Cross-Market Robustness of CNN + PPO for Multi-Stock Trading: Evidence from the United States, China, and India

Jizhi Wang

School of Economics, Huazhong University of Science and Technology, Wuhan, China

M202474992@hust.edu.cn

Abstract. This study examines the external validity of a convolutional neural network (CNN) feature extractor trained with Proximal Policy Optimization (PPO) for multi-stock daily trading across three equity universes: S&P50 (United States), SSE50 (China), and Nifty50 (India). This model was trained on 2023 data and evaluated for out-of-sample performance in 2024, using the same preprocessing, features, hyperparameters, and training protocol. It studied the sensitivity to lookback (10–50 trading days) with a window-expansion technique, and benchmark the CNN+PPO approach against a multilayer perceptron (MLP) baseline. The U.S. sample showed a clear advantage for CNN in shorter windows (particularly 20 days), as evidenced by increased risk-adjusted returns and more localized salience on recent channels. The magnitude of these gains is mixed on SSE50 and non-existent or negative on Nifty50. Complementary gradient saliency and permutation-importance diagnostics suggest that these cross-market differences in performance and saliency can be traced to a mismatch between the convolutional inductive bias and the presence or absence of short, cross-sectional motifs. Results suggest that architecture choice and deployment should be data-driven and informed by market diagnostics and feature engineering.

Keywords: deep reinforcement learning, PPO, convolutional neural networks, algorithmic trading, domain transfer, interpretability.

1 Introduction

Deep reinforcement learning (DRL) provides an end-to-end learning framework for extracting trading policies from historical market data by jointly learning representations and sequential decision rules [1]. Convolutional feature extractors have been proposed to exploit short-horizon spatio-temporal motifs in stacked "images" of prices and indicators [2][3]. And Proximal Policy Optimization (PPO) is a popular choice of policy-gradient optimizer in financial RL, due to empirical stability [4].

Prior research tends to test models within a single market. Exchanges vary in terms of liquidity, microstructure, and the underlying drivers of price movements, so a policy trained in one market is likely to fail if directly deployed to another without adaptation

X. Pan et al. (eds.), *Proceedings of the 2026 11th International Conference on Financial Innovation and Economic Development (ICFIED 2026)*, Advances in Economics, Business and Management Research 382,
https://doi.org/10.2991/978-94-6239-642-5_28

[5][6]. This paper offers a reproducible, cross-market evaluation of a CNN+PPO pipeline, along with interpretability diagnostics to diagnose transfer successes and failures across three structurally distinct markets.

Contributions. (1) A reproducible cross-market benchmark of CNN+PPO on S&P50, SSE50, and Nifty50 with identical pipelines; (2) paired gradient-saliency and permutation-importance analyses to link learned feature usage to performance; (3) practical diagnostics and deployment guidance for transferring DRL trading agents across markets.

2 Related Work

DRL for trading has evolved from shallow/tabular methods to deep actor/critic and critic-free portfolio-level approaches for multi-asset problems [7][8]. PPO is a popular optimizer in finance [4]. CNNs and hybrid architectures have been used for multivariate financial time series and may help when discriminative motifs are local [9][10]. However, convolutional inductive biases may harm transferability if the target domain lacks the same localized motifs.

Interpretability methods like gradient-based saliency and permutation importance offer complementary local and distributional perspectives on input usage [11][12][13]. These metrics are helpful as deployment gates in finance because they signal whether the model is using economically plausible and stable signals. Work on domain adaptation and meta-learning is relevant for transferability [14][15], but it falls beyond the empirical scope of this study.

3 Methods

3.1 Data and Markets

I evaluated three indices over 2023–2024:

United States (S&P50): Top 50 S&P constituents (train = 2023, test = 2024).

China (SSE50): SSE50 constituents, same split.

India (Nifty50): Nifty50 constituents, same split.

All pricing data were daily local-currency series. Chinese data were sourced from Tushare (using the Tushare Pro API), U.S./Indian data from Alpha Vantage (using the Alpha Vantage API). Tech indicators were computed using the usual libraries. All preprocessing was forward-fill with a zero fallback, to keep inputs contiguous.

3.2 Window Expansion

I trained and evaluated models using lookbacks of 10, 20, 30, 40, and 50 trading days to assess sensitivity to historical horizon. This mirrors the temporal-field hyperparameter approach in recent work [3].

3.3 Features

Input per instance: 50 tickers × 13 stock channels (Table 1) + 3 account features + 1 channel to fix shape. Specific features are shown in Table 1. The stacked input is treated as a 2-D image.

Table 1. 13 Stock Features

Feature name	Description
open	Opening price
high	Highest price
low	Lowest price
close	Closing price
vol	Trading volume
macd	MACD value
macd_signal	MACD signal line
macd_hist	MACD divergence histogram
boll_upper	Upper Bollinger Band
boll_lower	Lower Bollinger Band
dx	Directional Movement Index
rsi	Relative Strength Index
sma	Simple Moving Average

3.4 Trading Environment

I implemented a daily rebalancing MDP with discrete per-stock actions: sell 50% holdings, buy up to 50% of available cash, or hold. Initial capital: 10,000 in local currency. Transaction cost: 0.1% per share. Reward equals the daily change in account value.

3.5 Models and Training

Two feature extractors were compared:

CNN extractor: two convolutional layers + pooling + fully connected layer, as shown in Table 2.

MLP baseline: fully connected layers with the same output dimension to ensure parity in downstream policy capacity, as shown in Table 3.

Extractors feed a PPO policy (actor) trained with identical hyperparameters and random seed (SEED = 42) across markets and windows. Deterministic evaluation episodes used seeded offsets to ensure reproducibility. Implementation used common RL toolkits and the scientific Python stack.

Table 2. CNN Feature Extractor

Part	Layer	Output Shape
INPUT		[1,10,650]
	Conv_1	[64,10,650]
	BatchNorm2d	[64,10,650]
	ReLU	[64,10,650]
	MaxPool2d	[64,5,325]
CNN	Conv_2	[128,3,323]
	BatchNorm2d	[128,3,323]
	ReLU	[128,3,323]
	MaxPool2d	[128,1,161]
	Flatten	20608
	Linear	1024
FC	ReLU	1024
	Dropout	1024
	Linear	128

Table 3. MLP Feature Extractor

Part	Layer	Output Shape
INPUT		[1,10,650]
	Linear	1024
FC	ReLU	1024
	Dropout	1024
	Linear	128

4 Evaluation Metrics

For each run, I report: final account value, cumulative return, annualized return and volatility, annualized Sharpe ratio (zero risk-free rate for comparability), maximum drawdown (MD), and win rate (probability of positive daily returns). These standard metrics permit cross-market comparisons of risk-adjusted performance.

5 Experimental Results

Figures below present 2024 out-of-sample performance across windows for S&P50 (US), SSE50 (China), and Nifty50 (India). The summaries below report selected results considered most informative; full numeric tables are available upon request.

5.1 Summary--US (S&P50)

CNN yields positive returns, especially in 20 window days, as shown in Figure 1. 20-day CNN: final value 12,782 (cum. +27.8%), annualized return 33.1%, vol 19.37%, Sharpe 1.48, MD 12.92%, win rate 0.53. MLP 20-day: cum. +8.71%, Sharpe 0.77.

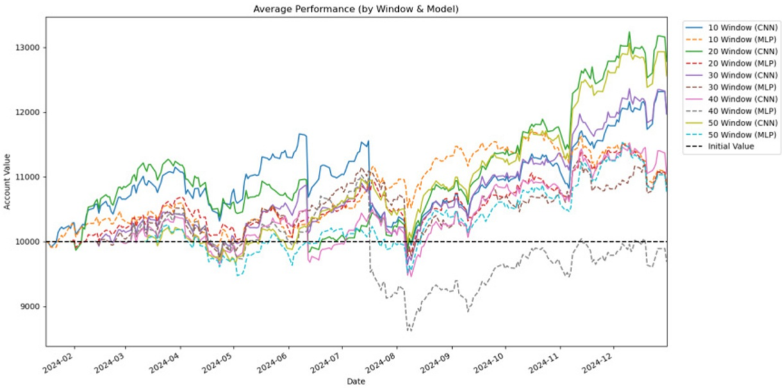


Fig. 1. Performance over time (US)

5.2 Summary--China (SSE50)

Both CNN and MLP produce positive returns, as shown in Figure 2. For example, 10-day CNN final value 15,536 (+55.36%, Sharpe 2.40); 20-day MLP sometimes exceeds CNN (e.g., 20-day MLP +61.28% vs. 20-day CNN +39.00%), indicating inconsistent architecture superiority.

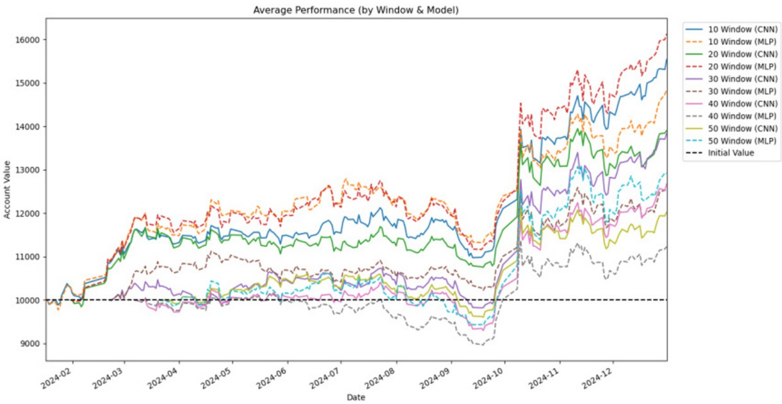


Fig. 2. Performance over time (China)

5.3 Summary--India (Nifty50)

CNN underperforms MLP in multiple windows, as shown in Figure 3. For example, 10-day CNN final value 9,895 (−1.05%), 30-day CNN +11.15% (Sharpe 0.84), while

MLP often posts small positive returns (e.g., 20-day MLP +12.29%, Sharpe 0.89). And CNN shows instability at longer windows (40–50 days).

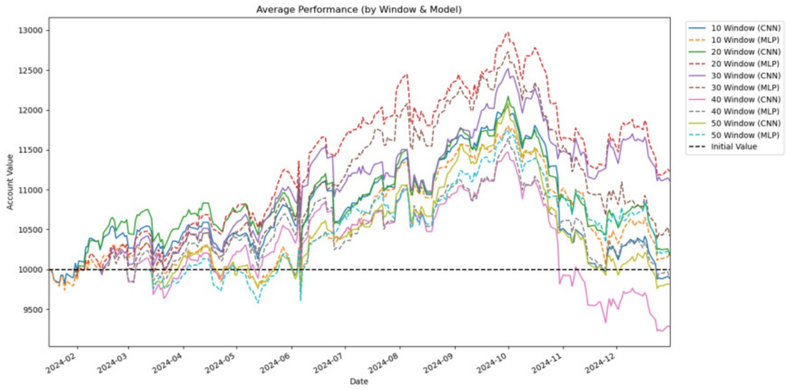


Fig. 3. Performance over time (India)

5.4 Summary

On S&P50, the CNN+PPO pipeline consistently produces superior risk-adjusted returns--particularly at short windows (notably 20 days). On SSE50, both extractors can perform well, but superiority is window-dependent. On Nifty50, CNN provides no systematic advantage and can underperform, especially at longer lookbacks.

6 Interpretability Analyses

To provide an intuition for cross-market performance gaps, this study have implemented two mutually-supporting diagnostics on the evaluation data (100 samples × 10 episodes × 20 repeats) for the 20 window days, as it performs well in Section 5:

Gradient saliency maps--absolute gradients of mean action logits w.r.t. inputs, averaged over samples to produce a (time × feature) sensitivity heatmap.

Permutation feature importance--for each time × feature cell, permute values across evaluation samples and record the mean change in the model output (mean absolute action logits).

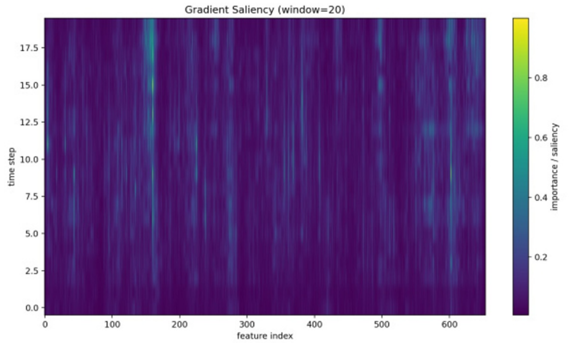
Both diagnostics were normalized for cross-panel visual comparison.

6.1 S&P50 (US)

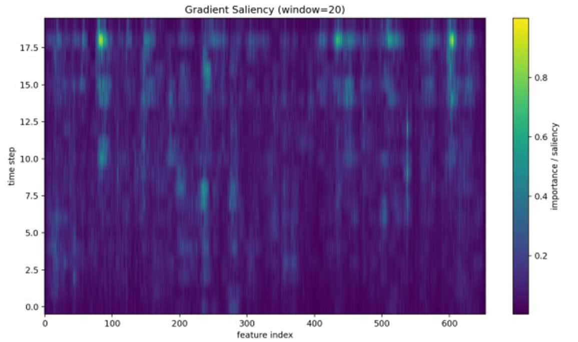
CNN saliency and permutation maps show concentrated sensitivity in the most recent 2–6 days (Figure 4a and 5a). The impact of permuting these cells on policy logits is large, showing that CNN depends on short, cross-sectional motifs. This concentration is the source of the CNN's advantage: convolutional filters can identify and pool spatially consistent short-horizon signals.

6.2 SSE50 (China)

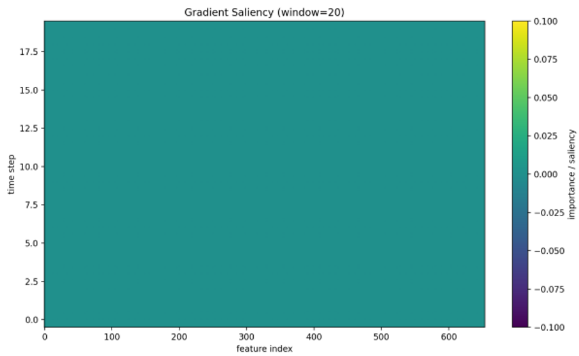
Saliency and permutation maps are more diffuse: importance is spread over time and features with weaker temporal concentration (Figure 4b and 5b). This implies a lower short-horizon SNR for different channels. Thus, CNN gains are smaller and less consistent across lookbacks.



a



b

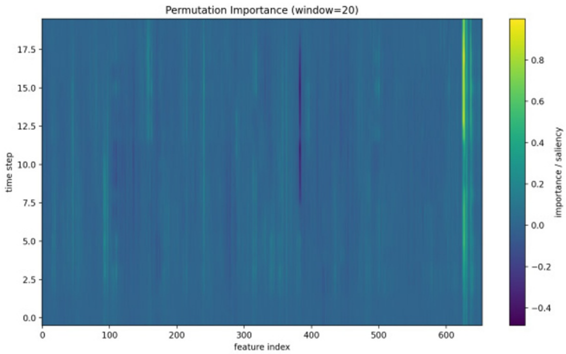


c

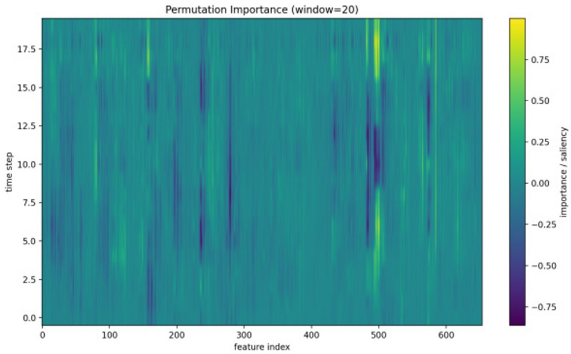
Fig. 4. (a-c). Gradient saliency maps (window = 20). Rows correspond to time steps within the 20-day input window (0 = oldest, 19 = most recent); columns index concatenated feature channels (features grouped by company). (a) S&P50 (b) SSE50 (c) Nifty50.

6.3 Nifty50 (India)

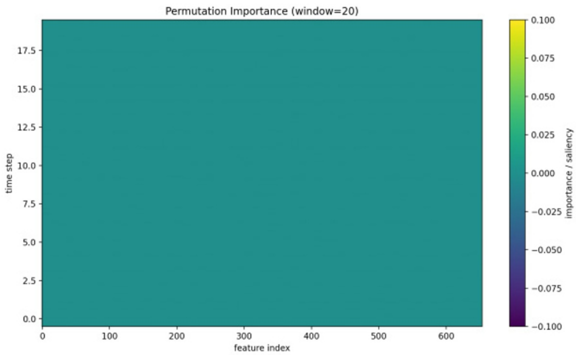
Maps are nearly flat at the depicted normalization: single time×feature perturbations cause little change in logits (Figures 4c and 5c). Short, cross-sectional motifs of the form exploited by the CNN are largely absent, which explains the model's poor transfer performance.



a



b

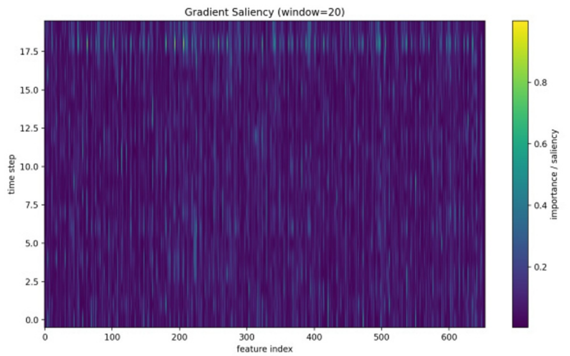


c

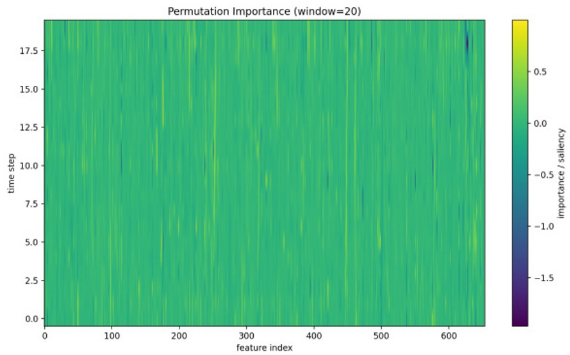
Fig. 5. (a-c). Permutation feature-importance heatmaps (window = 20). Positive values indicate that destroying that cell reduces the model output. (a) S&P50 (b) SSE50 (c) Nifty50.

6.4 MLP Comparison

MLP policies have flatter saliency and permutation profiles for all markets, in line with learning more distributed, longer-range, or aggregated relationships (Figure 6). MLPs outperform in markets with diffused signals, and CNNs outperform in markets with concentrated, short-horizon signals (S&P50).



a



b

Fig. 6. (a-b). MLP comparison on S&P50 (window = 20). (a) Gradient saliency averaged for the MLP policy on S&P50 (b) Permutation importance for the MLP policy on S&P50.

7 Discussion

7.1 Mechanism Linking Architecture to Outcome

The two diagnostics offer convergent evidence: CNNs are useful when markets display temporally local, cross-sectional structures amenable to pooling by convolutional filters. When no such structure exists, the CNN's inductive bias adds little benefit and can raise fragility (overfitting to spurious local patterns).

7.2 Practical Recommendations

Market validation: Always run tests and interpretability diagnostics on the target market before deployment.

Use interpretability as a gate: Require stable, economically plausible localized saliency + positive permutation impact for convolutional policies in the target market.

Match architecture to signal structure: If diagnostics show concentrated, short-horizon importance, favor CNNs or temporal convolutions; if importance is distributed, favor MLPs, linear models, or architectures designed for long-range dependencies.

Feature engineering matters: Markets where technical channels are less informative may require fundamentals, corporate actions, macro indicators, or order-book features to provide an exploitable signal.

7.3 Limitations

Interpretability diagnostics quantify sensitivity/dependence, not causal effect on realized profit; they do not replace execution-aware backtests (market impact, slippage, realistic execution). This research encompasses a recent timeframe (2023 for training and 2024 for testing) along with three indices; further temporal and cross-sectional replication studies are warranted.

7.4 Directions for Future Work

Domain adaptation, meta-learning, as well as hybrid architectures (convolutional front ends connected to attention or graph modules) also show good potential for transferability in the presence of related but noisy motifs [14][15][16].

8 Conclusion

Using identical preprocessing, features, and training procedures, CNN+PPO produced a marked advantage on S&P50--driven by concentrated short-horizon signals--but did not generalize reliably to SSE50 and Nifty50. Gradient saliency and permutation importance diagnostics explain these differences and provide practical checks for deployment. The main lesson is clear: single-market DRL results should not be assumed to generalize; market empirical tests and interpretability checks are essential, and architecture selection should be informed by the market's signal structure.

Acknowledgments

Thanks to the maintainers of the tools and data services used: Tushare and Alpha Vantage for data access, TA-Lib for indicators, and the open-source ML stack for reproducibility. I would also like to extend my gratitude to the professors for their invaluable suggestions pertaining to the experimental design. Portions of the manu-

script were language-polished with AI tools; scientific assertions and analyses are under the purview of the author's responsibility.

Author Note

The data and code required to replicate the experiments are accessible upon reasonable request. Portions of the manuscript were edited for clarity using an AI language model; all substantive intellectual contributions and interpretations are those of the author. The author declares no conflicts of interest.

References

1. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
2. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539.
3. Montazeri, S., Jumakhan, H., & Mirzaeinia, A. (2025). Finding optimal trading history in reinforcement learning for stock market trading. arXiv preprint arXiv:2502.12537. https://arxiv.org/abs/2502.12537.
4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. https://arxiv.org/abs/1707.06347.
5. Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. The Journal of Portfolio Management, 30(5), 15–29. https://doi.org/10.3905/jpm.2004.442611.
6. Hasbrouck, J. (2007). Empirical market microstructure: The institutions, economics, and econometrics of securities trading. Oxford University Press.
7. Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. IEEE Transactions on Neural Networks and Learning Systems, 28, 653–664. https://doi.org/10.1109/TNNLS.2016.2522401.
8. Zha, R., & Liu, B. (2025). A new DAPO algorithm for stock trading (arXiv:2505.06408). arXiv preprint arXiv:2505.06408. https://arxiv.org/abs/2505.06408.
9. Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Systems with Applications, 129, 273–285. https://doi.org/10.1016/j.eswa.2019.03.029.
10. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. Data Mining and Knowledge Discovery, 33, 917–963. https://doi.org/10.1007/s10618-019-00619-1.
11. Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034. https://arxiv.org/abs/1312.6034.
12. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.
13. Ismail, A., Gunady, A. A., Bravo, M. C., & Feizi, H. (2020). Benchmarking deep learning interpretability in time series predictions. arXiv preprint arXiv:2010.13924. https://arxiv.org/abs/2010.13924.

14. Cheng, M., Yang, J., Pan, T., Liu, Q., & Li, Z. (2024). ConvTimeNet: A deep hierarchical fully convolutional model for multivariate time series analysis. arXiv preprint arXiv:2403.01493. https://arxiv.org/abs/2403.01493.
15. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning (ICML) (pp. 1126–1135). Proceedings of Machine Learning Research, Vol. 70. https://proceedings.mlr.press/v70/finn17a.html.
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 5998–6008). https://arxiv.org/abs/1706.03762.