Chess Analysis Using Geometric Machine Learning And Topological Data Analysis

Charles Xu, Madelyn Stewart, Peter Yu, Ziyu Zhu

I. Abstract

Traditional methods of probing chess playing style and cheating have relied solely on move precision. In this project, we explore how the geometric structure of chess games can be used to study and classify the chess games of a particular player. Board positions from classical tournaments were one-hot encoded and then embedded into a lower dimensional space using an autoencoder with a contrastive loss to distinguish winning board positions from losing board positions. In this embedding, a single chess game is treated as a point cloud. To study the structure of the data, persistence diagrams corresponding to the chess games were calculated. Persistence entropy and persistence landscape features were calculated as topological data features. These features were fed into several classifiers to predict if a game belonged to a certain player, resulting in a modest classification accuracy of up to 64%. This work demonstrates that the structure of chess games contains valuable information that may be incorporated into existing chess cheating and style analysis methods.

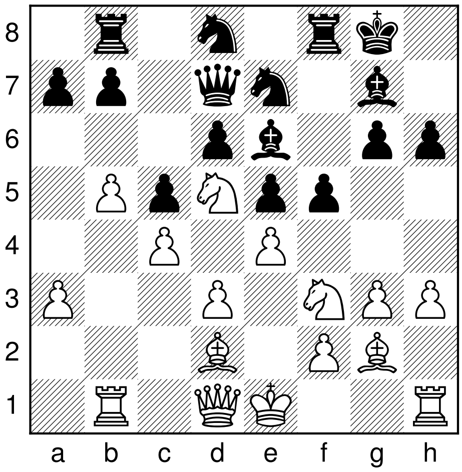
All code is made available at https://github.com/HungryAmoeba/chess_analysis.

Our presentation is viewable at https://tinyurl.com/chessanalysis.

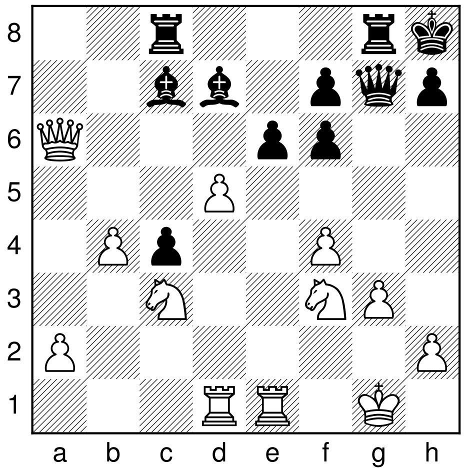
II. Introduction

Chess is a game famous for its complexities and endless possibilities. The extent of the possible game space remains an open question; Claude Shannon estimated the existence of 10^{120} legal chess games, making the 10^{80} estimated number of atoms in the universe seem small. All chess games begin in the same state: a predetermined board beginning with white's move. As the game unfolds, stark differences in player strategy and experience become apparent. Some game boards move quickly into the middle game, moving dynamically and eliminating pieces at a rapid rate; other games slowly move from opening while pieces seem to shuffle in place. The Soviet-American chess player Tigran Petrosian is famous for his solid and positional style, and his games tend to prefer closed positions.

An example of a closed position reached from the Czech Benoni opening is given below. In such positions, players will often rotate pieces, and transpose to similar (or identical) positions of a similar nature.



Mikhail Tal is a player whose games are characterized by their open tactical and aggressive play. An example of a position reached from one of his games is given below. The pieces have a high degree of mobility with many attacking chances from both sides. The game progressed quickly from this state, with large changes in the position and rapid piece exchanges:



We posit that chess has an underlying high-dimensional geometric structure granted by the rules of the game, and that this information can be used to uncover more about the structure of games, player techniques, and even discover possible attempts at cheating. In this project, we aim to use geometric machine learning techniques to analyze chess games. We employ board positions to represent chess games and attempt to predict whether games were played by a specific famous player.

Although modern chess engines result in precise play and quantitative evaluation of positions, they do not abstract away information about which positions or games are structurally similar to each other. In addition to providing new insights into positional play in chess, such information could be useful in cheat detection. Cheating in chess is an existential crisis for the sport, and the complexity of the game complicates cheating detection efforts. These problems have recently come into the public spotlight when high profile cheating allegations were launched against Hans Niemann. Niemann is an American chess grandmaster who won the 2021 World Open tournament and was ranked forty-fifth highest-rated player in the world. In response to the cheating allegations, Niemann has sued Magnus Carlsen and several others for defamation, asking for 100 million dollars in damages. Efforts to analyze the structure of chess games and identify similarities between games may prove more robust methodology to determine whether outside help is being utilized, as style could be more intrinsic to a player than move accuracy.

III. Background

Geometric machine learning methods have gained popularity over the past few years, with a rapid expansion of available high-dimensional data sets colloquially known as big data and a search for new techniques to understand correlations and represent the data in ways more easily conceptualized. These geometric methods are inspired by the Manifold Hypothesis, which states that many of these high-dimensional data sets are sampled from an underlying lower-dimensional manifold. This supports finding ways to accurately represent data in fewer dimensions which can in turn relay new discoveries about the structure of the point cloud. Games which significantly deviate from a player's manifold could be indicative of cheating or outside help.

We employ a few different techniques to study the structure of chess games. The first is Principal Component Analysis (PCA), a strategy of dimension reduction that identifies a basis that maximizes the variance of the data along the basis vectors and projects the data into a lower dimensional space using these vectors to embed the original data. We then move on to neural network embeddings of the data, employing an autoencoder with contrastive loss to parse through different game outcomes. Finally, we employed topological data analysis (TDA) techniques like persistence diagrams and related features. Persistence diagrams analyze the shape of data, quantifying at what point distinct features of the data structure are lost as we view it from a wider and wider scale.

We expect manifold learning and TDA to be helpful because differences in play style should become apparent using these methods. Positional players such as Petrosian should have games that progress very slowly through a latent space, and exhibit features at small length scales. Other aggressive and tactical players such as Tal seek moves with open up positions and lead to rapid changes in structure. It is expected that such players will have closed positions that progress quickly through a latent space, spread out quickly, and exhibit features and structures at larger length scales.

IV. Methods

Over the board tournament chess games from high rated players with classic time controls were downloaded from a database hosted by the publication ”The Week in Chess” in portable game notation (PGN) format. The pieces in chess are the pawn, rook, knight, bishop, queen, and king, and each piece exists for both the white and black players. Thus any piece (without distinguishing between duplicate pieces of the same color) can be one hot encoded in a 12 dimensional vector. Since there are 64 squares on a chess board, this means that any chess position can be encoded in a 768 dimensional vector. The vector was chosen to be flat rather than retain the spatial structure of the chess board because specifying neighbor proximity is not a good model of the dynamics of chess, where certain pieces such as the bishop, rook, queen and knight can exert their influence over longer ranges.

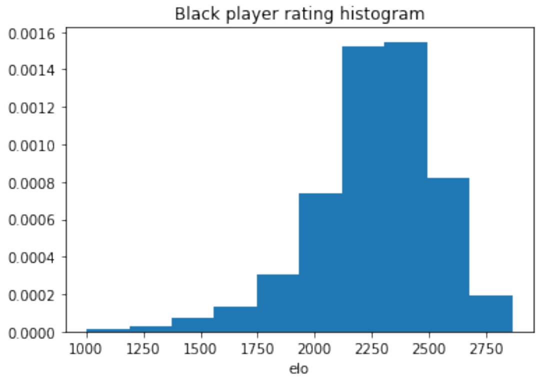


Fig. 1. Black player elo has a median of around 2250. The distribution of white elo ratings appears very similar. The sampled games include many advanced and titled players.

In total, over 13 million chess positions were downloaded. As a simple first attempt at understanding the underlying structure of the data, PCA was performed on the dataset, and the first two principal components were used for visualization purposes.

PCA is able to capture the stage of the game as shown in Figure 3. Points in earlier positions corresponding to darker colored points belong to the opening stages of the game and fall on the right side of the embedding. Middle game positions are in the middle of the embedding, and endgame positions are mapped onto the left half of the embedding. However, because there is likely non-linear structure in the data, other methods to learn a meaningful embedding are explored.

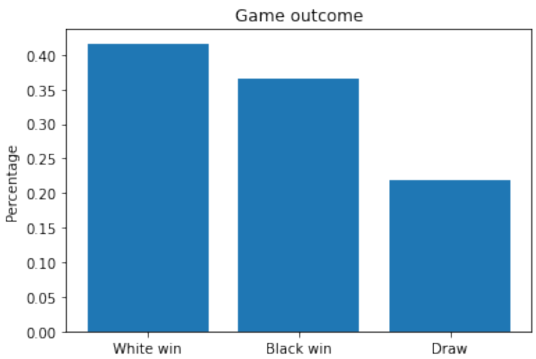


Fig. 2. A white win is the most common outcome of each game with an occurrence of slightly over 40%. Black wins roughly 35% of the time, and the remaining games are draws. Decisive results are the most common amongst these games, and may vary depending on nature of the tournaments that the games come from.

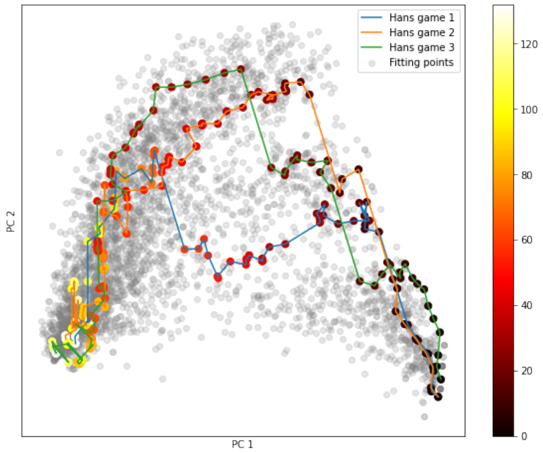


Fig. 3. PCA on the chess games. The grey points are randomly chosen chess positions which are used to learn the best principal components. Three chess games belonging to Hans Niemann are highlighted in the plot, with the color bar corresponding to the move number of the game that the position corresponds to.

A neural network autoencoder was used to learn embeddings which captured important features of the games. To avoid over-constraining the network, an embedding dimension of 10 was used. To visualize the embeddings, Potential of Heat-diffusion for Affinity-based Trajectory (PHATE) was used to visualize the embedding. PHATE preserves both local and global distances in addition to manifold structure. It also can successfully preserve patterns in data such as continual progressions, branches and clusters, making it an excellent tool for comparing and studying the progression of chess games.

Further work was done to capture information about the favorability of a given position. The final game outcome (white win, black win, or draw) was used as a label, and a contrastive loss $\mathcal{L}_C$ is applied:

$$\mathcal{L}_C = -\log \frac{\exp \mathrm{sim}(z_i, z_{n(i)})}{\sum_j^n \exp \mathrm{sim}(z_i, z_j)}$$

where z_i and $z_{n(i)}$ represent data points with the same class label, and z_i has a different class label from z_j. Cosine similarity was initially used as the similarity function. However because PHATE first calculates local similarities with the α-decay kernel applied to the euclidean distances between points, embeddings with high cosine similarity may still have large euclidean distances between points so the full power of the PHATE visualization may not be utilized. Instead, the contrastive loss similarity function was taken to be the euclidean distance passed through a Gaussian kernel.

We also examined topological data analysis over chess games, using the embedding created by our autoencoder with contrastive loss. Specifically, we sought to use TDA to distinguish between Hans Niemann games and other players. We limited our analysis to the last sixty board positions, filtering out any games with less than sixty moves and created persistence diagrams tracking homology dimensions 0, 1, and 2. A comparison between Euclidean Cech persistence and Vietoris Rips persistence on our data showed the latter was both more stable and less time intensive, so we proceeded with Vietoris Rips. We then employed a linear scaler of the diagrams, using the bottleneck metric. This metric quantifies similarity between persistence diagrams by calculating the shortest distance as measured by the supremum norm on $\mathbb{R}^2$ at which a perfect matching between points of two persistence diagrams that preserves the order of any points on the diagonal. In other words, the bottleneck scaler is the real number d such that any corresponding points between the persistence diagrams are at most d apart.

We derived persistence entropy descriptions of our data, which can be thought of as a measure of the "chaos" of the points in the persistence diagram. In exact terms, if $D = \{(b_i, d_i)\}_{i \in I}$ is the set of points of our persistence diagram, we calculate the entropy

$$E(D) = -\sum_{i \in I} p_i \log (p_i)$$

where

$$p_i = \frac{d_i - b_i}{\sum_{i \in I}(d_i - b_i)}.$$

We calculate the persistence landscape, an invertible mapping of a persistence diagram into real-valued functions space that allows for further analysis with statistical and machine learning tools. Using this landscape, we find the the argmax, or the input that returns the maximal value for each of the functions in our persistence landscape.

Finally, we tested the accuracy of five classifiers using the persistence entropy and persistence landscapes: C-Support Vector Classification (SVM), AdaBoost, Multi-layer Perceptron classifier, Random Forest, and Gaussian Process. We scored each classifier's performance over test data to quantify our results.

V. Results

The embeddings learned by the autoencoder with only a reconstruction loss and the embeddings learned

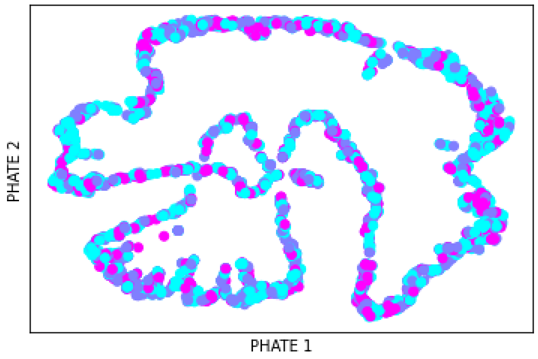


Fig. 4. Learned embeddings as visualized by PHATE when only a reconstruction loss is used to train the neural network. The three different colors represent the outcomes of the games.

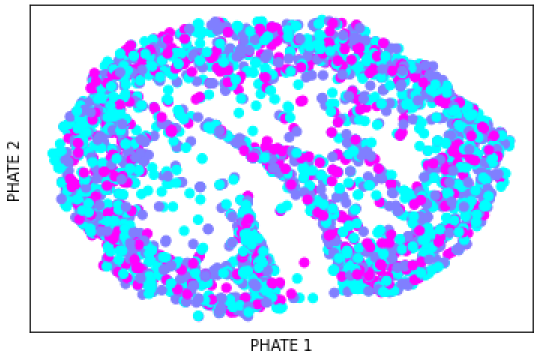


Fig. 5. The embedding of chess positions from an autoencoder with a contrastive loss. The latent space is not separated by game outcome, but has a markedly different structure from the embedding where no contrastive loss is applied.

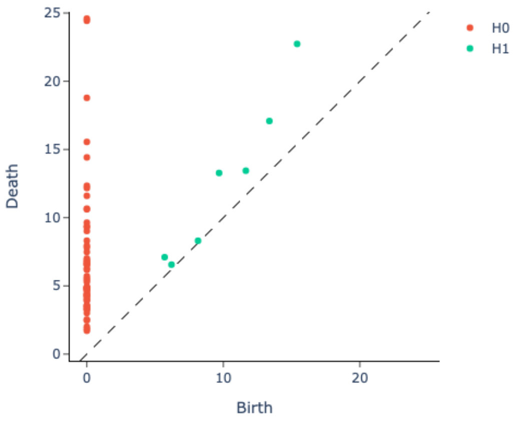


Fig. 6. A persistence diagram for a full chess game. Dimension 0, 1 and 2 features are tracked.

Feature	SVM	AdaBoost	MLP	RF	GP
PE	.6225	.6373	.6225	.6471	.6225
PL	.6127	.5637	.6078	.5147	.5490

TABLE I

PE stands for persistence entropy, PL stands for persistence landscape. SVM is support vector machine, AdaBoost is adaptive boosting, MLP is a multi-layer perceptron, RF is random forest, and GP is a Gaussian process classifier. The values reported are the percentage of games classified correctly (either as a Niemann game, or not a Niemann game).

by the autoencoder with an additional contrastive loss differed greatly. In the embedding layer, it is not possible to discern the final outcome of the game. Additionally, games do not follow smooth trajectories in this latent space.

Applying a contrastive loss to the autoencoder changes the shape and structure of the latent space, but does not allow for a clean separation of positions based on the final outcome of the game. In general, the same position can be associated to a win, loss, or draw with similar probability. This is particularly true of opening and middle game positions. Thus it is perhaps too ambitious of a task for the autoencoder to simultaneously learn embeddings and accurate position evaluations – a task that is usually performed by dedicated chess engines. Despite the visually unimpressive results, the contrastive loss changes the learned embedding to incorporate the additional available information about the game outcome.

The persistence diagrams were filtered. At this point, the data was divided in half. One half was games played by Hans Niemann, and the other half of games were from chess players who were also at a strong level. In addition to the persistence entropy and persistent landscapes, persistence images corresponding to each game was generated as shown in Figure 7. Using traditional computer vision techniques on persistence images could be a promising direction of future work. Persistence entropy and features from the persistence landscape were calculated from the persistence diagram and fed into several classifiers to see if it was possible to predict whether or not a game was played by Niemann. The results are reported in table 1.

For all classifiers, the persistence entropy produces higher classification accuracies than the persistence landscape although all accuracies were within 10% of each other. The highest accuracy achieved came from a random forest classifier over persistence entropy values, at a classification accuracy of 64.71%. Thus it appears that the persistence entropy carried more salient information to characterize the play style of Niemann when compared to other high level chess players. In theory this classifier could be used to predict cheating. If one game from Niemann was found to have a low probability of actually belonging to him, then the result would suggest the use of outside help or cheating.

Because the persistent entropy carries information which can aid in discerning a player's style from other players, the procedure described in this paper can extract general descriptors of play style. In the future,

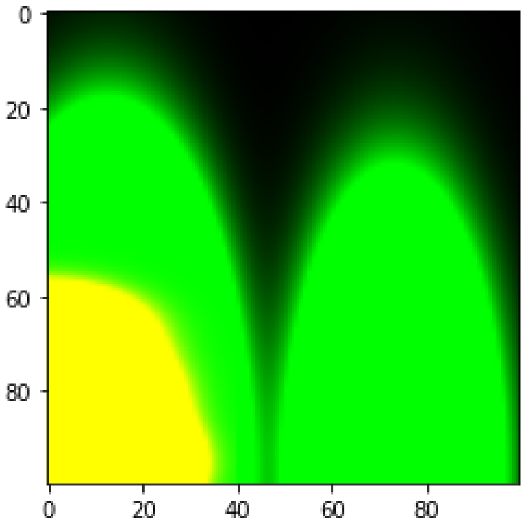


Fig. 7. Generated persistence images from point cloud data. The persistence image may contain more relevant information, and could result in higher classification accuracies in the future.

these features may also applied to further exploration. An example task could be clustering to see which players have games with similar structure.

VI. Conclusion

We started the project with the hope of developing a more concrete understanding of the abstract patterns of chess, and we eventually focused on performing geometric machine learning and topological data analysis to classify games of a particular chess player. We applied a variety of tools and techniques in the project, including PCA, autoencoder, PHATE, and TDA. Through our analysis, we obtained some interesting results, including embeddings of board positions and classifications of a player's cheating behavior. Most results are not conclusive, but overall, our project highlights the potential of machine learning to improve our understanding of chess. By continuing to develop and refine these techniques, we will be able to gain novel insights to the game built on the structure of games.