Supplementary Material

A. Derivation for loss equivalence

In this appendix section, we provide the derivation to show that if we would predict bounding boxes B_{t+l} in the trajectory network, instead of offsets between pairs of bounding boxes δ_{t+l}, Eq. (2) would reduce to Eq. (1) and the temporal ordering would not be enforced.

If we predict bounding boxes B_{t+l} and use $\delta_{t+k} = (B_{t+k} - B_{t+k-1}), k \in \{1, .., l\}$, the sum $\sum_{k=1}^{l} \delta_{t+k}$ can be rewritten as follows:

$$\begin{aligned}
\sum_{k=1}^{l} \delta_{t+k} &= \sum_{k=1}^{l} (B_{t+k} - B_{t+k-1}), \\
&= \sum_{k=1}^{l} (B_{t+k}) - \sum_{k=1}^{l} (B_{t+k-1}), \\
&= \sum_{k=1}^{l} (B_{t+k}) - \sum_{k=0}^{l-1} (B_{t+k}), \\
&= \sum_{k=1}^{l-1} (B_{t+k}) + B_{t+l} - \sum_{k=1}^{l-1} (B_{t+k}) - B_t, \\
&= B_{t+l} - B_t.
\end{aligned}$$

And we fill the above in Eq. (2). Then we have,

$$\begin{aligned}
L_{\sum}(B^*, \overleftrightarrow{\mathbb{T}_t}) &= \sum_{l=0}^{T} \mathcal{L}_1 \left((B^*_{t+l} - B_t) - \sum_{k=1}^{l} \delta_{t+k} \right), \\
&= \sum_{l=0}^{T} \mathcal{L}_1 \left(B^*_{t+l} - B_t - \sum_{k=1}^{l} \delta_{t+k} \right), \\
&= \sum_{l=0}^{T} \mathcal{L}_1 \left(B^*_{t+l} - B_t - B_{t+l} + B_t \right), \\
&= \sum_{l=0}^{T} \mathcal{L}_1 \left(B^*_{t+l} - B_{t+l} \right).
\end{aligned}$$

which is the same as Eq. (1):

$$L_{\text{bag}}(B^*, \{B_t, .., B_{t+T}\}) = \sum_{l=0}^{T} \mathcal{L}_1 (B^*_{t+l} - B_{t+l}).$$

B. Details for the *Simulated smooth motion*

In this section, we describe how we create the *Simulated smooth motion*: bounding boxes move between keyframes according to a smooth parabola, and the change of width and height is linearly interpolated. Given the center points of two keyframe digits (x_t, y_t) and (x_{t+T}, y_{t+T}), we choose the focus $F = (0, f), f = 8$ for the parabola, then the parabola can be written as,

$$y = \frac{1}{4f} x^2 - \frac{v_1}{2f} x + \frac{v_1^2}{4f} + v_2, \tag{5}$$

where the vertex is $V = (v_1, v_2)$. By filling in (x_t, y_t) and (x_{t+T}, y_{t+T}), we can get the value of v_1, v_2. For every pairwise neighbouring keyframes, we can have a parabola that acts as a simulated smooth trajectory for intermediate locations of digits. Here we show an example of having four keyframes and the simulated smooth motion as a parabola in Fig. 7. Because the digits move linearly in MovingDigits dataset, the digits of the keyframes stay on a linear line. We choose the focus of every second parabola sequence to be $F = (0, -8)$ to make all the parabola trajectories smoothly connected.

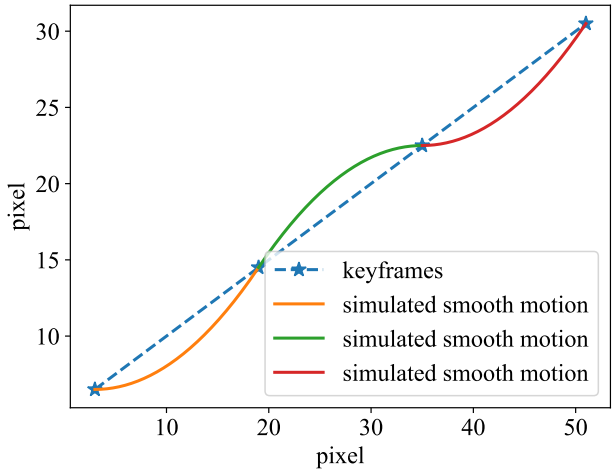

Figure 7. An example of simulated smooth motion generated by parabola functions. The parabola represents the trajectory of intermediate digit locations between every two keyframes. The simulated motion is smooth and continuous.