Page 8, Training
For each state s in this sequence we carry out the following procedure:

1. apply every possible transformation (12 in total) to the state s,
2. pass those 12 states to our current neural network, asking for value output. This gives us 12 values for every sub-state of s.
3. the target value for the state s is calculated as $y_{v_i} = \max_a(v_s(a) + R(A(s,a)))$, where $A(s,a)$ is the state after action a applied to the state s and $R(s)$ equals 1 if s is the goal state and -1 otherwise.
4. The target policy for state s is calculated using the same formula, but instead of max we take argmax: $y_{p_i} = \arg\max_a(v_s(a) + R(A(s,a)))$. This just means that our target policy will have 1 at the position of the maximum value for sub-state and 0 on all other positions.

This process is shown on the figure below, taken from the paper. The sequence of scrambles $x_0, x_1, \ldots, x_N$ is generated, where cube x_i is shown expanded. For this state x_i, we make targets for the policy and value heads from the expanded states by applying the formulas above.

Page 10, Model application

In addition, the value returned by the model is also used in this decision-making. The value is being tracked as the maximum from the current state's value and the value from their its children. This allows the most promising paths (from the model perspective) to be seen from the parent's states.

To summarize, the action to follow from a non-leaf tree is chosen by using the following formula: $A_t = \arg\max_a(U_{s_t}(a) + W_{s_t}(a))$, where $U_{s_t}(a) = cP_{s_t}(a)\frac{\sqrt{\sum_{a'} N_{s_t}(a')}}{1+N_{s_t}(a)}$. Here $N_{s_t}(a)$ is a count of times action a has been chosen in the state s_t. $P_{s_t}(a)$ is the policy returned by the model for state s_t and $W_{s_t}(a)$ is the maximum value returned by the model for all children's states of s_t under the branch a.

Page 16, Experiment results

After several experiments with this, I've come to the conclusion that this behavior is a result of the wrong value objective proposed in the method. Indeed, in formula $y_{v_i} = \max_a(v_s(a) + R(A(s,a)))$, the value $v_s(a)$ returned by the network is always added to the actual reward R(s), even for the goal state. With this, the actual values returned by the network could be anything: -100, 106 or 3.1415. This is not a great situation for neural network training, especially with the MSE objective.

To check this, I modified the method of the target value calculation by assigning zero target for the goal state:

$$y_{v_i} = \begin{cases} \max_a(v_s(a) + R(A(s,a))) & \text{if } s \text{ is not the goal} \\ 0 & \text{if } s \text{ is the goal state.} \end{cases}$$