ChessMatch-V2: A Style-Based Chess Recommender System for Personalized Game Study

Benjamín Varela [1] **Francisco Meza** [1 *] **Vicente Navarro** [1 *]

Abstract

We present ChessMatch-V2, a recommender system that analyzes a player's games to identify their style, match them with similar professional players, and suggest instructive games aligned with their tendencies. To represent individual games, we compare three embedding methods—SBERT, e5-large, and a graph neural network (GNN)—with GNN achieving the best performance. Using a curated dataset of 50 professional players, we embed each player based on the distribution of their games across style clusters obtained via K-Means. User embeddings are constructed similarly and compared via cosine similarity to retrieve similar players and games. ChessMatch-V2 enables style-focused learning and introduces a novel embedding approach based on cluster distributions, which can be generalized to represent entities through the structural patterns of their components in broader machine learning applications.

1. Introduction

Chess improvement often involves studying the games of grandmasters to understand strategic patterns and high-level decision-making. However, choosing which games to study is typically arbitrary and seldom tailored to the individual style of the learner. Personalized training systems could offer more efficient and engaging learning experiences.

While some frameworks attempt to classify player styles and guide training accordingly, they often rely on manual profiling or simplified typologies. Computational tools for chess learning tend to focus on specific areas like endgames or general educational adaptation, lacking style-aware content recommendation.

[*]Equal contribution [1]Pontificia Universidad Católica de Chile, Santiago, Chile. Correspondence to: Francisco Meza <fjmeza1@uc.cl>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

In this work, we introduce *ChessMatch-V2*, a data-driven recommender system that extracts a player's style from their game history, identifies similar professional players, and recommends relevant games for study. Our approach leverages deep learning embeddings, clustering algorithms, and a novel representation of players based on the distribution of their games across style clusters.

We build upon a preliminary prototype, ChessMatch-V1, and improve it by expanding the player pool, using more scalable and accurate embeddings, reducing computational overhead, and introducing a more robust evaluation methodology. Our contributions include the system design, comparison of embedding methods, and the proposal of a generalizable cluster-distribution representation.

2. Related Work

One of the most common ways to improve at chess is by studying games played by top players. However, these game selections are typically not tailored to the learner's own style.

A notable attempt to structure style-based training is *The 4 Player Types*, a model popularized by GMs Karsten Müller and Luis Engel, based on Lars Bo Hansen's typology. It divides players into four archetypes—activists, pragmatists, theoreticians, and reflectors—and suggests training adapted to each type (Müller & Engel, 2022). While insightful, this model depends on manual classification and lacks modern data-driven capabilities.

From a computational perspective, systems like EK-Chess (Zhuang et al., 2023) provide structured endgame instruction via knowledge graphs, and broader educational platforms explore machine learning-based personalized recommendations (Chen et al., 2024). However, none of these systems offer a mechanism to infer a player's style from full games and recommend professional matches accordingly.

ChessMatch (V1) (Varela & Álvarez, 2024), a prototype version of our system, used SVM-based embeddings from a classifier trained on 17 world champions. In ChessMatch-V2, we redesign the architecture to include more players, adopt faster and more expressive embeddings, represent players via cluster distributions, and evaluate the system with a more robust framework.

3. Proposed Solution

The goal of ChessMatch-V2 is to deliver a personalized chess study experience by recommending instructive games from professional players whose styles resemble that of the user. The system achieves this through a multi-step process involving game embeddings, clustering, player representation, and similarity-based retrieval.

3.1. Game Embedding and Clustering

The first step is to represent individual games as vectors. We explore three embedding approaches: SBERT, e5-large, and a custom Graph Neural Network (GNN). Once games are embedded, we apply K-Means clustering to group games into k latent style clusters (e.g., $k = 10$). These clusters are intended to capture abstract stylistic tendencies such as tactical aggression, positional maneuvering, or endgame focus.

3.2. Professional Player Representation

Each professional player in our dataset is represented by a distribution over the k game clusters. This is done by embedding all their available games, assigning each game to its closest cluster, and computing the normalized histogram of cluster assignments. The result is a k-dimensional vector capturing the stylistic fingerprint of the player.

Next, we apply K-Means again—this time on the set of player vectors—to group professionals into broader **style cluster**, enabling a higher-level classification of play styles.

3.3. User Representation and Style Matching

The same process is applied to the user: their games are embedded and assigned to the same game clusters, producing a distribution vector. This vector is then compared to the player embeddings using cosine similarity to retrieve the n most stylistically similar professional players.

Additionally, the user is assigned to a style cluster based on proximity in the player cluster space.

3.4. Game Recommendation

For each of the n retrieved players, we recommend m games that are most similar to the user's play. To do this, we compute the mean of the user's game embeddings and retrieve the top m most similar games from each professional using cosine similarity.

This approach ensures that the recommended games not only align with the user's overall style, but also reflect game-level patterns and motifs that the user is likely to benefit from studying.

4. Dataset

The dataset used for this study consists of 99,897 chess games played by 50 of the greatest players of all time, as ranked by Chess.com[1]. The games were downloaded from PGN Mentor[2], a repository of chess games in PGN format.

4.1. PGN Format

PGN (Portable Game Notation) is a widely used plain-text format for storing chess games. Each PGN file encodes both metadata and move sequences. The metadata includes information such as the players' names, event location, date, result, and ELO ratings, while the move section records the sequence of moves using algebraic notation.

Below is an example of how a typical PGN record appears:

```
[Event "Reykjavik WCh"]
[Site "Reykjavik ISL"]
[Date "1972.07.11"]
[Round "1"]
[White "Spassky, Boris V"]
[Black "Fischer, Robert J"]
[Result "1-0"]

1. e4 c5 2. Nf3 d6 3. Bb5+ Nd7 ...
```

This structure allows for easy parsing and analysis of game features and player metadata.

4.2. Player Selection and Game Distribution

To ensure a diverse and stylistically rich dataset, we selected 50 legendary players across different eras, styles, and regions. Each player's number of available games varies, depending on historical coverage and data availability.

Figure 1 presents a histogram showing the number of players whose total recorded games fall within specific intervals. The x-axis represents ranges of games (e.g., 0–500, 500–1000, etc.), while the y-axis indicates how many players fall into each range.

4.3. Game Characteristics

The games in the dataset span a wide historical range, from the late 19th century to the early 21st century, encompassing different eras of opening theory and strategic evolution. The average game length is 38 plies (i.e., 19 full moves per side),

[1] https://www.chess.com/article/view/best-chess-players-of-all-time
[2] https://www.pgnmentor.com/files.html

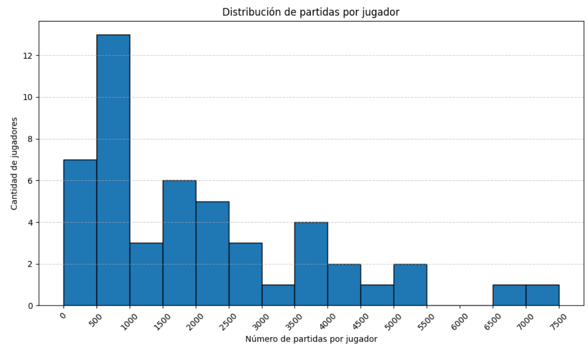


Figure 1. Number of players per range of total games played.

which is typical for competitive over-the-board chess.

This dataset forms the foundation for building player representations, training embedding models, and evaluating recommendation quality.

5. Methodology and Results

Our pipeline extracts from PGN files the sequence of moves made by the player of interest and converts each game into an embedding. We compare three embedding approaches—two language-based and one graph-based—each capturing different structural aspects of chess games.

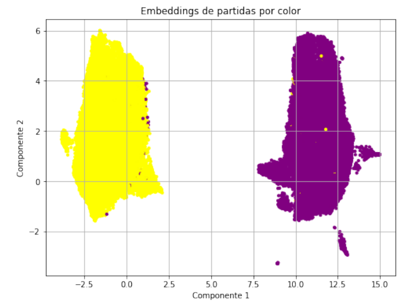
Why text? Inspired by Andreas Stöckl's article *"Chess2Vec: A Map of Chess Moves"* (Stöckl, 2020), where Word2Vec is used to embed individual moves based on their contextual similarity, we treat a chess game as a sentence and each move as a word. Just as Stöckl explores which moves are semantically close by analyzing their surrounding context—e.g., identifying what moves often follow or precede 1.d4—we adopt a similar perspective, representing the full sequence of a player's moves as a natural-language sentence suitable for language-model embeddings.

5.1. PGN Processing and Embedding Approaches

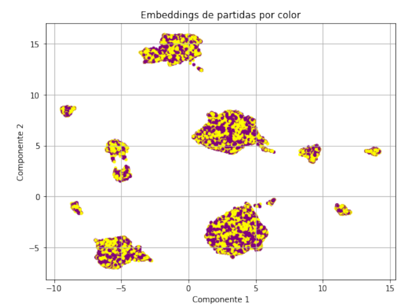
SBERT. ChessMatch-V1 used SBERT (Reimers & Gurevych, 2019) to embed chess games as sequences of moves in standard algebraic notation (SAN). SBERT produces 768-dimensional embeddings capturing tactical and strategic similarities between games.

e5-large-v2. For ChessMatch-V2, we explored e5-large-v2 (Wang et al., 2024), a more recent text encoder based on contrastive learning, which outputs 1024-dimensional embeddings with improved semantic separation.

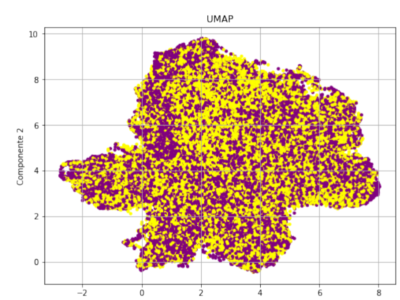
Graph Convolutional Network (GCN). To evaluate a non-textual approach, we used a GCN (Kipf & Welling, 2017) to embed games represented as sequences of board states (nodes) connected by moves (edges). Nodes encode piece positions; edges encode move details. Importantly, this GCN was not trained — we used random weights with a fixed seed for reproducibility.



(a) SBERT



(b) e5-large-v2



(c) GCN

Figure 2. UMAP projections of the three embedding approaches. Yellow = White games; Purple = Black games.

5.2. Player Embedding and Game Embedding Evaluation

We split the dataset into 80% training and 20% test games, stratified by player to preserve representation across both sets.

To evaluate the quality of each game embedding method, we tested how well they enable identification of players based on their games. For each method (SBERT, e5-large-v2, and GCN), we clustered the training games using K-Means with k varying from 2 to 60. Each player was embedded as the normalized distribution of their games across the k clusters.

For example, if 80% of a player's games fall into cluster 0 and 20% into cluster 1, their embedding is $\mathbf{p} = [0.8, 0.2]$.

Test set games were embedded using the same process, and the predicted player identity was obtained via cosine similarity with the training player embeddings. Accuracy was computed by measuring how often the closest match was the correct player. Since Louis-Charles Mahé de La Bourdonnais had no valid games, we evaluated over 49 players.

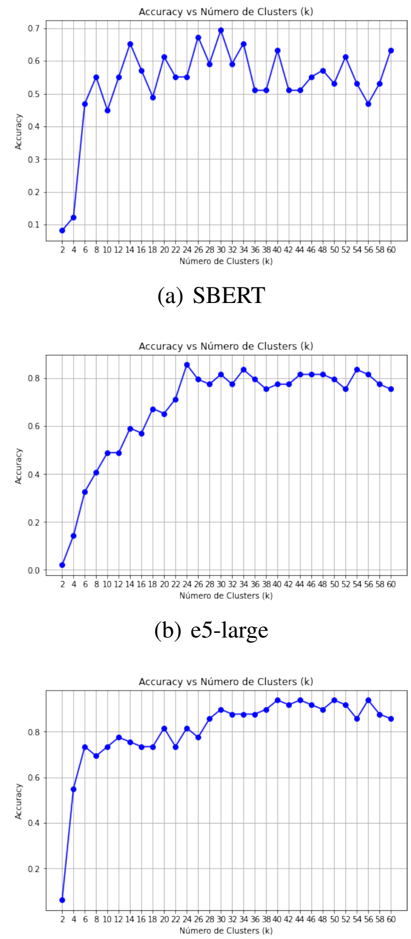
Although we initially used the elbow method (based on intra-cluster SSE) to estimate a suitable number of clusters k, this criterion did not align with downstream accuracy. Because our primary goal was to produce robust player embeddings for recommendation, we ultimately selected k based on top-1 classification accuracy rather than inertia minimization.

Figure 3 shows accuracy as a function of k for each embedding method. As an example, Figure 4 shows the elbow curve for GCN embeddings, where the SSE flattens early but accuracy continues to improve.

The results of our embedding comparison are summarized in Table 1. Beyond accuracy, we also report the file size of the generated embeddings and the computation time required. These additional metrics are critical for our goal of building an efficient recommender system that can scale to large datasets and provide fast responses to users.

Table 1. Embedding method comparison

Method	Optimal k	Accuracy (%)	Time (h)	Size (GB)
SBERT	30	69.4	4	1.12
e5-large-v2	24	85.7	6	1.2
GCN	40, 44, 50 and 56	**93.88**	**0.8**	**0.1441**



(a) SBERT

(b) e5-large

(c) GCN

Figure 3. Top-1 accuracy for identifying players based on their embedding, across different values of k (number of clusters).

5.3. Player Style Clustering

Based on our previous results, we fix the embedding method to the GCN with $k = 56$, which achieved the best performance across accuracy, computation time, and embedding size.

We apply a similar procedure to cluster player styles: using the training set player embeddings, we perform K-Means clustering to identify distinct style groups. Each test player embedding is then assigned to a style cluster based on proximity to the cluster centroids.

This approach allows us to evaluate whether, even if the exact player identity is not correctly predicted, the system can still correctly classify the player's style cluster. Such style-level prediction provides a useful fallback and further insight into player similarities beyond individual identities.

To improve clustering quality, we removed four players—Philidor, Morphy, Winawer, and Anderssen—who acted as outliers due to having significantly fewer games than the rest. Their limited data skewed the style distribution and negatively impacted cluster formation.

We evaluated the accuracy of style prediction using the remaining 45 players. After embedding and clustering with $k = 5$, we achieved **100% accuracy** in assigning the correct style cluster to each test player. This indicates that the learned style embeddings are highly discriminative and consistent, even under a reduced number of clusters and after the removal of outlier players.

To interpret the style clusters, we leveraged a Large Language Model (LLM) to generate preliminary descriptions based on the representative players in each group. These descriptions were then refined in collaboration with members of the chess team at Pontificia Universidad Católica de Chile, who provided domain expertise and practical insight. The final cluster interpretations are as follows:

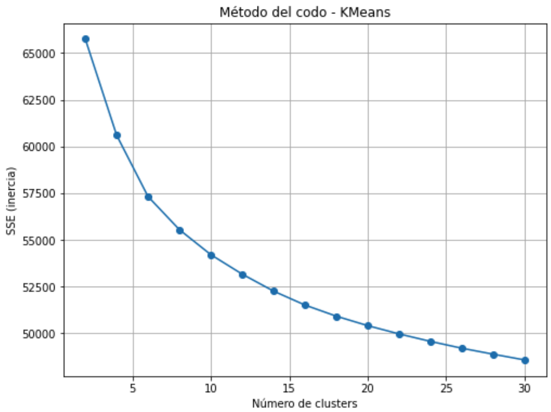


Figure 4. Elbow curve (within-cluster SSE) for GCN embeddings. While inertia flattens around $k = 8$, accuracy (Figure 3) continues improving beyond this point.

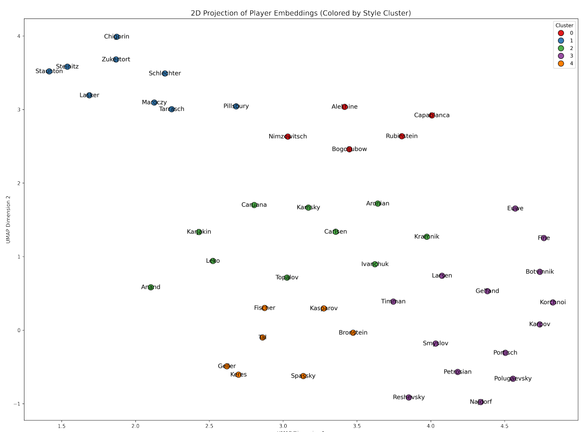


Figure 5. 2D UMAP projection of GCN-based player embeddings ($k = 5$), colored by style cluster. Each point represents a player, and proximity reflects similarity in game style. Clear separation suggests the embeddings capture consistent stylistic patterns.

- **Cluster 0 – "Pioneers of Modern Chess"**
  Description: Players who bridged classical and modern chess thought. Their style blends positional elegance with tactical creativity.
  Style: Structured creativity, refined technique, early hypermodern ideas.
  Examples: Alekhine, Bogoljubow, Capablanca, Nimzowitsch, Rubinstein

- **Cluster 1 – "Classical and Foundational School"**
  Description: Founding figures of positional chess, emphasizing classical principles like central control and harmonious development.
  Style: Classical, scientific, rigorous.
  Examples: Lasker, Steinitz, Tarrasch, Staunton, Chigorin

- **Cluster 2 – "21st Century Universal Masters"**
  Description: Modern top players with a universal approach, characterized by high theoretical preparation, precision, and adaptability.
  Style: Universal, multifaceted, technically adaptive.
  Examples: Carlsen, Anand, Kramnik, Caruana, Topalov

- **Cluster 3 – "Soviet Positional School"**
  Description: Strategically grounded players from the Soviet tradition, with deep understanding of middlegame structures and positional play.
  Style: Positional, structural, methodical.
  Examples: Botvinnik, Karpov, Petrosian, Smyslov, Gelfand

- **Cluster 4 – "Creative and Aggressive Geniuses"**
  Description: Brilliant attackers and intuitive calculators, known for dynamic, unpredictable, and spectacular play.
  Style: Tactical, aggressive, intuitive.
  Examples: Fischer, Kasparov, Tal, Bronstein, Spassky

5.4. Recommender System

The final recommender system is designed to provide personalized training material based on a user's playing style. It takes four inputs:

- Name of the user: Used only to filter the PGN file and extract the moves played by the user, ignoring the opponent's moves.
- A PGN file containing the user's games,
- An integer n specifying the number of similar professional players to recommend,
- An integer m indicating the number of games to recommend per matched player.

The pipeline proceeds as follows:

1. **Game embedding:** The user's PGN file is parsed, and only the moves played by the user are extracted. Each game is embedded using the pretrained GCN model with fixed random seed for reproducibility.
2. **Player embedding:** Using the same k-means model previously trained on professional games, the user's games are assigned to clusters. The user's player embedding is computed as the distribution of their games across the k clusters (e.g., for $k = 56$).
3. **Style assignment:** The user embedding is then passed through the style clustering model (trained on professional player embeddings). The predicted style cluster

is returned, along with its human-interpretable description (see previous section).

4. **Player recommendation:** The cosine similarity between the user embedding and the professional embeddings is computed, returning the top-n most similar players along with their similarity scores.

5. **Game recommendation:** To recommend specific instructive games, we first compute the mean of the user's individual game embeddings. For each of the top-n recommended players, we retrieve the top-m games that are closest (via cosine similarity) to this mean vector.

This system offers tailored game suggestions that match both the user's style and preferences, providing a practical and intuitive way to study chess through strategically aligned examples.

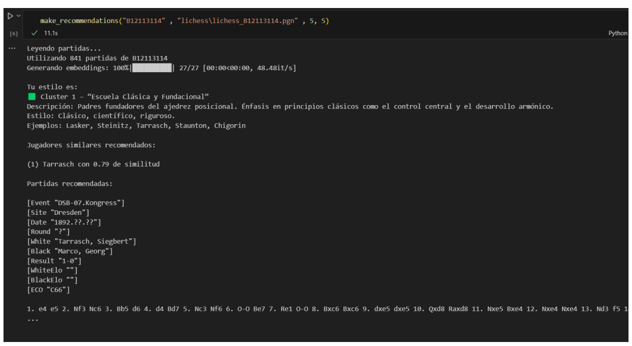


Figure 6. Example of the personalized game recommendation interface. The system shows the user's assigned playing style, the top recommended professional players, and the selected instructive games for study.

6. Conclusion

In this work, we introduced a novel embedding approach based on cluster distributions that effectively captures player style from chess games. Our experiments indicate that representing chess games as graphs, rather than purely as text sequences, yields superior embeddings for style analysis and player identification.

This cluster-based embedding method is not limited to chess; it can be generalized to other domains where complex entities are composed of multiple components or sequences.

The ChessMatch-V2 system demonstrates strong performance in offline evaluations, successfully matching users to professional players and recommending instructive games aligned with their style. Initial feedback from users has been positive, highlighting the system's potential as a personalized chess learning tool.

Limitations. All metrics reported are based on offline data. As ChessMatch-V2 is a new pipeline, we lack interaction data such as click logs or user ratings needed for standard recommender system metrics (e.g., Precision@k, Recall@k, nDCG, novelty, or serendipity). A live user study would be necessary to assess its impact more comprehensively—recording game engagement and gathering qualitative feedback.

Future work. A key direction for improvement is to develop embedding strategies that incorporate both the player's and the opponent's moves, enabling richer modeling of game dynamics while preserving the player's stylistic identity and avoiding opponent-induced drift. This could result in more nuanced and robust embeddings that better isolate the characteristics of the player of interest. Additionally, deploying the system in a live environment would allow us to collect real usage data, refine the recommendations, and evaluate impact through user engagement metrics.

References

Chen, W., Shen, Z., Pan, Y., Tan, K., and Wang, C. Applying machine learning algorithm to optimize personalized education recommendation system. *Journal of Technology and Performance in Engineering Systems (JTPES)*, 4(1), 2024. doi: 10.53469/jtpes.2024.04(01).14. URL `https://doi.org/10.53469/jtpes.2024.04(01).14`. Accessed: 2025-07-05.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*, 2017. URL `https://arxiv.org/abs/1609.02907`. arXiv:1609.02907.

Müller, K. and Engel, L. The 4 player types standard model. `https://shop.chessbase.com/en/products/engel_mueller_player_types_model?ref=RF5-TYB55LUOVF`, 2022. Accessed: 2025-07-05.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP–IJCNLP)*, pp. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. URL `https://aclanthology.org/D19-1410`.

Stöckl, A. Chess2vec: A map of chess moves. `https://medium.com/data-science/chess2vec-map-of-chess-moves-712906da4de9`, 2020. Accessed: 2025-07-06.

Varela, B. and Álvarez, E. Chessmatch (v1): Chess player recommender prototype. https://github.com/212113114/ChessMatch, 2024. Accessed: 2025-07-05.

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. Text embeddings by weakly supervised contrastive pretraining. https://arxiv.org/abs/2212.03533, 2024. arXiv:2212.03533.

Zhuang, Y., Zeng, L., Zhuang, J., Wang, S., and Liu, D. Ek-chess: Chess learning system based on top-level chess expert knowledge graph. In *2023 International Conference on Artificial Intelligence and Education (ICAIE)*, pp. 123–128, 2023.