A brief summary of the paper's goals or techniques introduced (if any).

The paper's goal is to build an artificial intelligence that is superior to traditional Go programs that most rely on monte Carlo tree search.

AlphaGo trains a supervised learning (SL) network to learn from expert human moves, trains a reinforcement learning (RL) policy network to adjust the policy towards winning games instead of predicting probabilities and trains a value network.

The SL network that alternates between convolutional layers and rectifier nonlinearities trains a policy network using a 13-layer. The last layer produces a soft-max probability distribution over different moves, which predicts the held-out test set with 57% accuracy.

In the RL policy network, a randomly selected previous iteration of a policy network is played against the current policy network to prevent overfitting, and weights are updated according to the stochastic gradient descent that maximizes outcome

the RL value network trains and outputs a single prediction that estimates the value of a position. In Go, due to the highly correlated nature of the success moves which may differ by one position, the key was in preventing overfitting by generating a massive set of distinct positions by playing and saving the games played between the RL policy networks.

All of the above are combined in a MCTS algorithm that selects actions by look-ahead search. While this technique does not differ from other Go programs available at the time of publication, gains from the other networks improved the overall strength of the program such that it needed to search fewer spaces even compared to Deep Blue in its chess match against Kasparov.

AlphaGo successfully won matches against previous Go programs, the distribution version of which had 100% winning rate. In 2015, it won 5 out of 5 matches against a Fan Hui, a professional Go player, a feat previously thought to be decades away.