Đại học Quốc gia Hà Nội - Trường Đại học Khoa học Tự nhiên

AI chơi cờ vây

Nhóm 20

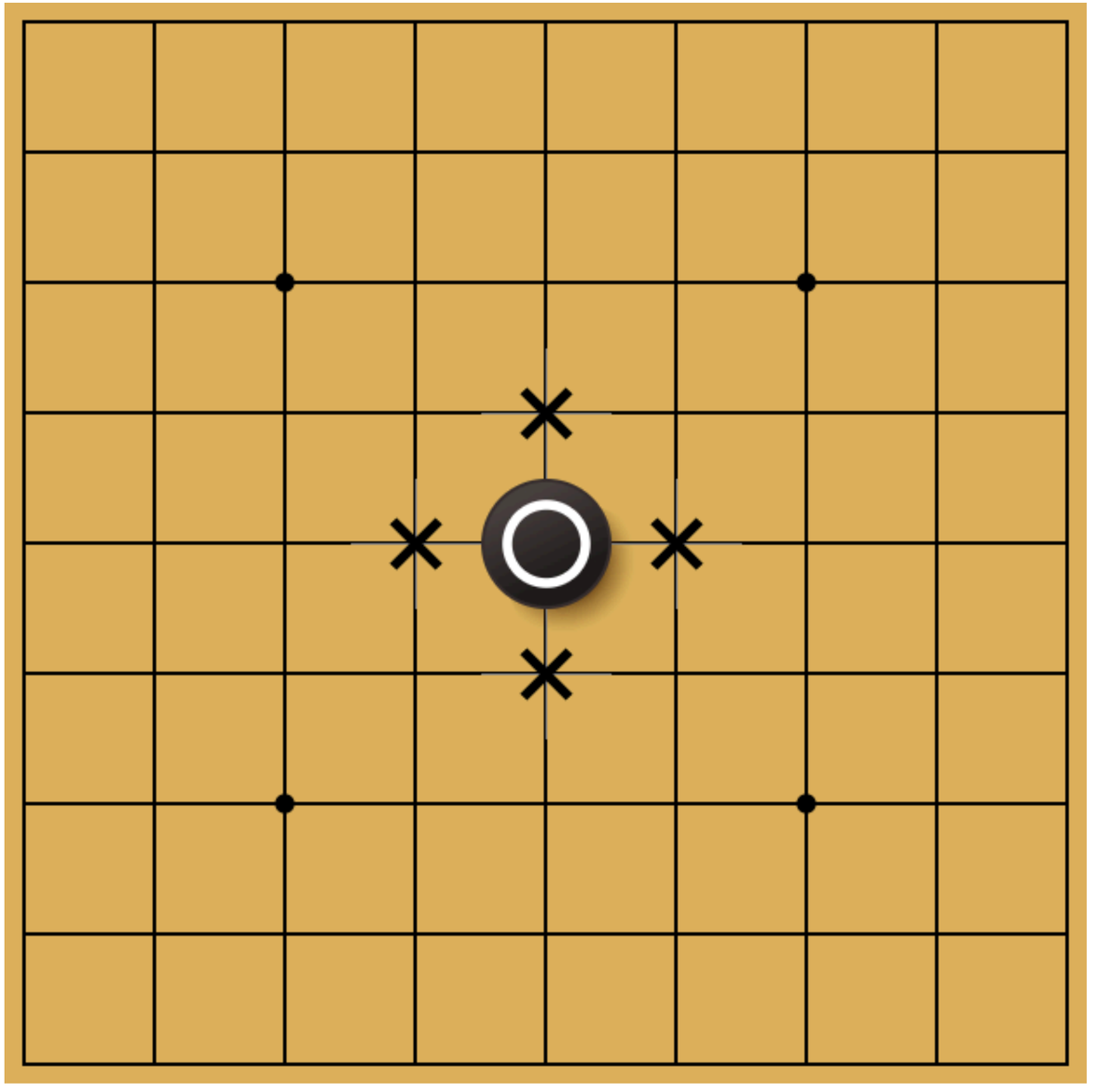
Hồ Quang Chung
Bùi Đức Hiếu
Nguyễn Khánh Toàn

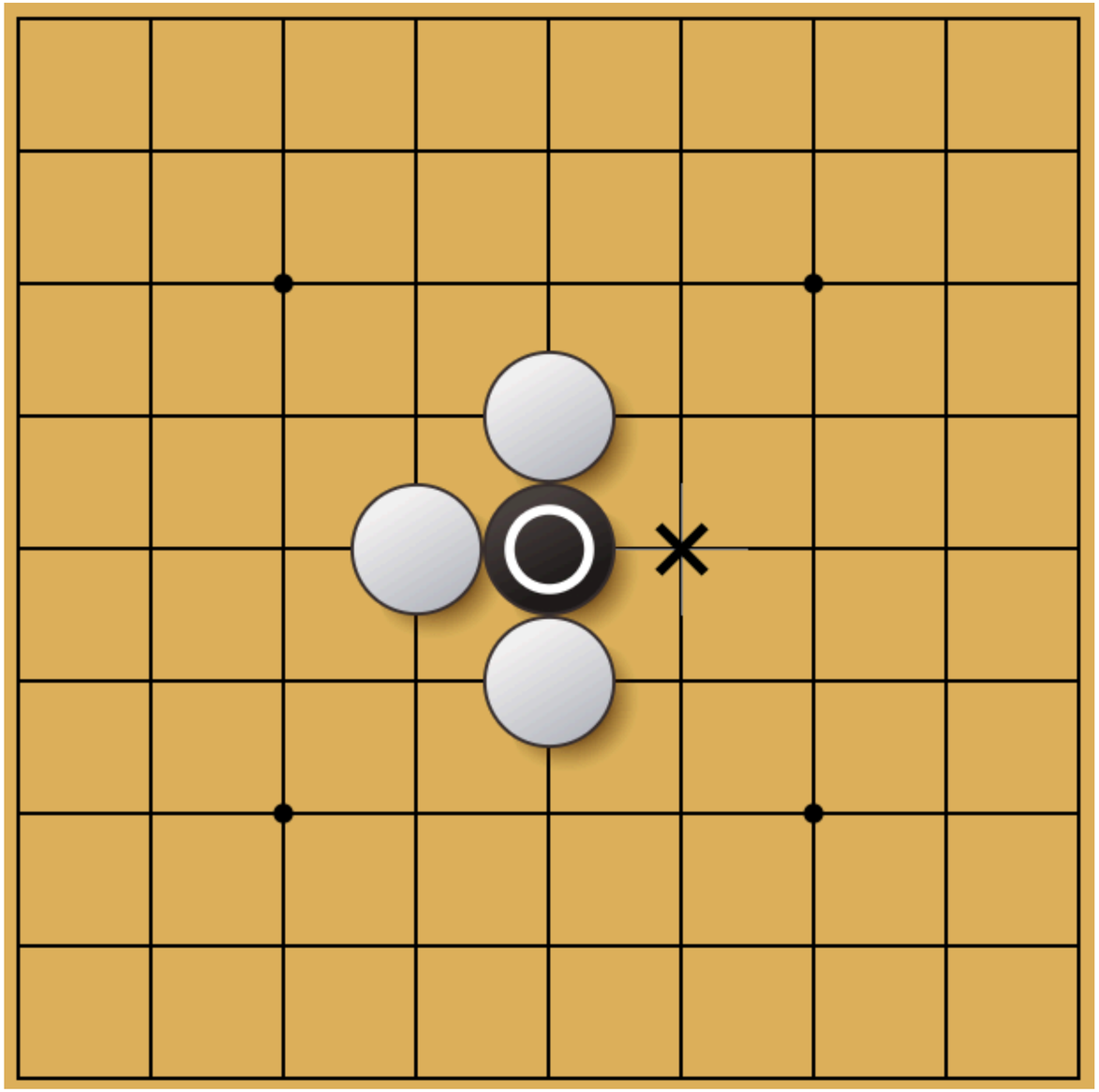
MAT3508 - Nhập môn Trí tuệ Nhân tạo

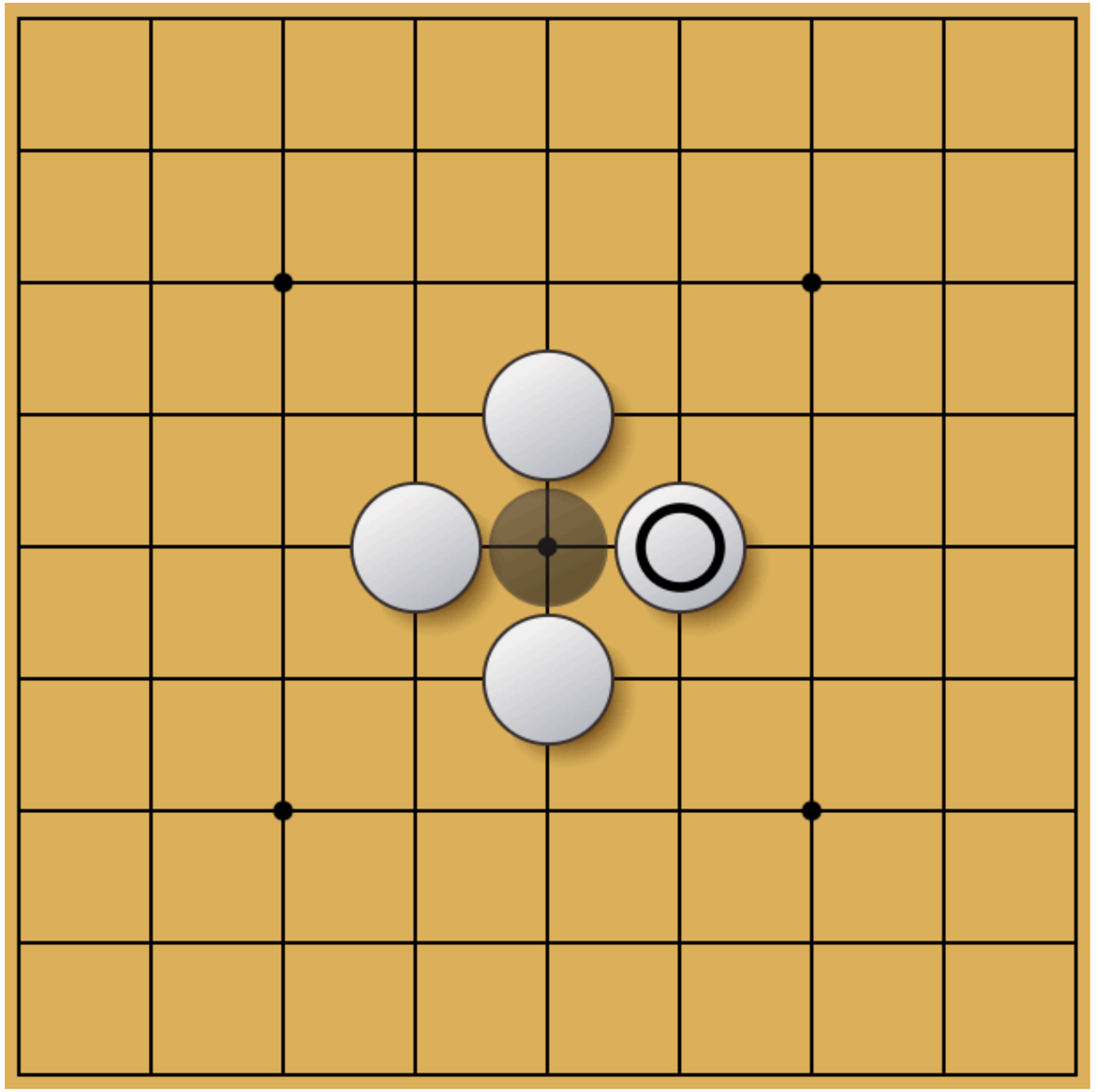
Đặt vấn đề

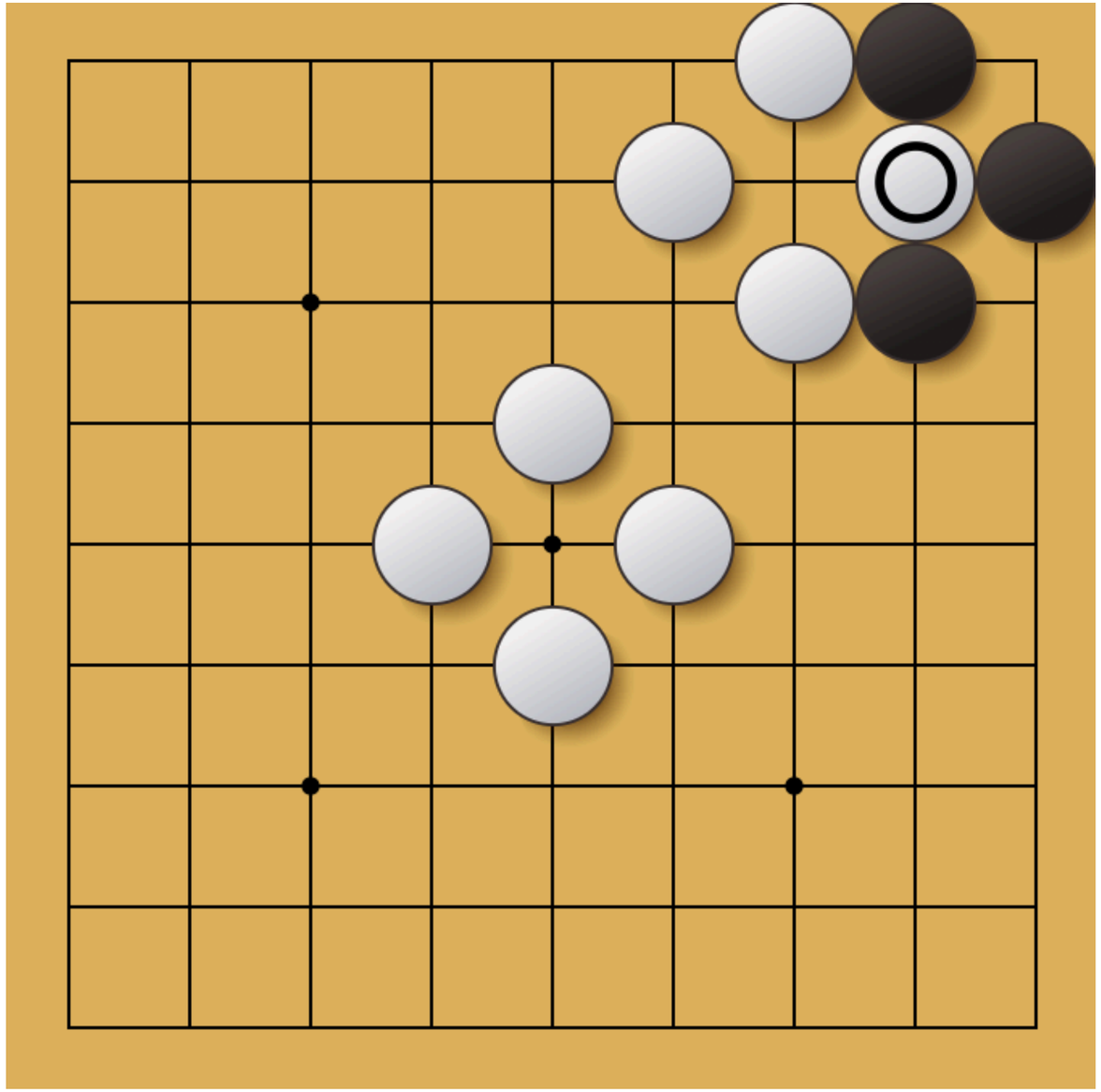
Trên bàn cờ 19 × 19, có tổng cộng 361 giao điểm. Ước tính số lượng trạng thái hợp lệ của bàn cờ lên đến khoảng 2,1 × 10^170, một con số khổng lồ vượt xa tổng số nguyên tử trong vũ trụ mà chúng ta quan sát được (khoảng 10^80)

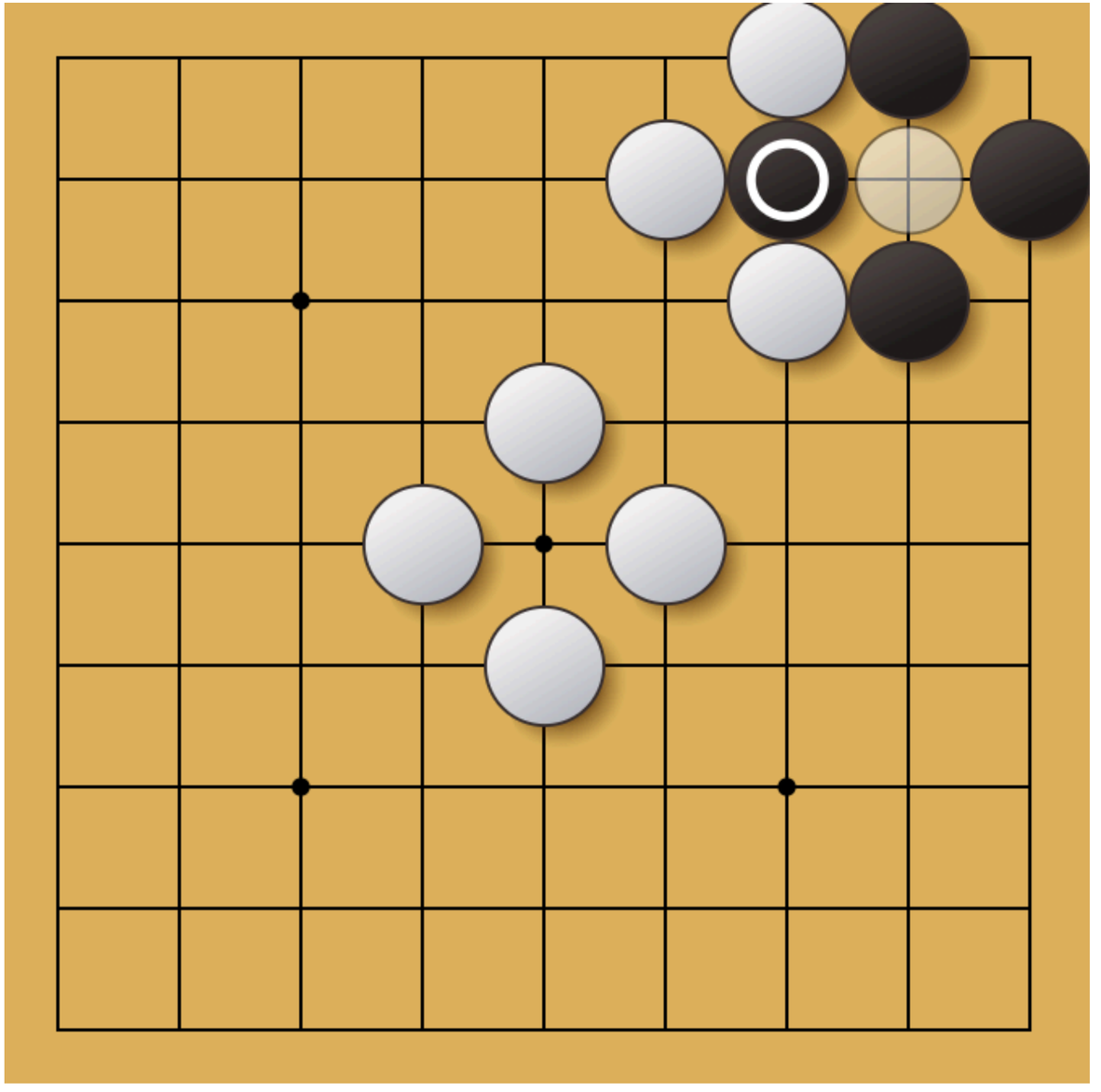
LUẬT CHƠI











Người chơi có thể chọn bỏ lượt. Khi cả 2 người cùng bỏ lượt sẽ đến phần tính điểm

Người ta thường chỉ đếm quân và đất của một bên (thường là bên Đen) để xác định kết quả:

- Nếu Tổng điểm của Đen (Đất + Quân) > 184.25 (thực tế là ≥ 185 trên bàn cờ): Đen Thắng.
- Nếu Tổng điểm của Đen ≤ 184.25: Trắng Thắng

02

Phương pháp

Table of contents - Phương pháp

Cơ sở lý thuyết

MTCS vs Alpha/Beta

Bùng nổ không gian trạng thái, MTCS duyệt cây theo xác suất

CNN + RESNET

Bàn cờ 19x19 có tính chất không gian tương tự một bức ảnh grayscale, và CNN xử lý rất tốt bài toán này. RESNET sẽ lo về vấn đề chiều sâu.

Mạng nơ-ron + MCTS (Biến thể PUCT)

. Mạng nơ-ron đóng vai trò "trực giác", còn MCTS đóng vai trò "suy luận logic".

Dual-head

Hiệu năng tính toán + Đa nhiệm

Kiến trúc hệ thống

Frontend - Reactjs

Đóng vai trò lớp Presentation. Xử lý tương tác người dùng, hiển thị bàn cờ, gửi dữ liệu nước đi và nhận kết quả từ Server

Backend - .NET 10

Đượcthiết kế theo mô hình Domain-Driven Design (DDD). Chịu trách nhiệm quản lý trạng thái ván đấu (State Management), thực thi luật cờ vây, và cung cấp API RESTful/WebSocket.

AI core

Module tích hợp trong Backend, sử dụng ONNX Runtime để chạy mô hình Deep Learning và thuật toán MCTS để đưa ra quyết định nước đi

Tiền xử lý dữ liệu

Biểu diễn dữ liệu (Input Features) Bàn cờ 19×19 được biểu diễn dướ idạng Tensor đa kênh(Multi-channel Tensor) với kích thước [Batch_Size,19,19,C]. Các kênh thông tin bao gồm:

Quân ta	1 nếu có quân, 0 nếu không.
Quân đối thủ	1 nếu có quân, 0 nếu không.
Vị trí trống	Đánh dấu các điểm chưa có quân
Lịch sử	Trạng thái bàn cờ tại các thời điểm t−1,t−2,... để giúp mạng nhận diện luật Ko (cướp lại)
Lượt đi	Plane chứa toàn số 1 (nếu Đen đi) hoặc 0 (nếu Trắng đi)

Dataset(github):
featurecat/go-dataset

Lọc:
Tập trung lọc các ván đấu của kỳ thủ chuyên nghiệp

Tổng quan về mô hình

3,329,125

Tông tham số (tất cả đều dùng được)

274.48

Total mult-adds (Units.GIGABYTES)

3999.93

Estimated Total Size (MB)

Nói thêm về biến thể PUCT

- **Policy Network P(s,a)** - Giảm chiều rộng: Giúp MCTS loại bỏ ngay lập tức hàng trăm nước đi vô nghĩa, chỉ tập trung khảo sát các nước đi có xác suất cao do mạng gợi ý.
- **Value Network V(s)** - Giảm chiều sâu: Thay vì phải mô phỏng đến hết ván cờ (Rollout) rất tốn thời gian, MCTS có thể dừng lại ở một nút lá và hỏi Value Network xem thế cờ đó ai đang ưu thế.

Công thức PUCT

Nước đi được chọn trong giai đoạn Selection dựa trên công thức tối đa hóa:

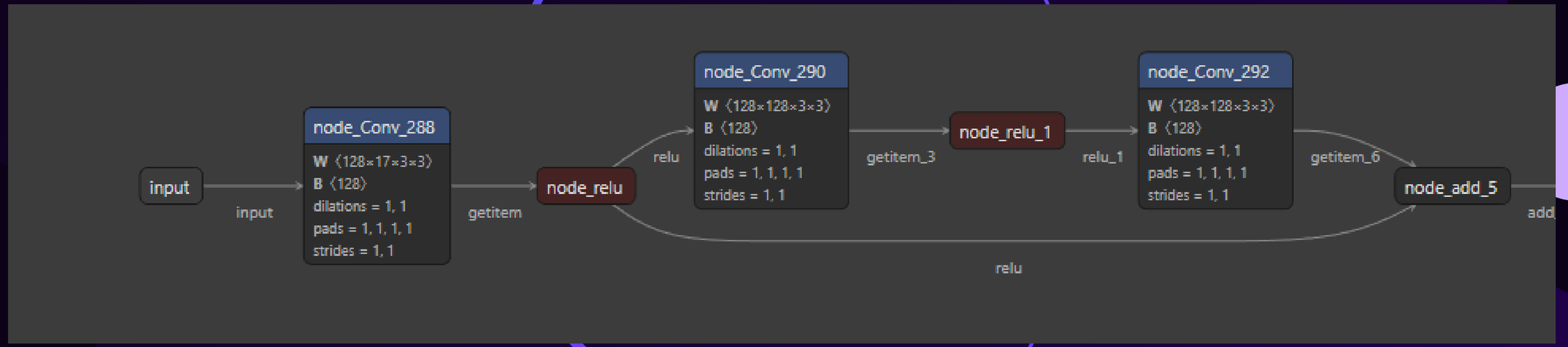
$$a_t = \text{argmax}_a(Q(s,a) + U(s,a)) \tag{2.5}$$

Với thành phần Upper Confidence Bound (U) khuyến khích khám phá:

$$U(s,a) = C_{puct} \cdot P(s,a) \cdot \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,a)} \tag{2.6}$$

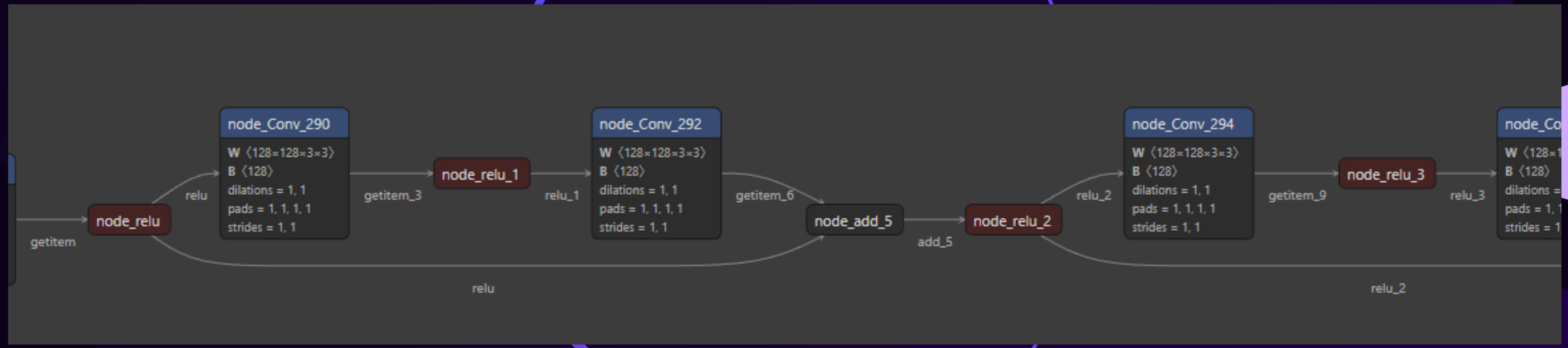
Trong đó:

- $Q(s,a)$: Giá trị trung bình (Mean Action Value).
- $P(s,a)$: Xác suất tiên nghiệm từ Policy Head.
- $N(s,a)$: Số lần nước đi a đã được thăm.



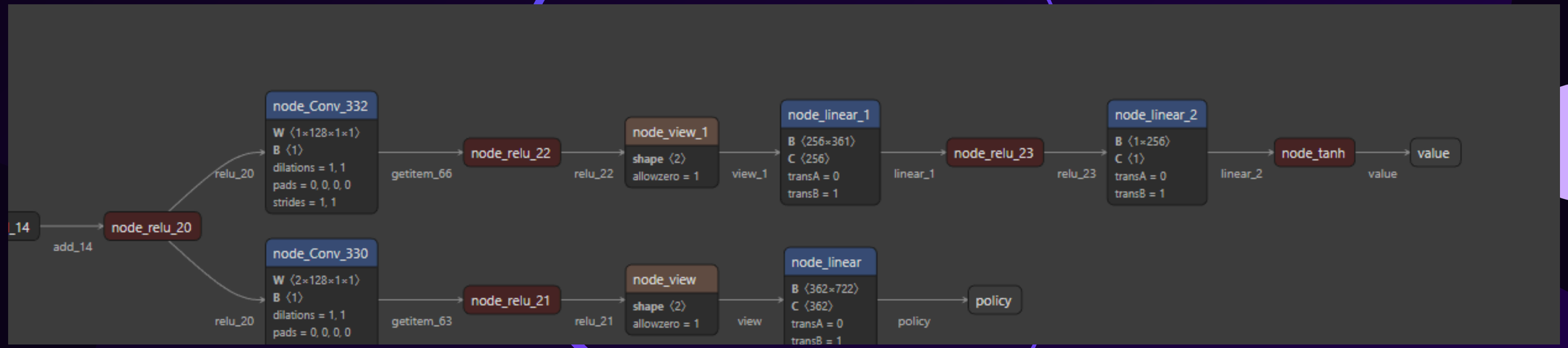
Input

Tensor có kích thước chuẩn là
[Batch_Size, 17, 19, 19]



Phần thân

10 Khối Dư (Residual Blocks)



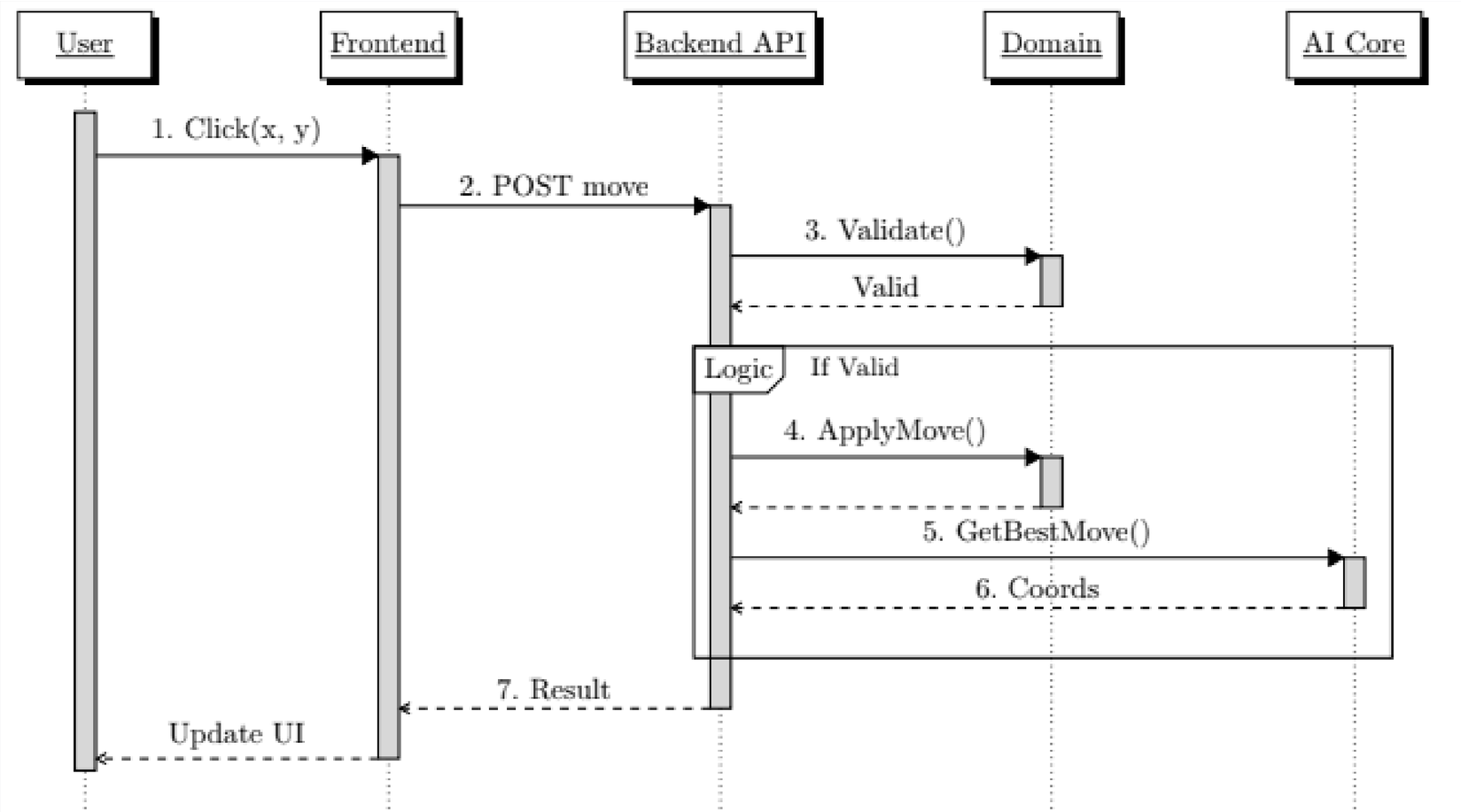
Phần đầu ra

Rẽ nhánh

Danh sách công nghệ

Thành phần	Công nghệ / Thư viện	Phiên bản (Dự kiến)
Frontend	ReactJS, TypeScript, Vite	React 18+, Vite 5.x
UI/UX	TailwindCSS, HTML5 Canvas (Render bàn cờ)	v3.4+
Backend Runtime	.NET 10 (C#)	Preview/Stable
Kiến trúc Backend	Domain-Driven Design (DDD), Clean Architecture	-
AI Inference	ONNX Runtime (CPU/GPU)	Microsoft.ML.OnnxRuntime
AI Training	Python, PyTorch, NumPy	PyTorch 2.x
Database	PostgreSQL hoặc SQLite (Dev)	Latest

Luồng xử lý nước đi



Hình 2.1: Biểu đồ tuần tự xử lý một lượt chơi

Quy trình huấn luyện

1

Tiền xử lý

Nguồn dữ liệu và Lọc

2

Huấn luyện Mô hình

Xây dựng kiến trúc mạng
và fine-tune

3

Đánh giá

Đánh giá mô hình đã đủ để
đưa vào hoạt động hay
chưa

4

Tích hợp Backend

Đây là bước "ghép não" AI
vào hệ thống game.

Tiền xử lý dữ liệu

Trích xuất đặc trưng (Feature Extraction)

Mỗi trạng thái bàn cờ được biểu diễn dưới dạng Tensor kích thước [17,19,19].

- 8 kênh đầu: Vị trí quân của người chơi hiện tại (tại thời điểm t,t−1,...,t−7). Lịch sử giúp nhận diện luật Ko và trạng thái động. 17
- 8 kênh tiếp theo: Vị trí quân của đối thủ (tương tự lịch sử 8 nước).
- 1 kênh cuối: Màu quân đi lượt này (Toàn 1 nếu Đen, toàn 0 nếu Trắng)

Kỹ thuật tối ưu bộ nhớ (Chunking & Lazy Loading)

Do dữ liệu sau khi giải nén rất lớn (hàng GB đến hàng chục GB), có khả năng không thể nạp toàn bộ vào RAM.

- Chunking: Lưu dữ liệu thành nhiều file nhỏ (ví dụ: features_0.npy, features_1.npy...), mỗi file chứa khoảng 2000-5000 ván.
- Memory Mapping: Sử dụng chế độ mmap_mode='r' của NumPy để đọc dữ liệu trực tiếp từ ổ cứng khi cần thiết, giữ mức tiêu thụ RAM ổn định

Training

Kiến trúc mạng

Sử dụng kiến trúc ResNet Dual-head lấy cảm hứng từ AlphaGo Zero.

Hàm mất mát

Mục tiêu tối ưu hóa tổng hợp: L(total) = L(policy) + L(value)

Fine-tuning

Khi cần cập nhật model với dữ liệu mới hoặc phong cách chơi mới

3

Đánh giá

Các chỉ số đánh giá

Policy Top-1 Accuracy

Tỷ lệ phần trăm số lần nước đi có xác suất cao nhất do AI dự đoán trùng khớp hoàn toàn với nước đi thực tế của kỳ thủ chuyên nghiệp

Policy Top-5 Accuracy

Tỷ lệ phần trăm số lần nước đi của kỳ thủ chuyên nghiệp nằm trong nhóm 5 nước đi có xác suất cao nhất do AI đề xuất

MSE

Sai số bình phương trung bình giữa giá trị dự đoán $v \in [-1,1]$ và kết quả thực tế $z \in \{-1,1\}$.

Value Sign Accuracy

Tỷ lệ số lần AI dự đoán đúng phe chiến thắng (cùng dấu với kết quả thực tế)

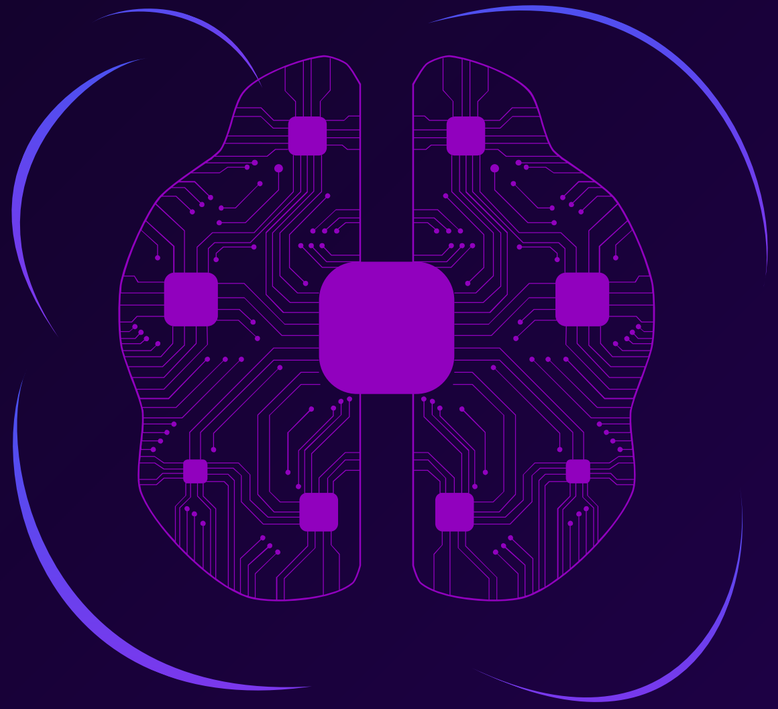
Mục tiêu cần đạt

>40%
Policy Top-1

>60%
Value Sign Acc

>75%
Policy Top-5

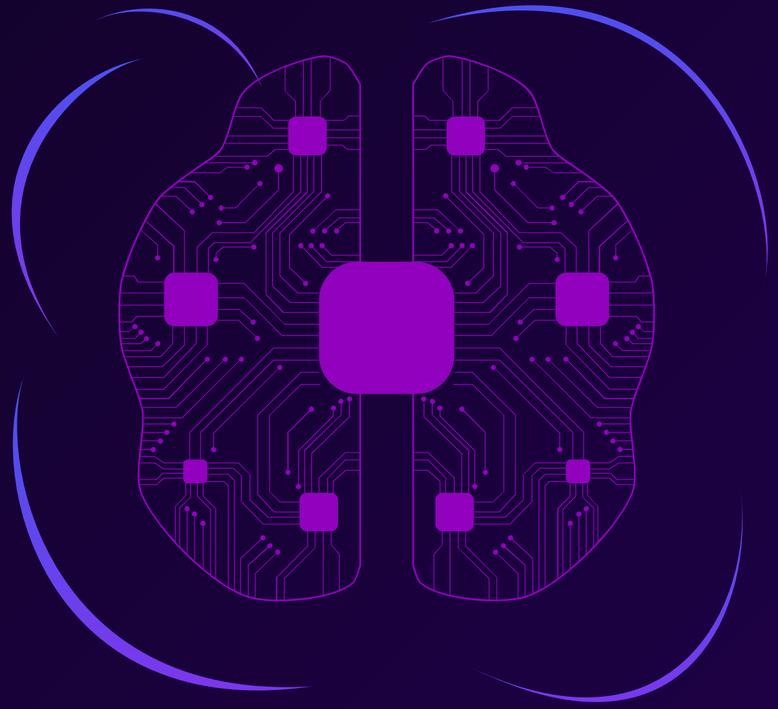
Kết quả huấn luyện



57,4%

Policy Top-1

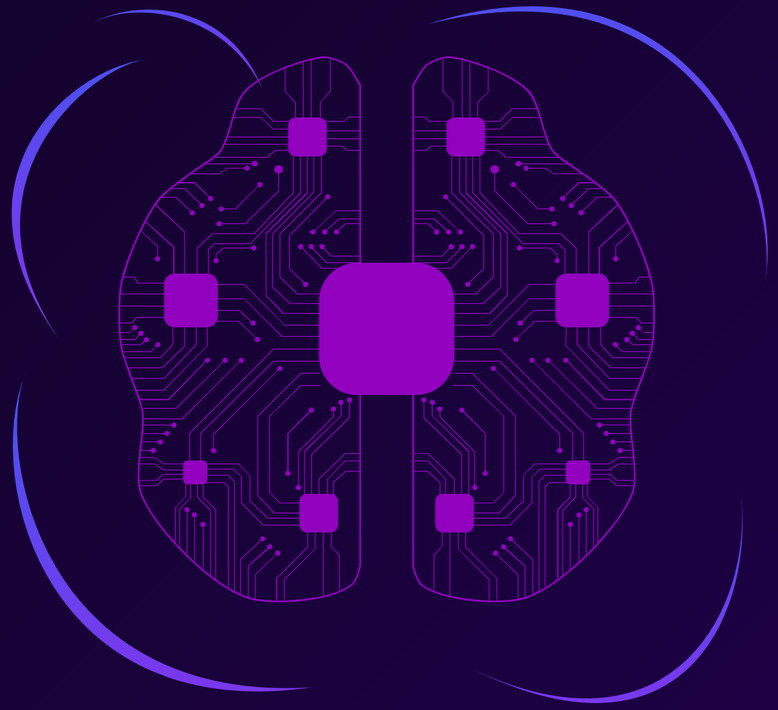
Kết quả huấn luyện



87.7%

Policy Top-5

Kết quả huấn luyện

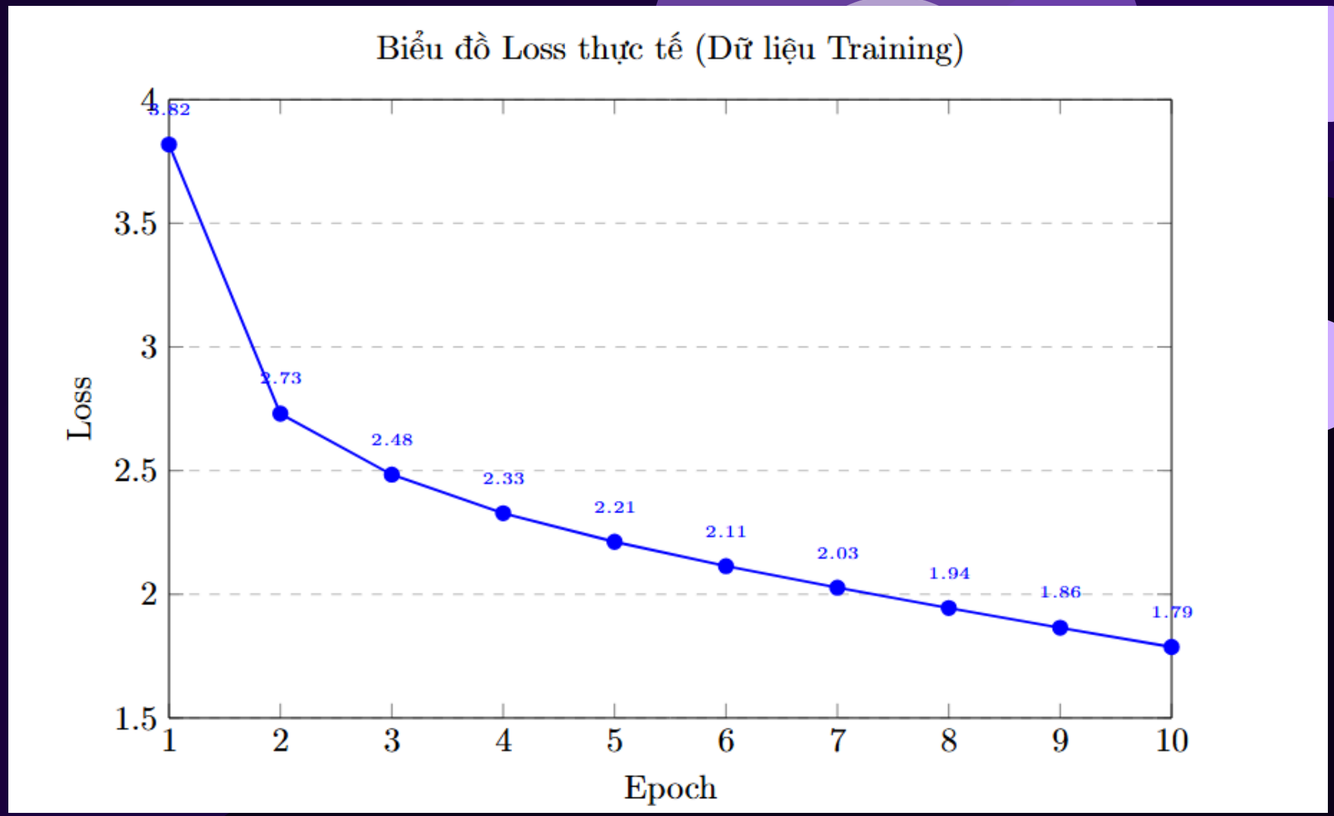


97.0%

Value Accuracy

Chỉ số qua các epoch

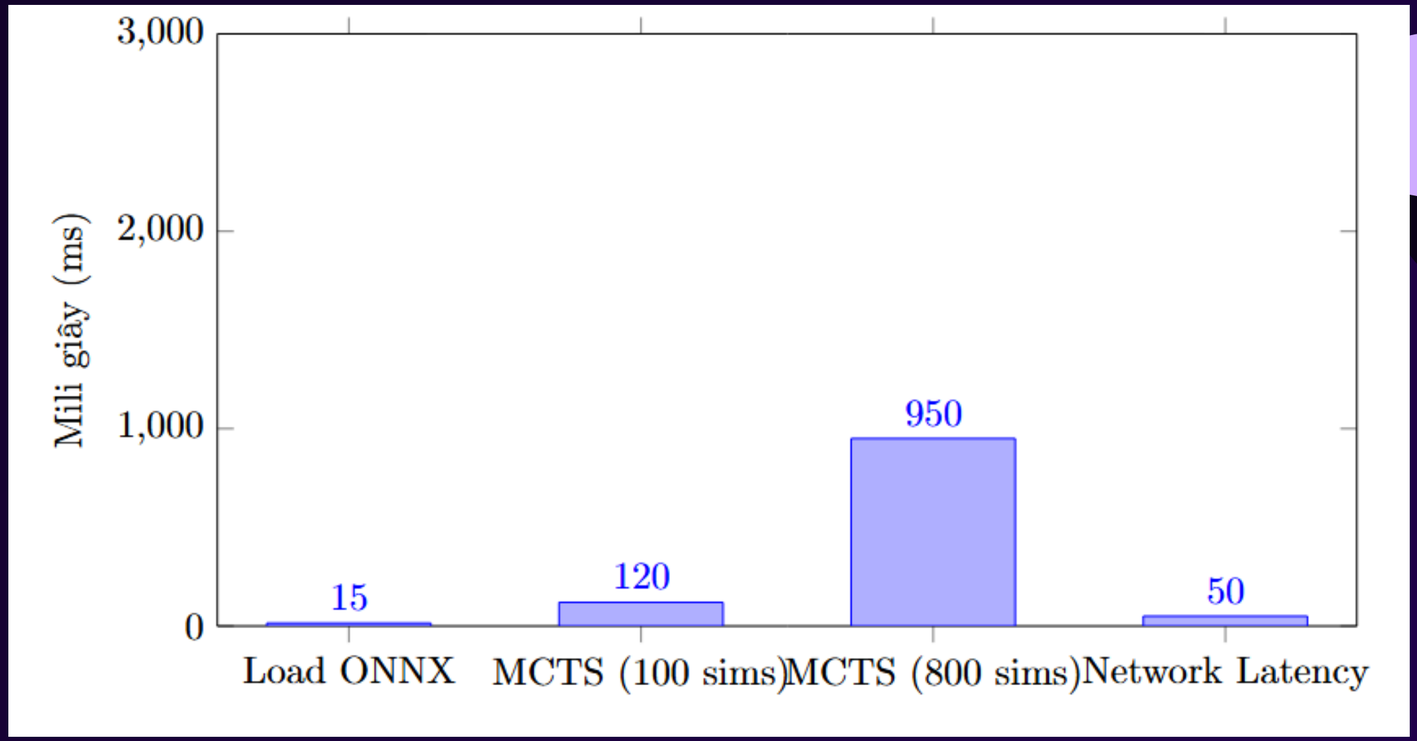
Quá trình huấn luyện được thực hiện với Batch Size = 256 trên Google Colab T4 GPU. Kết quả ghi nhận trong 10 Epochs đầu tiên cho thấy sự hội tụ tích cực của mô hình



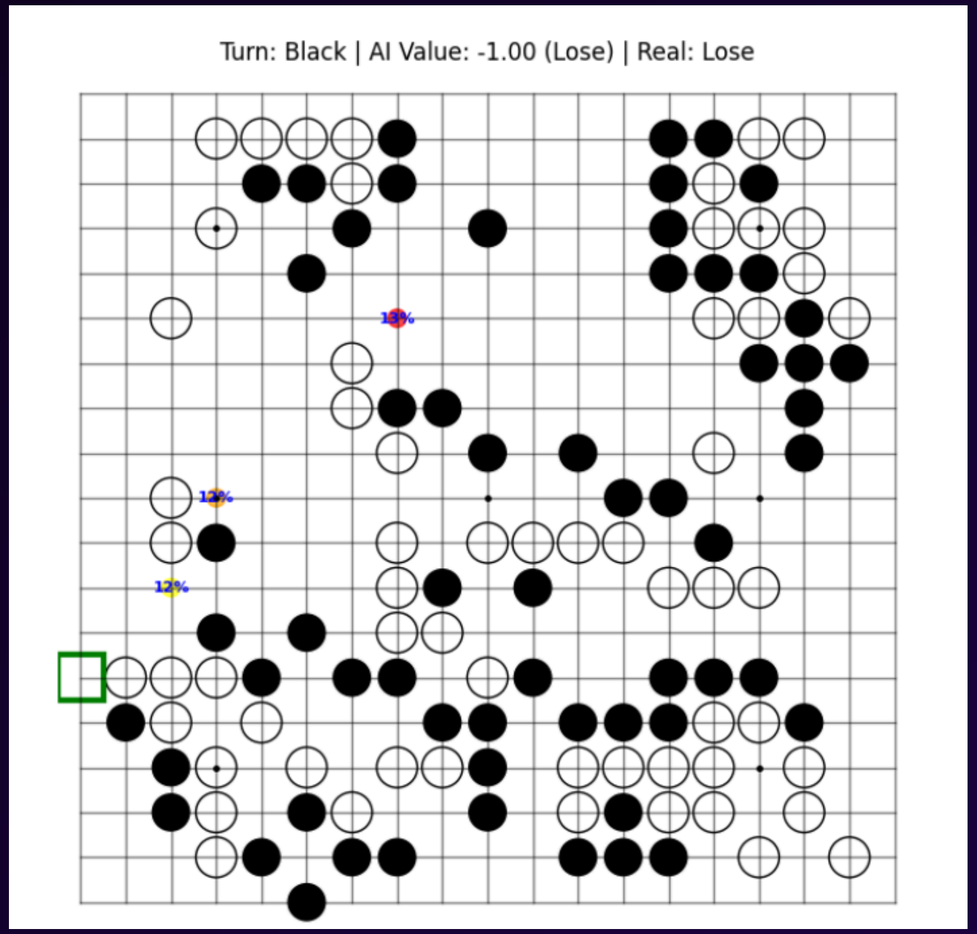
Epoch	Policy Top-1 (%)	Policy Top-5 (%)	Value Accuracy (%)	Value MSE
1	22.5%	45.1%	52.3%	0.401
3	35.8%	62.4%	58.1%	0.302
5	41.2%	70.5%	70.7%	0.254
8	46.8%	75.8%	85.4%	0.112
10 (Final)	**57.4%**	**87.7%**	**97.0%**	**0.074**

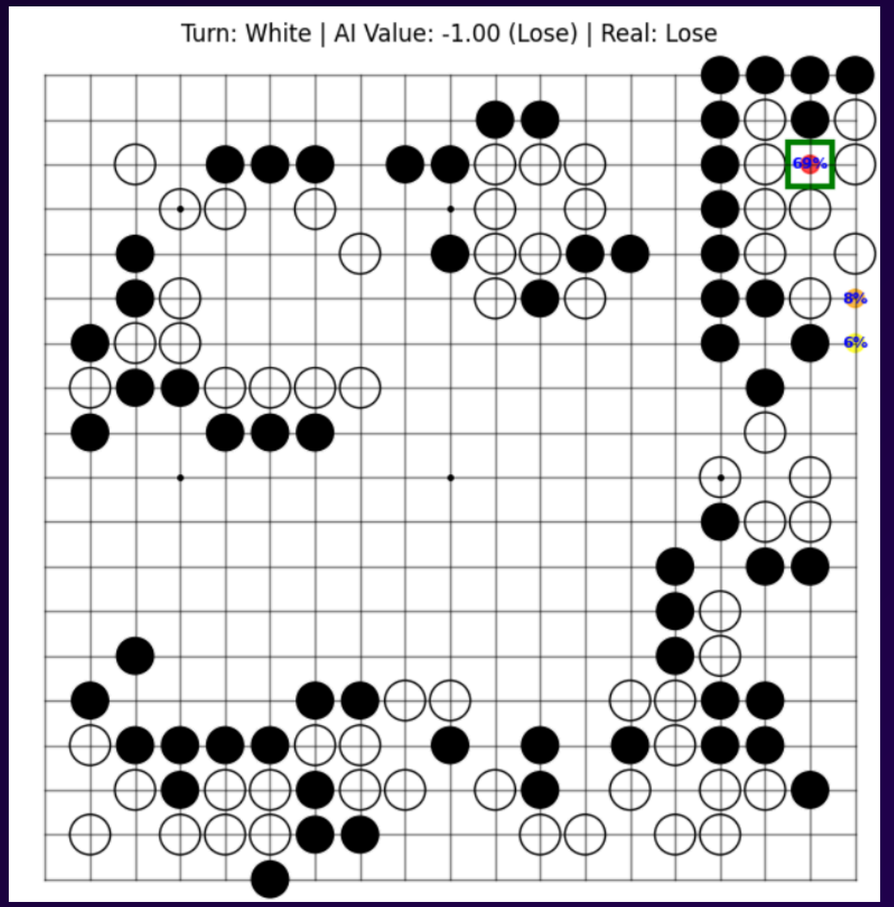
Mô hình dự đoán cực kỳ chính xác nước đi là tốt hay xấu cho thế cờ khi đến lần train thứ 10.

- Thời gian thực thi MCTS với 800 simulations là khoảng 950ms (gần 1 giây). Đây là mức thời gian chấp nhận được cho trải nghiệm người dùng Web thời gian thực.



- Việc sử dụng thư viện Microsoft.ML.OnnxRuntime trên .NET cho tốc độ inference trung bình khoảng 1.2ms/batch (với Batch=1 trên CPU), đảm bảo khả năng mở rộng khi có nhiều người chơi cùng lúc


Turn: Black | AI Value: -1.00 (Lose) | Real: Lose
13%
12%
12%


Turn: White | AI Value: -1.00 (Lose) | Real: Lose
8%
6%

Kết luận

- Kết quả huấn luyện xác nhận mô hình AI Cờ Vây hiệu quả và tin cậy. Mô hình ResNet Dual-head bắt chước hành vi chuyên nghiệp tốt và đánh giá tình thế vượt trội, hỗ trợ thuật toán MCTS.
- Mô hình đạt cân bằng giữa "Trực giác" (Policy Network 46% accuracy) và "Suy luận" (MCTS 800 sims). Hệ thống hoạt động ổn định trên .NET 10, sẵn sàng cho ứng dụng cờ vây trực tuyến bán chuyên nghiệp và mở rộng chế độ chơi.

Hướng phát triển

1. **Mở rộng Trải nghiệm Người dùng (UX Community):**
- Chế độ chơi giữa người với người qua server.
- Tính năng: Bảng xếp hạng, hệ thống Elo, lịch sử ván đấu.
- Chế độ phân tích ván đấu: AI phân tích và cung cấp điểm số, nước đi tối ưu.
- Trình chỉnh sửa ván cờ: Tải lên, lưu trữ, xem lại ván đấu định dạng SGF.
2. **Công cụ Huấn luyện:**
- Học theo chủ đề: Bài tập về sống/chết và khai cuộc với AI.
- Tính năng: Kiểm tra giải đố, gợi ý, hiển thị giải pháp tối ưu.
- Chế độ gợi ý và giải thích: AI cung cấp nước đi tốt nhất và giải thích.
3. **Tối ưu hóa Hiệu năng Tech Stack:**
- Triển khai phân tán: Kiến trúc microservices và GPU/TPU cho MCTS.
- Huấn luyện lại mô hình: Cải thiện AI qua chơi tự động.