2. Computation of $\Delta \boldsymbol{A}_{k+1}$ and $\Delta \boldsymbol{B}_{k+1}$

We illustrate the computation of $\Delta \boldsymbol{A}_{k+1}$ and $\Delta \boldsymbol{B}_{k+1}$. The component $\boldsymbol{D}_{k+1}^T \boldsymbol{E}_{1,k}$ of $\Delta \boldsymbol{A}_{1,k+1}$ is taken as an example to illustrate the recursion calculation manner.

$$\boldsymbol{D}_{k+1}^T \boldsymbol{E}_{1,k} = \boldsymbol{d}_{k+1}^T \Delta \boldsymbol{x}_{k+1}^p \boldsymbol{J}_k^T \boldsymbol{E}_{1,k} \tag{S1}$$

If we compute (S1) directly, the computational complexity is $O(kp^2 m_1^2)$ and increases continually. We should compute $\boldsymbol{J}_k^T \boldsymbol{E}_{1,k}$ recursively.

$$\begin{aligned}
\boldsymbol{J}_k^T \boldsymbol{E}_{1,k} = & \begin{bmatrix} \boldsymbol{J}_{k-1} \tilde{\boldsymbol{J}}_k \\ \Delta \boldsymbol{x}_k^p \boldsymbol{R}_{k-1} \tilde{\boldsymbol{J}}_k \end{bmatrix}^T \begin{bmatrix} \boldsymbol{E}_{1,k-1} - \boldsymbol{J}_{k-1} \left(\Delta \boldsymbol{x}_k^p\right)^T \boldsymbol{h}_k \\ \boldsymbol{h}_k \end{bmatrix} \\
= & \tilde{\boldsymbol{J}}_k^T \boldsymbol{J}_{k-1}^T \left(\boldsymbol{E}_{1,k-1} - \boldsymbol{J}_{k-1} \left(\Delta \boldsymbol{x}_k^p\right)^T \boldsymbol{h}_k \right) + \left(\Delta \boldsymbol{x}_k^p \boldsymbol{R}_{k-1} \tilde{\boldsymbol{J}}_k \right)^T \boldsymbol{h}_k \\
= & \tilde{\boldsymbol{J}}_k^T \boldsymbol{J}_{k-1}^T \boldsymbol{E}_{1,k-1} - \tilde{\boldsymbol{J}}_k^T \boldsymbol{J}_{k-1}^T \boldsymbol{J}_{k-1} \left(\Delta \boldsymbol{x}_k^p\right)^T \boldsymbol{h}_k + \tilde{\boldsymbol{J}}_k^T \boldsymbol{R}_{k-1}^T \left(\Delta \boldsymbol{x}_k^p\right)^T \boldsymbol{h}_k
\end{aligned} \tag{S2}$$

The recursion of $\boldsymbol{J}_k^T \boldsymbol{J}_k$ is:

$$\boldsymbol{J}_k^T \boldsymbol{J}_k = \tilde{\boldsymbol{J}}_k^T \boldsymbol{J}_{k-1}^T \boldsymbol{J}_{k-1} \tilde{\boldsymbol{J}}_k + \tilde{\boldsymbol{J}}_k^T \boldsymbol{R}_{k-1}^T \left(\Delta \boldsymbol{x}_k^p\right)^T \Delta \boldsymbol{x}_k^p \boldsymbol{R}_{k-1} \tilde{\boldsymbol{J}}_k \tag{S3}$$

As $\tilde{\boldsymbol{J}}_k = \boldsymbol{I} - \frac{(\Delta \boldsymbol{x}_k^p)^T \Delta \boldsymbol{x}_k^p \boldsymbol{R}_{k-1}}{1+c_k}$, $\boldsymbol{D}_{k+1}^T \boldsymbol{E}_{1,k}$ can be calculated recursively based on (S1)-(S3). Similarly, other components of $\Delta \boldsymbol{A}_{k+1}$ and $\Delta \boldsymbol{B}_{k+1}$ can also be computed recursively.

The computational complexity of $\tilde{\boldsymbol{J}}_k^T \boldsymbol{J}_{k-1}^T \boldsymbol{J}_{k-1} \tilde{\boldsymbol{J}}_k$ is $O(p^2 m_1^2)$ based on the associative law of multiplication. As $\Delta \boldsymbol{x}_k^p$ is a vector, the computational complexity of (S1) is $O(p^2 m_1^2)$ by reasonable arrangement of matrix calculation order. It is understandable because each component contains at least one vector. Similarly, other components of $\Delta \boldsymbol{A}_{k+1}$ and $\Delta \boldsymbol{B}_{k+1}$ need $O(p^2 m_1^2)$. In conclusion, the computational complexity of $\Delta \boldsymbol{A}_{k+1}$ and $\Delta \boldsymbol{B}_{k+1}$ is $O(p^2 m_1^2)$ per update, which is independent of the number of existing samples.