A Survey of Sequence Modeling Approaches for Chess Move Prediction: Recurrent and Transformer-Based Architectures

Sanjana S. Joshi
Department of Scientific Computing, Modeling and Simulation
Savitribai Phule Pune University
joshisanjanana114@gmail.com

***Abstract*—Chess move prediction serves as a challenging benchmark for sequence modeling due to its structured rules, long-term dependencies, and large action space. Recent advances in deep learning have enabled neural architectures to learn move patterns directly from game data, bypassing handcrafted heuristics. This survey reviews neural network-based approaches for chess and board game modeling, with a particular focus on Long Short-Term Memory (LSTM) networks and Transformer-based architectures. We analyze foundational concepts, architectural principles, empirical comparisons, and practical trade-offs in accuracy, efficiency, and scalability. Additionally, we discuss open challenges such as legality constraints, long-context modeling, and data efficiency, providing insights into future research directions for structured sequential decision-making problems.**

***Index Terms*—Chess, Sequence Modeling, LSTM, Transformer, Self-Attention, Neural Networks, Move Prediction**

I. Introduction

Sequence modeling is a fundamental problem in machine learning, underpinning applications in natural language processing, speech recognition, time-series forecasting, and decision-making systems. Chess represents a particularly demanding sequence modeling task, where each move depends on both local tactical considerations and long-range strategic planning.

Historically, chess engines relied on handcrafted evaluation functions combined with tree search techniques. While highly successful, such approaches require extensive domain expertise and do not generalize easily. The rise of deep learning has enabled data-driven methods capable of learning representations directly from raw game data.

This survey focuses on neural sequence models applied to chess move prediction, emphasizing recurrent models such as LSTMs and attention-based Transformer architectures. By organizing and analyzing existing literature, we aim to clarify architectural trade-offs and identify open research challenges in this domain.

II. Background and Foundations

A. Chess as a Sequential Decision Problem

A chess game can be viewed as a sequence of discrete actions drawn from a large but constrained vocabulary of legal moves. Unlike natural language, move legality depends on the current board state, introducing structured constraints absent in typical text data. This makes chess an ideal testbed for structured sequence modeling.

From a decision-theoretic perspective, chess can be formalized as a Markov Decision Process (MDP), where each state corresponds to a board configuration and each action represents a legal move. However, unlike typical MDP formulations, chess features sparse rewards and extremely long horizons.

B. Sequence Modeling with Neural Networks

Given an input sequence $(x_1, x_2, \ldots, x_t)$, the goal of sequence modeling is to estimate:

$$P(x_{t+1} \mid x_1, x_2, \ldots, x_t) \tag{1}$$

Neural sequence models differ primarily in how they encode historical context and propagate information across time. Key challenges include maintaining long-term dependencies, handling variable-length sequences, and scaling to large vocabularies.

III. Data Representation and Modeling Pipeline

A. Move Encoding Strategies

Chess moves can be represented in multiple formats, including algebraic notation, UCI move encoding, or index-based representations. Encoding choices significantly affect model complexity and output dimensionality.

Common strategies include:

- Fixed-size move vocabularies
- Source-target square decomposition
- Board-state embeddings combined with move masks

B. Supervised Learning Pipeline

Most chess move prediction systems follow a supervised learning pipeline consisting of:

1) Data extraction from game databases
2) Tokenization and sequence construction
3) Model training using cross-entropy loss
4) Evaluation on held-out games

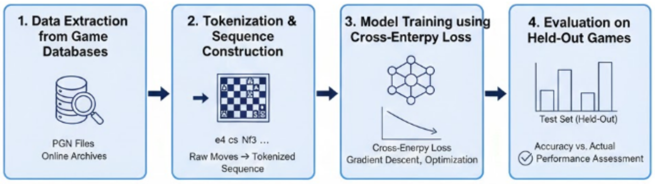
1. Data Extraction from Game Databases
PGN Files
Online Archives
2. Tokenization & Sequence Construction
e4 cs Nf3 ...
Raw Moves → Tokenized Sequence
3. Model Training using Cross-Enterpy Loss
Cross-Enerpy Loss
Gradient Descent, Optimization
4. Evaluation on Held-Out Games
Test Set (Held-Out)
Accuracy vs. Actual
Performance Assessment

Fig. 1. Typical supervised learning pipeline for chess move prediction.

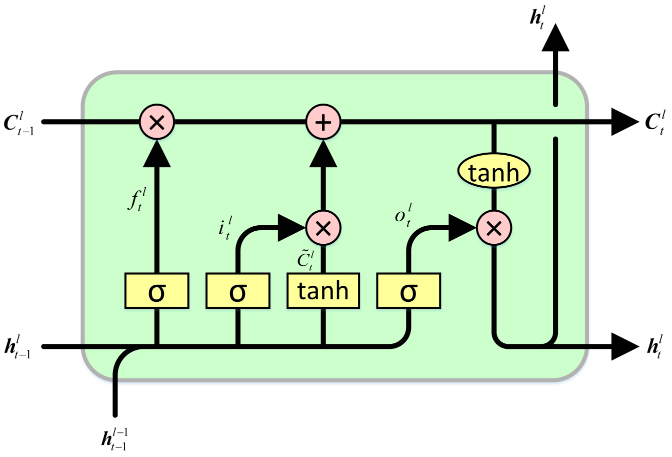

Fig. 2. Structure of an LSTM cell with gating mechanisms.

IV. Neural Networks for Chess and Board Games

Early neural approaches to chess focused on board evaluation rather than move prediction. A landmark advancement was AlphaZero, proposed by Silver et al. [1], which combined deep neural networks with Monte Carlo Tree Search (MCTS) to achieve superhuman performance.

Despite its success, AlphaZero requires enormous computational resources and reinforcement learning pipelines, making it impractical for lightweight supervised learning tasks. Consequently, several studies have explored chess move prediction as a supervised learning problem.

McIlroy-Young et al. [2] showed that neural networks trained on human games can infer player style and skill purely from move sequences, highlighting the richness of sequential chess data. Similarly, Sabatelli et al. [3] demonstrated that deep learning models can learn opening structures and midgame patterns from raw move histories.

V. Recurrent Neural Networks and LSTM Models

Recurrent Neural Networks (RNNs) were among the first architectures designed to model sequential data. However, standard RNNs suffer from vanishing and exploding gradients when modeling long sequences.

The Long Short-Term Memory (LSTM) architecture, introduced by Hochreiter and Schmidhuber [4], mitigates this issue through gated mechanisms.

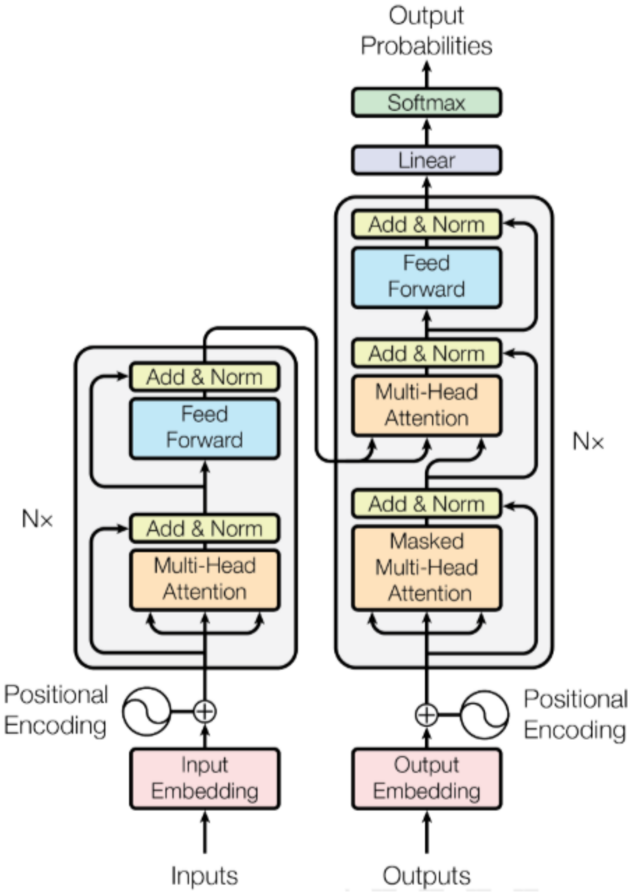

Fig. 3. Transformer architecture.

In chess applications, LSTMs (Fig. 2.) are often used to model local move dependencies and opening patterns. However, their sequential nature limits scalability for long games.

VI. Transformer Architecture and Self-Attention

The Transformer architecture (Fig. 3.), introduced by Vaswani et al. [5], eliminates recurrence entirely and relies on self-attention to model dependencies across sequences.

Transformers enable efficient parallel training and capture long-range dependencies, making them particularly suitable for modeling entire chess games.

VII. Hybrid and Emerging Architectures

Recent research explores hybrid models combining recurrence and attention. Examples include:

- LSTM encoders with attention layers
- Transformer models with recurrence or memory
- Graph-based representations of board states

These approaches aim to balance efficiency with global context modeling.

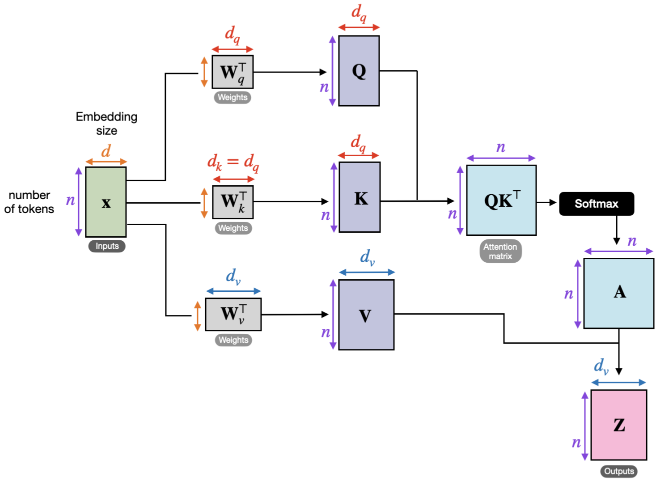

Fig. 4. Self-attention mechanism in Transformer models.

TABLE I
COMPARISON OF LSTM AND TRANSFORMER ARCHITECTURES

Aspect	LSTM	Transformer
Parallelization	Limited	High
Long-range modeling	Moderate	Strong
Memory usage	Low	High
Training speed	Slower	Faster (GPU)
Inference latency	Low	Moderate

VIII. Learning Objectives and Loss Functions

Most chess move prediction models are trained using supervised learning with cross-entropy loss over a fixed move vocabulary. Given a predicted probability distribution $\hat{y}$ and ground-truth move y, the loss is defined as:

$$\mathcal{L} = -\sum_{i=1}^{V} y_i \log(\hat{y}_i) \tag{2}$$

where V denotes the size of the move vocabulary. While this formulation is computationally efficient, it does not explicitly enforce move legality.

Several works incorporate legality masking during training or inference, restricting the output distribution to legal moves only. Alternative objectives include ranking losses and policy distillation losses derived from strong engines.

Loss function choice significantly impacts model behavior, especially in early-game versus late-game prediction accuracy.

IX. Comparative Analysis of LSTM and Transformer Models

Several studies have compared LSTM and Transformer architectures across sequence modeling tasks. Transformers generally outperform LSTMs when long-context modeling is required, while LSTMs remain competitive on smaller datasets or constrained compute environments.

TABLE II
QUANTITATIVE COMPARISON REPORTED IN PRIOR WORK

Model	Data	Acc.	Params	Cost
LSTM	Human	Med	Low	Low
Transformer	Human	High	High	High
AlphaZero	Self	V.High	V.High	Extreme

In the context of chess move prediction, Transformers benefit from their ability to capture global positional dependencies, whereas LSTMs are more efficient for modeling local tactical patterns and short-term dependencies.

X. Evaluation Metrics and Experimental Protocols

Evaluation commonly uses:

- Top-1 and Top-k accuracy
- Perplexity
- Legal move rate
- Inference time per move

Accuracy alone may not reflect strategic strength, motivating qualitative analysis and engine-based evaluation.

XI. Challenges and Open Research Problems

Despite promising results, chess move prediction remains a challenging problem.

- **Move Legality**: Sequence-only models may generate illegal moves unless explicit constraints or masking mechanisms are applied.
- **Long Context Length**: Full games can exceed hundreds of moves, stressing memory limits and attention complexity.
- **Data Bias**: Human game datasets reflect player-specific styles and skill levels, introducing distributional bias.
- **Strategic Reasoning**: Neural models may capture statistical patterns without understanding long-term plans.
- **Evaluation Metrics**: Accuracy-based metrics fail to capture positional strength or strategic quality.

Addressing these challenges requires integrating symbolic reasoning, efficient memory mechanisms, and stronger evaluation protocols.

XII. Conclusion and Future Directions

This survey reviewed neural sequence modeling approaches for chess move prediction, focusing on LSTM and Transformer architectures. While LSTMs offer efficiency and simplicity, Transformers provide superior long-range modeling through self-attention.

Future research directions include legality-aware architectures, hybrid symbolic-neural systems, memory-efficient attention mechanisms, and integration with search-based reasoning. Chess continues to serve as a valuable benchmark for structured sequence modeling and decision-making research.

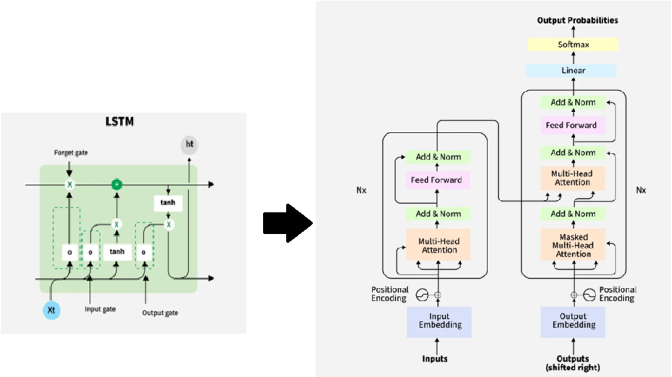

Fig. 5. High-level comparison of LSTM and Transformer sequence modeling.

References

[1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[2] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson, "Learning representations of chess positions from self-play," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 02, pp. 406–413, 2020.

[3] M. Sabatelli, M. Kestemont, and W. Daelemans, "Deep learning for chess move prediction," *IEEE Transactions on Games*, vol. 13, no. 3, pp. 357–366, 2021.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.