Appendix

1 Sliding Window

In this appendix, we provide detailed derivation for **equation (1)** and **equation (2)** in the main paper, i.e. the accumulated delay for fixed-size and dynamic sliding window approaches. Audio denoising can be divided into three steps: audio recording, denoised network processing and denoised audio playback. We show the fixed-size and dynamic sliding window examples in Fig. 1 and Fig. 2 respectively. In the figures, our timeline is from left to right.

We use block to indicate a certain time span. Each rectangle in the figures is regarded as a block. The blank area represents the waiting time between two neighbor blocks. We use R_i, N_i and P_i to indicate the recording, processing and playback block, use $s(block)$ and $e(block)$ to indicate the start time and end time of each block. In particular, we define t_i as the network processing time for a processing block, i.e. $t_i = e(N_i) - s(N_i)$, and d_i as the total delay of a certain audio block i. d_i is the time from the moment of receiving to the moment of outputting , i.e. $d_i = s(P_i) - s(R_i)$.

We first claim some facts for sliding window approach:
(1). There is no blank area for all recording block, i.e. $s(R_i) = e(R_{i-1})$ for any $i > 1$.
(2). The network will process current audio immediately when there exists unprocessed audio and the previous audio has been processed, i.e. $s(N_i) = \max(e(R_i),\ e(N_{i-1}))$, for any $i > 1$.
(3). The player will play current audio immediately when there exists unplayed audio and the previous audio has been played, i.e. $s(P_i) = \max(e(N_i),\ e(P_{i-1}))$, for any $i > 1$.
(4). The recording block and the playback contains the audio of the same length, i.e. $e(R_i) - s(R_i) = e(P_i) - s(P_i)$ for any $i > 0$. In particular, $e(R_i) - s(R_i) = L$ for any i in fixed-size sliding window approach. Here L is the fixed window size.
(5). Over time, the total delay will be accumulated, i.e. $d_i \leq d_{i-1}$ for any $i > 1$.

The following analysis shows the theoretical delay d_i, which in practices is smaller than the actual measured audio playback delay (D_A) as introduced in main paper.

1.1 Fixed-Size Sliding Window

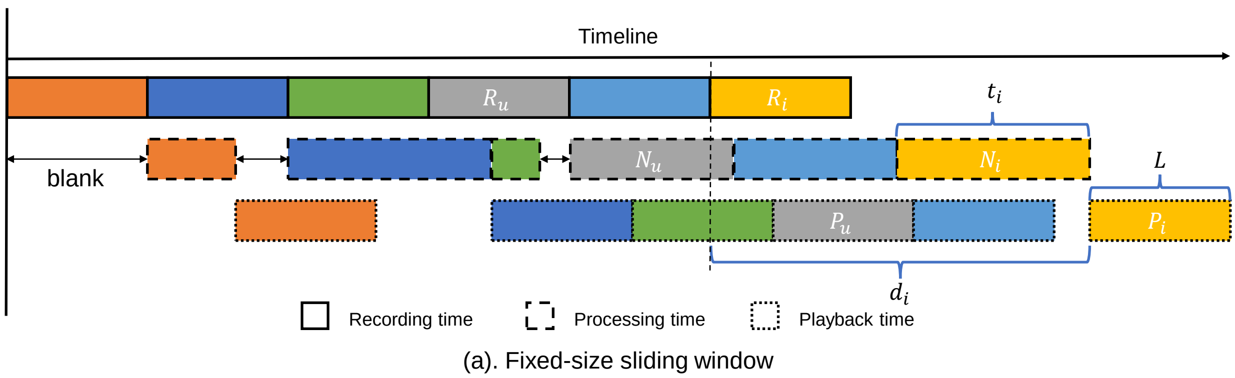


Figure 1: **Fixed-size sliding window**

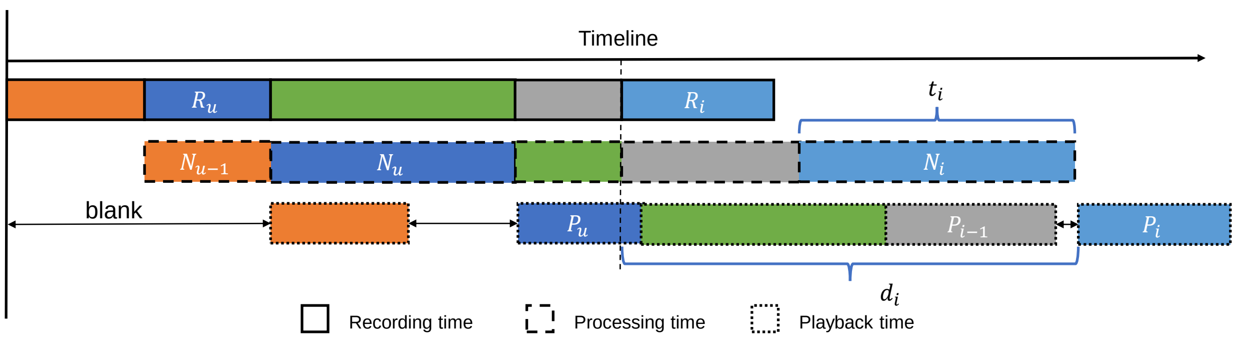
To calculate d_i in fixed-size sliding window approach, we first find the previous processing blank area, and the processing block N_u right after that blank. Thus, by fact (2), we have $e(R_u) = s(N_u)$. There are $i - u$ blocks with length L between R_i and R_u, thus we have $s(R_i) = S(R_u) + (i-u)L$ and $e(P_{i-1}) = e(P_{u-1}) + (i-u)L$. By fact (3), we can derive

$$\begin{aligned} d_i &= s(P_i) - s(R_i) \\ &= \max\left(e(N_i),\ e(P_{i-1})\right) - s(R_i) \\ &= \max\left(s(N_u) + \sum_{k=u}^{i} t_k,\ e(P_{u-1}) + (i-u)L)\right) - s(R_i) \\ &= \max\left(e(R_u) + \sum_{k=u}^{i} t_k,\ e(P_{u-1}) + (i-u)L)\right) - (s(R_u) + (i-u)L) \\ &= \max\left(s(R_u) + L + \sum_{k=u}^{i} t_k - s(R_u) - (i-u)L,\ e(P_{u-1}) - s(R_u)\right) \\ &= \max\left(2L + \sum_{k=u}^{i}(t_k - L),\ e(P_{u-1}) - e(R_{u-1})\right) \\ &= \max\left(2L + \sum_{k=u}^{i}(t_k - L),\ d_u\right) \end{aligned} \tag{1}$$

Through recursion, we can get

$$d_i = 2L + \max_{1 \le p \le q \le i} \sum_{k=p}^{q} (t_k - L) \tag{2}$$

1.2 Dynamic Sliding Window



(b). Dynamic sliding window

Figure 2: **Dynamic sliding window**

For dynamic sliding window approach, the recording time is always equal the previous processing time, i.e. $N_{i-1} = R_i = P_i$ for any $i > 1$. We have three cases that need to be analyzed.

Case 1: If there is no blank area between P_i and P_{i-1}, for any $i > 1$,

$$\begin{aligned} d_i &= s(P_i) - s(R_i) \\ &= e(P_{i-1}) - e(R_{i-1}) \\ &= d_{i-1} \end{aligned} \tag{3}$$

Case 2: If there is a blank area before P_i, and $i > 1$, then we can always find the previous playback blank area and the playback block P_u right after the blank area. By fact (3), we can derive

$$\begin{aligned} d_i &= s(P_i) - s(R_i) \\ &= \max\left(e(N_i),\ e(P_{i-1})\right) - s(R_i) \\ &= \max\left(s(N_u) + \sum_{k=u}^{i} t_k,\ s(P_u) + \sum_{k=u}^{i-1}(e(P_k) - s(P_k))\right) - \left(s(R_{u+1}) + \sum_{k=u+1}^{i-1}(e(P_k) - s(P_k))\right) \\ &= \max\left(s(N_u) + \sum_{k=u}^{i} t_k,\ s(P_u) + \sum_{k=u}^{i-1}(e(N_{k-1}) - s(N_{k-1}))\right) - \left(s(R_{u+1}) + \sum_{k=u+1}^{i-1}(e(N_k - 1) - s(N_k - 1))\right) \\ &= \max\left(s(N_u) + \sum_{k=u}^{i} t_k,\ s(P_u) + \sum_{k=u}^{i-1} t_{k-1}\right) - \left(s(R_{u+1}) + \sum_{k=u+1}^{i-1} t_{k-1}\right) \\ &= \max\left(\sum_{k=u}^{i} t_k,\ t_u + \sum_{k=u}^{i-1} t_{k-1}\right) - \sum_{k=u+1}^{i-1} t_{k-1} \\ &= \max\left(t_{i-1} + t_i,\ t_{u-1} + t_u\right) \end{aligned} \tag{4}$$

Through recursion, we can get

$$d_i = \max_{k \le i}\left(t_{k-1} + t_k\right), \quad i \ge 2. \tag{5}$$

Case 3: When $i = 1$, $d_1 = L_0 + t_1$, where L_0 is an initial window size to start the denoising process at the beginning.

In summary, we have

$$d_i = \begin{cases} L_0 + t_1, & i = 1 \\ \max_{k \le i}\left(t_{k-1} + t_k\right), & i \ge 2 \end{cases} \tag{6}$$