A. Supplementary Materials for Experiments

A.1. Implementation Details

Model Configuration There are some common model configurations for all tasks. For the NN^{feat} defined in (1), we set it as a FCN, where each convolution layer is composed via convolution, adaptive max-pooling, and ReLU and the convolution stride is set to 1 for all layers. For the RNN defined in (14), we set it as a Gated Recurrent Unit (GRU) [11] to capture long-range temporal dependencies. For the NN^{out} defined in (3), we set it as a Fully-Connected network (FC), where the ReLU is chosen as the activation function for each hidden layer. For the loss defined in (9), we set $\lambda = 1$. For the model configurations specified to each task, please see in Table 3. Note that to use attention, the receptive field of $\boldsymbol{c}_{t,m,n}$ is crafted as a local region on $\boldsymbol{X}_t$, i.e., 40×40 for MNIST-MOT and Sprites-MOT, and 44×24 for DukeMTMC (this can be calculated using the FCN hyper-parameters in Table 3).

Training Configuration For MNIST-MOT and Sprites-MOT, we split the data into a proportion of 90/5/5 for training/validation/test; for DukeMTMC, we split the provided training data into a proportion of 95/5 for training/validation. For all tasks, in each iteration we feed the model with a mini-batch of 64 subsequences of length 20. During the forward pass, the tracker states and confidences at the last time step are preserved to initialize the next iteration. To train the model, we minimize the averaged loss on the training set w.r.t. all model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{feat}, \boldsymbol{\theta}^{upd}, \boldsymbol{\theta}^{out}\}$ using Adam [32] with a learning rate of 5×10^{-4}. Early stopping is used to terminate training.

A.2. MNIST-MOT

As a pilot experiment, we focus on testing whether our model can robustly track the position and appearance of each object that can appear/disappear from the scene. Thus, we create a new MNIST-MOT dataset containing 2M frames, where each frame is of size 128×128×1, consisting of a black background and at most three moving digits. Each digit is a 28×28×1 image patch randomly drawn from the MNIST dataset [36], moves towards a random direction, and appears/disappears only once. When digits overlap, pixel values are added and clamped in $[0, 1]$. To solve this task, for TBA configurations we set the tracker number $I = 4$ and layer number $K = 1$, and fix the scale $s^x_{t,i} = s^y_{t,i} = 1$ and shape $\boldsymbol{Y}^s_{t,i} = \mathbf{1}$, thereby only compositing a single layer by adding up all transformed appearances. We also clamp the pixel values of the reconstructed frames in $[0, 1]$ for all configurations.

Training curves are shown in Fig. 9. The TBA, TBAc, and TBAc-noRep have similar validation losses which are slightly better than that of TBAc-noAtt. Similar to the results on Sprites-MOT, TBA converges the fastest, and TBAc-noMem has a significantly higher validation loss as all trackers are likely to focus on a same object, which affects the reconstruction.

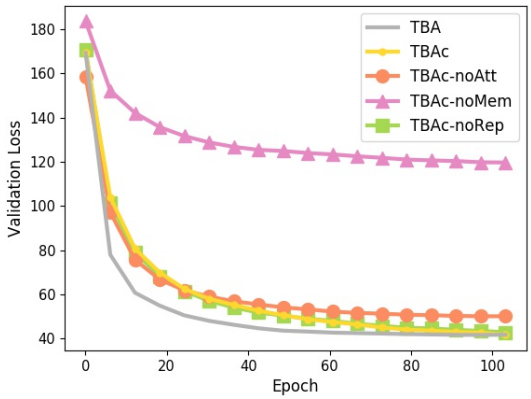

Figure 9: Training curves of different configurations on MNIST-MOT.

Qualitative results are shown in Fig. 10. Similar phenomena are observed as in Sprites-MOT, revealing the importance of the disabled mechanisms. Specifically, as temporal dependency is not considered in AIR, overlapped objects are failed to be disambiguated (Seq. 5).

We further quantitatively evaluate different configurations. Results are reported in Table 4, which are similar to those of the Sprites-MOT.

Table 3: Model configurations specified to each task, where 'conv h×w' denotes a convolution layer with kernel size h×w, 'fc' denotes a fully-connected layer, and 'out' denotes an output layer. Note that for NN^{feat}, the first layer has two additional channels than $\boldsymbol{X}_t$, which are the 2D image coordinates (as mentioned in Sec. 3.1).

Hyper-parameter	**MNIST-MOT**	**Sprites-MOT**	**DukeMTMC**
Size of $\boldsymbol{X}_t$: $[H, W, D]$	[128, 128, 1]	[128, 128, 3]	[108, 192, 3]
Size of $\boldsymbol{C}_t$: $[M, N, S]$	[8, 8, 50]	[8, 8, 20]	[9, 16, 200]
Size of $\boldsymbol{Y}^a_{t,i}$: $[U, V, D]$	[28, 28, 1]	[21, 21, 3]	[9, 23, 3]
Size of $\boldsymbol{h}_{t,i}$: R	200	80	800
Tracker number: I	4	4	10
Layer number: K	1	3	3
Coef. of $[\widehat{s}^x_{t,i}, \widehat{s}^y_{t,i}]$: $[\eta^x, \eta^y]$	[0, 0]	[0.2, 0.2]	[0.4, 0.4]
Layer sizes of NN^{feat} (FCN)	[128, 128, 3] (conv 5×5) [64, 64, 32] (conv 3×3) [32, 32, 64] (conv 1×1) [16, 16, 128] (conv 3×3) [8, 8, 256] (conv 1×1) [8, 8, 50] (out)	[128, 128, 5] (conv 5×5) [64, 64, 32] (conv 3×3) [32, 32, 64] (conv 1×1) [16, 16, 128] (conv 3×3) [8, 8, 256] (conv 1×1) [8, 8, 20] (out)	[108, 192, 5] (conv 5×5) [108, 192, 32] (conv 5×3) [36, 64, 128] (conv 5×3) [18, 32, 256] (conv 3×1) [9, 16, 512] (conv 1×1) [9, 16, 200] (out)
Layer sizes of NN^{out} (FC)	200 (fc) 397 (fc) 787 (out)	80 (fc) 377 (fc) 1772 (out)	800 (fc) 818 (fc) 836 (out)
Number of parameters	1.21 M	1.02 M	5.65 M

Table 4: Tracking performances of different configurations on MNIST-MOT.

Configuration	**IDF1↑**	**IDP↑**	**IDR↑**	**MOTA↑**	**MOTP↑**	**FAF↓**	**MT↑**	**ML↓**	**FP↓**	**FN↓**	**IDS↓**	**Frag↓**
TBA	99.6	99.6	99.6	99.5	78.4	0	978	0	49	49	22	7
TBAc	99.2	99.3	99.2	99.4	78.1	0.01	977	0	54	52	26	11
TBAc-noAtt	45.2	43.9	46.6	59.8	81.8	0.20	976	0	1,951	219	6,762	86
TBAc-noMem	0	–	0	0	–	0	0	983	0	22,219	0	0
TBAc-noRep	94.3	92.9	95.7	98.7	77.8	0.01	980	0	126	55	103	10

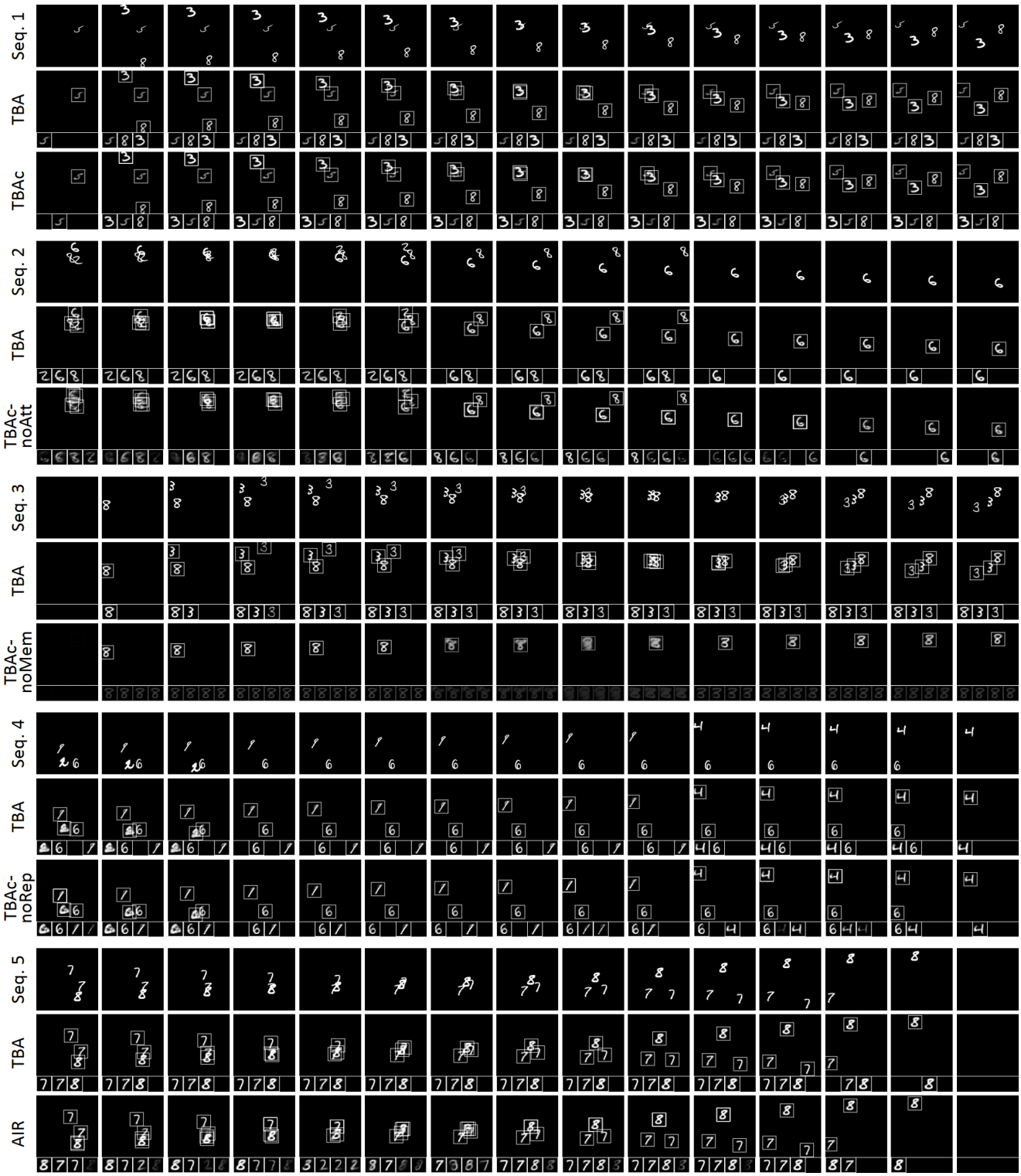

Figure 10: Qualitative results of different configurations on MNIST-MOT. For each configuration, we show the reconstructed frames (top) and the tracker outputs (bottom). For each frame, tracker outputs from left to right correspond to tracker 1 to I (here $I=4$), respectively. Each tracker output $\mathcal{Y}_{t,i}$ is visualized as $\left(y^c_{t,i}\ \boldsymbol{Y}^s_{t,i} \odot \boldsymbol{Y}^a_{t,i}\right) \in [0,1]^{U \times V \times D}$.