with the n-step are defined as follo

$$A\left(s_{n,t}, a_{n,t}; \theta, \theta_v\right) = Q^{(k)}\left(s_{n,t}, a_{n,t}; \theta, \theta_v\right) - V\left(s_{n,t}; \theta_v\right) \tag{10}$$

$$Q^{(k)}\left(s_{n,t}, a_{n,t}; \theta, \theta_v\right) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V^{\pi}\left(s_{t+k}; \theta_v\right) \tag{11}$$

n parallel tasks are set up. $s_{n,t}$ and $a_{n,t}$ are represented the state and action in environment $work_n (1 < n < N_n)$, respectively. The network performs gradient updates by small batch sampling from the replay memory. n works interact with the environment to generate batches size of $N * T_{max}$. The updates of policy gradient ∇_{θ}^{π} are presented in the equation below [42]:

$$\nabla_{\theta}^{\pi} \approx \frac{1}{N T_{\max}} \sum_{n=1}^{N} \sum_{t=1}^{T_{\max}} A\left(s_{n,t}, a_{n,t}; \theta, \theta_v\right) \nabla_{\theta} \log \pi\left(a_{n,t} \mid s_{n,t}; \theta\right) + \beta \nabla_{\theta} H\left(\pi\left(a_{n,t} \mid s_{n,t}; \theta\right)\right) \tag{12}$$

$\beta \nabla_{\theta} H(\pi(a_t \mid s_t; \theta))$ is the entropy of policy mod $\pi(a_t \mid s_t; \theta)$, increasing exploration and prevent the model from falling into a local optimum. The value gradient $\nabla_{\theta_v}^{V}$ is updated as follows [42]:

$$\nabla_{\theta_v}^{V} \approx \nabla_{\theta_v} \frac{1}{N T_{\max}} \sum_{n=1}^{N} \sum_{t=1}^{T_{\max}} \left(Q^{(T_n - t + 1)}\left(s_{n,t}, a_{n,t}; \theta, \theta_v\right) - V\left(s_{n,t}; \theta_v\right)\right)^2 \tag{13}$$

(a)PAAC-K Parameters The complete network architectu is shown in Fig. 1e and 1f. PAAC-K consists of a policy network F_1 and a value network F_2. F_1 and F_2 uses the circuit grid parameter matrix as input, F_1 returns an action in the current state, and F_2 returns a value evaluation (scalar) according to function $V(s_t; \theta_v)$.

The F_1 network is a multilayer perceptron (MLP) with 1 hidden cells. The input and hidden layers of the MLP connect to the tanh activation function, while the output layer connects to the softmax activation function. The policy model $\pi\left(a_t \mid s_t; \theta_p\right)$ returns the action with the highest probability distribution in vector form. The F_2 network firstly follows a convolutional layer for extraction characteristics. It is followed immediately by a maximum pooling layer and another convolution layer. Each convolutional layer connects the tanh activation function, the stride is 1, the kernel size is $5 * 5$, and the padding is default to 0 to prevent information loss. An MLP with 128 hidden cells is introduced after the convolutional layer. The input and hidden layers of the MLP connect the relu activation function, while the output layer has no activation function and returns the value scalar directly. The

circuit grid parameter matrix is processed by the Z score sta dardized before being fed into the neural network to improve the velocity of the gradient descent solution. The standardized formula is as follows:

$$y_i = \frac{x_i - }{\mathrm{std}(x)} \tag{14}$$

(b)Rew

The design performance of the filter is generally relate the centre frequency f_0', frequency band $f_1' \sim f_2'$, reflection loss, insertion loss, etc. Therefore, when setting the reward, the defined function should also be related to the above. The two steps are based on the stepwise training method. The first step is adjusting the centre frequency, and the second step is adjusting the reflection loss. The two reward functions, R' and R'', are implemented for the circuit design.

As shown in Fig. 6, the central frequency of the targe f_0', the bandpass band range of the target is $f_1' \sim f_2'$, and the centre frequency of the current state s_i is f_i. Reward $r_{1,i}$ satisfies the equation as follows:

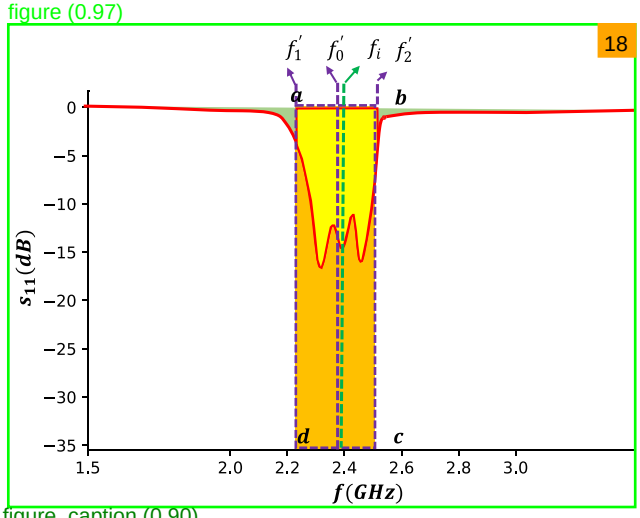
$$r_{1,i} = -|f_0' - f_i| \tag{15}$$

γ is the weight coefficient in the equation above. smaller the distance between f_i and f_0', the greater the reward. R' is the difference between the current and previous states. Then, R' is calculated as follows:

$$R' = r_{1,i} - r_1, \tag{16}$$

The yellow shadowed region A_i in state s_i is formed overlapping the quadrilateral $abcd$ and S_{11} curves, which is

figure (0.97)



figure_caption (0.90)

Fig. 6 Reward func