**Optimising Sentiment-based Approaches to News Analysis for Stock Prediction**

Vivian Gao

Shanghai Jiao Tong University

*Exchange Student*

*The University of Melbourne*

917030990021

Harry Lording

Shanghai Jiao Tong University

*Exchange Student*

*The University of New South Wales*

917030990012

1. Introduction
   1. **Background**

The complexity of stock market dynamics is representative of the volatility or stability of financial markets, and these movements are continuously influenced by multitudes of factors. These vary from financial reports; historical data; political and economic influences; and general market sentiment. Developments in predictive models serve as cornerstones for traders to gain an edge on each other however, fine-tuning existing algorithms and adapting these to ever-changing factors is fundamental to effective modelling.

The flaw with earlier studies on stock market predictions is that they are solely based on historical stock prices, a factor that has been debunked by later research. There is an underlying unpredictability as the market reacts in time to incoming information reflective in a random walk pattern and cannot be predicted with more than 50% accuracy when only taking historical prices into account[[1]](#endnote-1). As stock market prices are largely fluctuating, the efficient market hypothesis (EMH) states that financial market movements depend on news, current events and product releases and all these factors will have a significant impact on a company’s stock value[[2]](#endnote-2).

With the advent of the information age and the shortening length of the news cycle, accurately capturing this inflow of information into a model predictive of the market has become a key area of research. A whole industry has been formed around financial market sentiment detection[[3]](#endnote-3). As a result, many investment banks and hedge funds are trying to exploit the sentiments of investors to help make better predictions about the financial market.

Our response to this task is to build upon a proposed Hybrid Attention Networks (HAN) model[[4]](#endnote-4) with advancements in sentiment analysis through BERT[[5]](#endnote-5) and LSTM[[6]](#endnote-6) to predict the stock trend based on the sequence of recent related news. To validate the effectiveness of our approach, in this paper, we performed extensive experiments on real-world data. Compared to traditional approaches, the experiment results show that our proposed framework can increase the accuracy of stock prediction.

* 1. **Motivation**

Alongside advances in machine learning is their implementation within stock market analysis, with natural language processing and the use of deep learning being forefronts of research. The two key challenges to these models is primarily, its accuracy and thus, utility. Not only will the algorithm have to be accurate enough for implementation, the output of this program will have to provide value to traders upon existing methods of analysis. Secondly, these algorithms must have the capacity to react to real-time information, before the market, to make effective, and profitable predictions.

We have identified three key shortcomings of existing stock prediction models. The first, as highlighted previously, is their focus on purely historical data and thus, limited to only predictions from trends. The lack of responsiveness amongst these predictions limit their use to traders as they race to beat each other and thus, the market. Those that do implement sentiment analysis do so in a naive sense. To be able to capture the sentiment of news and react as the market does, it is key to analyse this information in a more human-like process. With this in mind, the input into our model will aim to not only capture the context of change in sentiments but also, be able to analyse the quality of this information.

Our goal through this project is to build upon existing implementations of advancements in machine learning to create a model for predicting the price of stocks that extrapolates from historical data whilst being highly responsive to news headlines through sentiment analysis.

1. Context and Related Work

The ability to predict price movements on the financial market would offer a lucrative competitive edge over other market participants. Therefore, it is not surprising that this topic has attracted much attention from both academic researchers and industrial practitioners.

Technical analysis deals with the time-series historic market data, such as trading price and volume, and make predictions based on that. Traditionally, the most widely used model in this direction is the Autoregressive (AR) model for linear and stationary time-series[[7]](#endnote-7). However, the non-linear and non-stationary nature of stock prices limits the applicability of AR models.

There are increasing research efforts focussed on exploiting advancements in deep neural networks to identify similar patterns. We have emphasised the research of Hu et al. (2018)4 to mine news sequence directly from text with hierarchical attention mechanisms for stock trend prediction. This work was able to address the issues regarding contextualising language data by considering the broader scope of the information and input.

When compared to the results of other leading models, the HAN framework by Hu achieved the greatest accuracy. However, we identified their use of word2vec as the word embedding layer a shortcoming of their research. According to the evaluation of natural language processing techniques, the embeddings of word2vec fell short when compared to paragraph vectors offered by doc2vec for capturing the sentiment of the broader text.[[8]](#endnote-8) More so, Google’s 2018 pre-trained language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers, offers cutting-edge language understanding5.

Recent advances in deep learning have brought a new wave of methods to this field where the Long Short-Term Memory (LSTM) recurrent neural network has been shown to be very effective[[9]](#endnote-9). When compared to the GRU model of Hu et al., the LSTM has a better return ratio indicating that it predicts trends better and has a more robust strategy but has lower certainty of prediction than LSTM’s.[[10]](#endnote-10)

Moving forwards, we will be experimenting with these proposals to determine the impact of these various machine learning methods towards the task of stock price prediction to provide a more robust and responsive framework.

1. Problem Formulation

We determine the problem of stock trend prediction as a classification problem and as such, have devised three classes to represent the movement of stock price. For a given date *t* and a pre-selected stock *s*, we can calculate its movement using the metric of its percentage of change by:

The three classes are as follows: *RISE*, *FALL*, and *STABLE,* representing upwards, downwards and insignificant movements in stock prices. For each date , our key input will a set of news articles of size where the set contains . Thus, our goal is as follows: when given a time sequence *N*, the stocks across that time *s* and a date *t*, we intent on taking the news input from that time sequence ( to ) which we will denote as to predict whether the stock price will fall into the classes of *RISE*, *FALL* or *STABLE*.

Due to the Hybrid nature of our attention model with two distinct attention mechanisms on a daily and contextual model, the frame of change is within a day by day basis. In future, to further improve the utility of this model for traders, improving the responsiveness to an hourly basis would capitalise on opportunities for greater returns.

1. Proposed Method
   1. **Hybrid Attention Network**

Based off of leading frameworks for stock prediction, our baseline model is the Hybrid Attention Network (HAN) by Hu et al.4 in 2018. We were able to review the analysis of machine-learning based stock prediction models by Xu et al[[11]](#endnote-11) with a summary of their performance in table 1.

Table 1. Comparison of Different Model Performances

|  |  |  |
| --- | --- | --- |
| **Models** | **Acc.** | **MCCC** |
| Rand | 50.89 | -0.002266 |
| ARIMA[[12]](#endnote-12) | 51.39 | -0.020588 |
| RandForest[[13]](#endnote-13) | 53.08 | 0.012929 |
| TSLDA[[14]](#endnote-14) | 54.07 | **0.065382** |
| HAN4 | **57.64** | 0.051800 |

The figures given are derived from their experimental environment and the second metric is the Matthews Correlation Coefficient (MCC)[[15]](#endnote-15) to avoid bias due to data skew. When given the confusion matrix classified as true positive, false positive, true negative and false negative, the MCC is calculated as follows:

We will briefly introduce the structure of Hybrid Attention Model from Hu et al. for baseline of our optimisation experiments. Beyond the higher accuracy scores, the model was the most effective at contextualising the news corpus and its impact and contributes to our project motivations.

A screenshot of a cell phone

Description automatically generated

Fig 1. Framework of the Hybrid Attention Network (HAN)

**News Embedding:** given the input of a news corpus sequence, a news embedding layer encodes each news into a news vector .

**News-Level Attention:** assigns an attention value to each news vector in a date and calculate the weighted mean of these news vectors as a corpus vector for this date.

**Sequential Modelling:** these corpus vectors are encoded through two variations of recurrent neural networks, either a bi-directional Gated Recurrent Units (GRU) or the Long Short-Term Memory (LSTM).

**Temporal Attention:** another temporal attention layer assigns an attention value to each date and calculate the weighted mean of these encoded corpus vectors to represent the overall sequential context information.

**Trend Prediction:** the classification is made by a discriminative network through a standard Multi-layer Perceptron (MLP), which takes as input and produces the three-class classification of the future stock trend.

* 1. **Natural Language Processing Techniques**

In order to improve upon the existing model, we intend on experimenting with a number of alterations. In the original study by Hu et al.4, an unsupervised word2vec was used as the word embedding layer in the network to reduce the complexity of the framework.

While members of the theory community have claimed word2vec should result in better performance than doc2vec, the financial research community have supported the use of doc2vec. More so, Google has recently released their paper on a new NLP technique, BERT5, pre-trained with greater understanding that previous models.

To supplement the utility of a stock prediction program, we have decided to take these suggestions onboard. Through comprehensive studies we will aim to prove which word embedding approach achieves better performance in the context of stock predictions.

* + 1. ***Word2Vec***

Word2vec [[16]](#endnote-16) is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. Each word is represented by a vector which is concatenated or averaged with other word vectors in a context, and the resulting vector is used to predict other words in the context. Hu et al. trained a word2vec on the vocabulary of their articles and then they computed the vector mean of those words to make a vector representation. However, upon further research, we concluded that word2vec would be unsuitable towards our project and opted for doc2vec and BERT as our NLP methods of choice.

* + 1. ***Doc2Vec***

Doc2vec[[17]](#endnote-17) model is based on word2vec, with only adding another vector (paragraph ID) to the input. Doc2vec is a Paragraph Vector, an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents Due to the natural of vectorizing language, the qualitative difference between doc2vec and word2vec document embeddings, remains unclear. However, each word in word2vec preserves its own semantic meaning. Thus, summing up all the vectors or averaging them will result in a vector which could have all the semantics preserved. However, averaging the words vectors loses the order of words, making it very similar to the concept of Bag of Words From this, we opted to use the paragraph vector variation for our experimentation.

* + 1. ***BERT***

BERT5 (Bidirectional Encoder Representations from Transformers) is a pre-trained NLP model that generates embeddings for a word based on the context it appears thus generating slightly different embeddings for each of its occurrence. This new model enables NLP models to better disambiguate between the correct sense of a given word providing more accurate representation of the semantic of language.

BERT is designed to pretrain deep bidirectional representations from unlabelled text by jointly conditioning on both left and right context in all layers. The revolutionary contribution of the BERT model is their inspiration from transformers - a sequence model that dispenses convolutions and recurrence. Instead, it uses attention to incorporate sequential information into sequence representation. As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications.

* 1. **Recurrent Neural Network Architecture**

To capture temporal context effects of news data, a recurrent neural network (RNN) was used. Due to the temporal nature of RNNs, they suffer from vanishing gradient. Currently there are two popular variations of RNNs that solve the vanishing gradient problem, Gated Recurrent Unit (GRU) and Long-Short Term Memory (LSTM). We will aim to compare the performance of using GRU and LSTM in the HAN framework.

A close up of a clock

Description automatically generated

Fig 2. Comparison of LSTM (left) and gated recurrent units (right).

, and are the input, forget and output gates, respectively. and denote the memory cell and the new memory cell content. and are the reset and update gates, and and are the activation and the candidate activation.

The original HAN framework opted for the use of Gated Recurrent Unit. Whilst similar, each approach each has their benefits. Having four gates (update, input, forget, and output) compared to the two gates that GRU uses (update and reset), the LSTM model in theory gives more control of the network in comparison to GRU[[18]](#endnote-18). Comparatively GRU doesn’t require maintenance of a memory unit and in many situations its performance has been relative to LSTM, while being simpler, computationally more efficient and faster to train.

* + 1. ***Long SHORT-TERM Memory***

The Recurrent Neural Networks (RNNs) architecture handles variable-length sequential input by way of a recurrent, shared hidden state. However, they were mostly impractical due to the vanishing and exploding gradient problems during training until the introduction of the long-short term memory (LSTM) recurrent unit. The LSTM does so via input, forget, and output gates; the input gate regulates how much of the new cell state to keep, the forget gate regulates how much of the existing memory to forget, and the output gate regulates how much of the cell state should be exposed to the next layers of the network.

LSTMs control the exposure of memory content (cell state) while GRUs expose the entire cell state to other units in the network. The LSTM unit has separate input and forget gates, while the GRU performs both of these operations together via its reset gate. However, LSTMs should in theory remember longer sequences than GRUs and outperform them in tasks requiring modelling long-distance relations[[19]](#endnote-19).

* + 1. ***Gated Recurrent Units***

The GRU[[20]](#endnote-20) is the newer generation of Recurrent Neural networks and is pretty similar to an LSTM. GRU’s removed the previous cell state and used the hidden state to transfer information. It also only has two gates, a reset gate and update gate. As GRU’s have fewer tensor operations, they are a little speedier to train then LSTM’s. There isn’t a clear winner which one is better. However, GRU use less training parameters and therefore use less memory, execute faster and train faster than LSTM's whereas LSTM is more accurate on dataset using longer sequences.

In order to capture the information from the past and future of a news as its context, we concatenate the latent vectors to encode the temporal sequence of corpus vectors. The study from Hu et al.4 utilises a GRU with a bidirectional setting, which can utilize the information from both past and future for prediction.

1. Experiment
   1. **Data Creation Process**
      1. ***Data Collection***

We chose to extract all news articles from the Wall Street Journal archive for the year 2015, using the BeautifulSoup Python package for parsing HTML and XML documents. In total over 40,000 articles were pulled from WSJ archives. The code for this process is located in the file data\_process.py. We opted for the WSJ for our experiment due to its scope of subject and global nature. The articles were scraped into plain text files that were placed into folders within their respective dates and labelled on a day-to-day basis. Not only would information regarding the company’s stock have a significant impact but also information about products and services and their broader sphere of influence has the potential to influence their stock price and are worth studying.

Regarding stock price data, all stock opening prices for companies of interested between the 1st jan 2015 to the 31st of December were recorded. Stock prices were downloaded from financial web resources as csv files.

* + 1. ***Data Pre-processing***

Preprocessing data involved organising the stock prices and news articles in a format in which connecting stock price movements with the corresponding news articles in the forecoming days before the price movement was a simple process. This involved numerous steps, outlined in the following paragraphs.

The first step in this process was to change the dating format used to organise the folders containing the scrapped news articles. Folders were changed from a standard date format to a four digit naming format. For example the number 0000 represented the first day scrapped, 0001 the second day and so forth. This format was used to complement how the stock data was numbered, with each stock price being donated a number instead of date as a label, 0 being the number for the stock price on the 1st of January.

The files were also renamed to make them easier to identify. At this point in the process each news article scrapped has its headline as its file name. Each article was renamed to the name of the folder it is located in, and in additional given an identification number for that day. For example an article could be named 0000\_1.txt which would donate the name of the first article for the day 1st of January 2015. As for the file 0010\_5.txt, it would represent the 5th article for the day of the 11th of January and so on.

The next step performed was to track which news articles mentioned which companies. Each article was scanned for a mentions of each company. To store this information a dictionary was used which was later picklizer (saved) for later use. Each company’s dictionary had keys for each respective day, and the values for each key was a list of article for which the company’s name was mentioned for that given day.

Processing the stock value data followed a similar process as outlined with the news articles. As stated before stock prices were gathered for the year 2015, and were saved as csv folders. To make the training and test sets and easier process, dictionaries were created with keys corresponding with the same day code format used to arrange the news article (0000 corresponds to the 1st January etc.). The values in the dictionaries were the movements of the price from the given day to the subsequent day.

Lastly the given natural language processing algorithm was trained. The Gensim framework was used create Doc2Vec and Word2Vec

* + 1. ***Dataset Creation***

Dataset creation refers to the process of turning unrelated dictionaries of stock movements and news articles into training and test sets. This process mostly concerned identifying periods where there was news activity for each company and matching that period with the corresponding stock movements. Once these periods of interest were identified, the 10 days prior news articles related to that company were gathered and vectorised using one of the natural language processing algorithms.

A 4 dimensional numpy array matrix was used as a input for our model. The 1st dimension of the matrix corresponds to the days for which we have articles regarding the company during the last 10 days. The 2nd dimension is the window, number of pasted days that we used to make the prediction. The 3rd dimension is the label for the articles that appear on the given day. The 4th dimension is the vector representing the news article.

Once the numpy matrixs were created for a given 10 day period they were matched with the corresponding stock movement to form x and y pairs for our model. These sets of matrixes and movements were then saved as as numpy files for future use in training and testing.

* 1. **Model**

To build the model keras and tensorflow frameworks were used. Such as in the reaserch paper, the HAN model composed of an attention layer, modelling news level attention, a second RNN layer, and lastly another attention layer encoding temporal attention.

* 1. **Word Embedding Comparison**

While initially an important goal of the project, a comparison between different word embedding techniques became irrelevant. In order for a natural language processing technique to be of use for the HAN model a paragraph vector was required as input. Doc2Vec is essentially an extension of Word2Vec in this regard, Doc2Vec is Word2Vec but adds the singular word vectors into a larger paragraph vector. As for BERT, at the time of doing this report the use of BERT for embedding sentence or paragraph was not developed, with most BERT models being used for word vector embedding. Some recents developments have been that of SentenceBert that uses sentence transformers to embed sentences, however this model did not integrate well with our model.

* 1. **Sequential Modelling Comparison**

Tests were set up between two RNNs architectures, GRU and LSTM. All independent factors were eliminated to ensure that a fair comparison was made between the two networks. This meant that both models were trained and tested under the same training and test sets, under the same NPL algorithm.

1. Evaluation

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Above is the results from the comparative testing between the HAN model implemented with the two different recurrent networks. Though similar in purpose, the results between the two networks are stark. While not perfect in its accuracy, which is too be expected when predicting the stock market, the GRU network does a good job at predicting trends. With each chart, there are distinct moments where it is clearly apparent the model is precieving the potential changes in the stock price, albeit with slight variations in timing.

In comparison, there are obviously issues regarding the use of a LSTM structure in the network. During the training of the network it is apparent that the network has become bias to predicting that the stock movement would not change. Interestingly the network also only predicted downward movements in the stock when it decided to deviate from its bias. However, when it did predict a downward movement in stock price the prediction was usually reasonable accurate, with a real downward movement occurring around the same time as the prediction.

It is clear for the purposes of the stock market prediction using a HAN network structure that GRU would be a better option. It is known that LSTM is more accurate for data in which sequences are long. Meanwhile GRU being the simpler framework converges faster than LSTM. It is likely that the sequence of data stretch of 10 day period was not long enough for LSTM to have an advantage over GRU. Given that a article of news over 10 days removded from the given day is not very likely to have any remaining impact on the price of a company stock, it is has led us to believe that the benefits of GRU outway LSTM.

1. Conclusion

Overall the HAN model performed very well. Under the correct hypertuning the model achieved upwards of 60% accuracy when trying to predict the stock market. While compared to other fields in artificial intelligence this may appear as a low rate of success, however with regards to the highly volatile nature of the stock market the model performed above expectations.

It is also clear that the choice of which RNN to implemented within the HAN model will also make a material difference. Between the GRU and LSTM architectures, the GRU significantly outformed the LSTM structure.

Future areas of research could go a number of different but equally interesting directions. One promising area of study is combining the exisiting HAN framework with technical indicators among other additional features. This could provide the network with a greater amount of relevant information that it could use to better contextual new corpus. Additionally, with many advances in NPL, among them being the development of BERT, the HAN network could see another potential improvement. As stated before a BERT model that could similarly embed paragraphs such as Doc2Vec could potentially be highly beneficial. This could be taken even further with the development of BERT models that are trained specifically for a financial news corpus.

1. References

   Qian, Bo, and Khaled Rasheed. "Stock market prediction

   with multiple classifiers." *Applied Intelligence* 26, no. 1 (2007): 25-33. [↑](#endnote-ref-1)
2. Fama, Eugene F. "The behavior of stock-market prices." *The journal of Business* 38, no. 1 (1965): 34-105. [↑](#endnote-ref-2)
3. Xing, Frank Z., Erik Cambria, and Roy E. Welsch. "Natural language based financial forecasting: a survey." *Artificial Intelligence Review* 50, no. 1 (2018): 49-73. [↑](#endnote-ref-3)
4. Hu, Ziniu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction." In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 261-269. ACM, 2018. [↑](#endnote-ref-4)
5. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv*:1810.04805 (2018). [↑](#endnote-ref-5)
6. Huynh, Huy D., L. Minh Dang, and Duc Duong. "A new model for stock price movements prediction using deep neural network." In *Proceedings of the Eighth International Symposium on Information and Communication*

   *Technology,* pp. 57-62. ACM, 2017. [↑](#endnote-ref-6)
7. L. Li, S. Leng, J. Yang, and M. Yu. Stock market autoregressive dynamics: A multinational comparative study with quatile regression. In *Mathematical Problems in Engineering*, 2016. [↑](#endnote-ref-7)
8. Lau, Jey Han, and Timothy Baldwin. "An empirical evaluation of doc2vec with practical insights into document embedding generation." *arXiv preprint arXiv:1607.05368* (2016). [↑](#endnote-ref-8)
9. Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. In *Big Data (Big Data), 2015 IEEE International Conference on. IEEE*, 2823–2824. [↑](#endnote-ref-9)
10. Sethia, Akhil, and Purva Raut. "Application of LSTM, GRU and ICA for stock price prediction." In *Information and Communication Technology for Intelligent Systems*, pp. 479-487. Springer, Singapore, 2019. [↑](#endnote-ref-10)
11. Xu, Yumo, and Shay B. Cohen. "Stock movement prediction from tweets and historical prices." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pp. 1970-1979. 2018. [↑](#endnote-ref-11)
12. Brown, Robert Goodell*. Smoothing, forecasting and prediction of discrete time series*. Courier Corporation, 2004. [↑](#endnote-ref-12)
13. Pagolu, Venkata Sasank, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. "Sentiment analysis of Twitter data for predicting stock market movements." In *2016 international conference on signal processing, communication, power and embedded system (SCOPES)*, pp. 1345-1350. IEEE, 2016. [↑](#endnote-ref-13)
14. Nguyen, Thien Hai, and Kiyoaki Shirai. "Topic modeling based sentiment analysis on social media for stock market prediction." In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1354-1364. 2015. [↑](#endnote-ref-14)
15. Matthews, Brian W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." *Biochimica et Biophysica Acta (BBA)-Protein Structure 405, no. 2* (1975): 442-451. [↑](#endnote-ref-15)
16. Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013). [↑](#endnote-ref-16)
17. Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." In *International conference on machine learning*, pp. 1188-1196. 2014. [↑](#endnote-ref-17)
18. Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014). [↑](#endnote-ref-18)
19. Yin, Wenpeng, Katharina Kann, Mo Yu, and Hinrich Schütze. "Comparative study of CNN and RNN for natural language processing." *arXiv preprint arXiv:1702.01923* (2017). [↑](#endnote-ref-19)
20. Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the properties of neural machine translation: Encoder-decoder approaches." *arXiv preprint arXiv:1409*.1259 (2014). [↑](#endnote-ref-20)