APPENDIX

1.1 Dataset Descriptions

We five real-world datasets and one synthetic dataset for comprehensive evaluations: Synthetic [35]: It is a synthetic dataset, generated by the MTS generator TSAGen [36]; MoCap [24]: Derived from the CMU motion capture repository. In this dataset, every motion is represented as a sequence of hundreds of frames; ActRecTut [6]:This dataset involves two participants performing hand movements with height gestures in daily life and 3D gestures while playing tennis; PAMAP2 [28]: Covering both basic (e.g., walking, sitting) and composite human activities (e.g., soccer), this dataset features data from eight individuals; UscHad [42]: This dataset encompasses 12 distinct human activities such as jumping and running, recorded for 14 individuals; UcrSeg [15]: This dataset encompasses diverse sources, including medical fields, insect studies, robotics, and power demand data.

A summary of the statistics is provided in Table 1, all the datasets can be available at http://bit.ly/3rMFJVv.

Table 1: Statistics of datasets used.

Dataset	MTS	State	Variate	Length (k)	State Duration (k)
Synthetic	100	5	4	9.3-23.7	0.1-3.9
MoCap	9	5-8	4	4.6-10.6	0.4-2.0
ActRecTut	2	6	10	31.4-32.6	0.02-5.1
PAMAP2	10	11	9	253-408	2.0-40.3
UscHad	70	12	6	25.4-56.3	0.6-13.5
UcrSeg	32	2-3	1	2-40	1-25

1.2 Implementation Details

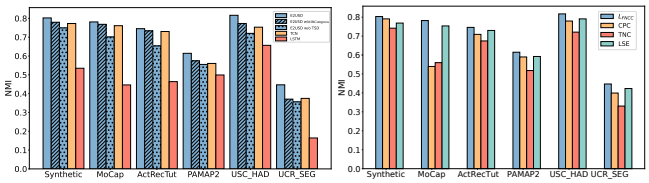
Experiments were conducted on a server with an NVIDIA Quadro RTX 8000 GPU and an Intel Xeon Gold 5215 CPU (2.50GHz). For FFTCOMPRESS, $Q = 33$, while in DDEM, $k_a = 5$, $C = 80$ random convolution kernels ($k_c = 3$), and final embedding dimension $D = 4$. FNCCLEARNING used $U = 20$ groups of windows (each with $V = 4$ neighboring windows) and threshold $\lambda = 0.5$. Sliding window sizes were 128, 256, or 512, dataset-dependent, with step size $B = 50$. Adam optimizer [20] was used with a learning rate of 0.003 for 20 epochs.

The MCU deployment uses an STM32H747 device [2] with a 480 MHz Arm Cortex-M7 core, 2 MB Flash memory, and 1 MB RAM, as presented in . The E2USD model was converted to ONNX format and translated to C code with X-CUBE-AI [3]. The C code was compiled using an ARM-specific version of GCC to create an executable binary.

Fig. 1: STM32H747 device.

1.3 NMIs of Component Study

In Fig. 2, we present the NMI comparisons for both encoders and losses. We note that the trends in NMIs are basically consistent with the corresponding ARIs shown in Fig. 7. This observation further substantiates the effectiveness of our proposed encoder and loss components.



(a) Encoder **(b) Loss**

Fig. 2: NMI comparison

1.4 Parameter Sensitivity Analysis

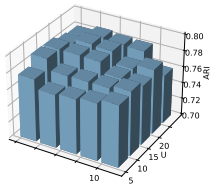


Fig. 3: Impact of U and V.

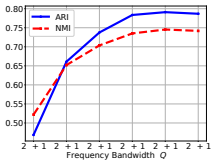


Fig. 4: Impact of Q.

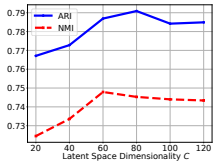


Fig. 5: Impact of C.

We conduct a comprehensive parameter sensitivity analysis using the ActRecTut dataset, focusing on assessing how these key parameters affect the ARI and NMI. The parameters under scrutiny are:

- U and V: These parameters jointly contribute to the computation of L_{FNCC}. More specifically, U designates the number of distinct window groups, whereas V defines the number of consecutive windows within each group.
- Q: This parameter serves as the size of frequency bandwidth of Energy-based Frequency Compression (EFC) on the FFTCOMPRESS. It has a direct bearing on the FFTCOMPRESS's compression rate.
- C: This parameter represents the dimensionality of the latent space of DDEM.

Impact of U and V: As shown in Fig. 3, altering V does not consistently influence ARI. This unexpected result could be attributed to larger V values capturing broader temporal scopes, thereby introducing false positives due to state transitions. This implies that while increasing V appears to be beneficial for capturing more data, it may inadvertently degrade performance. Conversely, ARI remains stable across a range of U values, highlighting the robustness of E2USD.

Impact of Q: Fig. 4 exhibits two key trends. Lower Q values compromise ARI and NMI due to aggressive data compression, causing the loss of essential information. On the other hand, elevating Q leads to performance plateaus or minor reductions, likely because of the introduction of noise.

Impact of C: As illustrated in Fig. 5, enlarging the embedding size C generally boosts both ARI and NMI, peaking at $C = 80$. Further increases in C result in diminishing returns and even minor performance setbacks.