APPENDIX A
PROOF OF FEEDBACK RECOVERING

Suppose that we have N feedback layers in total. The objective function S in Equation (6) can be expanded linearly from any feedback layer l as

$$S = \sum_{ijc} \alpha_{ijc}^{(l)} z_{ijc}^{(l)} x_{ijc}^{(l)}, \tag{1}$$

where $x_{ijc}^{(l)}$ is the input of the neuron (i, j) of the channel c in the feedback layer l, $z_{ijc}^{(l)} \in \{0, 1\}$, and $\alpha_{ijc}^{(l)}$ is the contribution weight that is determined by the neuron pathways from $z_{ijc}^{(l)}$ to the target neuron S. We denote the mapping function of the target S after updating the feedback layer l as S_l, and use subscript k to replace i, j, c for simplicity. And $w_{k'}^{(l-1)}$ denotes the convolutional weight between $x_{k'}^{(l-1)}$ and $x_k^{(l)}$ when the convolution operation is performed from layer $l-1$ to layer l. A sign function $\delta(x)$ is described as:

$$\delta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0. \end{cases} \tag{2}$$

To prove that the FR algorithm (Algorithm 1) can reach local optimum, we need to prove that S will keep increasing after each iteration, which means that we should prove $S_N \leq S_1$. Accordingly, we first prove that $S \leq S_N$ and $S_N \leq S_{N-1}$, then demonstrate that $S_l \leq S_{l-1}$ under the assumption of $S_{l+1} \leq S_l$, which following the rules of mathematical induction method.

1. When $l = N$:

We expand S with the N-th feedback layer as:

$$S = \sum_k \alpha_k^{(N)} z_k^{(N)} x_k^{(N)}, \tag{3}$$

where $x_k^{(N)}$ is an output of a neuron in the N-th ReLU layer, so $x_k^{(N)} \geq 0$.

Since $z_k^{(N)} \in \{0, 1\}$, we have

$$\alpha_k^{(N)} z_k^{(N)} x_k^{(N)} \leq \alpha_k^{(N)} \delta(\alpha_k^{(N)}) x_k^{(N)}. \tag{4}$$

Let $z_k^{(N)} \rightarrow z_k'^{(N)} = \delta(\alpha_k^{(N)})$

and

$$\alpha_k'^{(N)} = \alpha_k^{(N)} * z_k'^{(N)} \geq 0, \tag{5}$$

then

$$S \leq S_N = \sum_k \alpha_k'^{(N)} x_k^{(N)}. \tag{6}$$

After updating all $z_k^{(N)}$ in feedback layer N, S_N can be expressed by feedback layer $N-1$. Note that $\alpha_k^{(N-1)}$ is dependent on $\alpha_k^{(N)}$, and it will be changed to $\hat{\alpha}_k^{(N-1)}$ when $\alpha_k^{(N)}$ being modified, then

$$S_N = \sum_k \hat{\alpha}_k^{(N-1)} z_k^{(N-1)} x_k^{(N-1)}. \tag{7}$$

Update $z_k^{(N-1)}$ and $\hat{\alpha}_k^{(N-1)}$ to get S_{N-1} in the same way when we update $z_k^{(N)}$ and $\alpha_k^{(N)}$, then

$$S_N \leq S_{N-1}. \tag{8}$$

2. Let us assume that $S_{l+1} \leq S_l$:

Fix $z_k^{(N)}, z_k^{(N-1)}, \ldots, z_k^{(l+1)}$, then

$$S_l = \sum_k \alpha_k'^{(l)} x_k^{(l)}. \tag{9}$$

Note that $x_k^{(l)}$ can be expressed by $x_{k'}^{(l-1)}$ with convolutional weights $w_{k'}^{(l-1)}$:

$$x_k^{(l)} = relu(\textstyle\sum_{k'} w_{k'}^{(l-1)} z_{k'}^{(l-1)} x_{k'}^{(l-1)}). \tag{10}$$

If $\sum_{k'} w_{k'}^{(l-1)} z_{k'}^{(l-1)} x_{k'}^{(l-1)} < 0$, there will be a zero term in S_l which can be ignored. So we just care about the case when $\sum_{k'} w_{k'}^{(l-1)} z_{k'}^{(l-1)} x_{k'}^{(l-1)} \geq 0$, then

$$x_k^{(l)} = \textstyle\sum_{k'} w_{k'}^{(l-1)} z_{k'}^{(l-1)} x_{k'}^{(l-1)}. \tag{11}$$

So,

$$S_l = \textstyle\sum_k \alpha_k'^{(l)} \sum_{k'} w_{k'}^{(l-1)} z_{k'}^{(l-1)} x_{k'}^{(l-1)}. \tag{12}$$

Note that $\alpha_k'^{(l)} \geq 0$, then

$$S_l = \textstyle\sum_{k'} (\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) z_{k'}^{(l-1)} x_{k'}^{(l-1)}. \tag{13}$$

Next, update the gates of ReLU layer $l-1$ based on S_l.
Note that

$$\alpha_{k'}^{(l-1)} = \tfrac{\partial S_l}{\partial x_{k'}^{(l-1)}} = (\textstyle\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) z_{k'}^{(l-1)}. \tag{14}$$

Update $z_{k'}^{(l-1)} \rightarrow z_{k'}'^{(l-1)}$ by

$$z_{k'}'^{(l-1)} = \delta(\tfrac{\partial S_l}{\partial x_{k'}^{(l-1)}}) \tag{15}$$

and $\alpha_{k'}^{(l-1)} \rightarrow \alpha_{k'}'^{(l-1)}$ by

$$\begin{aligned} \alpha_{k'}'^{(l-1)} &= (\textstyle\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) * \delta(\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) \\ &= \frac{\partial S_l}{\partial x_{k'}^{(l-1)}} * \delta(\frac{\partial S_l}{\partial x_{k'}^{(l-1)}}) \\ &= (\textstyle\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) * z_{k'}'^{(l-1)} \\ &\geq (\textstyle\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) * z_{k'}^{(l-1)}. \end{aligned} \tag{16}$$

Note that $\alpha_{k'}'^{(l-1)} \geq 0$ and $x_{k'}^{(l-1)} \geq 0$, so

$$\begin{aligned} S_{l-1} &= \textstyle\sum_{k'} \alpha_{k'}'^{(l-1)} x_{k'}^{(l-1)} \\ &\geq \textstyle\sum_{k'} (\sum_k \alpha_k'^{(l)} w_{k'}^{(l-1)}) z_{k'}^{(l-1)} x_{k'}^{(l-1)} = S_l. \end{aligned} \tag{17}$$

That is

$$S_l \leq S_{l-1}. \tag{18}$$

Based on the above mathematical induction, after the first iteration, the following conclusion can be drawn:

$$S_N \leq S_1. \tag{19}$$

The score S will keep increasing until convergence.

APPENDIX B
PROOF OF FEEDBACK SELECTIVE PRUNING

To prove that the FSP algorithm (Algorithm 2) can also reach local optimum, the mathematical induction method is adopted again. In this case, we need to prove $S_1 \leq S_N$. We first prove that $S \leq S_1$ and $S_1 \leq S_2$, and then illustrate $S_l \leq S_{l+1}$ under the assumption of $S_{l-1} \leq S_l$.

1. When $l = 1$:

We expand S with the first feedback layer as:

$$S = \sum_k \alpha_k^{(1)} z_k^{(1)} x_k^{(1)}, \tag{20}$$

where $x_k^{(1)}$ is an output of a neuron in the first ReLU layer, so $x_k^{(1)} \geq 0$.
Since $z_k^{(1)} \in \{0, 1\}$, we have

$$\alpha_k^{(1)} z_k^{(1)} x_k^{(1)} \leq \alpha_k^{(1)} \delta(\alpha_k^{(1)}) x_k^{(1)}. \tag{21}$$

Let $z_k^{(1)} \rightarrow z_k'^{(1)} = \delta(\alpha_k^{(1)})$, then

$$x_k'^{(1)} = x_k^{(1)} * \delta(\alpha_k^{(1)}) \geq 0 \tag{22}$$

and thus

$$S \leq S_1 = \textstyle\sum_k \alpha_k^{(1)} x_k'^{(1)}. \tag{23}$$

After updating all $z_k^{(1)}$ in the first feedback layer, S_1 can be expanded with the second feedback layer. Note that $x_k^{(2)}$ will be changed to $\hat{x}_k^{(2)}$ because of $x_k^{(1)}$ being modified, that is

$$S_1 = \textstyle\sum_k \alpha_k^{(2)} z_k^{(2)} \hat{x}_k^{(2)}. \tag{24}$$

Update $z_k^{(2)}$ and $\hat{x}_k^{(2)}$ to get S_2 with the same way when we update $z_k^{(1)}$ and $x_k^{(1)}$, therefore:

$$S_1 \leq S_2. \tag{25}$$

2. Let us assume that $S_{l-1} \leq S_l$:
Fix $z_k^{(1)}, z_k^{(2)}, \ldots, z_k^{(l-1)}$, and then

$$S_l = \textstyle\sum_k \alpha_k^{(l)} x_k'^{(l)}. \tag{26}$$

The score S can be expressed by $x_k^{(l+1)}$, so

$$S_l = \textstyle\sum_k \alpha_k^{(l+1)} \hat{x}_k^{(l+1)} z_k^{(l+1)}, \tag{27}$$

where

$$\hat{x}_k^{(l+1)} = relu(\textstyle\sum_{k'} w_{k'}^{(l)} z_{k'}^{(l)} x_{k'}'^{(l)}) \geq 0. \tag{28}$$

Because

$$\begin{aligned} S_l &= \textstyle\sum_k \alpha_k^{(l+1)} \hat{x}_k^{(l+1)} z_k^{(l+1)} \\ &\leq \textstyle\sum_k \alpha_k^{(l+1)} \delta(\alpha_k^{(l+1)}) \hat{x}_k^{(l+1)} \\ &= \textstyle\sum_k \alpha_k^{(l+1)} x_k'^{(l+1)} = S_{l+1}, \end{aligned} \tag{29}$$

that is, $S_l \leq S_{l+1}$.
Then update $\hat{x}_k^{(l+1)} \rightarrow x_k'^{(l+1)}$ and $z_k^{(l+1)} \rightarrow z_k'^{(l+1)}$ with

$$\begin{aligned} x_k'^{(l+1)} &= \hat{x}_k^{(l+1)} \delta(\alpha_k^{(l+1)}) \\ z_k'^{(l+1)} &= \delta(\alpha_k^{(l+1)}). \end{aligned} \tag{30}$$

Based on the above mathematical induction method, after the first iteration, we have

$$S_1 \leq S_N. \tag{31}$$

The target S will keep increasing until convergence.