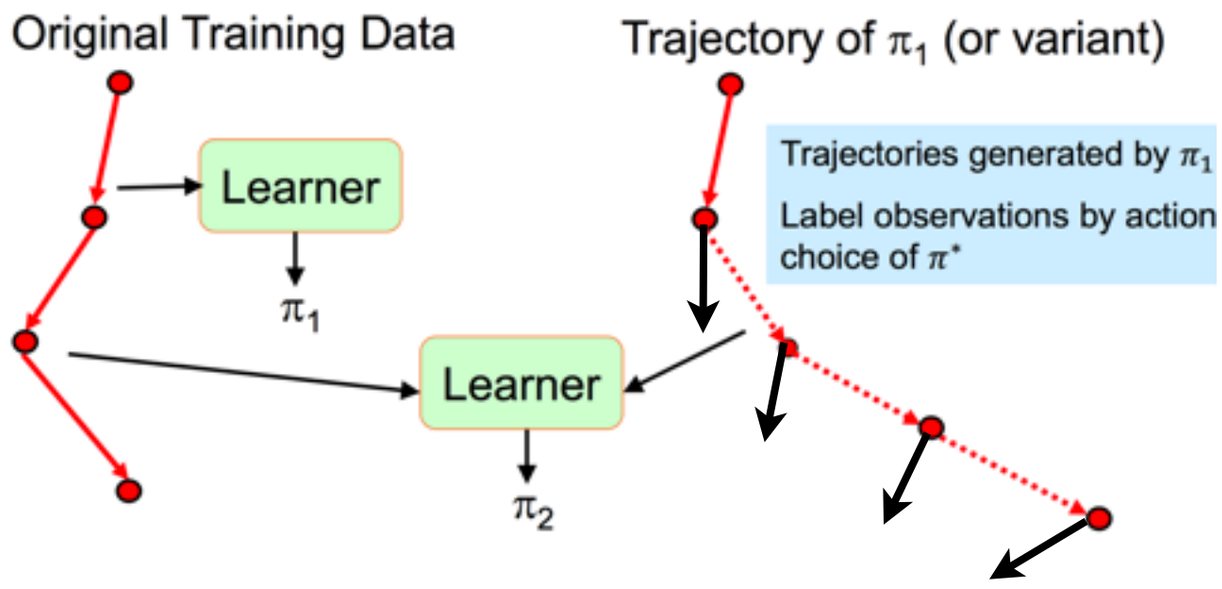
# Go as Seq. Decision Making

- Big Picture: Sequential Decision Making
  - OSU is one of the major players in this area (Alan, Prasad, Tom)
- Supervised Learning: follow gold trajectory
  - train Go from master plays
- Imitation Learning: everywhere-defined oracle (24/7 mentor)
  - ask professional player at each decision point
- Reinforcement Learning
  - no oracle, just reward
  - only need to know rules
  - more like real life!



# AlphaGo Zero

- AlphaGo uses both SL and RL; AlphaGo Zero only RL
  - human master plays are not always gold!

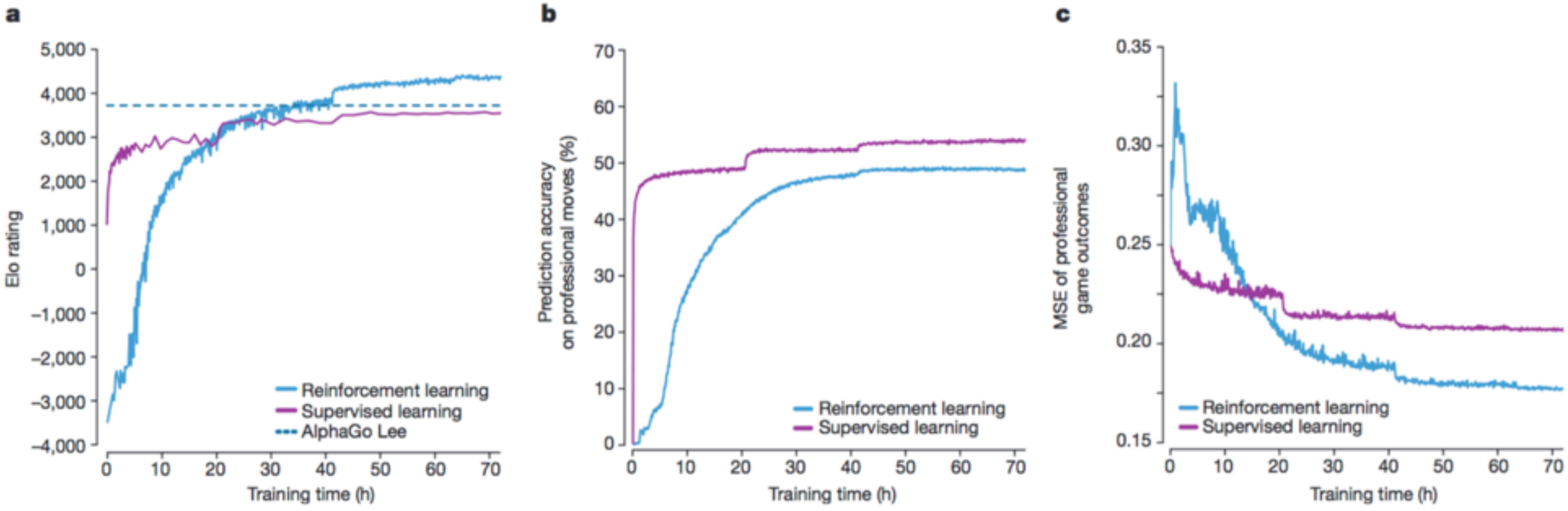



**Figure 3 | Empirical evaluation of AlphaGo Zero. a**, Performance of self-play reinforcement learning. The plot shows the performance of each MCTS player $\alpha_{\theta_i}$ from each iteration $i$ of reinforcement learning in AlphaGo Zero. Elo ratings were computed from evaluation games between different players, using 0.4 s of thinking time per move (see Methods). For comparison, a similar player trained by supervised learning from human data, using the KGS dataset, is also shown. **b**, Prediction accuracy on human professional moves. The plot shows the accuracy of the neural network $f_{\theta_i}$, at each iteration of self-play $i$, in predicting human professional moves from the GoKifu dataset. The accuracy measures the percentage of positions in which the neural network assigns the highest probability to the human move. The accuracy of a neural network trained by supervised learning is also shown. **c**, Mean-squared error (MSE) of human professional game outcomes. The plot shows the MSE of the neural network $f_{\theta_i}$, at each iteration of self-play $i$, in predicting the outcome of human professional games from the GoKifu dataset. The MSE is between the actual outcome $z \in \{-1, +1\}$ and the neural network value $v$, scaled by a factor of $\frac{1}{4}$ to the range of 0–1. The MSE of a neural network trained by supervised learning is also shown.

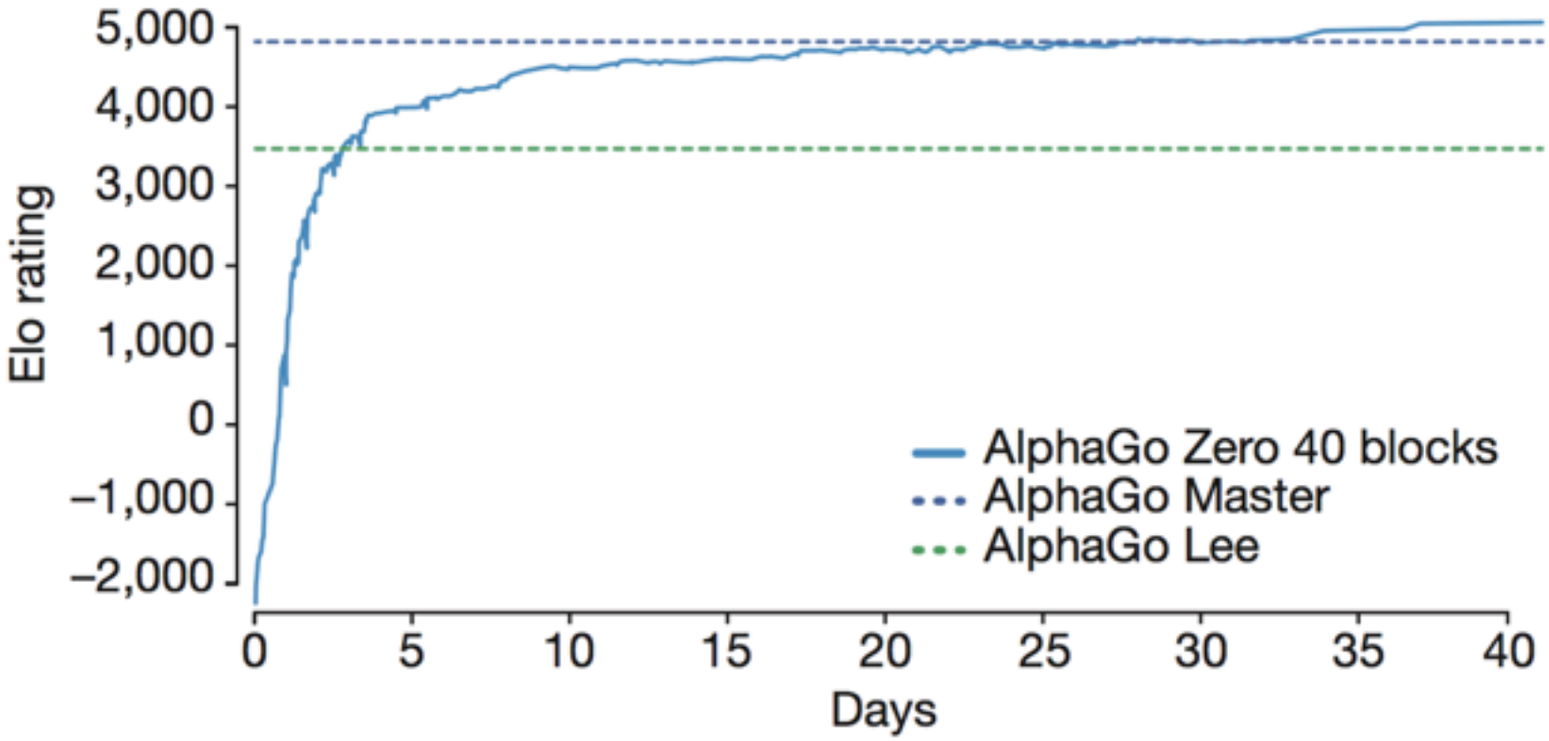
# Other Major Differences

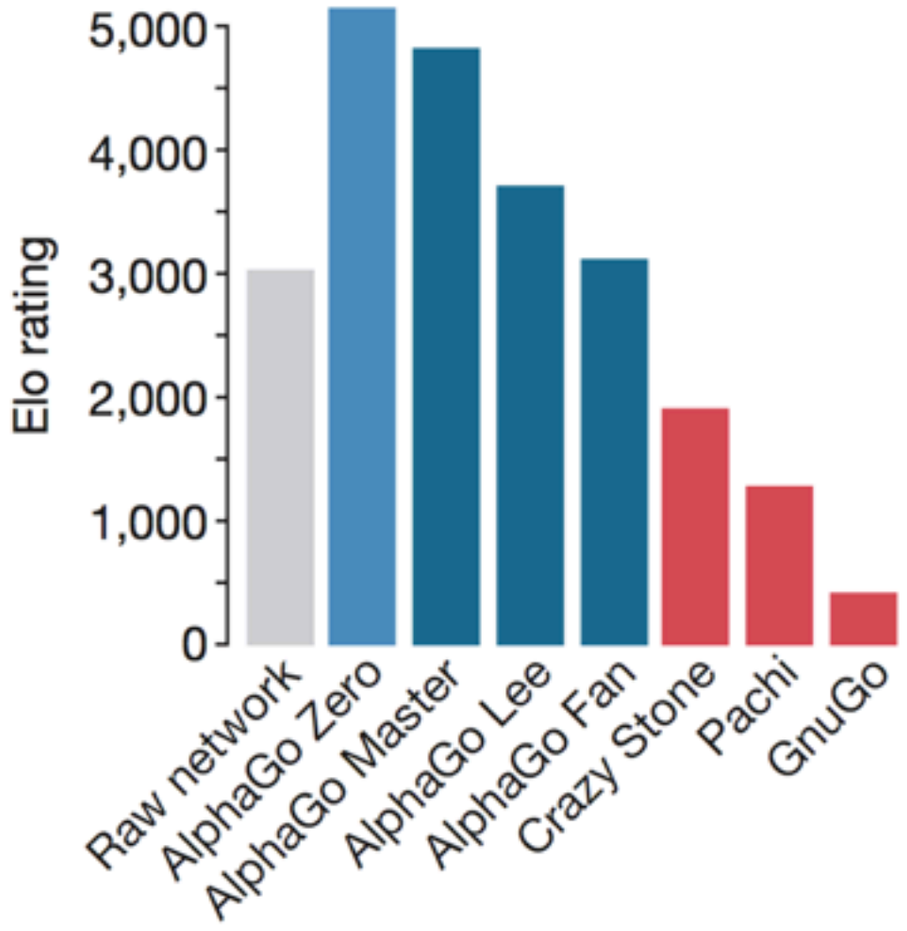
AlphaGo Zero only uses the black and white stones from the Go board as its input, whereas previous versions included a small number of hand-engineered features (liberty, ladder, ko, etc).

It uses one neural network rather than two. Earlier versions used a "policy network" to select the next move to play and a "value network" to predict the winner of the game from each position. These are combined in AlphaGo Zero, allowing it to be trained and evaluated more efficiently.

AlphaGo Zero does not use "rollouts" - fast, random games used by other Go programs to predict which player will win from the current board position. Instead, it relies on its high quality neural networks to evaluate positions.

# Zero vs. Master



a

b

0 12 24 36 48 60 72 h