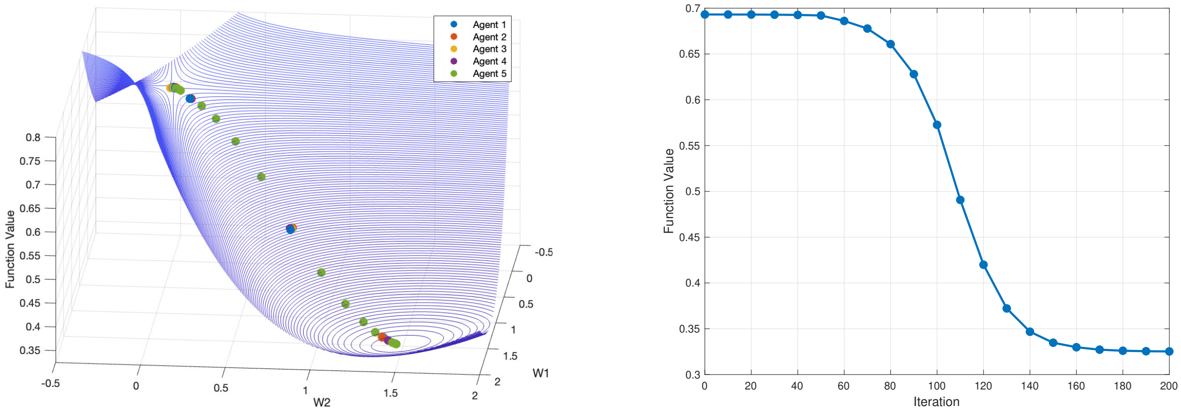

(a) **(b)**

Fig. 3. (a) Trajectories of all five agents when initialized on the saddle point (0,0). Note that all trajectories overlap with each other, implying perfect consensus among the agents. (b) Global function value dynamics. In this experiment with explicit saddle points, we consider a simple $\{0, 1\}$–classification task using a neural network with a single linear hidden layer and a logistic activation function. We use the cross-entropy loss function to train the network. We denote the feature vector as $\boldsymbol{h} \in \mathbb{R}^M$ and the binary class label as $y \in$ {-1, 1}. For the fully connected hidden layer, we represent the weights as $\boldsymbol{W}_2 \in \mathbb{R}^{L\times M}$ and $\boldsymbol{W}_1 \in \mathbb{R}^L$. The output is of the form: $\hat{y} = \frac{1}{1+e^{-\left\langle \boldsymbol{h}, \boldsymbol{W}_2^\top \boldsymbol{W}_1 \right\rangle}}$. Under the commonly used cross-entry loss function, the objective function is of the following form: $L(\boldsymbol{W}_1, \boldsymbol{W}_2) = \log(1 + e^{-y\left\langle \boldsymbol{h}, \boldsymbol{W}_2^\top \boldsymbol{W}_1 \right\rangle})$. We considered the scalar case with $L = M = 1$. Synthetic data samples ($\boldsymbol{h}$ and y) are randomly generated under the constraint $\boldsymbol{h}y = 1$ and the regularization parameter is set to $\rho = 0.1$. One can obtain analytically that there is a strict saddle point at $(0, 0)$. To visualize the evolution of optimization variables under our algorithm, we purposely initialized all the agents from the strict saddle point $(0, 0)$. The stepsize, noise amplitude, noise-injection interval and clipping threshold were set to $\alpha = 0.1$, $\theta = 0.1$, $\mathcal{K}_0 = 50$ and $c_0 = 1.0$, respectively. It can be seen that due to the noise and gradient clipping effect, all five agents collectively move along the descending direction, clearly indicating that our algorithm can ensure efficient escape from saddle points.