Online Appendix for

"Deep Tangency Portfolio"

Guanhao Feng Liang Jiang Junye Li Yizhi Song Yuanzhi Wang

Summary of Contents

- Section I presents detailed definitions of bond, equity, and option characteristics used in our empirical studies.
- Section II presents the implementation details of our deep learning algorithm and some robustness checks.
- Section III presents the results from two alternative sample split schemes.
- Section IV presents the performance of the deep tangency portfolio accounting for transaction costs.
- Section V presents the implementation details of machine learning (ML) algorithms.

I Bond, Equity, and Option Characteristics

Table A1: Description of 41 Bond Characteristics

Characteristics	Description
AGE	Time since issuance in years
RATING	Bond credit rating
T2M	The number of years to maturity
SIZE	Amount outstanding
DUR	Bond duration
VAR5	Value-at-risk 5% over past 3 years
VAR10	Value-at-risk 10% over past 3 years
LIQ_BPW	Liquidity measure of transitory price movements.
LIQ_ROLL	Roll's liquidity
LIQ_P_HL	High-low spread estimator
LIQ_P_FHT	Modified illiquidity measure based on zero returns
LIQ_AMIHUD	Amihud liquidity
LIQ_STD_AMIHUD	Standard deviation of Amihud daily liquidity
LIQ_TC_IQR	Interquartile range
MKT_BETA	Market beta
DEF_BETA	DEF factor beta
TERM_BETA	TERM factor beta
LIQ_BETA	Liquidity beta of bond illiquidity factor
DRF_BETA	Downside risk beta controlling bond market factor
CRF_BETA	Credit risk beta controlling bond market factor
LRF_BETA	Liquidity risk beta controlling bond market factor
VIX_BETA	VIX index beta
UNC_BETA	Macroeconomic Uncertainty Beta
STR	Short-term reversal t-1
VARIANCE	Variance of raw returns
SKEW	Skewness of raw returns
KURT	Kurtosis of raw returns
COSKEW	Systematic skewness with bond market
ISKEW	Idiosyncratic skewness
LIQ_RANGE	Simple high-low spread
LIQ_TRADE	Number of trades
MKT_RVAR	Market residual variance
TERM_DEF_RVAR	TERM DEF residual variance
TURN_CB	Bond Turnover
YTM	Yield-to-maturity
MOM6	Momentum from t-2 to t-6
MOM12	Momentum from t-7 to t-12
LTR	Long-term reversal from t-13 to t-48
BARQ	Average daily dollar volume in the 1-month period
STD_BARQ_1MOM	Standard deviation of dollar volume in the 1-month period
LIQ_RANGE_M	Simple high-low spread

Table A2: Description of 61 Equity Characteristics

Characteristics	Description
ABR	Abnormal returns around earnings announcement
ACC	Operating Accruals
ADM	Advertising Expense-to-market
AGR	Asset growth
ALM	Quarterly Asset Liquidity
ATO	Asset Turnover
BASPREAD	Bid-ask spread (3 months)
BETA	Beta (3 months)
BM	Book-to-market equity
BM_IA	Industry-adjusted book to market
CASH	Cash holdings
CASHDEBT	Cash to debt
CFP	Cashflow-to-price
CHCSHO	Change in shares outstanding
CHPMIA	Industry-adjusted change in profit margin
CHTX	Change in tax expense
CINVEST	Corporate investment
DEPR	Depreciation / PP&E
DOLVOL	Dollar trading volume
DY	Dividend yield
EP	Earnings-to-price
GMA	Gross profitability
GRLTNOA	Growth in long-term net operating assets
HERF	Industry sales concentration
HIRE	Employee growth rate
ILL	Illiquidity rolling (3 months)
LEV	Leverage
LGR	Growth in long-term debt
MAXRET	Maximum daily returns (3 months)
ME	Market equity
ME_IA	Industry-adjusted size
MOM12M	Cumulative Returns in the past (2-12) months
MOM1M	Previous month return
MOM36M	Cumulative Returns in the past (13-35) months
MOM60M	Cumulative Returns in the past (13-60) months
MOM6M	Cumulative Returns in the past (2-6) months
NI	Net Equity Issue
NINCR	Number of earnings increases
NOA	Net Operating Assets
OP	Operating profitability
PCTACC	Percent operating accruals
PM	Profit margin
PS	Performance Score
RD_SALE	R&D to sales
RDM	R&D Expense-to-market
RE	Revisions in analysts' earnings forecasts
RNA	Return on Net Operating Assets
ROA	Return on Assets
ROE	Return on Equity
RSUP	Revenue surprise
RVAR_CAPM	Residual variance - CAPM (3 months)
RVAR_FF3	Res. var. - Fama-French 3 factors (3 months)
SVAR	Return variance (3 months)

Description of 61 Equity Characteristics (continued)

Characteristics	Description
SEAS1A	1-Year Seasonality
SGR	Sales growth
SP	Sales-to-price
STD_DOLVOL	Std of dollar trading volume (3 months)
STD_TURN	Std. of Share turnover (3 months)
SUE	Unexpected quarterly earnings
TURN	Shares turnover
ZEROTRADE	Number of zero-trading days (3 months)

Table A3: Description of 30 Equity Option Characteristics

Characteristics	Description
IVSLOPE	Implied Volatility Slope
IVVOL	Volatility of atm implied volatility
IVRV	Implied and historical volatility spread
IVRV_RATIO	Ratio of implied to historical volatility
ATM_CIVPIV	Implied volatility spread
SKEWIV	Implied volatility skew
IVD	Implied volatility duration
DCIV	Change of implied volatility of atm call
DPIV	Change of implied volatility of atm put
ATM-DCIVPIV	Change of implied volatility spread
NOPT	Number of traded options
SO	Stock-option volume ratio
DSO	Stock-option dollar volume ratio
VOL	Option Trading Volume
PCRATIO	Put-call ratio
PBA	Proportional bid-ask spread
TOI	Total open interest
MFVU	Option-implied upside semivariance
MFVD	Option-implied downside semivariance
RNS1M	1-month risk-neutral skewness
RNK1M	1-month risk-neutral kurtosis
IVARUD30	Option-implied variance asymmetry
RNS3M	3-month risk-neutral skewness
RNK3M	3-month risk-neutral kurtosis
RNS6M	6-month risk-neutral skewness
RNK6M	6-month risk-neutral kurtosis
RNS9M	9-month risk-neutral skewness
RNK9M	9-month risk-neutral kurtosis
RNS12M	12-month risk-neutral skewness
RNK12M	12-month risk-neutral kurtosis

II Implementation Details

II.1 Data Cleaning

As shown in Table 1, we provide the summary statistics for all our Bond-Month observations (before standardization and imputation). Since the raw bond return data began in July 2002 and at least 1 year of data is required to initialize risk characteristics, we start with data from July 2004. After standardizing the raw characteristics, we filter out bonds with a maturity of less than 2 years and impute missing values with 0. The equity and option characteristics are merged into bond data by the "PERMNO". If the firm has issued multiple stocks simultaneously, we keep only the earliest-issued stock and merge it.

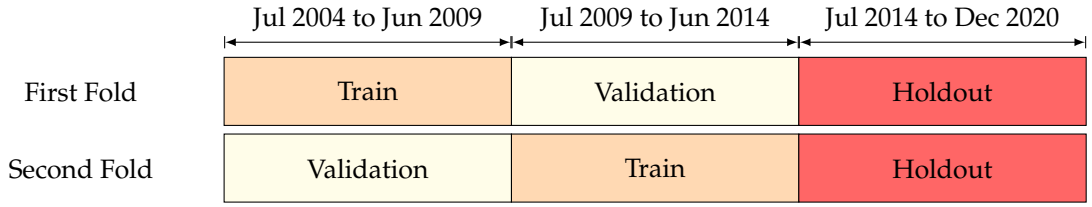
II.2 Model Training

We divide the TRACE dataset into two parts: we train on samples from July 2004 to June 2014 and test on samples from July 2014 to December 2020. We train the neural network's structure and parameters during the in-sample period and determine the portfolio's weights based on the in-sample factor statistics. To avoid the effect of outliers during training, we also winsorize the in-sample bond returns within the $ret_{low,t}$ and $ret_{up,t}$ bounds at each month t, where $ret_{low,t}$ and $ret_{up,t}$ are the cross-sectional 5% and 95% quantiles of month t, respectively. Each month, bond returns that are lower/higher than the monthly low/up quantiles of cross-sectional returns will be revised to $ret_{low,t}$ and $ret_{up,t}$, respectively. It is important to emphasize that winsorization is only applied to in-sample data. All out-of-sample (OOS) results are tested on NON-Winsorized data, which may contain potential extreme values.

We use both equal- and value-weighted market factors as our benchmarks. For the IPCA/RP-PCA model, we train the PCA structure using balanced individual bond returns from July 2004 to June 2014, and compute the factor values for both the in-

Figure A1: Two-Fold Cross Validation

This figure illustrates the deterministic two-fold cross-validation scheme used to select the tuning parameters. Specifically, we determine the tuning parameters using the sample from July 2004 to June 2014. The deterministic design divides the sample into two consecutive fold samples. We train our neural network separately on one, then calculate the fit results with different tuning parameters on the other. We calculate the average out-of-sample loss and select the parameter pair with the best performance.



sample and OOS periods using this trained structure. Similarly, our neural network takes the individual bond data from July 2004 to June 2014 as a training set for our neural network structure, and we determine the tuning parameters through a two-fold cross-validation process, as shown in Figure A1, for which we will provide more details in Appendix II.3.

In ranking the bond returns, we choose $a_1 = 50$ and $a_2 = 8$ in Equation (16) such that at each time, approximately 50 % to 70% of assets are in the middle rank and have negligible weights, similar to the traditional sorting procedure (see Figure A2).[13]

After training, we use the fixed network structure and input the pairwise characteristics to generate in-sample and OOS estimates. All results in Section II are computed using PyTorch 1.10.2 and parallelized across a server with 96 Intel(R) Xeon (R) Gold 6230 @ 2.10GHz CPUs and 314 GB of RAM.

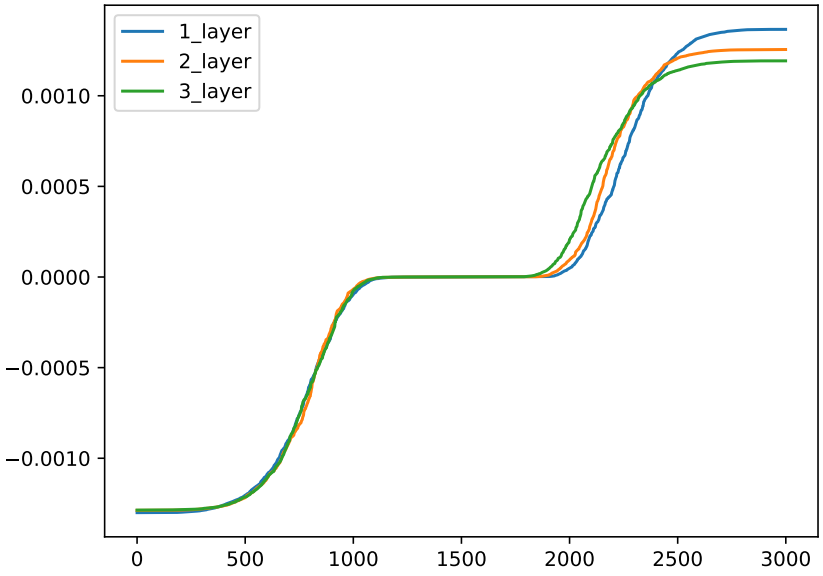
II.3 Tuning Parameters Selection

In this subsection, we outline our procedure for selecting tuning parameters. To determine the optimal tuning parameters for our network, we employ a two-fold cross-validation approach (as shown in Figure A1) as follows. We first divide the sample

[13]We select a_1 and a_2 sequentially rather than through a full two-dimensional grid because both parameters primarily govern the concentration of the long and short tails. Moderate changes around the selected values produce similar S-shaped weighting profiles and do not change the qualitative ranking of the network architectures. The pair $a_1 = 50$ and $a_2 = 8$ minimizes the average validation loss among the values considered.

Figure A2: Softmax Ranking ($a_1 = 50$ and $a_2 = 8$)

This figure plots the individual corporate bonds' weights generated by the softmax rankings of deep characteristics for different layers.



period from July 2004 to June 2014 into two consecutive fold samples of equal length. We train our neural network on one fold and evaluate the fitted results using different tuning parameters on the other fold. We compute the average loss over the validation samples and select the parameter pairs that yield the smallest loss. Finally, we retrain the model using the chosen tuning parameters. This approach ensures that the data used is completely in-sample, eliminating any look-ahead bias that may affect our OOS trading results discussed in the main text. When implementing, we start with a reasonable set of initial tuning parameters and test additional points adjacent to these sets. We evaluate a total of 120 combinations of tuning parameters, which are shown in Table A4.

III Alternative Sample Splits

III.1 Reversed Sample

To ensure robustness in OOS evaluation, we reverse the training, validation, and forecasting periods in this subsection. In other words, we use the end periods of the

Table A4: Tuning Parameter Selection in the Empirical Analysis

This table presents the network tuning parameters used in our empirical analysis.

Notation	Tuning Parameters	Candidates	Chosen
a_1	Value in equation (16)	50	50
a_2	Value in equation (16)	5,8,10	8
HDN	Number of nodes in the hidden layer	66	66
BTCH	Batch size, in months	10,30	10
LR	Learning rate	1e-3, 5*1e-3, 1e-2, 5*1e-2, 1e-1	1e-3
DR	Dropout rate	0.05, 0.1, 0.2, 0.3	0.1
EPCH	Number of optimization epochs	400	400
OPT	Optimization method	Adam	Adam

original sample as training and validation sets and the beginning periods as the OOS evaluation periods.

We use the same approach as in Section 2.2 to build the neural network. Panel A in Table A5 shows that, 1) the deep tangency portfolios have significantly higher OOS Sharpe ratios than the market portfolio; 2) the 2-layer deep tangency portfolios achieve the highest OOS Sharpe ratio in most cases, both of which are consistent with our previous findings.

III.2 Sample Randomly Split

To further ensure robustness in OOS evaluation, we follow Fama and French (2018) to randomly split the dataset into three parts sequentially. Specifically, in every quarter (three months), the data is randomly assigned to one of three groups. This process continues until each month's data is assigned to a group. Figure A3 illustrates how we divide the dataset into three groups, represented by different colors. We then employ the same approach as in Section 2.2 to construct the neural network and implement a two-fold cross-validation scheme: for each test group, the other two groups are used for training and validation. Panel B in Table A5 shows the OOS performance for all three groups. We find that: 1) the deep tangent portfolios have significantly higher OOS Sharpe ratios than the market portfolio; 2) the layer-2 deep tangent portfolios achieve the highest OOS Sharpe ratio in most cases. These results are consistent with our previous findings.

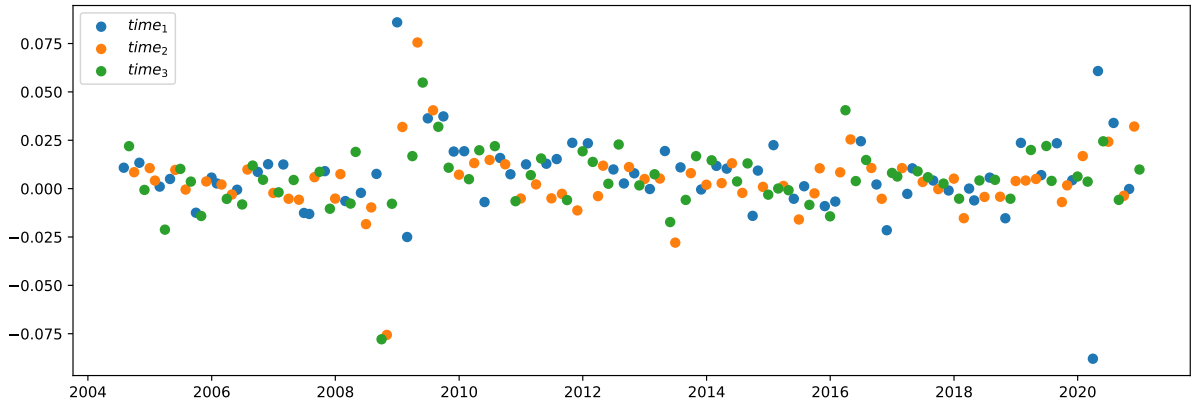
Table A5: Deep Tangency Portfolios in Two Alternative Sample Split Schemes

This table reports the out-of-sample annualized Sharpe ratios of the deep tangency portfolios using multiple-layer deep factors from two alternative sample split schemes. L_i represents a neural network with i layers. We respectively describe two panels as follows: (i) Panel A presents the performance of the reversed sample. We use samples from January 2011 to December 2020 for model training and validation, and from July 2004 to December 2010 for out-of-sample testing; and (ii) Panel B displays the performance for the randomly split sample. We follow Fama and French (2018) to randomly split the dataset into three parts sequentially. Specifically, in every quarter (three months), the data is randomly assigned to one of three groups. We employ the same approach as in Section 2.2 to construct the neural network and implement a two-fold cross-validation scheme: for each test group, the other two groups are used for training and validation. For example, the "$time_1$" column displays the out-of-sample performance when groups 2 and 3 are used for training and validation, respectively, and group 1 is used for the out-of-sample evaluation. The "all" column displays the out-of-sample Sharpe ratios for all three groups. Additionally, in both panels, "MKT" represents the market factor, which is compared with the other three multiple-layer deep tangency portfolios. Moreover, in Panels A, the test of the Sharpe ratio improvement by including an additional deep factor is based on Barillas and Shanken (2017). ***, **, and * denote the levels of significance of 1%, 5%, and 10%, respectively.

	Equal-Weighted				Value-Weighted			
	Panel A: Reversed Sample							
	MKT	L_1	L_2	L_3	MKT	L_1	L_2	L_3
SR	0.87	1.41***	1.59***	1.26***	0.78	1.27***	1.50***	1.39***
	Panel B: Sample Randomly Split							
	$time_1$	$time_2$	$time_3$	all	$time_1$	$time_2$	$time_3$	all
MKT	1.14	0.73	1.06	0.98	1.11	0.74	0.91	0.93
L_1	2.82	1.76	2.47	2.19	3.30	1.97	0.99	1.70
L_2	3.07	2.83	2.17	2.65	2.85	2.37	2.08	2.41
L_3	2.47	1.93	2.21	2.18	2.22	2.26	2.47	2.31

Figure A3: Bond Market Return for the Sample Randomly Split

This figure plots the bond market return from July 2004 to December 2020. Different colors represent three randomly split groups, denoted by $time_1, time_2$, and $time_3$, respectively.



IV Deep Tangency Portfolios Accounting for Transaction Costs

Following Kelly, Palhares, and Pruitt (2023), we incorporate transaction costs based on turnover. After the weights of a deep factor are generated by high-dimensional characteristics as in Equation (14), we replace the raw weights $w_d(z_{t-1})$ by the "slowed down" weights $\widetilde{w}_d(z_{t-1})$ according to exponential smoothing as follows:

$$\widetilde{w}_d(z_{t-1}) = (1-\gamma)w_d(z_{t-1}) + \gamma\widetilde{w}_d(z_{t-2}), \tag{25}$$

where γ is a tuning hyperparameter that can control the degree of smoothing, and we initialize with $\widetilde{w}_d(z_{-1}) = w_d(z_0)$. Subsequently, we calculate the net deep factor returns, $\widetilde{R}_{d,t}$, by subtracting the costs from gross portfolio returns in Equation (18),

$$\begin{aligned}\widetilde{R}_{d,t} &= R_{d,t} - R_{TC,t} \\ &= \widetilde{w}_d(z_{t-1})r_t - \lambda \times \left|\widetilde{w}_d(z_{t-1}) - \widetilde{w}_d(z_{t-2})\right|,\end{aligned} \tag{26}$$

where $|\widetilde{w}_d(z_{t-1}) - \widetilde{w}_d(z_{t-2})|$ is the portfolio turnover as the ℓ_1 distance between consecutive smoothed weight vectors, and λ represents the constant transaction costs. To be conservative, we apply one-way costs of two representative levels ($\lambda = 15$ and 20 bps) to all trades. Finally, we implement the same approach as in Section 2.2 and add a range constraint for the tangency portfolio weights to construct the deep tangency portfolio.[14]

Table A6 reports Sharpe ratios of the resulting deep tangency portfolios across different network layers, smoothing parameters, and transaction cost levels. A central finding is that incorporating transaction costs does not alter the main conclusions of our paper. Across both equal-weighted and value-weighted specifications, the relative ranking of models remains stable. In particular, the 2-layer network consistently delivers the highest Sharpe ratios, even after accounting for turnover-based trading

[14]The case including transaction costs is more complicated. To avoid extreme short positions (e.g., Ferreira and Santa-Clara, 2011; Thornton and Valente, 2012) after including transaction costs, we restrict the tangency portfolio weights to be between -2 and 3.

Table A6: Deep Tangency Portfolio Accounting for Transaction Cost

This table presents the out-of-sample annualized Sharpe ratios of the deep tangency portfolios constructed using multiple-layer deep factors accounting for transaction costs. L_i represents a neural network with i layer(s), and γ is the parameter for turnover levels. "MKT" is the market factor, which serves as the benchmark for comparison with the other three multiple-layer deep tangency portfolios. The test of the Sharpe ratio improvement by including an additional deep factor is based on Barillas and Shanken (2017). ***, **, and * denote the level of significance of 1%, 5%, and 10%, respectively. We take the sample from July 2004 to June 2014 for model training and validation, and the sample from July 2014 to December 2020 for out-of-sample testing.

	Equal-Weighted				Value-Weighted			
γ	MKT	L_1	L_2	L_3	MKT	L_1	L_2	L_3
				Panel A: 15 Basis Points				
0.0	0.85	0.15	1.50***	0.28	0.90	0.12	1.91***	0.40
0.1	0.85	0.44	1.34***	0.38	0.90	0.34	1.62***	0.41
0.3	0.85	0.77	1.14***	0.86	0.90	0.85	1.62***	0.90
0.5	0.85	0.74	1.06***	0.71	0.90	0.62	1.07***	0.78
0.7	0.85	0.89	1.18***	0.98***	0.90	0.91	1.15***	0.95
0.9	0.85	0.85	0.93**	0.88	0.90	0.93	0.98**	0.93
				Panel B: 20 Basis Points				
0.0	0.85	0.14	1.52***	1.01***	0.90	0.24	1.87***	1.06***
0.1	0.85	0.75	1.54***	0.31	0.90	0.27	1.92***	0.28
0.3	0.85	0.38	1.17***	0.97***	0.90	0.51	1.23***	1.00**
0.5	0.85	0.83	1.08***	0.90	0.90	0.99**	1.18***	0.93
0.7	0.85	0.77	1.00***	0.86	0.90	0.88	0.99**	0.89
0.9	0.85	0.85	0.92*	0.87	0.90	0.93	0.97*	0.90

costs and weight smoothing.

More broadly, the results indicate that the performance of the deep factors is robust to realistic implementation frictions. While higher transaction costs and stronger smoothing naturally attenuate returns, they do not eliminate the economic advantage of deeper architectures. The persistence of the 2-layer model's outperformance suggests that its superior risk-adjusted returns are not driven by excessive trading, but by more informative and stable cross-sectional signals. Taken together, these findings reinforce the empirical relevance of deep tangency portfolios when transaction costs are explicitly accounted for.

V The Implementations of Standard Machine Learning Methods

In Section 4.5, we present a forecast-implied long-short portfolio based on predicted bond returns from four ML methods: Lasso, Ridge, principal component anal-

ysis regression (PCA), and standard neural networks (NN) to predict the asset returns for comparison directly. Our implementation follows Feng, He, Wang, and Wu (2025).

PCA is a common dimension reduction technique used in empirical asset pricing to address the bias-variance trade-off. It involves using a low-dimensional linear transformation of the predictors to construct a predictive model. The procedure consists of two steps: the first combines predictors with a small set of linear combinations that best preserve the covariance structure, and the second involves using the first K components in multiple regressions. In our applications, K is set to 5 for PCA.

Lasso and ridge are linear regularized regressions used in ML finance to maintain interpretability. They add a penalty over least squares to preserve the predictors without transforming them. Lasso performs variable selection, while ridge shrinks the regression coefficients of useless predictors to small values. A tuning parameter controls the penalty weight, with a larger penalty weight imposing more shrinkage on the coefficients. A two-fold cross-validation is used to determine all tuning parameters.

For the standard neural network (NN), we use the same input characteristics and two-fold cross-validation scheme as in our deep learning network. We also add regularization penalty terms to the loss function to limit the complexity of the neural networks, stabilize estimates, and avoid overfitting.