.35



COO: Tanh

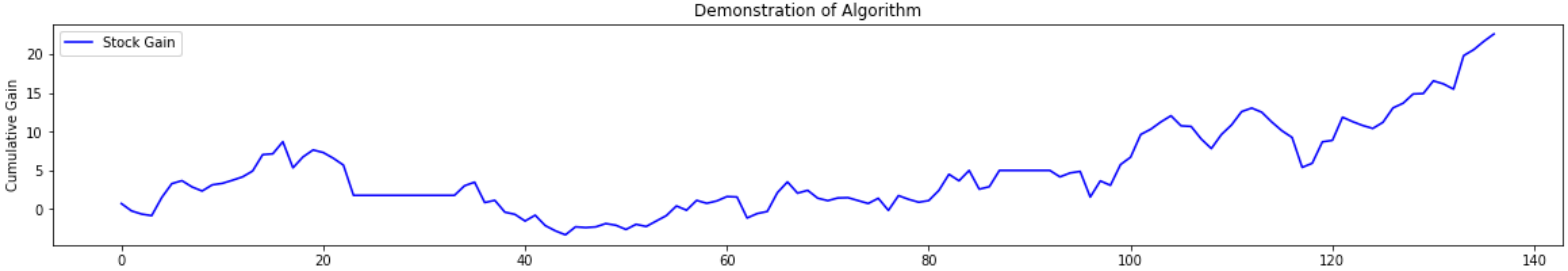
Demo Stock ticker: COO, change in closing price during testing period: $19.35



ABBV: One Hot

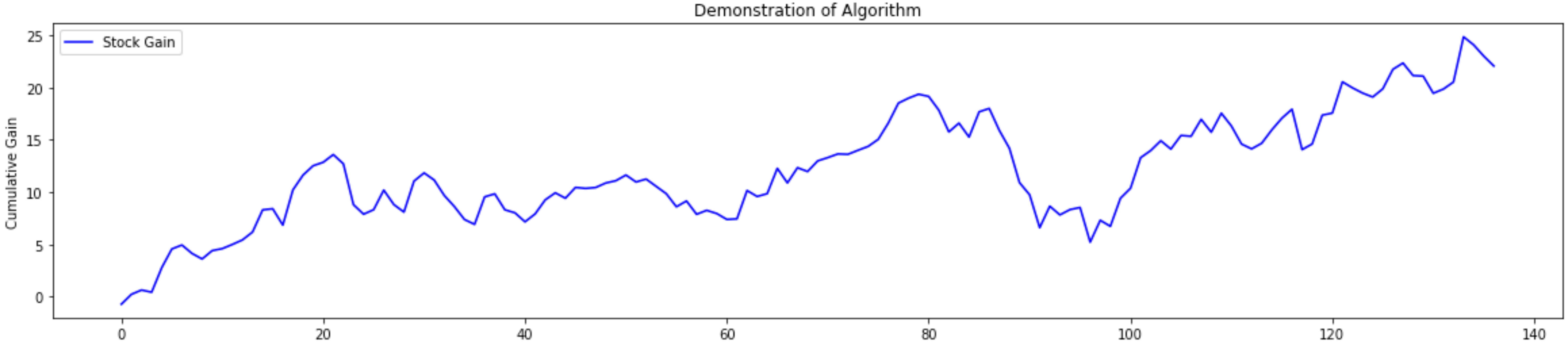
```
Demo Stock ticker: ABBV, change in closing price during testing period: $-10.21
```

```
demo(net, demo_iter , device, 17, 'DLRL')
```



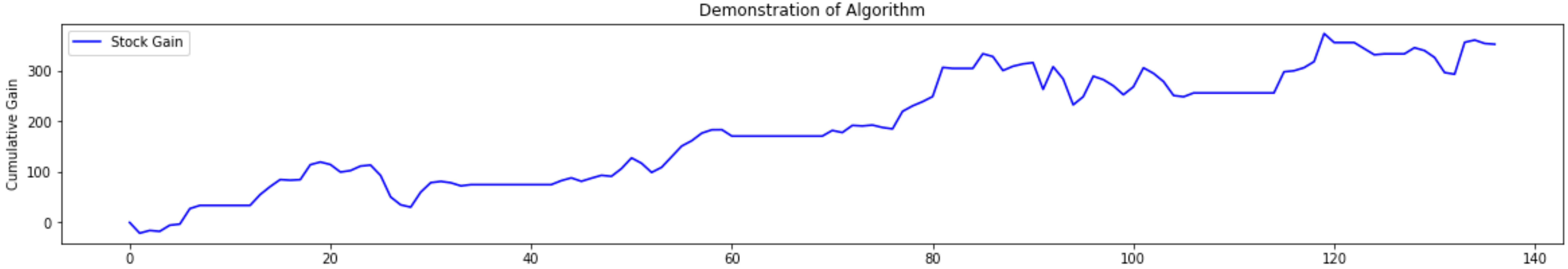
ABBV: Tanh

```
Demo Stock ticker: ABBV, change in closing price during testing period: $-10.21
```



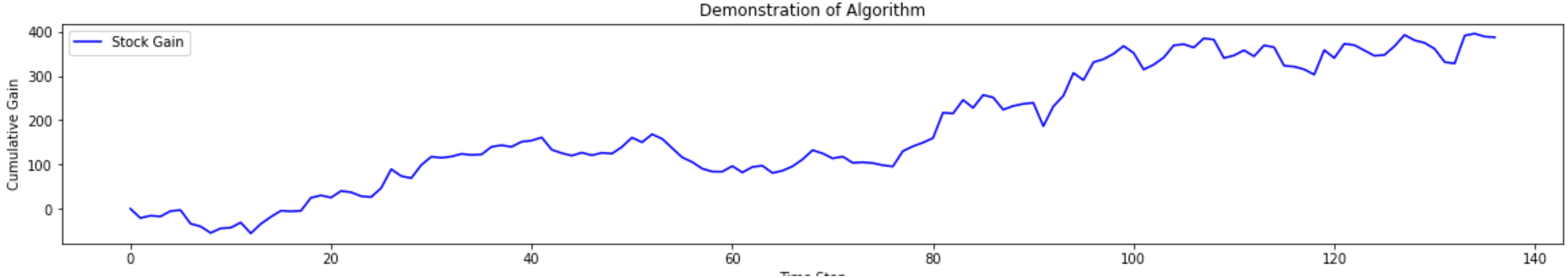
GOOG: One Hot

Demo Stock ticker: GOOG, change in closing price during testing period: $-64.59



GOOG: Tanh

Demo Stock ticker: GOOG, change in closing price during testing period: $-64.59



Thank You!