The computational complexity of the convolutional layers can be written as

$$\mathcal{C}_{\mathrm{CL}} = \mathcal{O}\left(\sum_{l=1}^{N_{\mathrm{CL}}} D_x^{(l)} D_y^{(l)} F_x^{(l)} F_y^{(l)} N_{\mathrm{SF}}^{(l-1)} N_{\mathrm{SF}}^{(l)}\right)$$

where $D_x^{(l)}$, $D_y^{(l)}$ are the column and row sizes of each output feature map, $F_x^{(l)}$, $F_y^{(l)}$ are the 2-D filter size of the l-th layer. $N_{SF}^{(l-1)}$ and $N_{SF}^{(l)}$ denote the number of input and output filter channels of the l-th layer respectively. Thus, the complexity of the two convolutional layers with $256@2 \times 2$ and $512@2 \times 2$ filters approximately becomes

$$\begin{aligned}\mathcal{C}_{\mathrm{CL}} &= \mathcal{O}\left(k_1 \cdot (N_t - 1)(3N_r - 1) + 2k_1 \cdot (N_t - 2)(3N_r - 2)\right) \\ &\approx \mathcal{O}\left(N_t N_r\right)\end{aligned}$$

where $k_1 = 2^2 \cdot 256$, is a constant.

The computational complexity of the fully connected layers can be written as

$$\mathcal{C}_{\mathrm{FCL}} = \mathcal{O}\left(\sum_{l=1}^{N_{\mathrm{FCL}}} L^{(l-1)} L^{(l)}\right)$$

where L is the number of units. It can be calculated that the computational complexity of FCL can be approximated as $\mathcal{C}_{\mathrm{FCL}} \approx \mathcal{O}(N_s^2 + N_t N_r N_s)$

Hence the computational complexity of JBFNet is $\mathcal{C} = \mathcal{C}_{\mathrm{CL}} + \mathcal{C}_{\mathrm{FCL}}$, which approximately is

$$\mathcal{C} \approx \mathcal{O}\left(N_s^2 + N_t N_r N_s\right)$$

It is easy to see that the proposed DL-based approach has low computational complexity.