Supplementary Data for Multi-SpaCE: Multi-Objective Subsequence-based Sparse Counterfactual Explanations for Multivariate Time Series Classification

S1. Model details

S1.1. Black-box Classifier

We trained InceptionTime with default hyperparameters following the implementation in `https://github.com/hfawaz/InceptionTime`

S1.2. Autoencoder

We trained eight different autoencoder architectures for each dataset to better adapt the models to the specific characteristics of the data, serving as a lighter alternative to grid search or Bayesian optimization. Each model randomly splits 10% of the training data for validation and is trained over 200 epochs, using early stopping with a patience of 30 epochs. Additionally, learning rate reduction is applied when the validation loss plateaus, with a patience of 10 epochs. The initial learning rate is set to 0.001, the batch size to 32, and a dropout rate of 20% is applied across all models.

Both the encoder and decoder are based on convolutional layers with fixed strides of 2 and include a Dense layer in between, enforcing a target compression rate of the input space to either 6.25% or 12.5%. If the dimensionality at the output of the encoder prevents achieving the desired compression rate, the model is not trained, and its performance is represented with a "-" in the results tables. The architectures vary based on the following parameters:

- **Shallow:** One convolutional layer with 16 channels and a kernel size of 7. The decoder performs the inverse operations.
- **Simple:** Two convolutional layers with 16 and 32 channels, with kernel sizes of 7 and 5, respectively. The decoder performs the inverse operations.
- **Intermediate:** Three convolutional layers with 16, 32, and 64 channels, with kernel sizes of 7, 5, and 3, respectively. The decoder performs the inverse operations.
- **Complex:** Four convolutional layers with 16, 32, 64, and 128 channels, with kernel sizes of 7, 5, 5, and 3, respectively. The decoder performs the inverse operations.

Tables S1 and S2 present the results for each architecture on the multivariate and univariate datasets, respectively.

Table S1: Autoencoder reconstruction errors on multivariate test sets.

Dataset	Shallow (6,25%)	Shallow (12,5%)	Simple (6,25%)	Simple (12,5%)	Interm. (6,25%)	Interm. (12,5%)	Complex. (6,25%)	Complex. (12,5%)
PEMS-SF	**0.02**	0.02	0.02	0.02	0.02	0.02	0.02	0.02
UWave	-	-	-	0.38	0.39	**0.07**	0.11	0.09
NATOPS	0.14	**0.1**	0.11	0.11	0.13	0.13	0.14	0.14
Cricket	-	0.56	0.49	0.32	0.3	0.21	0.19	**0.17**
AWR	0.56	0.53	0.43	0.33	0.33	**0.23**	0.28	0.28
Epilepsy	-	-	-	0.42	0.43	**0.31**	0.36	0.34
BasicMotions	-	1.42	1.59	**1.34**	1.63	1.58	1.91	1.79
RacketSports	-	2.76	2.77	**2.57**	2.76	2.73	2.81	2.76
PenDigits	-	-	-	5.03	6.57	4.25	6.29	**4.04**
SR-SCP1	-	9.93	**4.96**	5.16	6.06	7.48	6.05	5.94

Table S2: Autoencoder reconstruction errors on univariate test sets.

Dataset	Interm. (12,5%)	Complex (6,25%)	Complex (12,5%)
HandOutlines	**0.0**	0.0	0.0
Strawberry	0.02	0.02	**0.01**
NonInvasiveFatalECGThorax2	0.02	0.03	**0.02**
ProximalPhalanxOutlineCorrect	0.02	0.02	**0.02**
Plane	0.09	0.1	**0.08**
ECG5000	0.08	0.1	**0.08**
CinCECGTorso	**0.11**	0.16	0.15
FordA	**0.16**	0.27	0.16
ECG200	**0.17**	0.2	0.18
ItalyPowerDemand	0.26	**0.19**	0.21
TwoPatterns	**0.21**	0.29	0.22
Gunpoint	0.93	0.29	**0.28**
Phoneme	**0.28**	0.44	0.3
FacesUCR	0.44	0.52	**0.41**
Coffee	**0.63**	0.73	0.82
CBF	**0.68**	0.75	0.75

S1.3. Isolation Forest

We conduct a grid search over the following parameters to optimize the Isolation Forest (IF) models:

- n_estimators: [100, 200, 400, 500]
- contamination: [0.05, 0.1, 0.2, 0.4]
- max_features: [0.1, 0.2, 0.4, 0.5]

The model with the highest silhouette score on the test set is selected. The chosen model is then used to calculate the Outlier Score on the main paper. This process is performed independently for each dataset.

S1.4. Local Outlier Factor

We conduct a grid search over the following parameters to optimize the Local Outlier Factor (LOF) models:

- n_neighbors: [1, 5, 10, 20, 50].
- contamination: [0.05, 0.1, 0.2, 0.4].
- p: [1, 2].

The model with the highest silhouette score on the test set is selected. The chosen model is then used to calculate the Outlier Score, on the main paper. This process is performed independently for each dataset.

S2. Sparsity and Contiguity results

As discussed on the main paper, sparsity and contiguity must be jointly evaluated. Sparsity minimizes the number of changes applied to the original instance, while contiguity ensures that these changes are coherent. Several methods impose constraints on the structure of changes. For example, NG and COMTE restrict modifications to a single subsequence per channel, resulting in counterfactuals with a low number of subsequences. This is evident in Table S3, where these methods achieve the lowest Number of Subsequences scores. Glacier also produces counterfactuals with a single subsequence in the univariate setting, although this outcome is due to altering every point of the original instance, leading to the worst sparsity scores overall. In the multivariate setting, Multi-SpaCE achieves the second lowest rank, while on the univariate setting, it achieves the third lowest rank.

Table S3: NoS results.

(a) NoS for Multivariate Datasets

Dataset	COMTE	AB-CF	DiscoX	Multi-SpaCE
AWR	**2.97**	10.6	52.78	6.7
BasicMotions	**3.0**	24.11	15.33	4.55
Cricket	**3.0**	6.88	-	50.86
Epilepsy	**4.47**	7.87	21.95	4.68
NATOPS	**3.24**	47.28	33.18	10.65
PEMS-SF	**86.81**	1241.13	-	1187.22
PenDigits	2.87	3.13	**1.31**	1.69
RacketSports	**2.78**	8.23	10.88	2.86
SR-SCP1	**3.93**	10.69	354.09	19.3
UWave	**2.58**	6.01	30.5	3.36
Average Rank	**1.3**	3.9	4.25	2.5

(b) NoS for Univariate Datasets

Dataset	NG	Glacier	Glacier(AE)	AB-CF	DiscoX	Multi-SpaCE
Coffee	**1.0**	**1.0**	**1.0**	1.82	10.93	2.86
ECG200	**1.0**	**1.0**	**1.0**	1.99	5.71	1.38
FordA	**1.0**	**1.0**	**1.0**	2.39	10.85	4.14
Gunpoint	**1.0**	**1.0**	**1.0**	1.91	4.95	1.48
HandOutlines	**1.0**	**1.0**	**1.0**	2.37	90.98	2.5
ItalyPower	**1.0**	**1.0**	**1.0**	2.67	1.79	1.24
PPOC	**1.0**	**1.0**	**1.0**	1.78	5.26	1.39
Strawberry	**1.0**	**1.0**	**1.0**	1.82	10.81	1.84
CBF	**1.0**	-	-	2.24	5.75	1.78
CinCECGTorso	**1.0**	-	-	2.26	25.7	3.34
TwoPatterns	**1.0**	-	-	2.15	5.12	1.39
ECG5000	**1.0**	-	-	1.57	5.41	1.19
Plane	**1.0**	-	-	1.8	5.74	1.89
FacesUCR	**1.0**	-	-	2.9	3.96	2.2
NI-ECG2	**1.0**	-	-	1.99	13.39	3.39
Average Rank	**1.0**	**1.0**	**1.0**	4.2	6.0	3.8

Imposing restrictions on how to change the original instance to form the counterfactual can result in requiring more alterations to achieve the desired outcome, negatively affecting sparsity. Consequently, NG and COMTE perform worse than Multi-SpaCE in terms of sparsity, as shown in Table S4. Multi-SpaCE consistently achieves the best sparsity performance across univariate datasets and has the lowest rank for multivariate datasets.

This relation between contiguity and sparsity motivated the need for their joint evaluation. Multi-SpaCE demonstrates superiority over other methods in the literature when both metrics are considered together.

Table S4: Sparsity for Multivariate Datasets.

(a) Multivariate Datasets

Dataset	COMTE	AB-CF	DiscoX	Multi-SpaCE
AWR	0.33	0.9	**0.11**	0.17
BasicMotions	0.5	0.56	**0.12**	0.19
Cricket	0.5	0.97	-	**0.31**
Epilepsy	0.8	0.77	**0.27**	0.29
NATOPS	0.14	0.73	**0.04**	0.11
PEMS-SF	**0.06**	0.49	-	0.15
PenDigits	0.82	0.72	0.35	**0.27**
RacketSports	0.46	0.93	0.13	**0.12**
SR-SCP1	0.33	0.87	0.14	**0.06**
UWave	0.86	0.53	0.27	**0.16**
Average Rank	3.9	4.5	2.0	**1.5**

(b) Univariate Datasets

Dataset	NG	Glacier	Glacier(AE)	AB-CF	DiscoX	Multi-SpaCE
Coffee	0.22	1.0	1.0	0.51	0.3	**0.09**
ECG200	0.34	1.0	1.0	0.48	0.46	**0.11**
FordA	0.43	1.0	1.0	0.64	0.58	**0.09**
Gunpoint	0.23	1.0	1.0	0.56	0.37	**0.14**
HandOutlines	0.31	1.0	1.0	0.48	0.39	**0.01**
ItalyPower	0.71	1.0	1.0	0.55	0.51	**0.25**
PPOC	0.45	1.0	1.0	0.53	0.35	**0.06**
Strawberry	0.41	1.0	1.0	0.55	0.42	**0.06**
CBF	0.31	-	-	0.44	0.36	**0.2**
CinCECGTorso	0.27	-	-	0.5	0.37	**0.17**
TwoPatterns	0.29	-	-	0.59	0.3	**0.11**
ECG5000	0.26	-	-	0.81	0.52	**0.19**
Plane	0.42	-	-	0.57	0.55	**0.22**
FacesUCR	0.37	-	-	0.57	0.61	**0.19**
NI-ECG2	0.33	-	-	0.52	0.54	**0.09**
Average Rank	3.2	6.0	6.0	4.8	4.0	**1.07**

S3. Execution time

We analyze the execution times of all the methods considered in the experiments. We rely on parallelization by dividing the 100 instances to be explained for each dataset into 20 chunks of 5 instances. Counterfactuals are then generated using 10 parallel processes for each method. Execution times are evaluated with respect to both the length of the time series and the number of input channels. To examine trends, we fit an ordinary least squares regression model, observing the relationship between input characteristics and execution times. Given the variability in dataset input dimensions, we use logarithmic scales for both the x- and y-axes. Results are presented in Figure S1 and Figure S2.

Multi-SpaCE generally falls in the middle of the spectrum, being faster than DiscoX, COMTE, and Glacier, but slower than NG and AB-CF. The results reveal that, in general, execution times are more sensitive to increases in time series length than to the number of input channels.

Due to multiprocessing, we observed typical overheads, including bottlenecks caused by GPU sharing across processes, which resulted in imperfect parallel executions. Consequently, the reported execution times per method may not exactly reflect those expected in a standard environment without parallelization. To evaluate this effect, we repeated the experiments on multivariate datasets without parallelization. The results of this experiment are reported in Table S5.

Table S5: Average execution time (in seconds) when using normal and parallel experimentation.

Dataset	Multi-SpaCE(parallel)	Multi-SpaCE
AWR	57.27	26.67
BasicMotions	34.45	21.86
Cricket	220.96	94.67
Epilepsy	47.11	21.98
NATOPS	35.81	18.79
PEMS-SF	408.85	178.45
PenDigits	31.82	16.93
RacketSports	32.23	17.67
SR-SCP1	104.44	42.41
UWave	66.32	25.61

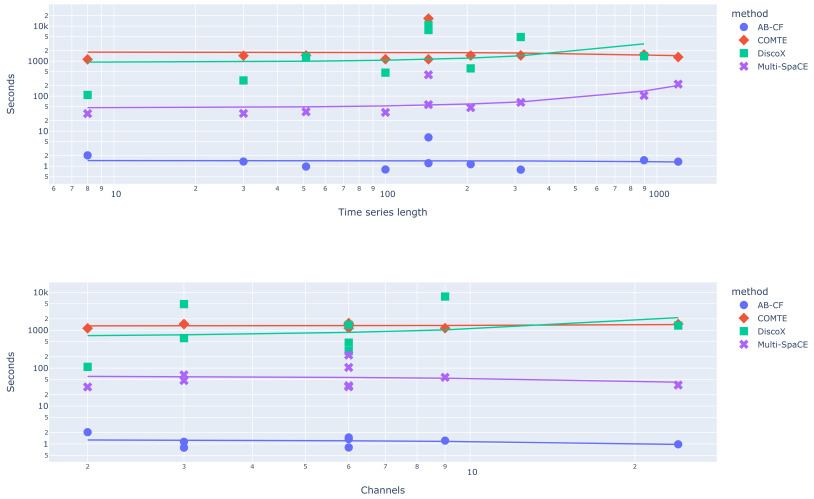

Figure S1: Execution times depending on the length and the number of channels in multivariate datasets.

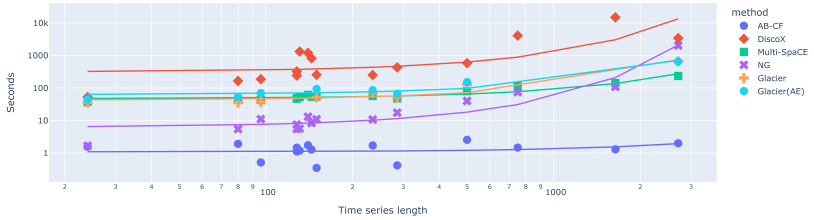

Figure S2: Execution times depending on the length in univariate datasets.