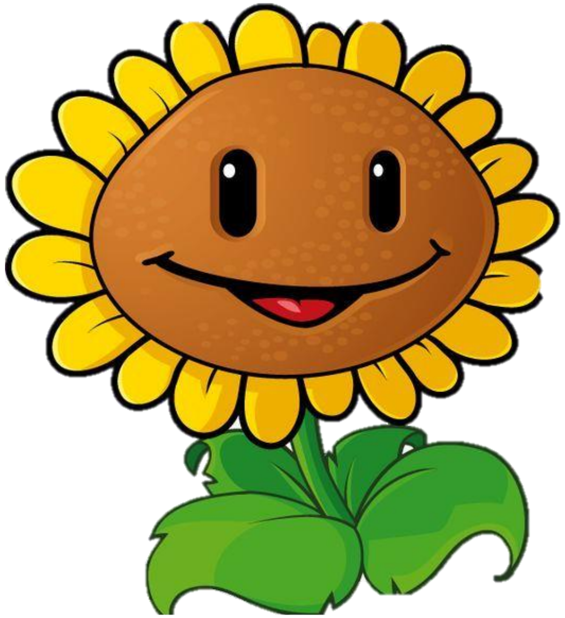
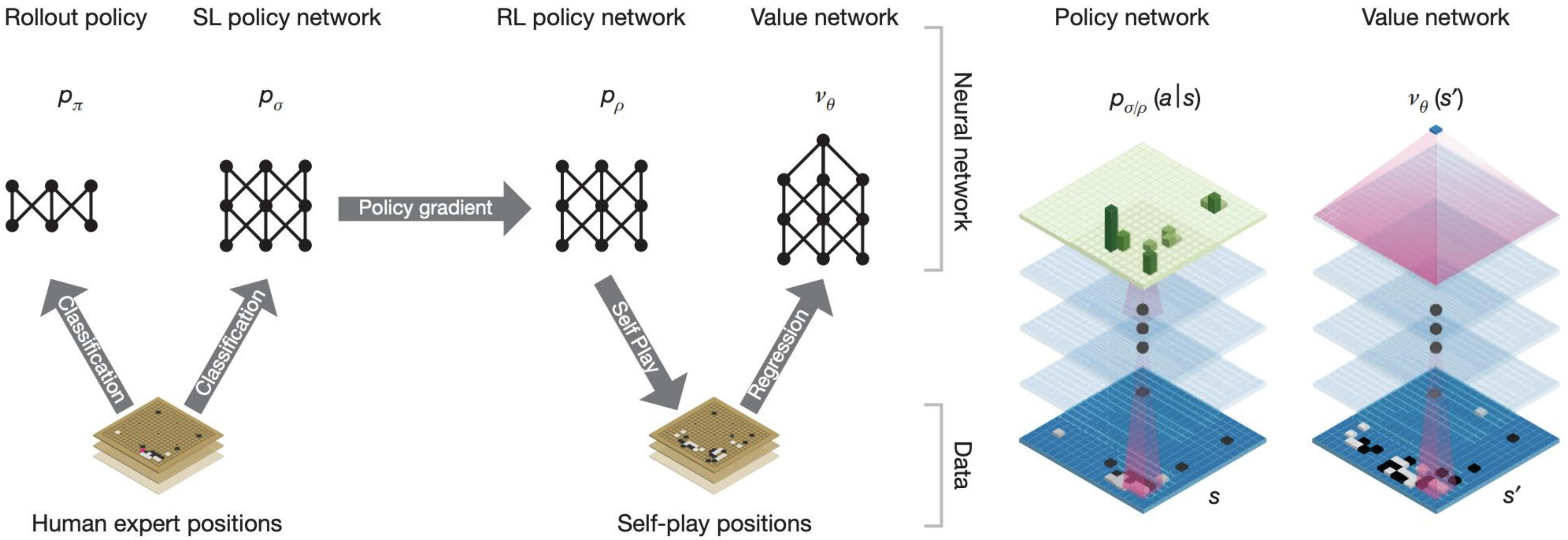
Integrating Learning and GOFAI systems

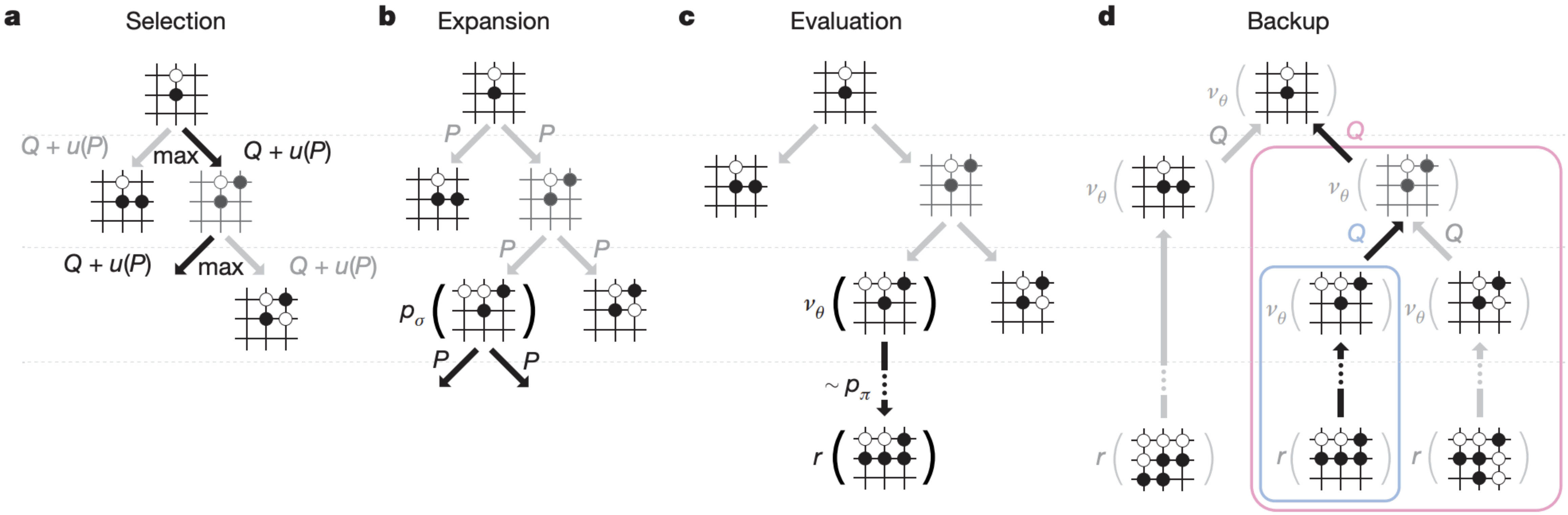


AlphaGo



Learn a Value function and a Policy function, use in MCTS

MCTS in AlphaGo



Selection

$$a_t = \operatorname*{argmax}_a (Q(s_t, a) + u(s_t, a))$$

$$u(s,a) \propto \frac{P(s,a)}{1 + N(s,a)}$$

Policy Network

Total number of simulations/visits

$$N(s,a) = \sum_{i=1}^{n} 1(s,a,i)$$

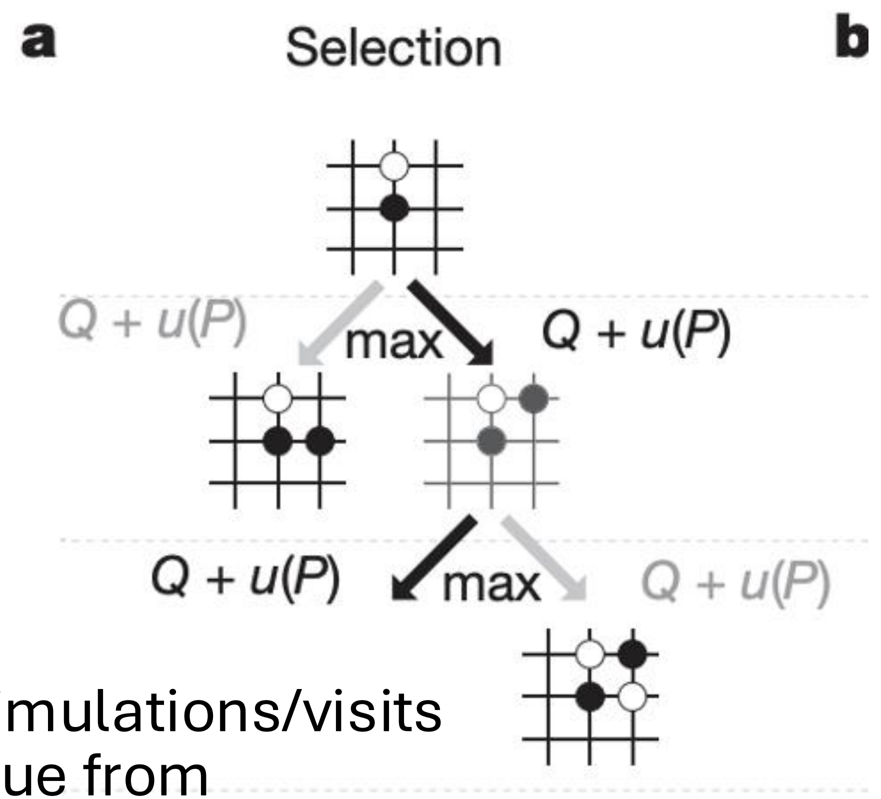
Average value from that state (including value estimated with Value network and result of simulations)

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{n} 1(s,a,i) V(s_L^i)$$

Value Network

Simulation result

$$V(s_L) = (1 - \lambda) v_\theta(s_L) + \lambda z_L$$

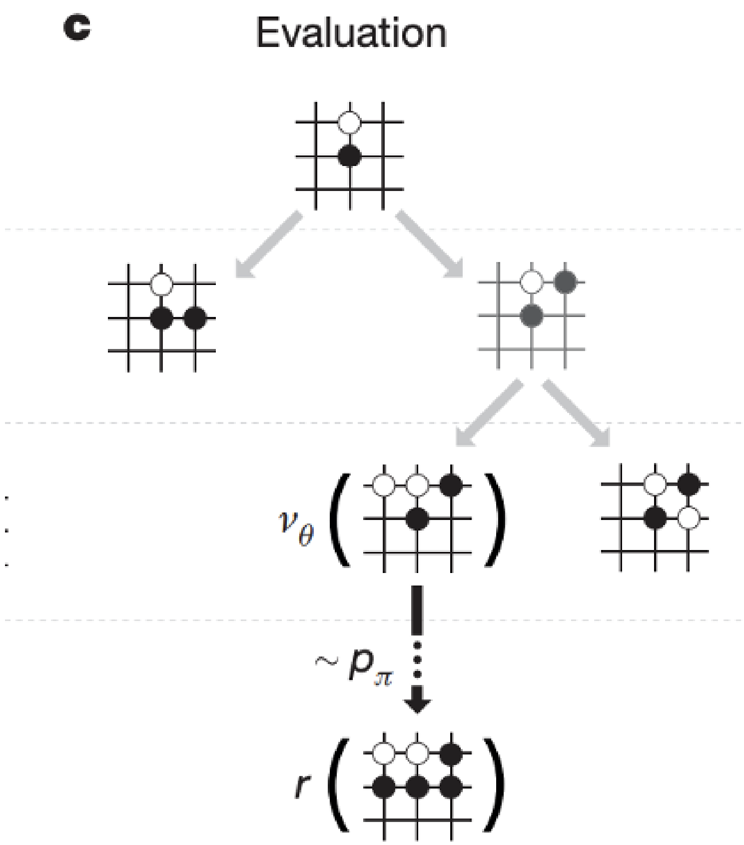


Evaluation

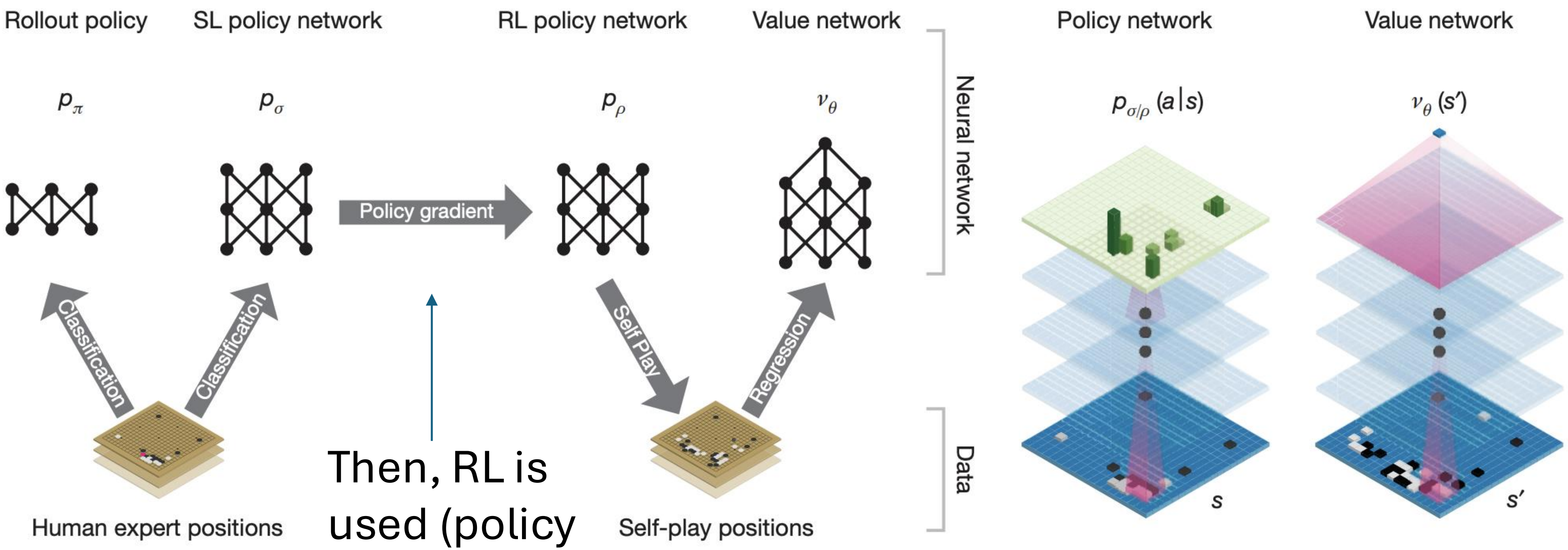
Instead of running random simulations, use a rollout policy p_π.

This is a **simple and fast** rollout policy.

It doesn't need to be perfect, it just needs to be fast. Anything better than random will help.



Training Value and Policy Networks



Then, RL is used (policy gradient algorithm)

Initially trained with database of human games

Policy Network: Predict actions
Value Network: Predict Values (win/lose from position)

Self-Play

How do you set up a RL environment for Go? Who is your opponent?

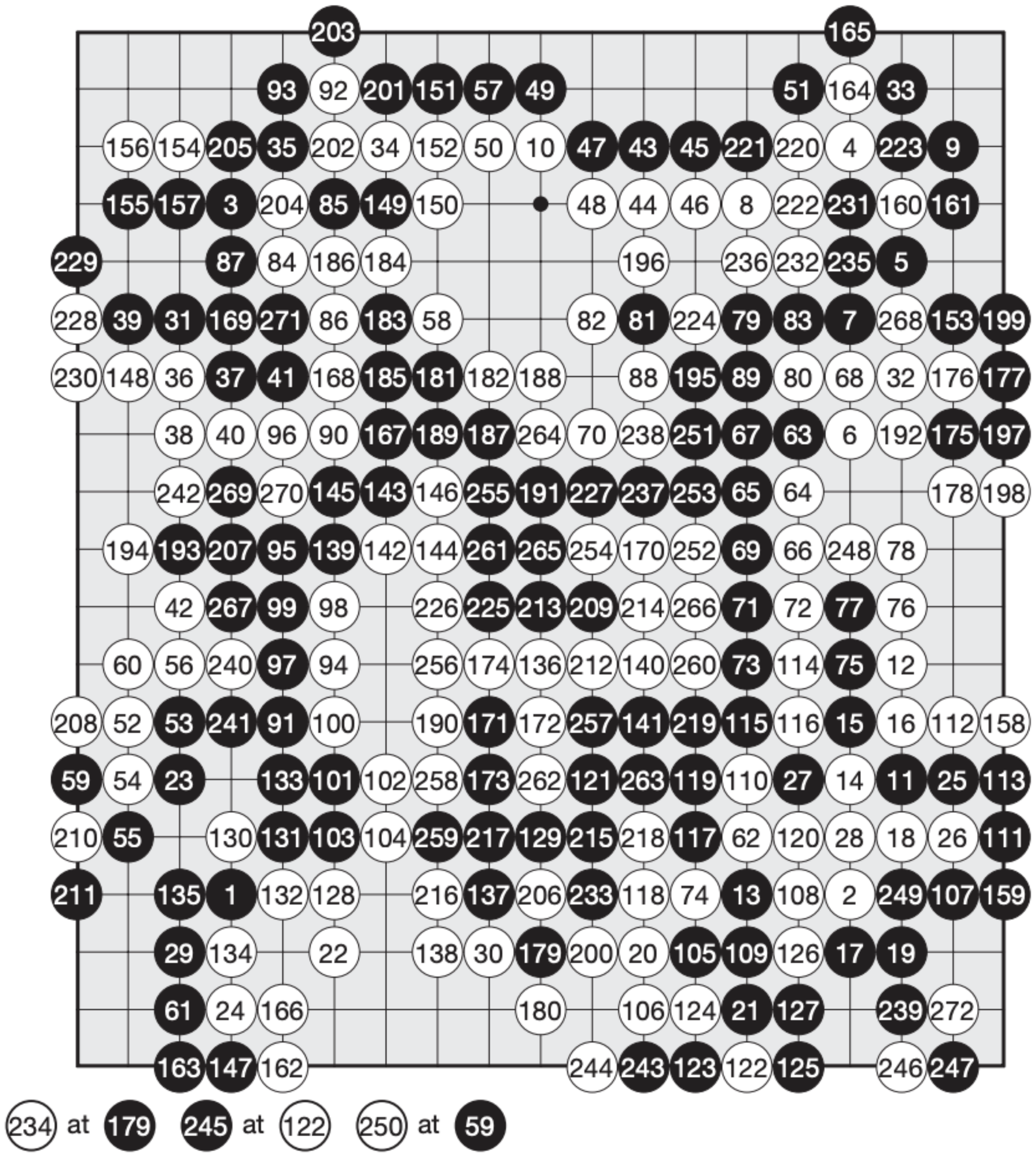
Yourself!

AlphaGo repeatedly played against itself and improved its parameters using Reinforcement Learning

AlphaGo

Could defeat the European Champion!

Game 1
Fan Hui (Black), AlphaGo (White)
AlphaGo wins by 2.5 points



AlphaGo

Problems with AlphaGo

- Supervised training makes it specific to Go
- Reliant on data gathered from humans

What if even the best human players are playing Go "wrong"?

Science cover, December 2018

AlphaZero

Next iteration:

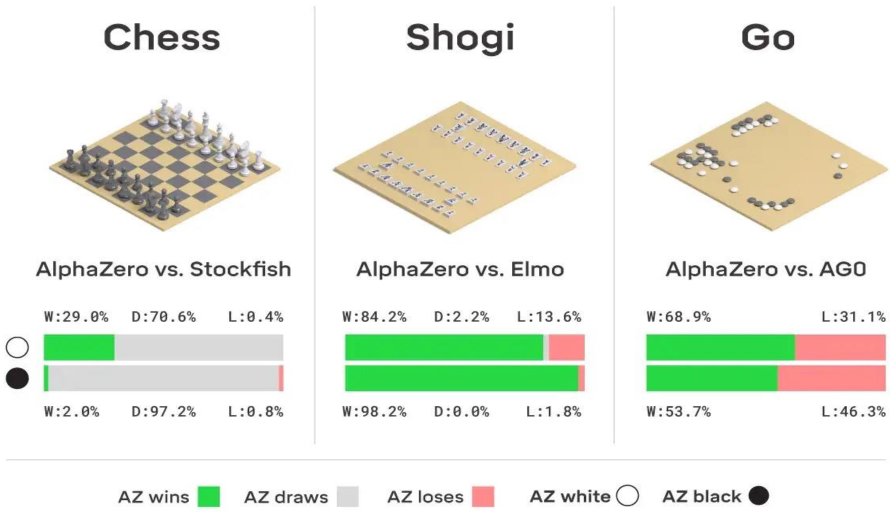
- Only uses Reinforcement Learning (no supervised learning)
- No longer specific to the game of Go

Science cover, December 2018

AlphaZero

AlphaZero defeated Lee Sedol (18 world titles) 4-1 in March 2016.

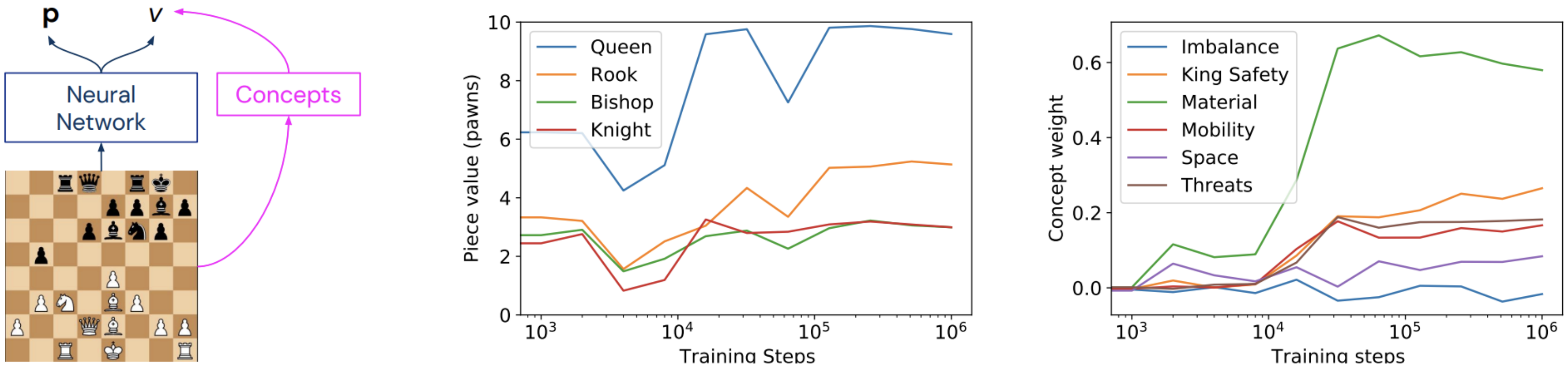
In Chess, AlphaZero passed Stockfish (briefly) to become the best chess engine.



How the Artificial-Intelligence Program AlphaZero Mastered Its Games, The New Yorker

AlphaZero For Chess

AlphaZero confirmed heuristic values for each piece that humans had been using for hundreds of years



(a) Value regression methodology: we train a generalized linear model on concepts to predict AlphaZero's value head for each neural network checkpoint.

(b) Piece value weights converge to values close to those predicted by conventional theory.

(c) Material predicts value early in training, with more subtle concepts such as mobility and king safety emerging later.

Figure 4. Value regression from human-defined concepts over time.

Source: Acquisition of Chess Knowledge in AlphaZero, McGrath et al. 2022

Why should you care?

What does the future of AI hold?

Past Trends in AI:
- Search
- KRR
- Optimization
- Sequential Decision Making
- Reasoning on Uncertainty

Current Trends in AI:
- Deep Learning
- LLMs
- Diffusion Models

Don’t Reinvent the Wheel

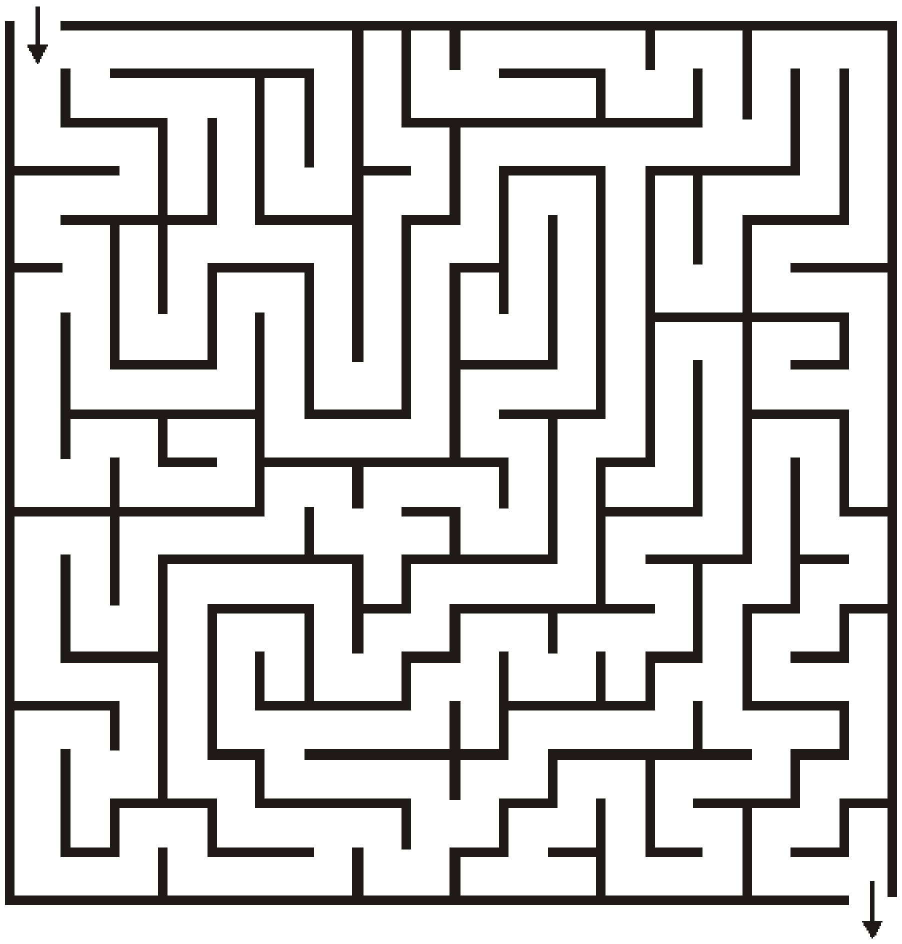
You’d like to solve a Maze.

You can either:

1. Ask an LLM to solve it
2. Use A*

AI has provably correct, optimal, and otherwise efficient algorithms for many tasks.

Use the right tool for the right job!



Artificial Intelligence

The goal of the field is to produce **intelligent systems**

What these systems look like, or how they are made, matters less than how well they perform.

Systems will always involve multiple components working together. There is room for Deep Learning components and traditional AI to work together.

Homework 10

How should we can we leverage LLMs for planning?

We already have efficient planning algorithms and formulations (e.g., PDDL)

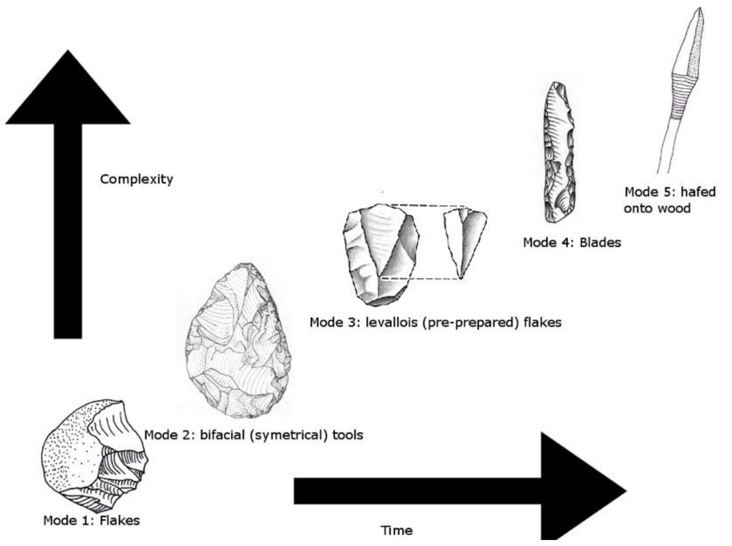
But these formulations can be hard to interact with

Leverage the strength of LLMs (they are easy to interact with in natural language) with the strength of existing planning algorithms

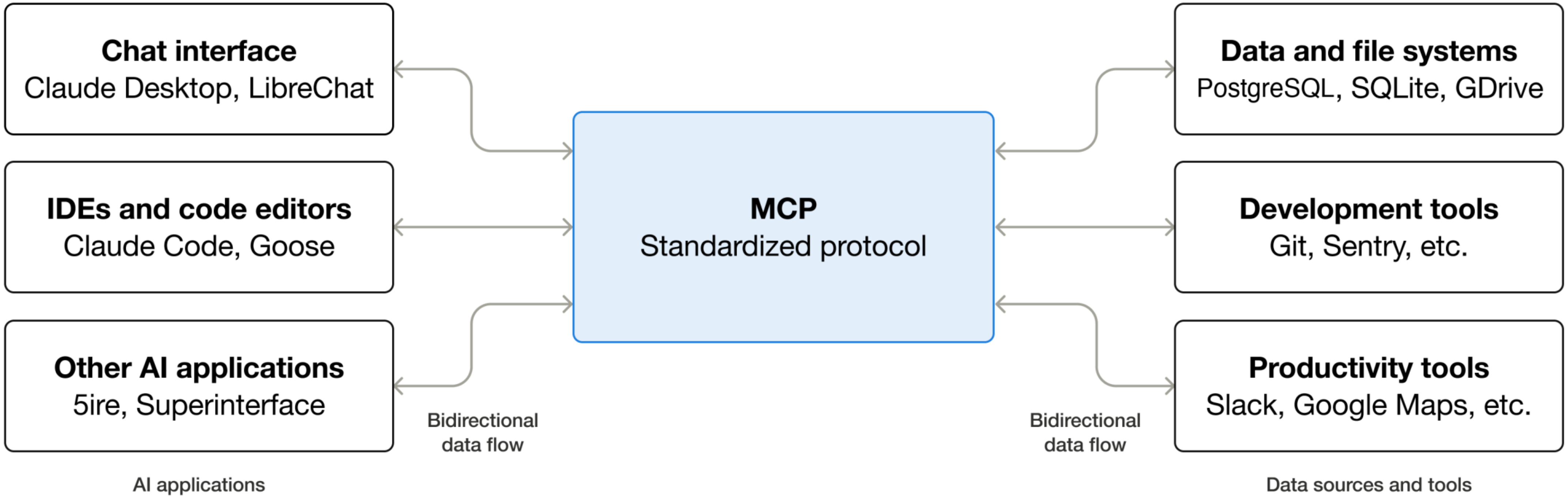
Tool Use

Humans are great at **building** and **using** tools
This forms one of the key properties of our intelligence

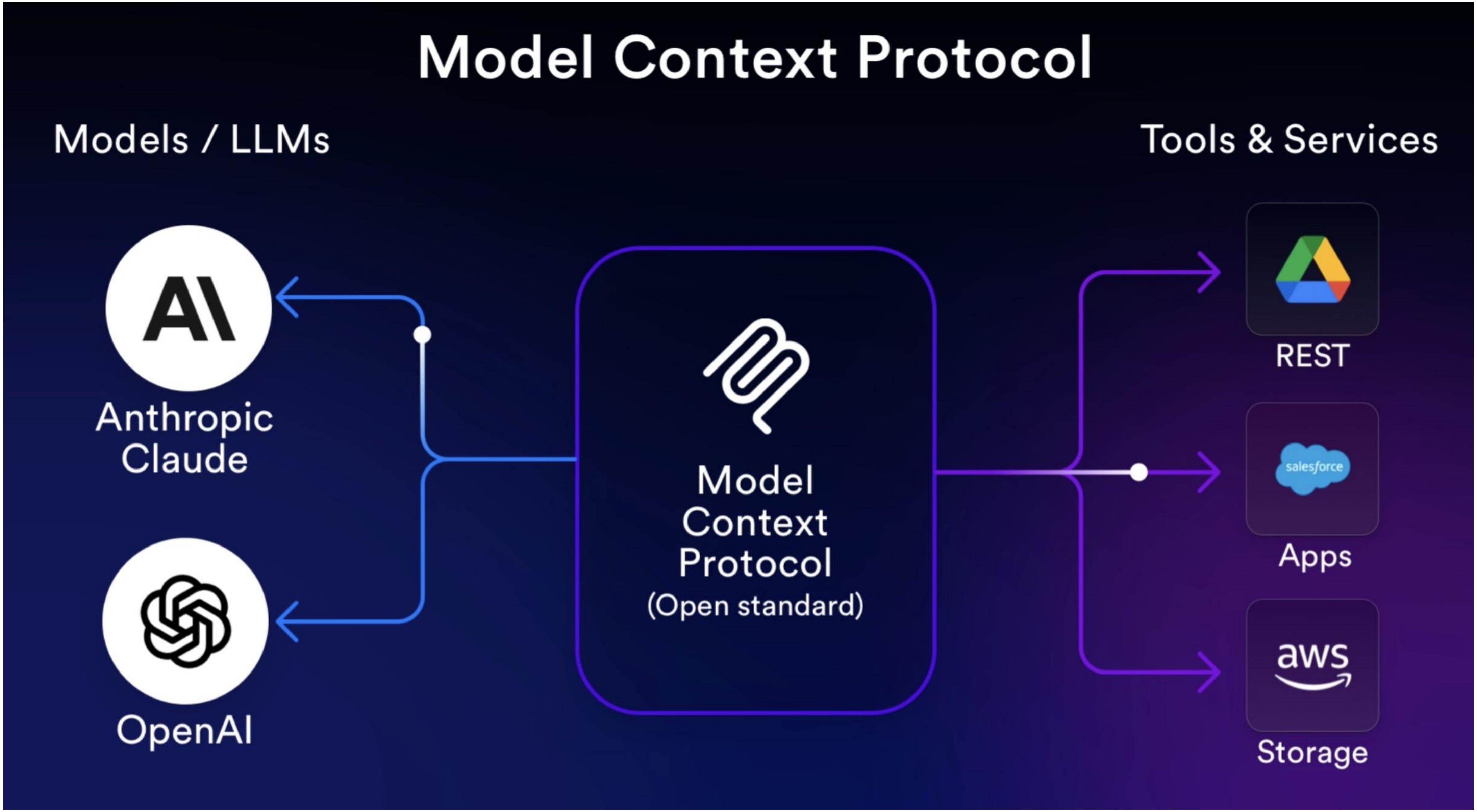
LLMs should be capable of using **existing tools** and building **new tools**



Model Context Protocols (MCP)



Source: https://modelcontextprotocol.io/docs/getting-started/intro



http://lightflows.co.uk/blog/what-is-model-context-protocol-in-ai-an-introduction-to-its-rol

Model Context Protocols

MCPs define three primitives that servers expose:

1. Tools: Tools available for a specific server (e.g., SQL query, terminal command, git operations)
2. Resources: Data sources available
3. Prompts: System prompts or prompt templates that may be helpful

MCP Clients allow 3 main features:

1. Sampling: the server may query the client's LLM, (e.g., which of these flights is best fo
2. Roots: The client can specify which files or directories the server should focus on
3. Elicitation: The client can request input from the user

MCP Example

We don’t need LLMs to be able to add or multiply, we can provide tools that they can use that do that task for them.

```python
from mcp.server.fastmcp import FastMCP

# Create a FastMCP server
mcp = FastMCP("Calculator MCP Server")

@mcp.tool()
def add(a: float, b: float) -> float:
    """Add two numbers together and return the result."
    return a + b

@mcp.tool()
def subtract(a: float, b: float) -> float:
    """Subtract b from a and return the result."""
    return a - b

@mcp.tool()
def multiply(a: float, b: float) -> float:
    """Multiply two numbers together and return the res
    return a * b
```

Source: https://github.com/microsoft/mcp-for-beginners/blob/main/03-GettingStarted/samples/python/mcp_calculator_server.py

Revisiting the field