Appendix3: Random Seeds and Reproducibility (Unified Explanation)

Notice: This appendix serves as the sole declaration of random seeds and reproducibility; it will not be repeated in the main text.

1.Fixed Dataset Splitting Seed (S_split)

Each subset contains 850 images. The splitting strategy is as follows: randomly select 150 images for Test using S_split, then randomly select 150 images for Val from the remaining samples using S_split+1, and the remaining 550 images will be used for Train.

Before splitting, the samples are stably sorted by their relative paths. The results are saved in train.list, val.list, and test.list, along with metadata files. The dataset will not be re-split during the training phase.

Dataset	S_split	Train	val	test
WSD-E1	202501	550	150	150
WSD-E3	202501	550	150	150
WSD-E5	202501	550	150	150
WSD-F1	202501	550	150	150
total		2200	600	600

Note: In the table, S_split+1 is used solely to randomly select the validation set from the remaining samples, in order to reduce the correlation between the two-step sampling process.

2.Independent Experiment Training Seed (S_train)

On the same fixed split, to assess the impact of training randomness (parameter initialization, mini-batch shuffling, data augmentation, Dropout masks, etc.), K independent training runs are conducted.

In this study, S_train $\in$ {0, 1, 2, 3, 4} (K=5). Only S_train is changed for each run, with other data and hyperparameters remaining consistent. The validation/testing phases are kept deterministic.

If model comparisons are made, paired evaluations are performed within the same S_train group to enhance statistical power.

Run -number	**Run-1**	**Run-1**	**Run-1**	**Run-1**	**Run-1**
S_train	0	1	2	3	4

3.Metric Summary and Statistical Testing

For the results of the same model across K independent runs, report the mean ± standard deviation (mean ± SD).

For model comparisons, a paired design is used (within the same S_train group), and either paired t-test or Wilcoxon signed-rank test can be employed.

4.Reproducibility Best Practices

Before training, synchronize the random state of random, NumPy, and the deep learning framework. Specify a generator for DataLoader and derive sub-seeds in worker_init.

Disable automatic acceleration strategies that may introduce non-determinism (e.g., disable cudnn.benchmark, enable deterministic operators if necessary).

Record in the metadata: S_split, S_train, framework and dependency versions, GPU/driver information, and key training hyperparameters.