Data-Efficient Realized Volatility Forecasting with Vision Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent work in financial machine learning has shown the virtue of complexity: the phenomenon by which deep learning methods capable of learning highly nonlinear relationships outperform simpler approaches in financial forecasting. While transformer architectures like Informer have shown promise for financial time series forecasting, the application of transformer models for options data remains largely unexplored. We conduct preliminary studies towards the development of a transformer model for options data by training the Vision Transformer (ViT) architecture, typically used in modern image recognition and classification systems, to predict the realized volatility of an asset over the next 30 days from its implied volatility surface (augmented with date information) for a single day. We show that the ViT can learn seasonal patterns and nonlinear features from the IV surface, suggesting a promising direction for model development.

1 Background

The implied volatility surface (IV surface) of an optionable asset encodes information about market dynamics and sentiment, the future realized volatility of the asset, and the probability distribution of its return Bali et al. [2022]. Traders construct features from the IV surface to infer this information using options pricing theory or empirical observations. Recent work in financial machine learning has also discovered the virtue of complexity: Gu et al. [2020], Didisheim et al. [2023] the existence of highly nonlinear features in financial data which can be extracted using neural networks, contradicting prior assumptions that financial returns can be explained by a small number of predictive factors. However, machine learning methods are difficult to apply to financial data because the data itself is noisy and limited in scale. For example, our entire preprocessed dataset totals 6.1 GB while text corpora used to train frontier LLMs contain multiple terabytes of data Liu et al. [2024].

2 Prior Work

The use of neural networks to identify nonlinear patterns in financial data is investigated extensively in Gu et al. [2020]. Other examples include overparametrized factor models with more factors than assets under observation Didisheim et al. [2023], Transformer-based time series forecasting Zhou et al. [2021], and structured approaches to machine learning in finance Dixon and Halperin [2019]. Neural networks have also been applied to generate smooth, arbitrage-free IV surfaces from raw option prices Ackerer et al. [2020], Wiedemann et al. [2025]. However, fewer researchers have investigated deep learning for predictions from IV surfaces. Previous approaches include the use of hand-constructed features Neuhierl et al. [2022] or convolutional neural networks (CNNs) Kelly et al. [2023], with the latter using the IV surface on the last trading day of the month to predict the monthly return of the next month. We train Vision Transformer (ViT) models on IV surfaces, treating them

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

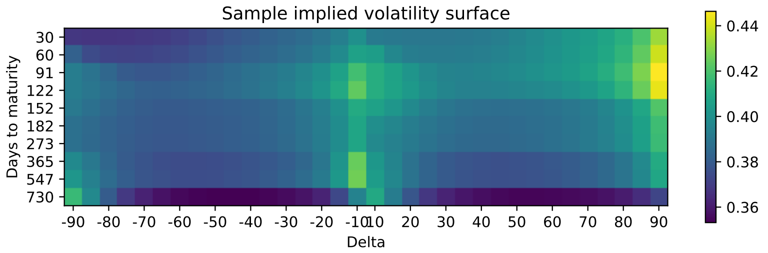

Figure 1: IV surface for NVDA stock on 2021-04-13, presented as a one-channel image instead of the traditional three-dimensional surface. Negative deltas correspond to puts.

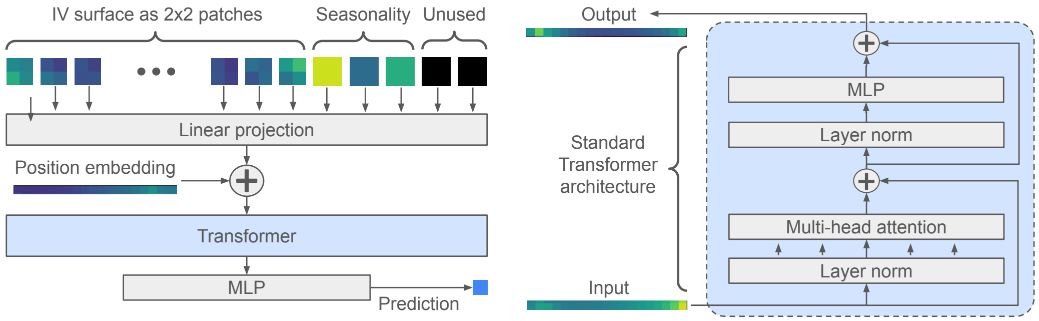

Figure 2: Vision transformer architecture (left) on our data, with more detailed schematic of the standard transformer architecture used in our model (right). In the deep Vision Transformer, the MLP layer in the Transformer model is repeated.

as small, single-channel images. ViTs on images require less computational cost than CNNs and provide more stable training performance Dosovitskiy et al. [2021]; thus we hypothesize they will be more robust to noisy data and outliers.

3 Methodology

3.1 Data preparation

We use the OptionMetrics IvyDB implied volatility, a grid of smoothed interpolated values with implied δ of the option on one axis and number of days to maturity on the other; and realized volatility calculated by OptionMetrics over $n = 28$ calendar days, using the standard deviation on the daily log return. Following Kelly et al. [2023], we split data by year and month and drop any samples with incomplete data, producing a full dataset of 4,259,070 rows between 2012 and 2022. We augment the IV surface with the month, day, and day of the week of the observation, scaled to values in $[0, 1]$, to allow the model to capture seasonal trends (Figure 1). See Appendix A for details.

3.2 Model Architecture

We tested both deep and wide ViT architectures, adapting the original ViT model Dosovitskiy et al. [2021] for single-channel matrices of size 10×36 instead of traditional images[1]. Because the ViT outputs a vector, we add a small four-layer MLP to produce the final real-valued prediction. We study the performance of this model on our dataset, varying the number of layers and the number of parameters per layer. Model scaling is of particular interest, as we show that small models can be trained on limited data and achieve strong performance in the task of forecasting realized volatility. We compare our model against a baseline multilayer perceptron (MLP) on the flattened IV surface, observing that the ViT architecture outperforms the MLP. The MLP is also more difficult to train, requiring early stopping, batch normalization, and multiple training attempts with the best model selected at the end.

[1] Model definitions and code to reproduce all results will be released with the final version of this paper.

Model	ViT_0.005M_wide	ViT_0.12M_deep	ViT_0.17M_wide
# Params	46466	122114	170754
Model	ViT_0.5M_deep	ViT_0.5M_wide	ViT_1.7M
# Params	469506	545282	1732610

Table 1: Model definitions with number of parameters; full definitions are in Appendix B.

3.3 Training

We define train and test sets such that if the test year is y_i, the corresponding n training years are $y_{i-n}, y_{i-n+1}, ..., y_{i-1}$. We pay close attention to the choice of optimizer, learning rate schedule, and loss function to achieve more stable and efficient training. Prior work has applied early stopping, regularization, and ensembling to overcome the challenges of training on financial data Gu et al. [2020]. We apply batch normalization, a form of regularization, and Xavier initialization Glorot and Bengio [2010] in the MLP prediction component of our model. Our training procedure follows the process used to train text foundation models such as DeepSeek-v3 DeepSeek-AI et al. [2025] and Llama Grattafiori et al. [2024], scaled down for the available data. We use a cosine annealing learning rate scheduler[2], introduced in Loshchilov and Hutter [2017] and shown to achieve strong performance on ImageNet Goyal et al. [2018] with large batch sizes. (We use a batch size of 2048 for all experiments.) We use the AdamW optimizer Loshchilov and Hutter [2019] and select the best-performing model from all training epochs. To reduce the impact of outliers, we use the Huber loss 1, a convex loss function that is quadratic for small values of $\hat{y} - y$ and linear for large values. We report the model's R^2 on unseen test data.

$$\ell_{\text{huber}}(\hat{y}, y) = \begin{cases} \frac{1}{2}\left(\hat{y} - y\right)^2 & |\hat{y} - y| \leq d \\ d \cdot \left(|\hat{y} - y| - \frac{1}{2}d\right) & \text{otherwise} \end{cases} \tag{1}$$

We evaluate both deep and wide ViT models of varying sizes. The deep models have four MLP layers in the Transformer module (see Figure 2), while the wide models have one MLP layer in this module.

4 Results

We study the performance of the models listed in Table 1 to predict the realized volatility of an asset over the next 30 days when trained on varying dataset sizes. Because one of the challenges for financial machine learning is the availability of data, we evaluate model performance when trained on one, four, or ten years of data, finding that smaller models (.05-.17M) can perform well when trained on smaller datasets but collapse on large ones, while the 0.5M models do not improve on the smaller models. The 1.7M model yields the best performance but requires the full ten years of training data (Figure 3). Additionally, all models perform poorly if the training and test data are dissimilar (Figure 4). In this case, the small models provide an advantage as they could be retrained as new data becomes available. Model performance varies across different market conditions, with all models showing reduced performance when tested on 2020 data (Figure 4). The best model is ViT_1.7M trained on 2012-2021 data, which achieves $R^2 = 0.41$ on the 2022 test set (Figure 3). All ViT models reach their maximum R^2 within one or two epochs, with further training causing overfitting (Appendix C).

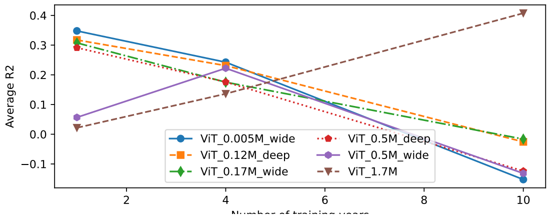

Figure 3: Effect of dataset size on model R^2. Where multiple train-test splits are possible, average R^2 is reported.

Key findings: ViT models can extract nonlinear features from IV surfaces, with small models requiring as little as one year of data to train. Despite the limited availability of market data, **designing model architectures and training processes to fit the available data can enable the development of transformer models for financial forecasting tasks.**

[2] https://github.com/katsura-jp/pytorch-cosine-annealing-with-warmup, called with parameters `scheduler scheduler_first_cycle_steps = 200, scheduler_max_lr = 0.01, scheduler_min_lr = 0.001, scheduler_warmup_steps = 100, scheduler_gamma = 0.95.`

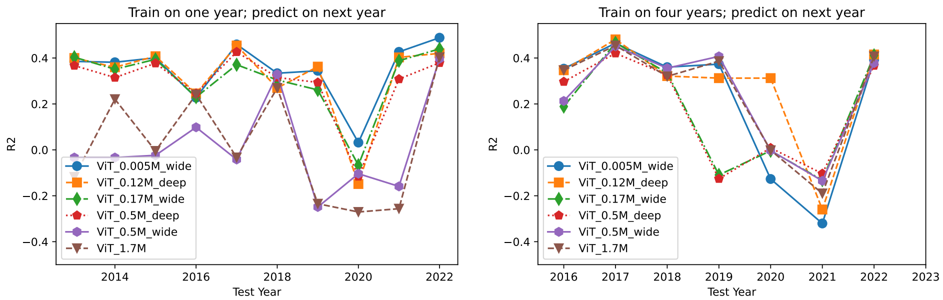

Figure 4: Training on one year (left) or four (right) and predicting the 30-day realized volatility on the next year, for data between 2012-2022. The performance drop for the test year 2020 reflects market disruption during the COVID pandemic. In practice one could iteratively retrain the models; note that the small models recover their performance on the 2021 test sample.

5 Ablation Testing

We test our ViT model against two ablations: the ViT model on the IV surface with no seasonality augmentation, and an MLP-only model with roughly the same number of parameters. Full definitions of these models are provided in Appendix B. Removing the seasonality information has a small negative effect, suggesting the model is primarily extracting nonlinear patterns from the IV surface. The MLP-only models do not perform well, with larger model sizes actually yielding worse performance.

Model	# Train Years	Baseline R^2	No seasonality
ViT_0.5M_deep	4	0.35	0.35
ViT_0.5M_wide	4	0.37	0.35
ViT_1.7M	10	0.41	0.38

Table 2: Effect of removing seasonality information.

Model	ViT Params	Baseline R^2	MLP Model	MLP Only R^2	MLP Params
ViT_0.12M_deep	122114	0.27	MLP_0_12	0.29	114842
ViT_0.17M_wide	170754	0.37	MLP_0_17	0.29	174722
ViT_0.5M_deep	469506	0.35	MLP_0_5	0.17	515252
ViT_0.5M_wide	545282	0.37	MLP_0_5	0.17	515252

Table 3: Comparison between ViT and MLP-only architectures with similar parameter counts. All models were trained on 4 years of data from 2018-2021 and tested on 2022.

6 Planned and Future Work

This is an ongoing project with many interesting directions. Our top priority is to study the potential for transfer learning on IV surfaces: the ability to fine-tune a model or retrain only the final stages of the model, such as regressor or classifier layers, to predict a different target value. ViT models trained on image datasets exhibit this property and are often fine-tuned for specific classification tasks in medical or scientific imaging Li et al. [2021]. We are also interested in investigating whether the ViT model can learn output vectors that generalize to other prediction tasks if the MLP predictor is retrained. We tested for this capability using the task of predicting the asset's return over the next 28 days and did not observe it; however this task is more difficult than predicting the realized volatility. Following our theme of applying foundation model techniques to financial data, we could compare ensembling, applied in Kelly et al. [2023], against a Mixture-of-Experts architecture DeepSeek-AI et al. [2025]. Finally, as there is clearly a link between model size, dataset size and performance, a theoretical understanding of the information content of IV surfaces could provide guidance for optimal data sampling to improve model performance.

References

Damien Ackerer, Natasa Tagasovska, and Thibault Vatter. Deep smoothing of the implied volatility surface, 2020. URL https://arxiv.org/abs/1906.05065.

Turan G Bali, Fousseni Chabi-Yo, and Scott Murray. A factor model for stock returns based on option prices. *Available at SSRN 3487947*, 2022.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Antoine Didisheim, Shikun (Barry) Ke, Bryan T. Kelly, and Semyon Malamud. Complexity in factor pricing models. NBER Working Papers 31689, National Bureau of Economic Research, Inc, Sep 2023. URL https://ideas.repec.org/p/nbr/nberwo/31689.html.

Matthew Dixon and Igor Halperin. The four horsemen of machine learning in finance. *SSRN Electronic Journal*, 01 2019. doi: 10.2139/ssrn.3453564.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. URL https://arxiv.org/abs/1706.02677.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak,

Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle

Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 02 2020. ISSN 0893-9454. doi: 10.1093/rfs/hhaa009. URL `https://doi.org/10.1093/rfs/hhaa009`.

Bryan T. Kelly, Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu. Deep learning from implied volatility surfaces. Swiss Finance Institute Research Paper Series 23-60, Swiss Finance Institute, Aug 2023. URL `https://ideas.repec.org/p/chf/rpseri/rp2360.html`.

Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollár, Kaiming He, and Ross B. Girshick. Benchmarking detection transfer learning with vision transformers. *CoRR*, abs/2111.11429, 2021. URL `https://arxiv.org/abs/2111.11429`.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. Datasets for large language models: A comprehensive survey, 2024. URL `https://arxiv.org/abs/2402.18041`.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. URL `https://arxiv.org/abs/1608.03983`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL `https://arxiv.org/abs/1711.05101`.

Andreas Neuhierl, Xiaoxiao Tang, Rasmus Tangsgaard Varneskov, and Guofu Zhou. Option characteristics as cross-sectional predictors. LawFin Working Paper 37, Frankfurt a. M., 2022. URL `https://hdl.handle.net/10419/261467`. urn:nbn:de:hebis:30:3-652441.

Ruben Wiedemann, Antoine Jacquier, and Lukas Gonon. Operator deep smoothing for implied volatility, 2025. URL `https://arxiv.org/abs/2406.11520`.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021. URL `https://arxiv.org/abs/2012.07436`.

A Data Preparation

A.1 Identification of Valid Assets

While the IV surfaces and realized volatilities are present in the OptionMetrics IvyDB dataset, we are also interested in predicting the future returns of an asset, which can be calculated using the daily returns in the CRSP (Center for Research in Security Prices) dataset. However, OptionMetrics uses the primary key `secid` and CRSP uses the primary key `cusip`. These keys do not have a one-to-one mapping because it is possible for assets to be delisted, to be added to the dataset during a calendar month, or to change primary keys (for example, due to company mergers or acquisitions). Because we batch data by month and year, we can construct a one-to-one mapping between `secid` and `cusip` using the WRDS link tables (`wrdsapps_link_crsp_optionm` and the `stocknames__v2` table, which provide the start and end dates during which each primary key is active. For each month of data we drop any rows where the `cusip` or `secid` is valid for only part of the month.

A.2 Data Collection

Using our table of valid primary keys, we downloaded raw data from OptionMetrics IvyDB, using the Volatility Surfaces and Realized Volatility tables, and the end-of-day return from the CSRP Stock dataset. The IV surface dataset contains smoothed, interpolated data on standardized calls and puts, with expirations of 10,30, 60, 91, 122, 152, 182, 273, 365, 547, and 730 calendar days, at deltas of 0.10, 0.15,0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75,0.80, 0.85, 0.90 (negative deltas for puts). We fuse this data on the primary key and date, producing a total of 120 parquet files (12 months each from 2012 to 2022).

Rows are dropped if there is missing data in the IV surface or invalid values in the CRSP stock price or return values, indicating assets that did not trade on a particular day. We construct the IV surface using all available δ values and days-to-expiry for both calls and puts. We do not attempt to filter for outliers in the IV surfaces or other data.

Our final dataset consists of 4,259,070 rows, distributed across months as shown in Figure 5.

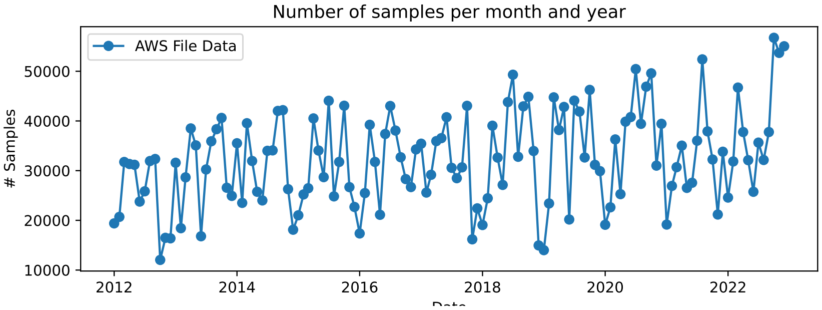

Figure 5: Number of samples per month of data.

B Model Definitions

Table 4 lists the parameters used to initialize the various model architectures.

The Vision Transformer architecture is taken from the standard PyTorch implementation[3], modified to accept tensors of size $\mathbb{R}^{1\times10\times36}$ instead of square RGB images.

All models considered take input of size 10×36, operate on 2×2 image patches, following the approach in Kelly et al. [2023], and produce a single real-valued prediction. Models consist of a Vision Transformer (ViT) followed by a 4-layer multilayer perceptron (MLP) to convert the ViT vector output into a single prediction.

Model	# ViT Layers	# Heads	# ViT hidden dim.	ViT MLP dim.	ViT Dropout	ViT output size	MLP hidden dim.
`ViT_0.005M_wide`	1	8	64	64	0.1	64	64
`ViT_0.12M_deep`	4	8	64	64	0.1	64	64
`ViT_0.17M_wide`	1	16	128	128	0.1	128	128
`ViT_0.5M_deep`	4	16	128	128	0.1	128	128
`ViT_0.5M_wide`	1	16	256	256	0.1	256	128
`ViT_1.7M`	4	16	256	256	0.1	256	128

Table 4: Summary of model parameters for all model sizes.

To conduct the MLP-only ablation experiment, we define the following alternate models, selected to match the parameter sizes of the ViT models.

- `MLP_0_12`: A four-layer MLP with input size = 360 (to match the flattened IV surface) and hidden size = 180.
- `MLP_0_17`: A four-layer MLP with input size = 360 and hidden size = 240.
- `MLP_0_5`: An eight-layer MLP with input size = 360 and hidden size = 350.

B.1 `torchinfo` summary of `MLP_0_5` model

This summary was generated with an input and hidden size of 360, matching the size used in the ablation test.

```
===========================================================================
Layer (type:depth-idx)                  Output Shape             Param #
===========================================================================
DeepMLP                                  [1, 1]                   --
Linear: 1-1                            [1, 360]                 129,960
BatchNorm1d: 1-2                       [1, 360]                 720
ReLU: 1-3                              [1, 360]                 --
Linear: 1-4                            [1, 360]                 129,960
BatchNorm1d: 1-5                       [1, 360]                 720
ReLU: 1-6                              [1, 360]                 --
Linear: 1-7                            [1, 360]                 129,960
BatchNorm1d: 1-8                       [1, 360]                 720
ReLU: 1-9                              [1, 360]                 --
Linear: 1-10                           [1, 180]                 64,980
BatchNorm1d: 1-11                      [1, 180]                 360
ReLU: 1-12                             [1, 180]                 --
Linear: 1-13                           [1, 180]                 32,580
BatchNorm1d: 1-14                      [1, 180]                 360
ReLU: 1-15                             [1, 180]                 --
Linear: 1-16                           [1, 90]                  16,290
BatchNorm1d: 1-17                      [1, 90]                  180
```

[3] https://docs.pytorch.org/vision/main/models/vision_transformer.html

```
ReLU: 1-18                             [1, 90]                    --
Linear: 1-19                           [1, 90]                    8,190
BatchNorm1d: 1-20                      [1, 90]                    180
ReLU: 1-21                             [1, 90]                    --
Linear: 1-22                           [1, 1]                     91
============================================================================
Total params: 515,251
Trainable params: 515,251
Non-trainable params: 0
Total mult-adds (Units.MEGABYTES): 0.52
============================================================================
```

B.2 `torchinfo` summary of all other MLP models

This summary was generated with an input size of 256 and a hidden size of 128.

```
============================================================================
Layer (type:depth-idx)                   Output Shape              Param #
============================================================================
SimpleMLP                                 [1, 1]                     --
Linear: 1-1                            [1, 128]                   32,896
BatchNorm1d: 1-2                       [1, 128]                   256
ReLU: 1-3                              [1, 128]                   --
Linear: 1-4                            [1, 128]                   16,512
BatchNorm1d: 1-5                       [1, 128]                   256
ReLU: 1-6                              [1, 128]                   --
Linear: 1-7                            [1, 64]                    8,256
BatchNorm1d: 1-8                       [1, 64]                    128
ReLU: 1-9                              [1, 64]                    --
Linear: 1-10                           [1, 1]                     65
============================================================================
Total params: 58,369
Trainable params: 58,369
Non-trainable params: 0
Total mult-adds (Units.MEGABYTES): 0.06
============================================================================
```

C Additional Results

Figure 6 shows the training trajectories for small and large ViT models, showing how many models achieve their full performance on one or two epochs.

We observe that while training on one year of data can produce good results, they are often inconsistent. Further, we observe the expected relationship between the number of model parameters and the data required to train the model; the smallest models perform best on small datasets, and the largest model requires the full dataset: 10 years of training data, with 1 year of test data.

Figure 7 shows the Huber loss, recorded at each batch, for some samples of 1, 4, and 10-year training runs. We observe a few loss spikes, which may be caused when the cosine learning rate increases, although in most cases the learning rate increase does not cause a spike.

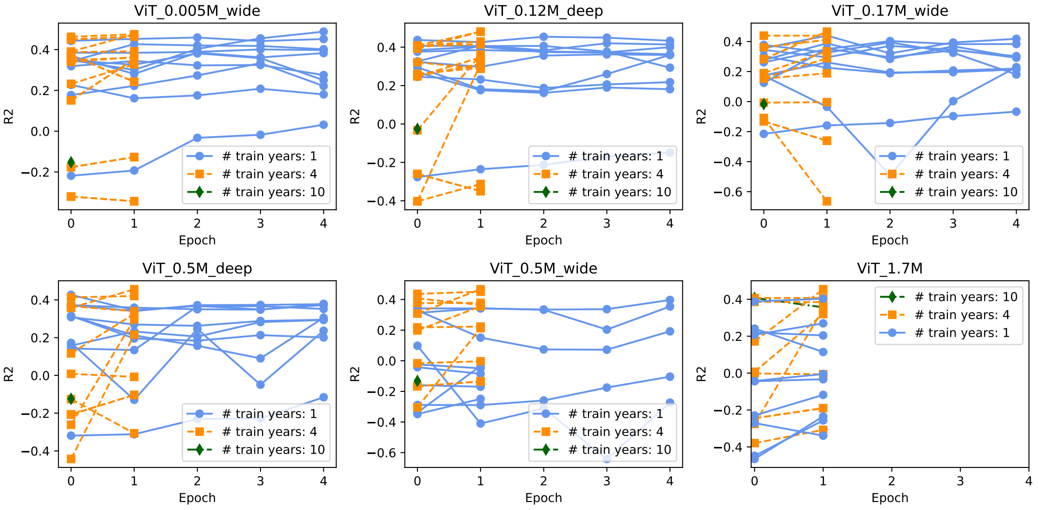

Figure 6: R^2 plotted over the number of training epochs, broken down by ViT model type. All of the train-on-one-year, test-on-one-year models exhibit poor performance when tested on 2020 data, corresponding to a single low or negative R^2 trajectory observed in each plot.

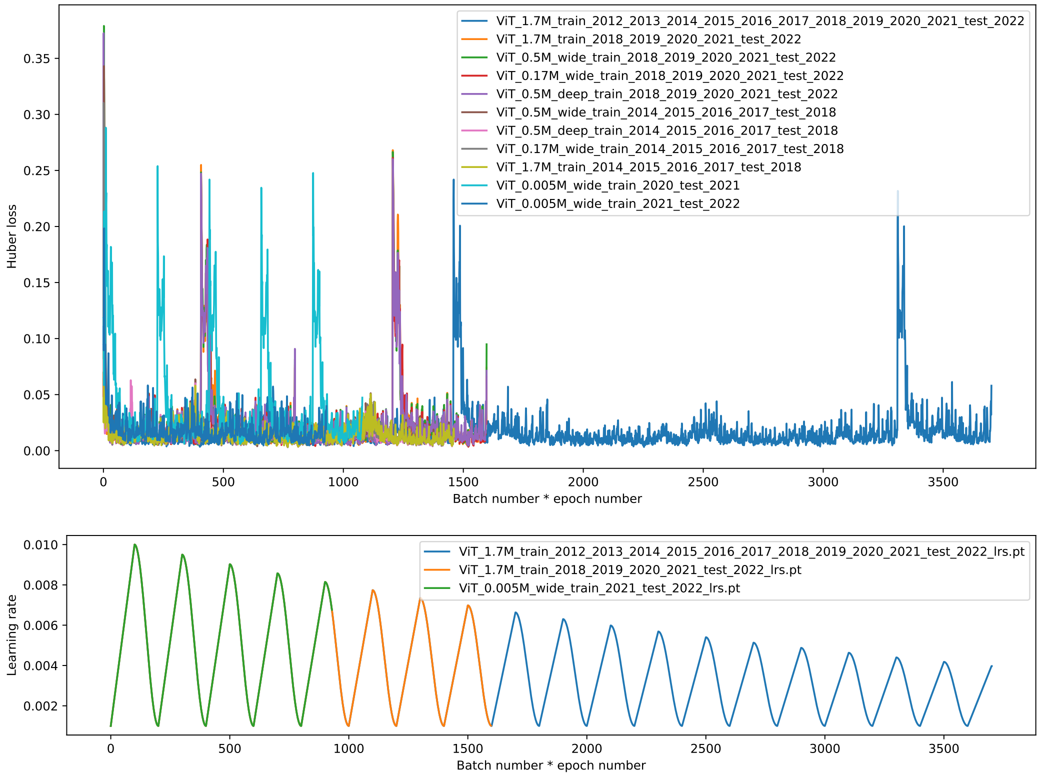

Figure 7: Huber loss and cosine learning rate for several sampled training runs.