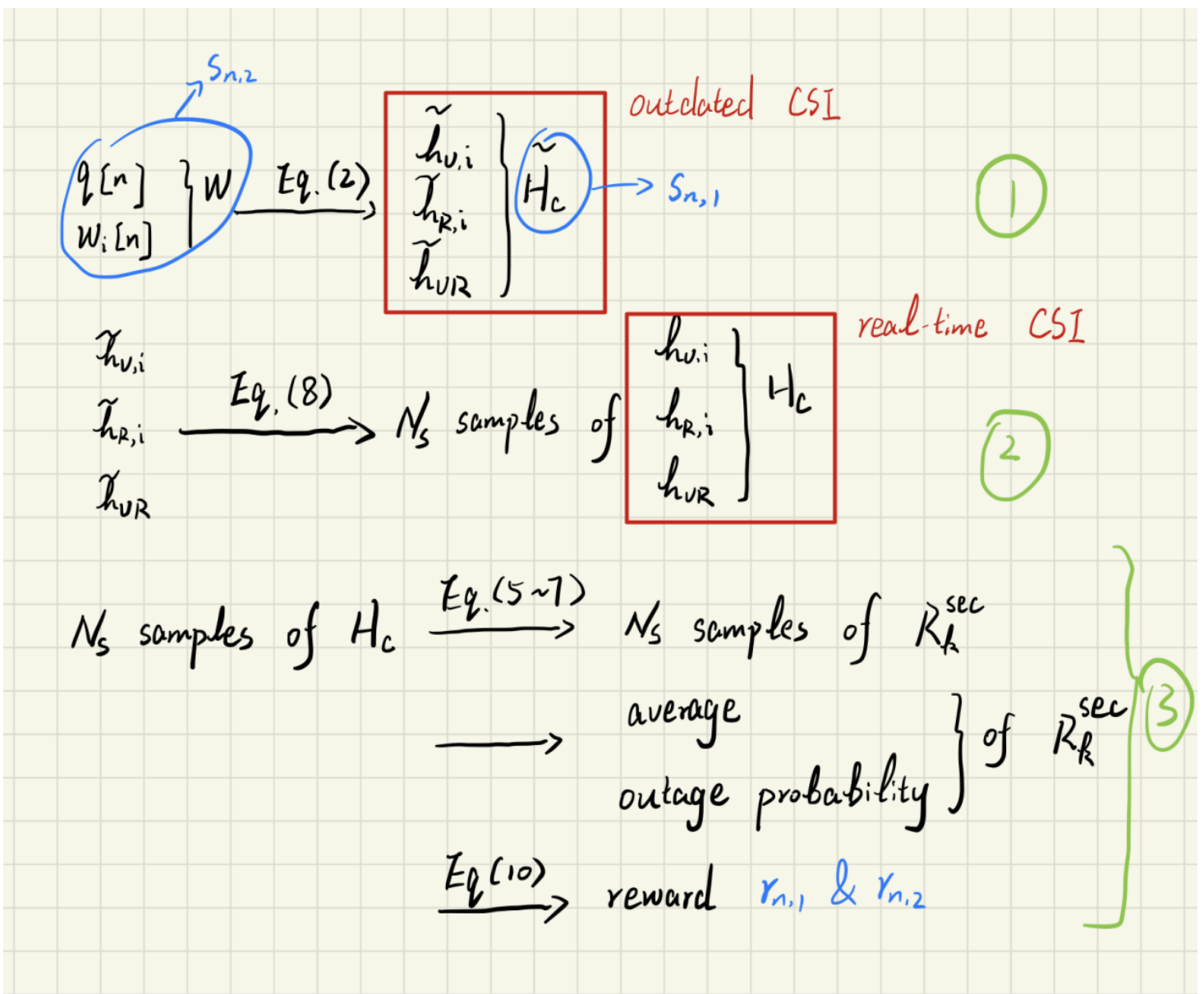


Q&A

1. Do you generate new channel coefficients in the line that the agent is observing the environment?
If so, what is the procedure to do this?

Yes.

At the very first of each 'step' (n -th time slot), the 1-st agent needs to observe the CSI \mathbf{H}_C (which is actually the outdated CSI $\widetilde{\mathbf{H}}_C$, for simplicity it's been written as \mathbf{H}_C in this paper) and the 2-nd agent need to observe the position information \mathbf{W} . We do generate new states according to the following procedures:



As we use the 3D SV channel, at the first step, the outdated CSI $\widetilde{\mathbf{H}}_C$ is generated by Eq.(2). Then the distribution of the real-time CSI can be expressed by Eq.(8). And N_s samples of the real-time CSI are generated. (We model the outdated CSI $\widetilde{\mathbf{H}}_C$ as a fixed variable during each step and the real-time CSI \mathbf{H}_C as a statistical variable, because the outdated CSI is assumed to be feedback to the UAV, however, the real-time CSI remains unknown at the UAV.) Following the distribution, N_s samples of CSI are used to calculate the average value and the outage probability of R_k^{sec} . At last the reward $r_{n,1}$ and $r_{n,2}$ can be calculated.

2. Is the position of components of the network fixed?

The network is designed to be at the UAV, and the UAV's position is the network's position.

3. Also, how do you pre-process the input data to the neural networks?

$$G = \text{Im}(G) + \text{Re}(G) = G_i + G_r$$

$$G' = (G_i | G_r)$$

$$\text{Vec}(G') \xrightarrow{\text{to Tensor}} \text{Input}$$

As well as W

All the variables are divided into the real part and the imaginary part. Then both of them are vectorized to 1-D vectors and are turned into tensors to fit the network input. Note that we did not use normalization when pre-processing the data.