Regarding the next iteration of Alpha Go, popularly known as AlphaGo Zero. The paper can be found here.

The original paper's summary, for comparison purposes, titled "AlphaGo Paper updated reading", is located in this repository.

The paper introduces AlphaGo Zero, a system that starts with no understanding of the game, learning entirely through self-play. By providing only the basic rules of Go, the system sets out to discover every strategy on its own, which differs from the previous paper, in which expert human moves were used for training.

Self-Play:

Instead of relying on pre-existing databases of expert moves, the system plays games against itself, gradually improving its performance with each game. This self-reinforcing cycle allows the AI to discover and refine strategies over time, building an understanding of the game that goes well beyond what has been passed down by human players. By continuously learning from its own experiences, the AI develops a unique and highly effective style of play.

Neural Network Architecture:

Previous versions of AlphaGo used two separate networks: one for choosing moves and another for evaluating positions (see the AlphaGo summary for more information). In contrast, AlphaGo Zero uses a single deep residual network that outputs both a set of move probabilities and a value estimate indicating the likelihood of winning from any given position.

Monte Carlo Tree Search:

In each move, the AI performs an MCTS that uses the neural network's predictions to guide the search for the best moves. Unlike earlier versions that relied on separate rollout policies to simulate complete games from a given position, AlphaGo Zero forgoes these rollouts entirely. Instead, it uses the neural network to evaluate positions directly.

A rollout in earlier systems involved simulating the remainder of a game from a certain point by following a predefined policy until the end of the game. These simulated outcomes were then used to estimate the value of a position. AlphaGo Zero skips this step. It directly uses its deep network to predict the outcome, avoiding the inefficiencies that can come from simulating many random or suboptimal moves. This shift not only speeds up the search but also leads to more precise evaluations.

Reinforcement Learning Network:

As the AI plays games against itself, it collects data on each move and its eventual outcome. This data is then used to update the neural network via a loss function that balances two goals:

aligning the network's move probabilities with those suggested by MCTS, and ensuring that its value estimates match the actual game results. The system evolves from making random moves, to making expert ones which exceed even those levels of human expertise.