$$\mathbf{w}_t^u \leftarrow \gamma \mathbf{w}_{t-1}^u + \mathbf{w}_t^r + \mathbf{w}_t^w. \tag{5}$$

Here, γ is a decay parameter and $\mathbf{w}_t^r$ is computed as in (3). The *least-used* weights, $\mathbf{w}_t^{lu}$, for a given time-step can then be computed using $\mathbf{w}_t^u$. First, we introduce the notation $m(\mathbf{v}, n)$ to denote the n^{th} smallest element of the vector $\mathbf{v}$. Elements of $\mathbf{w}_t^{lu}$ are set accordingly:

$$w_t^{lu}(i) = \begin{cases} 0 & \text{if } w_t^u(i) > m(\mathbf{w}_t^u, n) \\ 1 & \text{if } w_t^u(i) \leq m(\mathbf{w}_t^u, n) \end{cases}, \tag{6}$$

where n is set to equal the number of reads to memory. To obtain the write weights $\mathbf{w}_t^w$, a learnable sigmoid gate parameter is used to compute a convex combination of the previous read weights and previous least-used weights:

$$\mathbf{w}_t^w \leftarrow \sigma(\alpha)\mathbf{w}_{t-1}^r + (1 - \sigma(\alpha))\mathbf{w}_{t-1}^{lu}. \tag{7}$$

Here, $\sigma(\cdot)$ is a sigmoid function, $\frac{1}{1+e^{-x}}$, and α is a scalar gate parameter to interpolate between the weights. Prior to writing to memory, the least used memory location is computed from $\mathbf{w}_{t-1}^u$ and is set to zero. Writing to memory then occurs in accordance with the computed vector of write weights:

$$\mathbf{M}_t(i) \leftarrow \mathbf{M}_{t-1}(i) + w_t^w(i)\mathbf{k}_t, \forall i \tag{8}$$