[1] **Hemanth Volikatla**

Reinforcement Learning Approaches for Exploring Discrepancy Variations: Implications for Supply Chain Optimization

Abstract: - Combinatorics discrepancy theory studies the ways in which a given state deviates from the ideal. A set system's disparity in its classical form has been shown to be bound in certain situations. Though they may not hold for other types of discrepancy, no mathematician has been able to produce an example that defies these limits as of yet. Due to the large number of set systems (kmn for m sets and n elements, each choosing one of k possible values), it is not viable to use traditional techniques, such as human reasoning or computer brute forcing, to identify the few set systems with a significant discrepancy. However, in a 2021 preprint, Adam Zsolt Wagner shows how the deep cross-entropy method—a popular Neural Network (NNs)-based Reinforcement-Learning (RL) technique—could effectively find examples that disprove open conjectures in two other combinatorial subfields: pattern avoidance and graph theory. Therefore, in light of these encouraging findings, we wondered if Wagner's method could help us find examples that defy the upper bound of discrepancy variations. To start answering this question, we examined whether Wagner's approach might be extended to find set systems that exhibit significant inconsistencies, either as classical or variant (prefix and fractional in our instance) systems. Our findings could help approximation algorithms, which employ discrepancy theory to tackle certain problems, as well as discrepancy theory itself.

Keywords: Combinatorics, Set Systems, Supply Chain, Wagner's Method

1. Introduction

Within the field of combinatorics, discrepancy theory explores the differences that exist between a real-world situation and the ideal condition that an individual would like to attain. The investigation of deviations from set systems, particularly when quantified in classical forms, is at the core of discrepancy theory. These classical forms, which often adhere to certain bounds, provide a mathematical problem that has not yet been completely solved, especially when taking into account variations of discrepancies such as classical, prefix, and fractional discrepancies. The volume of set systems in this sector is enormous and grows exponentially with the number of sets and elements involved. As a result, traditional techniques such as manual analysis or brute-force calculation are not sufficient to detect instances with large disagreement. Particularly in the aftermath of advances in Artificial Intelligence (AI) and Machine Learning (ML), the fascination with discrepancy theory and its difficulties has prompted a move towards novel computational techniques [1]–[3]. Such as Reinforcement Learning (RL) is a potentially fruitful approach that has gained recognition for its achievements in a number of fields, including financial market research, vehicle navigation, and even conquering difficult strategic games like chess and Go [4]–[16]. These accomplishments demonstrate how RL may be used to handle intricate, strategic decision-making processes, much like combinatorial decision-making in discrepancy theory. Adam Zsolt Wagner's groundbreaking work [17], presented in his 2021 preprint, illustrated the effectiveness of the deep cross-entropy method—a RL technique—by using it to find counterexamples in graph theory. This accomplishment opens the door to further research into the possibility of using comparable methods to discrepancy theory in an effort to find high discrepancy set systems that may defy current theoretical limitations. Wagner's deep cross-entropy approach is being modified for use in discrepancy theory as part of our study, with an emphasis on finding set systems that exhibit high levels of discrepancy. In order to identify the set systems with the largest discrepancy values, the methodology combines combinatorial analysis and ML algorithms to sort through the enormous number of potential set systems. This technique is promising not only for improving our knowledge of discrepancy theory but also for applying it to adjacent areas where discrepancy measurements are important, such as approximation algorithms. Through the implementation of this research, we hope to improve the deep cross-entropy method's accuracy and computational efficiency while adjusting it to the intricate needs of discrepancy analysis. This means creating specialised Neural Networks (NNs) models for fractional combinatorial analysis, enhancing signal creation, and optimising the process both algorithmically and at the machine level. Early experiments have produced promising results: the NNs is able to identify set systems with high levels of discrepancy in fractional, prefix, and classical categories. In addition, new directions in combinatorial analysis could be triggered by cross-disciplinary innovation that draws inspiration from disciplines like physics or chemistry. Successfully integrating discrepancy theory with RL techniques has significant ramifications. Such developments could transform computational tactics in many sectors that depend on discrepancy measures, beyond the mathematical satisfaction of finding high-discrepancy set systems. This includes improving the computational tools used in scientific and technical applications as well as creating approximation methods that are more effective.

[1] Independent Researcher, Alpharetta, Georgia, USA, Email: hemanthvolikatla@gmail.com

The paper is as follows; in the following section, we'll look at the study's backdrop. The related works are presented in Section 3. The pre and post methodology are provided in Sections 4 and 5, respectively. The experimental analysis is done in Section 6, and we wrap up the paper with some conclusions and ideas for future research in Section 7.

2. Background

A branch of combinatorics called discrepancy theory studies the difference between desired and actual states in set systems. According to this theory, the amount of imbalance that can occur when elements are not distributed equally across overlapping sets is measured as discrepancy. Lower and upper limitations have been determined for classical discrepancy; however, it is still unclear whether these bounds apply to varieties of discrepancy such as fractional and prefix. The goal of classical discrepancy analysis is to find the absolute difference between the counts of red and blue elements in each set of *m* sets with *n* elements, all of which are coloured either red or blue. To portray the disparity of the set system, the goal is to identify the colouring that minimises this disagreement across all sets. This study addresses the use of a NNs method to detect high-discrepancy set systems in both classical and variant forms. The method is influenced by [17] research on the deep cross-entropy method. The objective is to investigate if set systems that beyond established discrepancy bounds can be found using such computational techniques, thus refuting current hypotheses. The concept of discrepancy goes beyond its traditional definition to include variations such as fractional and prefix discrepancies, which bring new challenges and avenues for study. Fractional discrepancy provides a richer mathematical landscape for analysis by allowing set elements to be fractionally included. The computation of discrepancy is further complicated by prefix discrepancy, which adds a maximisation step across subsets of items. Therefore, these variants raise unanswered problems, especially about the applicability of the traditional discrepancy bounds. Spencer's theorem[2], in particular, offers an upper constraint on discrepancies for set systems, contingent on the number of sets and elements; however, its validity with respect to fractional and prefix discrepancies is still questionable. The discrepancy problem is modelled in this study using NNs, more precisely via Multi Layer Perceptrons (MLPs), as function approximators. These networks adjust the weights of their input vectors to approximate the function mapping to the relevant disparities through layers of linear and nonlinear transformations (activation functions). In the framework of discrepancy theory, we modify [17] deep cross-entropy approach, which was first used in graph theory. The approach may be able to find very discrepant set systems, especially in the less-explored areas of fractional and prefix discrepancies, which could provide novel perspectives and push the boundaries of current mathematical understanding. The achievements of ML in diverse complex systems analysis, which imply that it is feasible to find high-discrepancy set systems that may defy accepted theoretical limitations, serve as the driving force behind this computational exploration. Essentially, the process consists of repeatedly modifying the NNs parameters in order to reduce the variation across sets. The goal is to find configurations that may highlight mathematical characteristics of these systems that may have been missed in the past. A deeper comprehension of the underlying complexity of fractional and prefix discrepancies in particular can be provided by examining these mathematical concepts in more detail. Finding high-discrepancy set systems is only one goal of using NNs in this situation; another is to comprehend the underlying structures and processes that give rise to these differences. By exposing nuances in discrepancy theory that have escaped conventional mathematical study, this computational experiment may offer a new angle on this age-old mathematical conundrum. Therefore, this study aims to explore the depths of discrepancy theory by utilising NNs and representing the connection between computational learning and mathematical theory. We aim to question established paradigms in the discipline and gain new insights by converting the mathematical structures of discrepancy into computational models. A promising direction towards furthering our grasp of discrepancy theory is the use of NNs to explore high-discrepancy set structures. It shows how computer techniques can be used to supplement classical mathematical analysis, which could result in new discoveries about combinatorial structures and their characteristics. This examination into the use of NNs in discrepancy theory research not only broadens the methodological toolkit that mathematicians can employ, but it also fits with the larger trend of combining computational and analytical methods in scientific research.

3. Related Works

When encoding the parameters of a combinatorial optimization—variable-length inputs are required—which a NNs can handle. Recurrent Neural Networks (RNNs) are used as the front-end model in PtrNets, which were first introduced by [18], one of the first neural combinatorial optimization techniques. A succession of points is placed on the plane, and each point's Cartesian coordinates is processed one at a time by the RNN. The RNN technique is carried out by [19], who improve the Ptr-Net technique by combining it with RL. By keeping the RNN decoder to handle changing input size and eliminating the RNN encoder, [20] further improve this method. Graph—is one of the main applications of the Graph Neural Network (GNN) class of encoding networks for neural combinatorial optimization. In an arbitrary graph with an arbitrary number of nodes, GNN learns the message forwarding policy between nodes. Among the first to demonstrate that a GNN-based framework can solve a wide variety of combinatorial optimization issues on graphs uniformly are [21]. By using a more advanced Graph Convolutional

[2] According to Spencer's theorem, a series of polygons with progressively shorter lengths can be used to approximate any continuous curve.

Network (GCN) structure, [22] expand on the work [23]. GCN is used well by [24] to address combinatorial optimization issues on graphs with billions of edges. [25] enhance the quality of the combinatorial optimization solutions by applying unsupervised learning on GNN. The encoding networks that are most pertinent to our research are those that utilise the Transformer encoder. [26] solve the numerous routing issues by concurrently encoding the Cartesian coordinates of all the nodes provided by the problem using this kind of encoder. Using the same encoder as [26], [27] generate solutions with a markedly higher quality by utilising RL training and inference methods that are better suited for combinatorial optimisations. However, in order to solve MIP instances, where constraints and variables constitute a bipartite network weighted by constraint coefficients, [28] learn efficient branch-and-bound variable selection policies. The weapon-target assignment problem is a well-known combinatorial optimization problem that is solved by [29] using a bipartite graph. Given a matrix that specifies the value that each bidder attributes to each item up for sale, [30] determine appropriate auction regulations.

4. Pre-Methodology

In 2021, [17] published a study that presented a new way to apply the deep cross-entropy approach to graph theory problems, demonstrating how effective it is for solving high-dimensionality combinatorial problems. This approach lacks sensitivity to the scale of combinatorial problems and has a specific learning mechanism that makes it stand out from other swift algorithms. It is therefore different from more conventional methods such as human deduction or extensive computer brute forcing. Potential counter examples are produced via the deep cross-entropy approach through a series of repetitive processes. The procedure begins by generating a large number of random graphs, using a probability distribution function based on NNs to direct the process. Then, each graph is assessed and given a score according to how relevant it is to the particular conjecture; the graphs with the highest scores are chosen to be examined further. After that, the process iteratively improves the distribution function's weight parameters to raise the possibility of creating graphs that closely resemble the highest-scoring examples. The technique creates an adjacency matrix that is representative of a graph by employing NNs as a probability distribution function to systematically identify the best vector element at each step and produce these graphs. An auxiliary vector reflecting the current prediction emphasis helps with this procedure, which calls for an in-depth understanding of the graph's structure. The NNs can correctly determine the order of the graph's edge predictions thanks to this dual-vector input mechanism. Once a predetermined number of random graphs have been created, these are evaluated and given a score that indicates how likely it is that they will serve as counter examples to the conjecture being studied. Based on the details of the conjecture, the scoring function is computed with the goal of guiding the learning process to produce graphs with the necessary counter example properties. By repeatedly iterating, the approach gets closer to discovering a legitimate counter example by fine-tuning the probability distribution to favour the formation of high-scoring graphs. [17] method involves changing the NNs weights in the third stage to improve the network's capacity to produce graphs that more closely match the highest-scoring examples from earlier iterations. The efficacy of the technology mostly stems from this learning process, which allows the NNs to adjust and enhance its graph formation strategy over time. Although there are easier techniques to create random graphs, this NNs-based method excels because of its capacity for dynamic learning. The network iteratively improves its probability distribution, gradually favouring graph layouts that achieve high scores for the specified metrics. This methodical weight adjustment, which embodies the idea of a function approximator, progressively transforms the network's output towards the ideal solution under the guidance of gradient descent, back propagation, and cross-entropy principles. Tailored to the complex requirements of proving conjectures in graph theory, Wagner's deep cross-entropy approach is an advanced fusion of NNs technology with probabilistic modelling. Through iterative learning and adaptation, the method presents a viable way to find counter examples in combinatorial problems, demonstrating how ML techniques can progress mathematical study.

5. Post-Methodology

[17] code must first be implemented and modified to fit discrepancy theory, effectively making sure the fundamental setup functions, before we can search for matrices with a high discrepancy. Next, we must ensure that the scoring function is well-defined in order to provide the model with sufficient signal to discriminate between desirable and less desirable incidence matrices. Finally, we optimise for speed because we don't want to have to wait weeks to get any noteworthy results.

5.1 Basic Setup

[17] finds counter examples in graph theory using the deep cross-entropy method, but his code is built in a way that makes it easily adaptable to other combinatorics subfields. As a result, the majority of our setup choices and the majority of our code were developed on his foundation. Python is used to write [17] code, as well as ours, presumably because it is a simple language with a wealth of excellent ML and NNs libraries. The calc score function, which accepts an instance as input and returns the instance's score, was the only part of the code that needed to be changed in order to make the method work with prefix discrepancy. The objective is to identify high (prefix) discrepancy matrices, so we designed our scoring function (calc score) to accept an incidence matrix as input and output the prefix discrepancy of the matrix (Equation 7) as a score. In this manner, matrices with extremely high prefix discrepancy should be found by the network over time. Should this transpire, we will have effectively demonstrated the applicability of [17] approach to discrepancy theory, and therefore potentially to

refute some of the conjectured upper bounds. [17] code adaptation for fractional discrepancy was a little more difficult because the NNs now has to produce continuous numbers rather than just zeros and ones. The scoring function (calc score) is defined similarly to the prefix discrepancy function, with the exception that it now yields the fractional discrepancy as the score.

5.2 Further Improvements

We wanted to tackle the difficult prefix and fractional discrepancy issues in combinatorial set systems by modifying [17] deep cross-entropy method. [17] approach, first used in graph theory, scores random graphs, iterates the creation process based on highest scores, and employs NNs to find counter examples to conjectures. This strategy is distinguished from more conventional brute force or heuristic methods by its ability to handle the large dimensionality of combinatorial problems with resilience and by offering a unique learning process. Initially, we faced a challenge where the NNs was unable to distinguish between more and less desired matrices in situations when there was a prefix difference. Similar to classical discrepancy, prefix discrepancy is quantified as a discrete quantity that is constant amongst identically sized matrices. For example, a lot of 7 × 7 matrices usually receive a prefix discrepancy score of 2. Because of this, in the event that every matrix receives the same score, the NNs will not receive the feedback it needs to determine which matrices are better. As a result, the learning process is no more efficient than random generation. We changed the scoring system to take into account both the number of colorings that led to the given discrepancy and the discrepancy value in order to improve our methodology. This intricate scoring system is shown in Equation (1).

$$calc\_score(A) = prefix - disc(A) - 0.0001 \cdot ln(count)$$
(1)

In this case, "count" refers to the total number of colorings that resulted in the observed difference. Through this modification, the NNs will be directed to concentrate on more promising candidates by giving preference to matrices that obtain their score with fewer colorings. This will produce a stronger learning signal. Getting the network to create and learn from fractional matrices was the issue for fractional discrepancy. The outputs of the network were first directly translated into the matrix elements by mapping them using a sigmoid function to a continuous range between zero and one. But this process produced identical matrices, which significantly reduced the network's capacity to process a variety of input. While each matrix was based on the predictions of the network, we made sure that each one would differ somewhat from the others by adding a noise component to the output values to add variability. This strategy was beneficial at first, but it eventually caused stagnation. Unlike in binary contexts, where it can output values close to zero or one to show high confidence, the network found it difficult to convey confidence in its predictions. In order to get around this, we used an image recognition technique and built up the network to produce a set of probabilities over a specified range of values (e.g., 0.0 to 1.0 in 0.1 increments). By varying the probability distribution over these steps, this technique helped the network convey confidence and uncertainty more effectively and enabled a more sophisticated manner of producing fractional matrices. This was essential because [17] process was iterative. To improve performance, we carried out a number of machine-level adjustments. The Numba just-in-time compiler was the first tool we used. It greatly increases the speed at which Python code can be executed, particularly when NumPy functions and loops are used. In order to lessen memory utilisation and speed up processing, we additionally optimised our data structures by designating lower precision data types where appropriate. We also parallelized the matrix generation and evaluation operations because we knew they were independent jobs. By scaling the computation across numerous CPUs, we were able to significantly reduce the time needed for each iterative cycle of the approach by utilising multi-threading and cloud computing resources. In addition to expanding the [17] application to a wider range of combinatorial issues, we have also increased the method's efficiency and efficacy in discovering high-discrepancy matrices by modifying it to address prefix and fractional discrepancies in set systems.

6. Experimental Analysis

We put the strategies outlined in Section 5 into practice and assessed their effectiveness in identifying significant discrepancies for each of the three variants—fractional, prefix, and classical. At first, we employed the same NNs as [17], who employed three hidden layers with 128, 64, and 4 neurons, respectively. Ultimately, we tripled the number of neurons in each layer to increase learning for prefix and fractional inconsistencies.

6.1 Classical Discrepancy

We wanted to confirm whether our NNs can approach the top bounds proved for classical discrepancy, even though our ultimate goal is to find counter examples for prefix and fractional discrepancy. It is practically impossible to create a zero-one matrix of size 7 × 7 with a (classical) discrepancy greater than 3 since 7 × 7 matrices have an upper bound of discrepancy of 3. However, the majority of 7 × 7 matrices have a disparity of 2. Thus, our goal was to determine if our network could locate one of the few 7 × 7 matrices that had a classical discrepancy of 3, or in other words, hit the upper bound. As seen in Fig. 1, our NNs gets off to a good start by consistently identifying matrices with higher scores. These matrices all have a classical discrepancy of 2, although the number of colouring decreases over time. But rather than going all the way to 3, the score rises quickly before plateauing, for unknown reasons, at a value equal to four colorings after 500 iterations. The fact that our NNs can automatically identify patterns in the data to maximise the provided score function without prior knowledge of the issue setting is still an intriguing outcome, though. Finding even one 7 × 7 matrix with a classical discrepancy of 3 is difficult because there are very few of them. The model is obviously capable of learning, so even if it did not achieve the established upper bound, it might still be able to find counter examples to easier issues. Maybe our choices of parameters wasn't the best ones. For instance, altering the learning rate could cause the NNs to locate these few matrices with a three-discrepancy. In actuality, insufficient learning rate causes even well-designed NNs models to become trapped in local optima. As a result, we think it would be worthwhile to repeat the experiment with various learning rates (or other significant parameters). However, there was sufficient evidence for this theory to proceed to the forms of discrepancy.

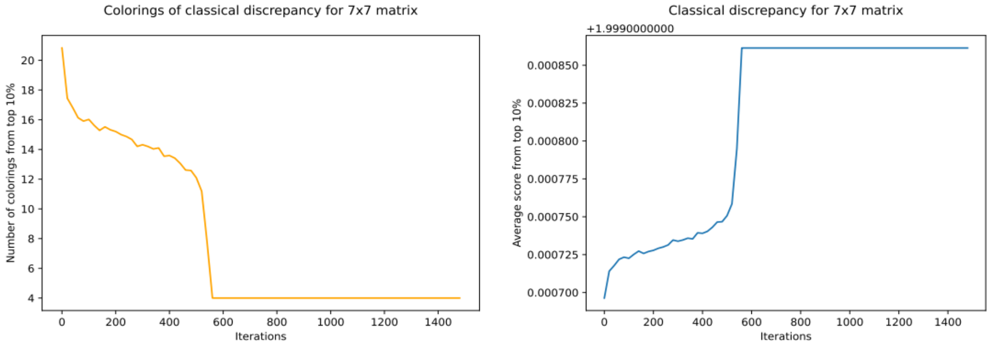

Fig. 1. Our NN learns to reduce the number of colorings for matrices with a classical disparity of 2 (left) to just 4 in roughly 500 iterations, which corresponds to scores of roughly 1.999700 to 1.999861, respectively (right)

6.2 Prefix Discrepancy

Instead of focusing on finding an incidence matrix with a large classical discrepancy, we repeated the experiment and tried to find one with a high prefix discrepancy. The results show that our model performs exceptionally well in identifying matrices with a prefix discrepancy of 3 (refer to Fig. 2, left). After just 30 iterations, the average score of the top 10% was almost 3. Before discovering more than 100 matrices with a prefix discrepancy of 3, the network only produced about 30,000 random matrices due to the NN's 1000 random matrix generation every iteration. We conducted the experiment 100 times because the deep crossentropy approach is random. As a result, we obtained an average of 21,540 random matrices before identifying one with a prefix discrepancy of 3. On the other hand, we required more than three million random matrices (or more than 100 trials on average) when we sampled from a uniform distribution before we discovered a matrix with a prefix discrepancy of three (refer to Table 1). These preliminary findings are really promising: it is evident that the NNs can learn constantly over time without any prior problem knowledge. At approximately 23 iterations, the most remarkable learning occurs (refer to Fig. 2, left). This is because, once the NNs has identified one matrix with a prefix discrepancy of 3, it can rapidly produce over 100 other matrices with the same disparity, which causes the average score of the top 10% to grow significantly. Incorporating the count variable into the scoring function (calc score) also produced the desired results: the network consistently improves the top 10% score, which indicates that fewer colorings from the initial iteration are being generated by the network (refer to Fig. 2, right). Later, the NNs finds a matrix with a prefix discrepancy of 3, so it can eventually go from a few colorings that result in a prefix discrepancy of 2 to zero colouring with a prefix discrepancy of 2. It's unclear why this jump is feasible for prefix discrepancy but not for classical discrepancy. Maybe there are more matrices with a prefix discrepancy of 3, indicating that, in situations when the task is marginally easier than in the prior trial, our model performs flawlessly. Or maybe, compared to classical discrepancy, prefix discrepancy is less susceptible to learning rate fluctuations. To achieve our ultimate objective, we will now swap out the 7 × 7 matrices for $m \times n$ matrices with $m << n$.

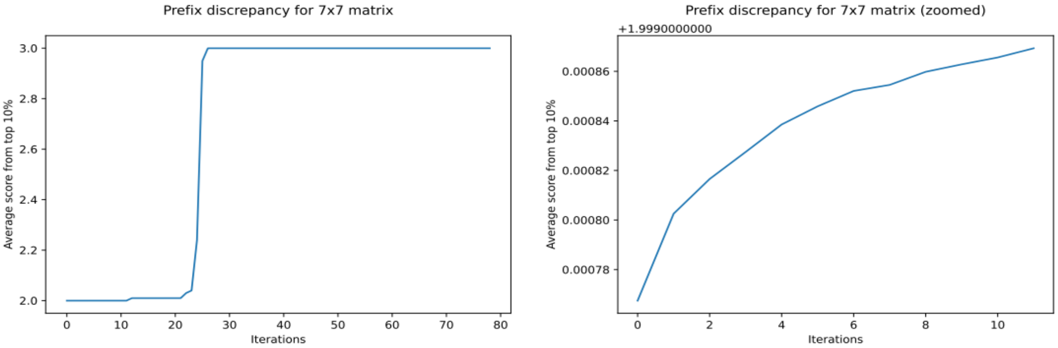

Fig. 2. Compared to classical discrepancy, our model performed better on prefix discrepancy. It could produce 10% of matrices with a prefix discrepancy of 3 (left) in just 30 iterations. The signal problem was resolved when the number of colorings was included to the scoring function. This allowed the NN to learn from the very first

iteration on each subsequent iteration, even in cases when most or all of the matrices still had a prefix discrepancy of 2 (right)

Table 1. We found that our NN approach was over 150 times more efficient than a naive approach for identifying a single matrix with a prefix discrepancy of 3. Our NN technique was nearly ten thousand times more efficient for finding a hundred such matrices. The displayed results are averages from more than 100 trials

Task	Method	# matrices
Find 1 matrix	pref-disc. 3 naive	141,584
	NN	21,540
Find 100 matrices	pref-disc. 3 naive	314,158,400
	NN	28,510

6.3 Fractional Discrepancy

Lastly, we assessed the network's ability to identify 7 × 7 matrices with a substantial fractional disparity. Yet, we did not apply [17] suggested NNs model this time. Alternatively, we employed the fractional combinatorics model we recently constructed. In addition, we did not utilise the scoring function because fractional discrepancy does not have the problem of needing to "jump" from one discrete discrepancy to the subsequent discrete discrepancy, such as from a discrepancy of 2 to a discrepancy of 3. Over the first 60,000 iterations, as shown in Fig. 3, our new NNs model was able to continuously learn and improve the average top 10% score; however, after that, it appeared to plateau at a top 10% average of 2.00. The new model we created effectively learned how to identify matrices with fractional values that result in a high disparity, which is an already encouraging outcome. Surprisingly, the outcome we got is even better than we had anticipated—the score actually hovers slightly above the 2.00 mark, indicating that the network can consistently identify at least a few matrices with a fractional discrepancy higher than 2. It's interesting to note that binary matrices with a classical discrepancy of three were produced when we substituted the closest integer (zero or one) for each fractional value in these matrices. As a result, our model was able to locate matrices with the largest potential classical discrepancy indirectly. Our novel model made it possible for us to reward desired behaviour with a signal that was more discriminating, which resulted in these incredibly difficult-to-find binary matrices after discretization. We believe more experiments should be conducted to address these pertinent topics. However, it is yet unknown if this characteristic also applies to matrices larger than 7 × 7. Moreover, modifying a few of the input parameters could produce a different—possibly even better—result. For example, obtaining these binary matrices directly (without discretization) is one possibility.

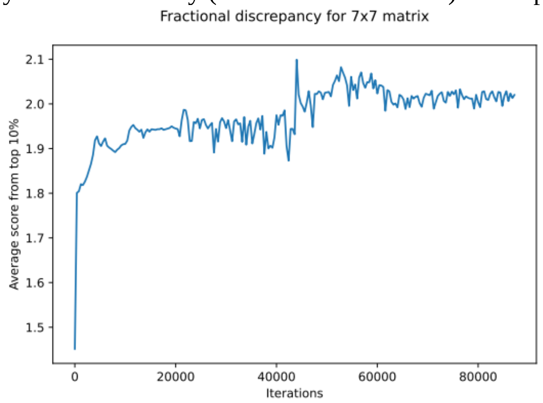

Fig. 3. Our new model was able to detect matrices with a fractional disparity bigger than 2, even though it required up to 60000 iterations

7. Conclusions and Future Works

We explored the use of the deep cross-entropy method, which was originally applied in graph theory, to solve discrepancy theory problems, with a focus on matrices with high discrepancy levels (classical, prefix, and fractional), after being inspired by Wagner's preprint (2021). Through machine-level and algorithmic optimisations, as well as enhanced signal generation—especially with a unique NNs model specifically designed for fractional combinatorics—we increased the program's efficiency. Promising results from studies showed that the NNs could recognise matrices with greater differences in all three categories. Notably, in binary situations, our fractional model outperformed Wagner's method and produced higher signal, demonstrating the promise of this approach for detecting high-discrepancy examples. These promising results support the need for additional study. In order to fully assess the model's effectiveness and discover more information, further steps should involve undertaking extensive testing with different matrix sizes, NNs parameters, and expansion into higher dimensions. Furthermore, multidisciplinary methods influenced by disciplines such as chemistry may present new ways to improve the model. This advanced NNs model goes beyond discrepancy theory and may prove useful in challenging current theories in fractional combinatorics and related fields.

References

[1] N. Kamuni, I. G. A. Cruz, Y. Jaipalreddy, R. Kumar, and V. K. Pandey, “Fuzzy Intrusion Detection Method and Zero-Knowledge Authentication for Internet of Things Networks,” *International Journal of Intelligent Systems and*

Applications in Engineering, vol. 12, no. 16s, pp. 289–296, Feb. 2024, Accessed: Mar. 31, 2024. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/4821

[2] N. Kamuni, S. Chintala, N. Kunchakuri, J. S. A. Narasimharaju, and V. Kumar, "Advancing Audio Fingerprinting Accuracy Addressing Background Noise and Distortion Challenges," *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pp. 341–345, Feb. 2024, doi: 10.1109/ICSC59802.2024.00064.

[3] Kamuni, N., Shah, H., Chintala, S., Kunchakuri, N., & Alla, S. (2024, February). Enhancing End-to-End Multi-Task Dialogue Systems: A Study on Intrinsic Motivation Reinforcement Learning Algorithms for Improved Training and Adaptability. In 2024 IEEE 18th International Conference on Semantic Computing (ICSC) (pp. 335-340). IEEE.

[4] Alla, S., Soltanisehat, L., Tatar, U., & Keskin, O. (2018). Blockchain technology in electronic healthcare systems. In IIE Annual Conference. Proceedings (pp. 901-906). Institute of Industrial and Systems Engineers (IISE).

[5] Lakshminarayanan, V., Ravikumar, A., Sriraman, H., Alla, S., & Chattu, V. K. (2023). Health care equity through intelligent edge computing and augmented reality/virtual reality: a systematic review. Journal of Multidisciplinary Healthcare, 2839-2859.

[6] Alla, S., Soltanisehat, L., & Taylor, A. (2018, July). A Comparative Study of Various AI Based Breast Cancer DetectionTechniques. In IX International Conference on Complex Systems (p. 213).

[7] Alla, S. (2019). A Statistical Analysis of Surgeons' Preference on Robot-Assisted Surgeries. In IIE Annual Conference. Proceedings (pp. 1385-1390). Institute of Industrial and Systems Engineers (IISE).

[8] G. S. Kashyap, A. E. I. Brownlee, O. C. Phukan, K. Malik, and S. Wazir, "Roulette-Wheel Selection-Based PSO Algorithm for Solving the Vehicle Routing Problem with Time Windows," Jun. 2023, Accessed: Jul. 04, 2023. [Online]. Available: https://arxiv.org/abs/2306.02308v1

[9] M. Kanojia, P. Kamani, G. S. Kashyap, S. Naz, S. Wazir, and A. Chauhan, "Alternative Agriculture Land-Use Transformation Pathways by Partial-Equilibrium Agricultural Sector Model: A Mathematical Approach," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.11632v1

[10] G. S. Kashyap *et al.*, "Revolutionizing Agriculture: A Comprehensive Review of Artificial Intelligence Techniques in Farming," Feb. 2024, doi: 10.21203/RS.3.RS-3984385/V1.

[11] G. S. Kashyap, K. Malik, S. Wazir, and R. Khan, "Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing," *Multimedia Tools and Applications*, vol. 81, no. 25, pp. 36685–36698, Oct. 2022, doi: 10.1007/s11042-021-11558-9.

[12] S. Wazir, G. S. Kashyap, K. Malik, and A. E. I. Brownlee, "Predicting the Infection Level of COVID-19 Virus Using Normal Distribution-Based Approximation Model and PSO," Springer, Cham, 2023, pp. 75–91. doi: 10.1007/978-3-031-33183-1_5.

[13] S. Wazir, G. S. Kashyap, and P. Saxena, "MLOps: A Review," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.10908v1

[14] G. S. Kashyap, D. Mahajan, O. C. Phukan, A. Kumar, A. E. I. Brownlee, and J. Gao, "From Simulations to Reality: Enhancing Multi-Robot Exploration for Urban Search and Rescue," Nov. 2023, Accessed: Dec. 03, 2023. [Online]. Available: https://arxiv.org/abs/2311.16958v1

[15] S. Naz and G. S. Kashyap, "Enhancing the predictive capability of a mathematical model for pseudomonas aeruginosa through artificial neural networks," *International Journal of Information Technology 2024*, pp. 1–10, Feb. 2024, doi: 10.1007/S41870-023-01721-W.

[16] G. S. Kashyap, A. Siddiqui, R. Siddiqui, K. Malik, S. Wazir, and A. E. I. Brownlee, "Prediction of Suicidal Risk Using Machine Learning Models." Dec. 25, 2021. Accessed: Feb. 04, 2024. [Online]. Available: https://papers.ssrn.com/abstract=4709789

[17] G. S. Kashyap *et al.*, "Detection of a facemask in real-time using deep learning methods: Prevention of Covid 19," Jan. 2024, Accessed: Feb. 04, 2024. [Online]. Available: https://arxiv.org/abs/2401.15675v1

[18] P. Kaur, G. S. Kashyap, A. Kumar, M. T. Nafis, S. Kumar, and V. Shokeen, "From Text to Transformation: A Comprehensive Review of Large Language Models' Versatility," Feb. 2024, Accessed: Mar. 21, 2024. [Online]. Available: https://arxiv.org/abs/2402.16142v1

[19] H. Habib, G. S. Kashyap, N. Tabassum, and T. Nafis, "Stock Price Prediction Using Artificial Intelligence Based on LSTM– Deep Learning Model," in *Artificial Intelligence & Blockchain in Cyber Physical Systems: Technologies & Applications*, CRC Press, 2023, pp. 93–99. doi: 10.1201/9781003190301-6.

[20] N. Marwah, V. K. Singh, G. S. Kashyap, and S. Wazir, "An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning," *International Journal of Information Technology (Singapore)*, vol. 15, no. 4, pp. 2317–2327, May 2023, doi: 10.1007/s41870-023-01264-0.

[21] A. Z. Wagner, "Constructions in combinatorics via neural networks," Apr. 2021, Accessed: Apr. 15, 2024. [Online]. Available: https://arxiv.org/abs/2104.14516v1

[22] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, vol. 2015-Janua, pp. 2692–2700.

[23] M. Gori, G. Monfardini, and F. Scarselli, "A new model for earning in raph domains," in *Proceedings of the International Joint Conference on Neural Networks*, 2005, vol. 2, pp. 729–734. doi: 10.1109/IJCNN.2005.1555942.

[24] M. Nazari, A. Oroojlooy, M. Takáč, and L. V Snyder, "Reinforcement learning for solving the vehicle routing problem," in *Advances in Neural Information Processing Systems*, 2018, vol. 2018-Decem, pp. 9839–9849.

[25] H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-Decem, pp. 6349–6359.

[26] Z. Li, Q. Chen, and V. Koltun, "Combinatorial optimization with graph convolutional networks and guided tree search," in *Advances in Neural Information Processing Systems*, 2018, vol. 2018-Decem, pp. 539–548.

[27] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Sep. 2016, Accessed: Apr. 15,

2024. [Online]. Available: https://arxiv.org/abs/1609.02907v4

[28] S. Manchanda, A. Mittal, A. Dhawan, S. Medya, S. Ranu, and A. Singh, “GCOMB: Learning budget-constrained combinatorial algorithms over billion-sized graphs,” in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-Decem, pp. 20000–20011.

[29] N. Karalias and A. Loukas, “Erdos ? Goes neural: An unsupervised learning framework for combinatorial optimization on graphs,” in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-Decem, pp. 6659–6672. Accessed: Apr. 15, 2024. [Online]. Available: https://github.com/Stalence/erdos_neu

[30] W. Kool, H. Van Hoof, and M. Welling, “Attention, learn to solve routing problems!,” in *7th International Conference on Learning Representations, ICLR 2019*, Mar. 2019. Accessed: Jul. 04, 2023. [Online]. Available: https://arxiv.org/abs/1803.08475v3

[31] Y. D. Kwon, J. Choo, B. Kim, I. Yoon, Y. Gwon, and S. Min, “POMO: Policy optimization with multiple optima for reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-Decem, pp. 21188–21198.

[32] M. Seyfi, A. Banitalebi-Dehkordi, Z. Zhou, and Y. Zhang, “Exact Combinatorial Optimization with Temporo-Attentional Graph Neural Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2023, vol. 14172 LNAI, pp. 268–283. doi: 10.1007/978-3-031-43421-1_16.

[33] D. Gibbons, C. C. Lim, and P. Shi, “Deep learning for bipartite assignment problems,” in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2019, vol. 2019-Octob, pp. 2318–2325. doi: 10.1109/SMC.2019.8914228.

[34] P. Dütting, Z. Feng, H. Narasimhan, D. C. Parkes, and S. S. Ravindranath, “Optimal auctions through deep learning,” *Communications of the ACM*, vol. 64, no. 8, pp. 109–116, May 2021, doi: 10.1145/3470442.