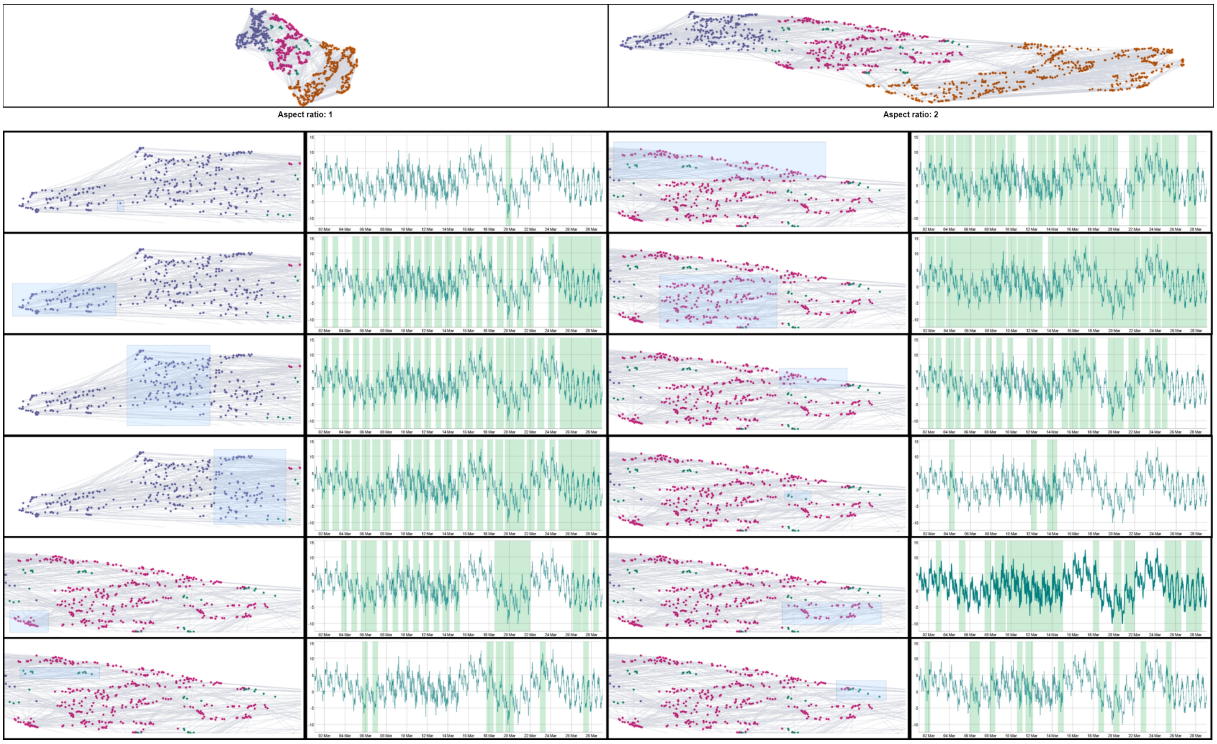

Figure 4.20: Global view of the embeddings of the zero-shot and the fine-tuned large models for S1.

Analysis of S2 using the six versions of MOMENT

The S2 dataset contains two point anomalies. In the figure 4.21 both of them are detected in the MOMENT-small's embedding space, but the interrelations between clusters make it difficult to detect them in a single view, losing interpretability.

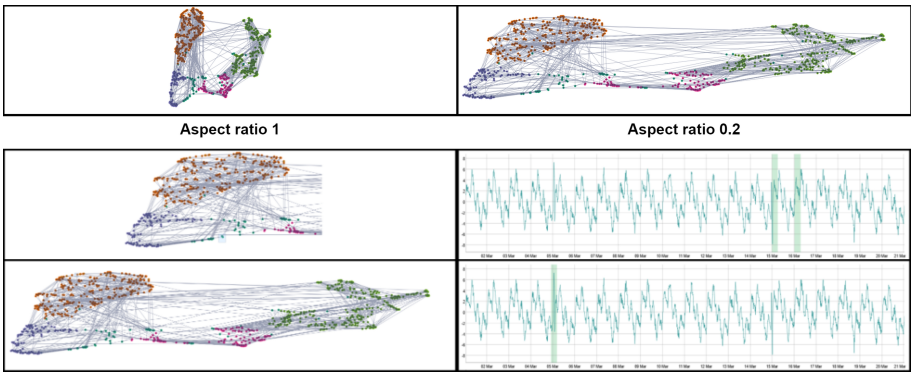

Figure 4.21: Global view of the embeddings of the zero-shot small model for S2.

When comparing the left and right sections of the TS, anomalies appear in other regions of the embedding space. These points are not connected temporally, but rather joined by short-term shapes. It seems that temporal correlation is being lost and each window is treated more like a word in a sentence, making it difficult to detect large parts, so anomalies are easy to check (see Fig. A.14). In this part, both anomalies are found. In addition, the right part of the TS, reveals more patterns and the presence of both anomalies of the TS (see Fig. A.15).

After fine tuning MOMENT-small, the embedding space does not change so much, with no improvement of its interpretability (See Fig. 4.22).

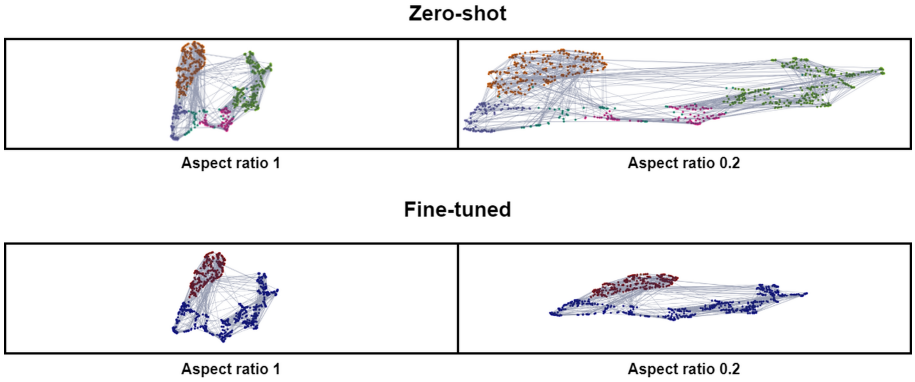

Figure 4.22: MOMENT-small fine-tuned and zero-shot embeddings for S2.

Figure A.9 shows the MOMENT-base model results on more defined clusters. The difference between the zero-shot and the fine-tuned models is still not that much, as the only change is the rotation of positions between the brown and green clusters in the right part. Again, MOMENT seems to correctly detect small patterns but it makes difficult to check the anomalies as they are not in the apart-zones but, at least, the anomaly by the right is at a corner of the pink cluster at an edge point (very similar "input-output" directions, like if they were the same line). This makes it easier to detect the anomalies, but not as easy as with the embeddings of MTSAE.

The embedding space of the MOMENT-large model is similar to the previous ones but seems to be more accurate in the right part with the new purple cluster (Fig. A.10). Focusing on the pink cluster, both anomalies can be detected in its upper corner, which is a great advance to directly check the anomalies (Fig. A.16). Figures A.17, A.18, A.19 and A.20 show the insider view of the clusters. It is interesting that again the anomalies appear in different clusters at the same time: pink and yellow. The fine-tuned version has a very similar shape. In this case, the anomalies are in the middle of the pink cluster A.21, difficult to detect.

Thus, in summary, anomalies are not that easy to detect. Table 4.7 shows an overview of the analysis for each version of the dataset.

Analysis of S3 using the six versions of MOMENT

This dataset is used to check the interpretation of trends through the visual projection of the embedding space. The zero-shot and fine-tuned versions of MOMENT-small have very similar projections, with small rotations. Taking a loop from left to right to the different clusters, no trend is detected, showing a spring effect (see Figs. 4.23, A.23).

The base projections are very similar to the small ones in both the zero-shot and the fine-tuned versions. Although it seems to be getting the trend when selecting the two clusters as a full object, getting inside the blue part and going through the lines, the spring effect re, showing that the trend is not related to the position of the points in the embedding space (see Figs. A.22, A.23).

The large version also shows really few differences between the zero-shot and fine-tuned embedding projections. This time, the trend seems to be detected, but it still has a spring effect in the blue cluster (see A.24).

Table 4.7: Comparison of zero-shot vs. fine-tune versuibs for S2 (Anomaly detection) using Moment-Small, Moment-Base, and Moment-Large.

Model	Training Type	Advantages	Disadvantages
MOMENT-Small	Zero-shot	✓ Clearly defined clusters. ✓ Captures general patterns in S2.	✗ Clusters are highly intertwined. ✗ Anomalies detected but hard to isolate.
	Fine-tuned	✓ No significant improvement.	✗ Anomalies remain hard to isolate. ✗ Cluster structure unchanged.
MOMENT-Small	Zero-shot	✓ One anomaly detected at a cluster edge.	✗ The anomalies appear in different clusters at the same time.
	Fine-tuned	✓ Minor improvement in anomaly detection.	✗ Clusters rotated but similar (don't add/summarize clear information).
MOMENT-Large	Zero-shot	✓ Most defined clusters. ✓ Anomalies detected at cluster edges.	✗ Anomalies still appear in multiple clusters.
	Fine-tuned	No significant improvement.	✗ Cluster structure remains nearly identical. ✗ The anomalies move to the midle of a cluster, making them more difficult to detect in a visual inspection.

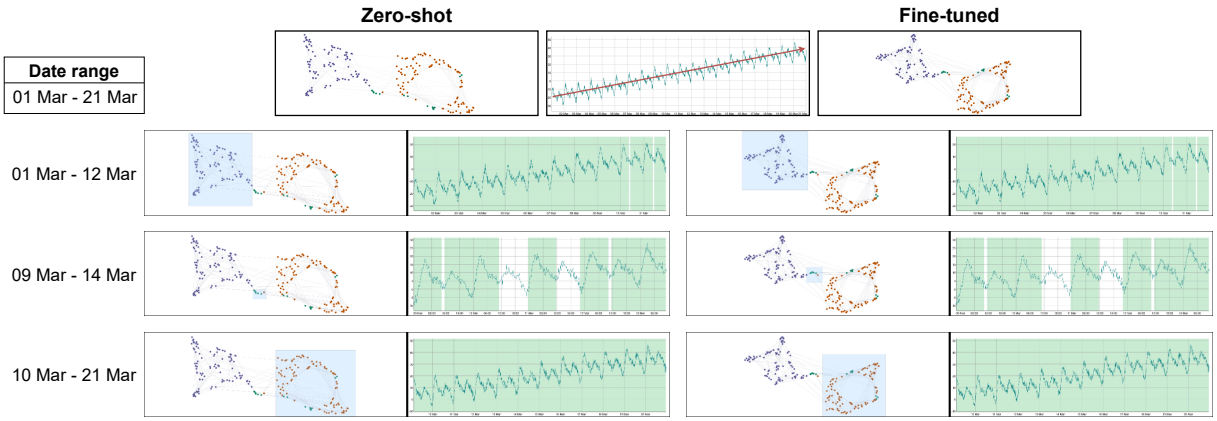

Figure 4.23: Embeddings projections of MOMENT-small applied to S3.

Analysis of Kohl's using the six versions of MOMENT

The Kohl's TS is used to analyze both segmentation and trend. The model is supposed to fit both the upper trend and the segmentation between peaks and plains.

The MOMENT-small-zero-shot versions show two distinct clusters. However, the green one (in the bottom) is near to detect all the peaks (slightly displaced to the right) and the first plain. Those peaks seem possible to reconstruct by using the other two small green clusters. However, when the rest of the embedding space is selected, the segmentation is observed to not to be that clear, since the blue part contains the full TS. In addition, the "linear" pattern of the trend within the embedding space is not detected. The fine-tuned version has more defined clusters. The 'plains' are near to be learned in the green part. However, it still has parts of the peaks and the first plain is shared among all the clusters. It is slightly easier to detect, but there is no clear trend or segmentation pattern in the plot (see Fig. 4.24).

Table 4.8: Comparison of zero-shot vs. Fine-tune for S3 (Trends) using Moment-Small, Moment-Base, and Moment-Large.

Model	Training Type	Advantages	Disadvantages
MOMENT-Small	Zero-shot	✓ Defined but interconnected clusters.	✗ No clear trend detected, with an spring effect.
	Fine-tuned	No significant changes.	✗ Spring effect remains in cluster visualization.
MOMENT-Base	Zero-shot	✓ Some structures suggest trend detection.	✗ Still no fully defined trends.
	Fine-tuned	✓ Slightly clearer structure.	✗ Trend still mixed within clusters.
MOMENT-Large	Zero-shot	✓ Best model for detecting trend-like structures.	✗ Spring effect still present.
	Fine-tuned	No significant improvement.	No significant differences with the zero-shot version.

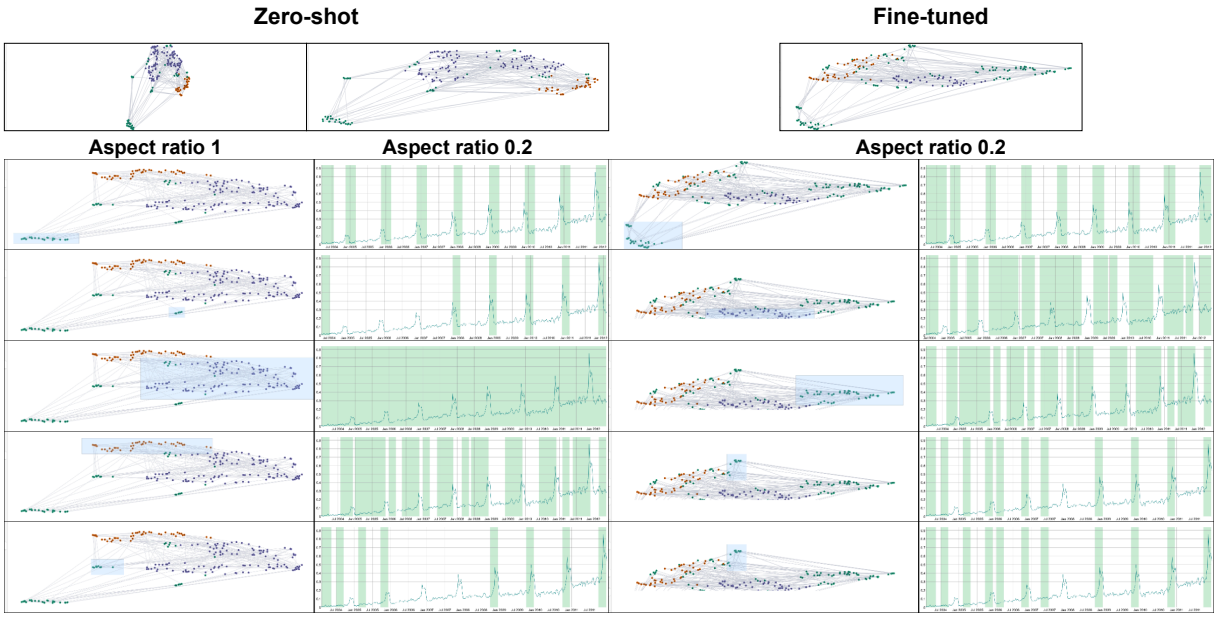

Figure 4.24: Embeddings projections of MOMENT-small applied to Kohls.

The MOMENT-base model seems to be clearly detecting the peaks not that well the plains, correctly detecting them as a pattern more than a segmentation. In fact, the blue clusters get the full TS again instead of just the plain parts. Again, no trend visual pattern is found A.25. For the MOMENT-large version, see that the zero-shot and the fine-tuned models are really similar, with a rotation in the right part (and a subtle more definition). This time, the fine-tune and the zero-shot models are more different (see Figs. A.26, A.27). The zero-shot model shows a more clear segmentation between peaks and plains (except for the first plain), showing a great performance at detecting the plains while detecting both patterns and the intermediates. The fine-tuned version seemed to be distinct, but the results are really similar.

Analysis of M-Toy using the six versions of MOMENT

The M-Toy dataset is used for analyzing sequence anomalies in multivariate timeseries. The small versions can detect the two anomalies in the data set. In both the fine-tuned and the zero-shot

Table 4.9: Comparison of zero-shot vs. Fine-tune for Kohl's using Moment-Small, Moment-Base, and Moment-Large.

Model	Training Type	Advantages	Disadvantages
MOMENT-Small	Zero-shot	✓ Clear separation of peaks.	✗ Trend not fully detected. ✗ First plain is mixed within clusters. ✗ Not clear separation of the plains.
	Fine-tuned	✓ More defined clusters. ✓ Plains are closer to be detected.	✗ First plain shared between all the clusters. ✗ Plain section still containing peaks. ✗ Still no clear trend detection.
MOMENT-Base	Zero-shot	✓ Peaks are well defined.	✗ Plains are not clearly detected.
	Fine-tuned	✓ Slight improvement in segmentation.	✗ No clear improvement in trend detection.
MOMENT-Large	Zero-shot	✓ Best model for segmentation. ✓ Peaks and plains clusters are visually distinct.	✗ Trend is still not fully learned.
	Fine-tuned	No major improvement.	Trend detection remains similar to the zero-shot version.

versions the anomaly by the left is really easy to check as and edge point but the right anomaly is detected in the middle of a cluster, making it more likely a transition point and thus difficult to detect (see Fig. 4.25).

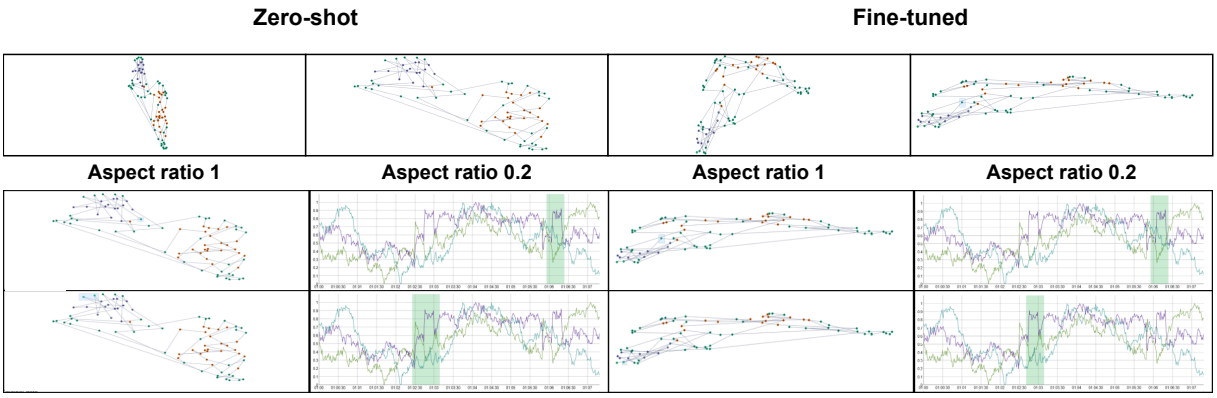

Figure 4.25: Embeddings projections of MOMENT-small applied to `M-Toy`.

The base zero-shot version is more difficult to analyze. However, when applying the fine tune, the anomalies become easier to detect and become really close to them, showing a relation between both patterns (see Fig A.28).

The large zero-shot version embedding space is more difficult to read (see Figs. A.29, A.30), especially for the anomaly on the left, as it gets mixed with intermediate patterns (which may also be interesting). The large-fine-tuned version results in a similar plot with a better interpretability of the anomaly on the right. However, the anomaly on the left is even more difficult to detect within the embedding space.

Table 4.10: Comparison of zero-shot vs. fine-tuned versions for `M-Toy` using Moment-Small, Moment-Base, and Moment-Large.

Model	Training Type	Advantages	Disadvantages
MOMENT-Small	Zero-shot	✓ Detects the left anomaly as an edge point.	✗ Right anomaly is mixed within clusters.
	Fine-tuned	✓ Left anomaly remains easy to detect.	✗ No major improvement in right anomaly visibility.
MOMENT-Base	Zero-shot	✓ Anomalies are more structured than in Small.	✗ Right anomaly is still not well-separated.
	Fine-tuned	✓ Right anomaly becomes slightly more distinguishable.	✗ Left anomaly detection remains unchanged.
MOMENT-Large	Zero-shot	✓ Best structured clusters. ✓ Left anomaly still visible.	✗ Right anomaly is harder to isolate.
	Fine-tuned	No significant improvement.	✗ Right anomaly remains difficult to distinguish.

Table 4.11: Summary of model performance across different tasks (anomaly detection, pattern detection, segmentation, trend detection).

Model	Anomaly Detection	Pattern Detection	Segmentation	Trend Detection
MOMENT-Small	✗ Anomalies detected but difficult to isolate	✓ Some patterns detected.	✗ Weak segmentation.	✗ No visible trend detection.
MOMENT-Base	✓ Minor improvements in detecting anomalies at edges.	✓ Patterns are clearer than using Small.	✗ Some, but incomplete.	✗ Trend detection is suggested but unclear.
MOMENT-Large	✗ Anomalies more difficult to detect than with base.	✓ Most structured pattern detection.	✗ No clear segments	✗ Still lacks clear trend detection.

4.2.6 Are Time Series Foundation models useful for visual embedding projections

Table 4.11 shows how MOMENT's embeddings are not that easy to check visually in order to detetect the timeseries charasteristics, thus giving a negative answer to **RQ2.1**. The best performance is obtained for short pattern detection. Also, the clusters are really interconnected, showing that the time correlation may not be correctly detected.

Focusing on the research questions **RQ2.2** and **RQ2.3**, the loss performance of the fine-tuned version is directly related to the percentage of the dataset used. However, after selecting the best cases within the statistical analysis, the model does not show great differences within the MTSAE inspection within the UMAP followed by PCA projections version. Thus, it seems that the improvement in terms of loss is definitely not related to the fine-tuning level (in fine-tuning percentages), but this opens the question if MOMENT is able to detect the same patterns than MTSAE if trained with the full training dataset. However, this is against the idea of having a faster interaction due to the less need of training (still, it remains to be evaluated wether a full training of MOMENT in any of its versions is faster than MTSAE or not). Thinking about whether the introduction of TSFM into DeepVATS is worthit or not, it seems that the answer is "not yet", or maybe "not this way" as the obtained projections plot is not even close to being interpretable.

4.2.7 Future lines to enhance MOMENT-DeepVATS

The integration of MOMENT into DeepVATS has not yet demonstrated the anticipated level of efficacy. Although it is an improvement in execution time, it does not reach the expected capabilities in capturing data behaviors within the embedding space when compared to MTSAE. Despite these limitations, the analysis has led to the identification of four promising directions for further analysis. These future lines aim to improve the interpretability of the embeddings produced by the fine-tuned version of MOMENT through various methodological adjustments, each addressing different aspects of potential optimization within the current framework. This are the future lines:

- **FL3. Modify the loss distance function.** The first line is related to the distance used to calculate the loss in the optimization phase. Selecting other losses such as soft-DTW loss [6], which can lead to a better fit to the TS at shape. This option may not result in much difference anyway as the MSE distance should be enough for a fine-tuning but the question to solve behind is "is the distance in the fine tune related to the visual performance within a specific task?".

- **FL4. Using other projection techniques.** The second line is related to the use of other projection techniques, as the original authors showed time correlation and trend detection within the de PCA and t-SNE projections, resulting in different patterns than MTSAE-UMAP. Thus, as a future line, the PCA (without UMAP) and t-SNE projections will be tested within the datasets to check if the selection of the projection technique is directly related to the results obtained.

- **FL5. Data preprocessing.** The third line is related to the following question that arises: Is is true that TSFM can detect everything without preprocessing the data? As it is really cheap in time and computation, the next step is to add in the visualization app the option to fine-tune the model with a preprocessed version of the dataset using classical preprocessing techniques and get the embeddings for the original dataset afterward. This option should result in clearer projection plots as the task is simplified but would add a preprocessing time, depending on the size of the TS.

- **FL6. Freezing layers.** The fourth line relates to the precision of the fine-tuning process. As a transformer encoder, MOMENT consists of multiple layers. Then, it becomes possible can analyze which layer is the most affected by the fine-tuning and how much is the embedding space modified. Also, the impact of freezing specific layers (especially the output layers that depend more on the task than on the learned insights, as in the recommended fine-tuning settings of Goswami et al. [1]) on better visual precision in detriment of the loss improvement by reinforcing the variation of the rest of the model layer's weights.

This Future Line (FL), whose content and relation with the research questions are summarized in Figure 4.26, can be summarized in a final third hypothesis: "the PP of TSFM can enhanced through the application to preprocessed thata, the use of task-specific projection techniques or the modification of the fine-tuning process. This enhancement leads to better interactive VA tools for large TS".

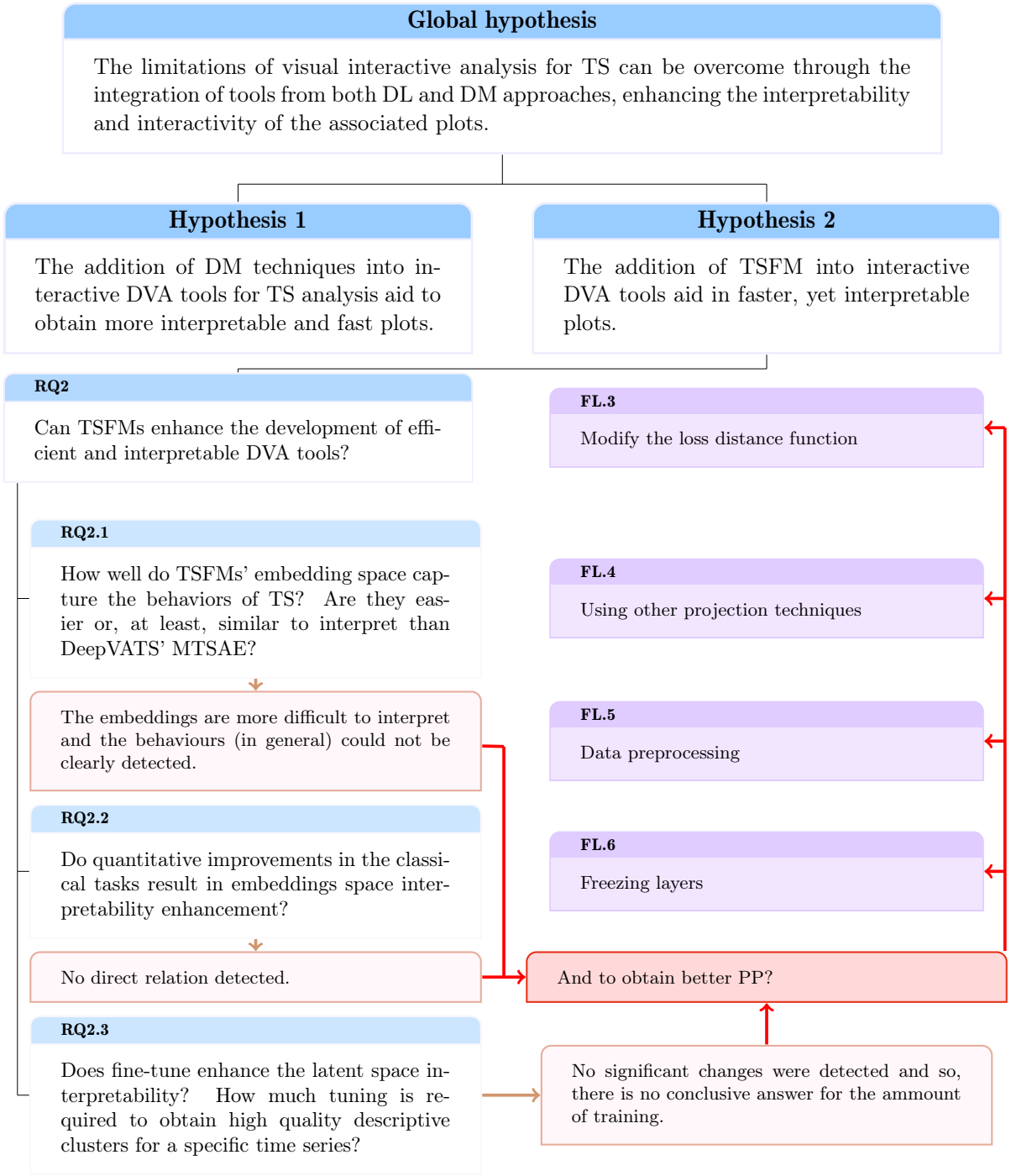

Figure 4.26: Future lines II: schema showing the relation between the hypothesis, the research questions of the DL path, their answers and the associated future lines.

4.2.8 Analysis of MOMENT with soft-DTW distance as loss for Kohl's dataset

The comparison of two TSs is directly related to a similarity function, and this can be accomplished using various distance measures. Traditional metrics like the Euclidean distance assume that discrete TSs consist of evenly spaced points in time and are perfectly aligned along the time axis. However, in certain fields, although TSs may share similarities in amplitude and shape, they can be misaligned over time. Consequently, similar patterns might appear at different times, resulting in varying degrees of temporal distortion, or time warping, across different sequences, due to their lack of temporal alignment.

Although most of algorithms rely on the Euclidean distance, conventional distances prove inadequate as they are highly sensitive to minor temporal distortions and generally cannot handle TSs of unequal lengths without some preprocessing. DTW is a widely used technique to measure the similarity between TSs by finding an optimal alignment between them. Unlike the Euclidean distance, DTW can handle sequences of different lengths and is robust to shifts or dilatations across the time dimension, enabling the identification of shifted patterns in-between TSs (see Fig. 4.27). DTW has demonstrated superiority in most fields, especially for tasks like: clustering, classification, and similarity search [320, 321].

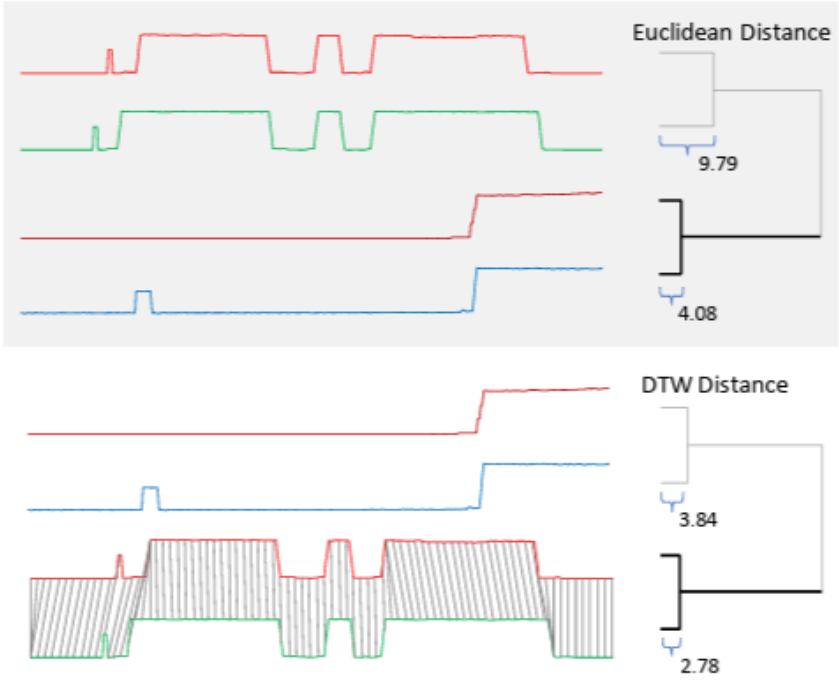

Figure 4.27: Euclidean and DTW distance for two pairs of sequences, showing how DTW can detect shifted TSs as similar. Figures obtained from [15].

However, the classical DTW relies on a nondifferentiable minimum operator, which makes it unsuitable for direct integration into gradient-based optimization frameworks. This change allows us to test if interchanging the loss function (currently MSE) in MOMENT enhances its ability to detect similar patterns, relying on more explainable embedding space.

This section introduces DTW and its differentiable version, as well as an analysis of MOMENT using dtw as a first step in analyzing the line **FL3**.

Dynamic Time Warping

DTW is a seminal TSs comparison technique that has been used for speech and word recognition since the 1970s [108]. This technique is especially useful when sequences vary in speed or duration (different lengths or different positions of the patterns). Unlike the euclidean distance, which compares the sequences point by point and requires them to have the same length, DTW introduces flexibility through a process called "warping".

Warping refers to the alignment of points between two TSs by dynamically adjusting (stretching or compressing) their temporal axes. This allows DTW to effectively handle cases where similar patterns occur at different speeds of drifted positions, providing a more robust and meaningful measure of similarity.

Formally, let X^n, Y^m be two multivariate TSs taking values in $\Omega \subset \mathbb{R}^p$.. For this series, $\mathcal{A}_{n,m} \subset \{0,1\}^{n \times m}$ can be considered for the set of (binary) **alignment matrices**; that is, paths on a $n \times m$ matrix that connectw the upper-left $(1,1)$ matrix entry to the lower-right $(m-m)$ using only $\downarrow, \rightarrow, \searrow$ moves. These matrices represent ways of linking the points of the TSs, allowing to "align the timestamps" so that patterns are detected when shifted or streched. Our goal is to define a cost for each of its routes so that it can be selected as the best path, being its cost te DTW distance value.

To do so, the **subsitution-cost** function can be defined as a differentiable function $\delta : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^+$ which will be, in most cases, the quadratic Euclidean distance between two vectors. Its matrix representation $\Delta(X,Y) := (\delta(x_i, y_j)) \in \mathbb{R}^{(n \times m)}$ is called the **cost matrix**.

Now, the inner product $< A, \delta(x,y) >$ of that matrix with an alignment matrix $A \in \mathcal{A}_{n,m}$ gives the **score** of A, being

$$DTW(X,Y) := \min_{A \in \mathcal{A}_{n,m}} < A, \Delta(x,y) > .$$

For example, let $t = \{1,2,3,4,5,6,7\}$, X^7 be the TS withiin t taking values $\{1,3,4,8,8,2,1\}$ and Y^7 the series taking values $\{2,2,4,7,5,2,0\}$. Then, its cost matrix, using the euclidean distance as substitution cost function results in

$$\Delta = \begin{bmatrix} 1 & 1 & 9 & 36 & 16 & 1 & 1 \\ 1 & 1 & 1 & 16 & 4 & 1 & 9 \\ 4 & 4 & 0 & 9 & 1 & 4 & 16 \\ 49 & 49 & 25 & 4 & 16 & 49 & 81 \\ 36 & 36 & 16 & 1 & 9 & 36 & 64 \\ 0 & 0 & 4 & 25 & 9 & 0 & 4 \\ 1 & 1 & 9 & 36 & 16 & 1 & 1 \end{bmatrix}$$

The set of alignment matrices will be made up of many different paths. Consider these three examples of alignment matrices (paths), which can be visually observed in Fig. 4.28:

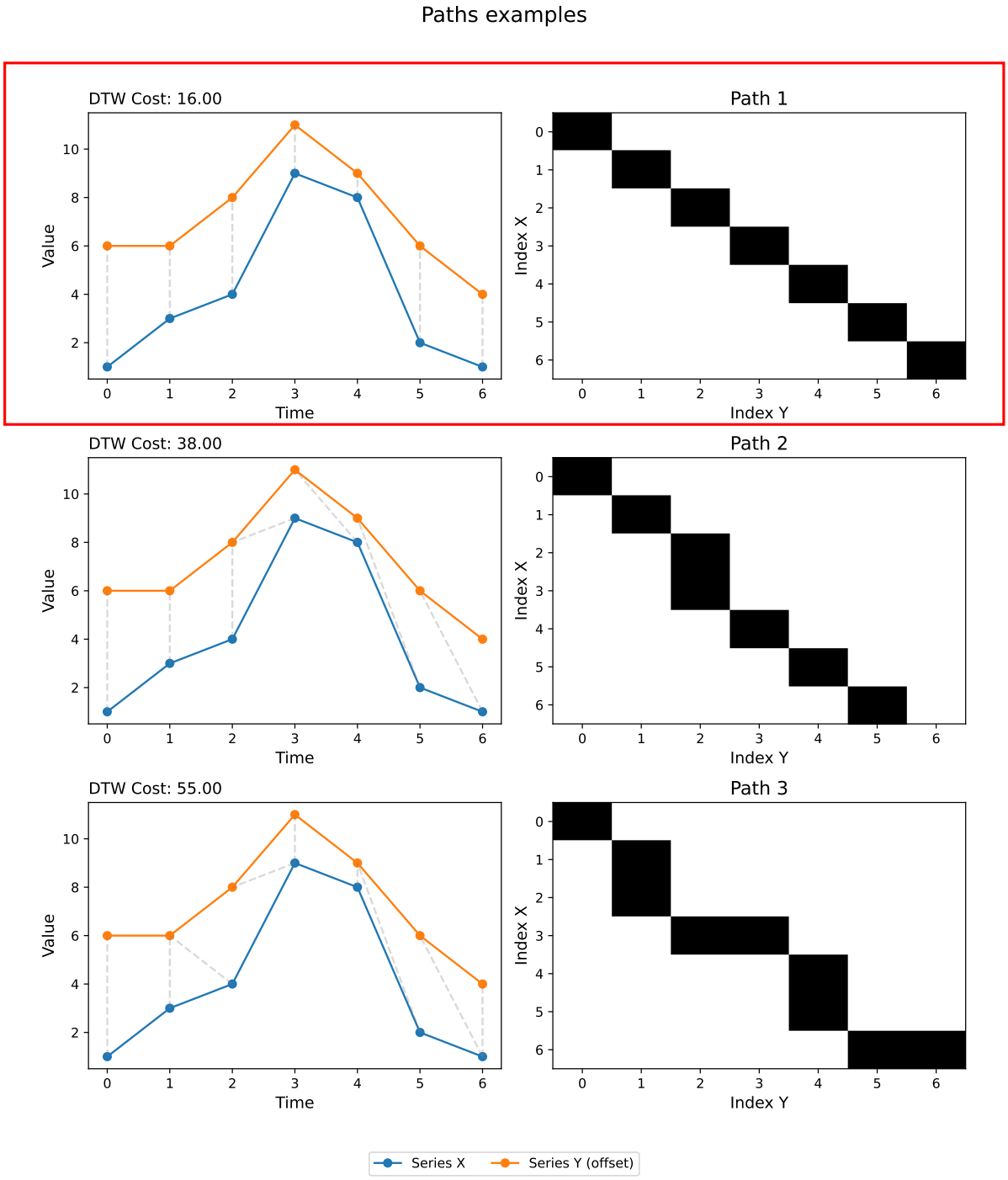

Figure 4.28: DTW and Euclidean distance calculation between two TSs.

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The scores (inner products of these paths with the cost matrix) are:

$$\text{Score}(P_1) =< P_1, \Delta >= 1+1+0+4+9+0+1 = 16,$$

$$\text{Score}(P_2) = 1+1+0+25+1+9+0 = 37,$$

$$\text{Score}(P_3) = 1+1+4+(25+4)+9+9+(1+1) = 55.$$

Thus, P_1 is the best path, followed by P_2 and finally, P_3. Getting the minimum value between all possible paths, the DTW distance, returns its cost. In this case, the best path results to be P_1, being the equivalent to directly apply the Euclidean distance as it supposes no time warping. Thus, the distance between X and Y was $dtw(X,Y) = 16$

One of the main advantages of this measure is that it is applicable to different sized TSs. Fig. 4.29 shows an example. In this case, the best path did not result in the identity matrix, thus resulting in a distance different from the Euclidean one. In this case, the first three points are "joined with" to fix the blue series to the orange one.

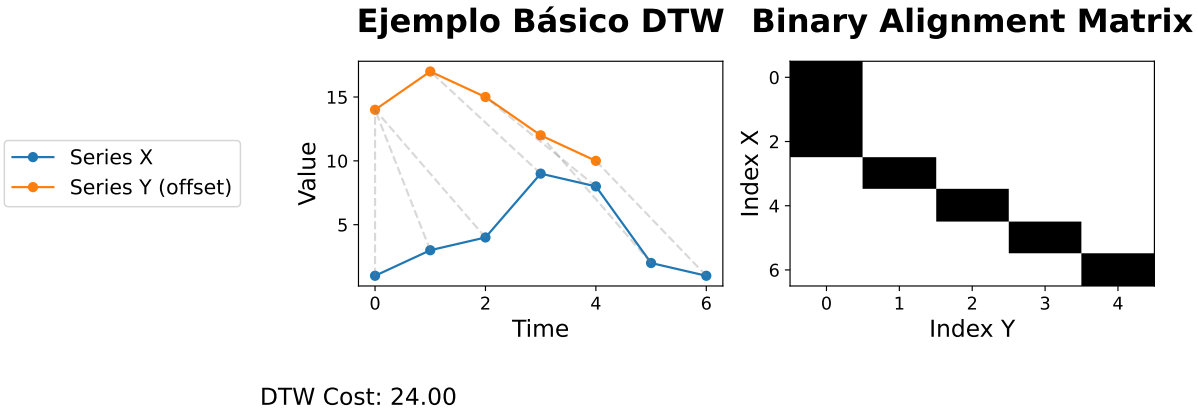

Figure 4.29: DTW and Euclidean distance calculation between two TSs.

Soft-DTW Formulation

However, in order to use a similarity measure as loss function, it must be differentiable and the minimum function is not. Soft-DTW is a smooth variant of DTW that overcomes this limitation by replacing the hard minimum operator with a soft-minimum function [6]. The soft-minimum function is defined as:

$$\min^{\gamma}(a_1, a_2, \ldots, a_n) = \begin{cases} \min\{a_i\}_{i=1}^{n}, & \text{if } \gamma = 0, \\ -\gamma \log\left(\sum_{i=1}^{n} \exp\left(-\frac{a_i}{\gamma}\right)\right), & \text{if } \gamma > 0. \end{cases}$$

Thus, soft-DTW is defined as:

$$dtw^{\gamma}(X,Y) = \min_{A \in \mathcal{A}_{n,m}} < A, \Delta(X,Y) >$$

In this way, the soft-minimum introduces a smoothing parameter $\gamma > 0$. For small values of

γ, the soft-minimum approximates the classical minimum, while larger values provide a more averaged cost. This smoothing makes the measure function differentiable and so enables its use as a loss function in DL models. Figure 4.30 shows an example of how the predictions of a multilayer perceptron fit the expected values better when using soft-DTW rather than the mean squared error.

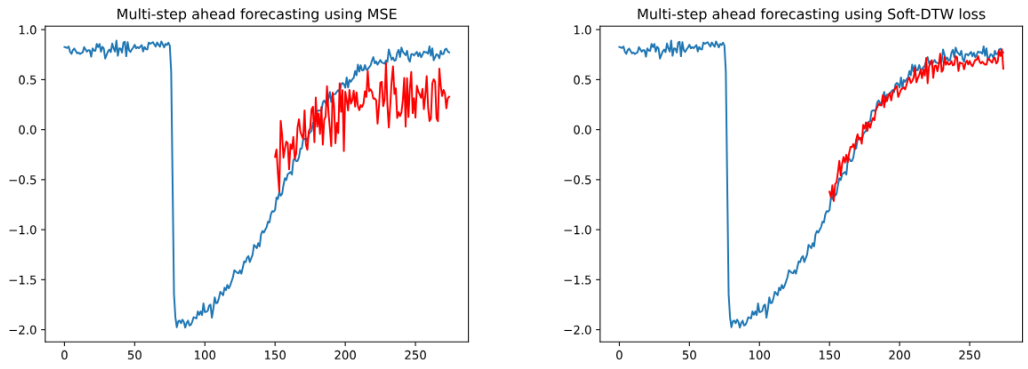

Figure 4.30: Examples of predictions of a multilayer perceptron using MSE and soft-DTW as loss in tslearn package. Figures obtained from `https://short.upm.es/2182w`.

Statistical analysis of kohl's cases using dtw

To accomplish this task, as a first approximation, `tslearn`'s `SoftDTWLossPyTorch` function is used with `gamma=5.0` as smoothing parameter. The present analysis could be extended by modifying the value of gamma and checking whether the embeddins are more explainable or not.

Following the steps for the previous statistical analysis, the improvement per case in each case is really similar to the MSE version. This time, the MOMENT-small model improves up to 22%, MOMENT-base up to 10% (with really similar plots to MSE) and MOMENT-large stops at 2%, reducing the improvement to half of the case in MSE (see Figs. 4.31, 4.12).

The correlation matrices do not change that much, not giving a real idea of the parameters that could be affecting to the enhance of loss improvement. However, there is a part that really makes the difference: the best epochs are completely different from one another, resulting that MOMENT-small should be trained for 19 epochs, MOMENT-base for 18 epochs, and MOMENT-large 9 epochs (See Figs. 4.32, 4.33).

Table 4.12: Best parameter values for Small, Base, and Large models based on feature importance analysis.

Parameter	Small	Base	Large
masked_percent	25	50	50
best_epoch	19	18	9
n_windows	1	5	1
dataset_percent	30	20	20

Once confirmed, the number of windows remains irrelevant 4.34, the feature importance is done

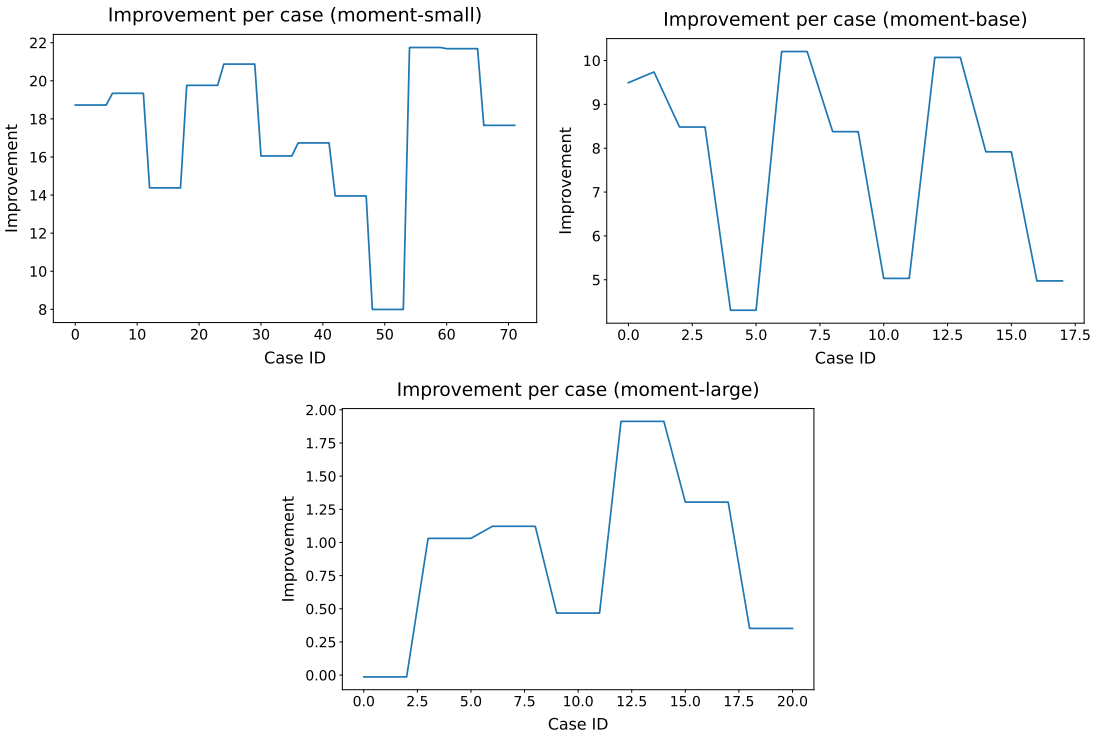

Figure 4.31: Improvements plot of the statistical analysis of MOMENT using soft-DTW distance as loss.

again, getting the parameters within 4.34. Taking into account the computational time, the best epoch frequencies, and the irrelevance of the number of windows, the final selected parameters are selected (see Tb. 4.13).

Table 4.13: Final parameter selection for MOMENT-small, MOMENT-base, and MOMENT-large models for the visual experimentation,

Parameter	Small	Base	Large
masked_percent	25	15	75
best_epoch	17	13	10
n_windows	1	5	1
dataset_percent	15	25	20

Analyzing `Kohl's` with DTW-MOMENT-small

The embedding space after using the soft-DTW function as loss has changed a lot compared to that obtained within Fig. 4.24. However, even if one is really close to have two clusters with the segmentation, it is still not perfect, as the plains zones still have the right peaks. However, peaks are really good detected within the left part of the embedding space projection.

As there is no line, but only a big cycle, the projection does not allow to check for trends in a direct way. However, it is true that the TS can be explored following the line in an ordered path, but the last part of the circle joins the initial part of the TS, thus not allowing visually checking

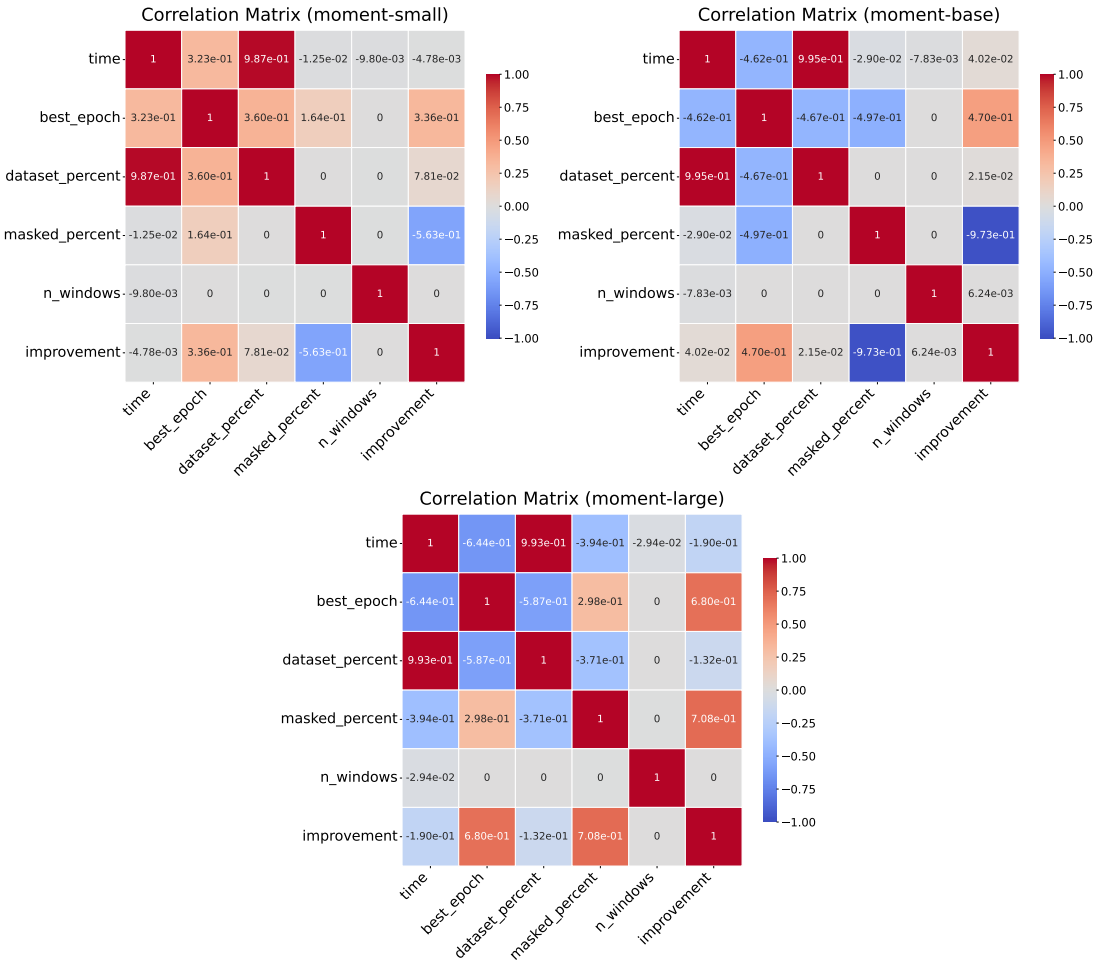

Figure 4.32: Correlations of statistical analysis of MOMENT using soft-DTW distance as loss.

a trend looking only at the embedding space but forcing us to check the TS to visually check the position to know that there is a property there (see Fig. 4.35).

4.2.9 Final analysis of MOMENT-DeepVATS and the contributions of DTW

The DTW distance introduces a temporal alignment mechanism that allows sequence comparisons under non-linear time transformation. This makes it particularly suitable for scenarios. where the same pattern may occur at different speeds or be shifted across time in different sequences. This property makes it great for segmentation, pattern detection, and trend modeling. In the context of MOMENT-DeepVATS, DTW was incorporated to enhance MOMENT sensitivity to such temporal distortions through its application as a loss function in the fine-tune phase. The goal was to check whether this modification of the loss distance resulted in a better interpretability of the PP.

As the different versions of MOMENT did not show great topological differences when applied to the same dataset, the MOMENT-small version was selected for testing because it is the faster one. The evaluation was done using the `Kohl's` dataset, which includes both pattern and trend detection. The embeddings became clearly distinct. However, it remained difficult to visually detect global TS trend or clearly detect repeated patterns.

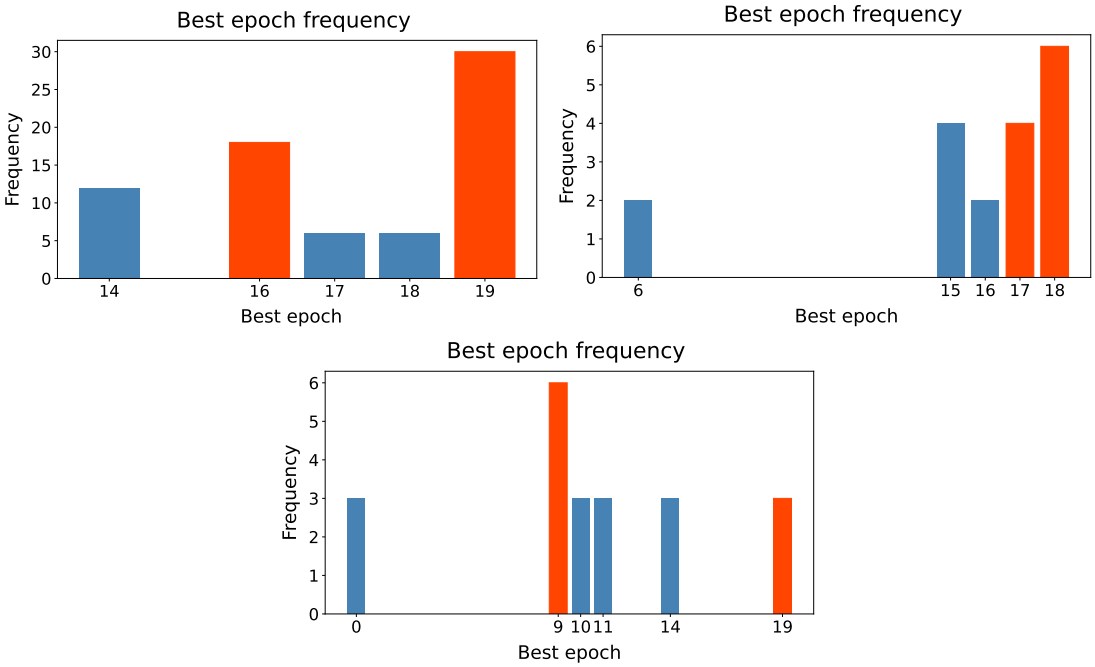

Figure 4.33: Epoch frequencies of statistical analysis of MOMENT using soft-DTW distance as loss.

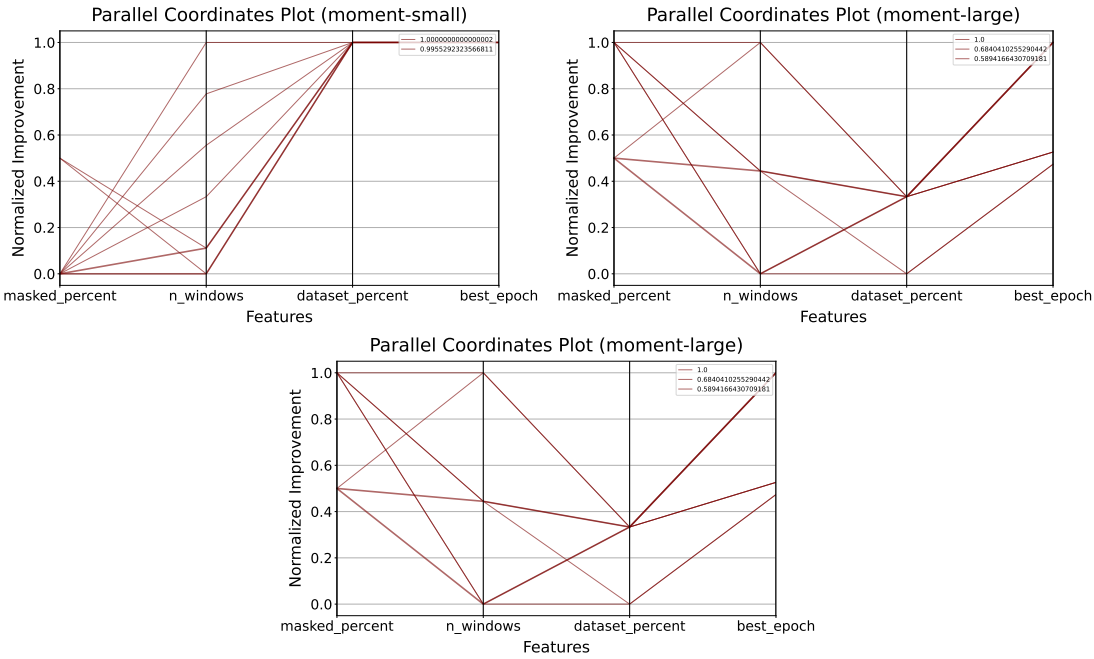

Figure 4.34: Parallel coordinates plot of the statistical analysis of MOMENT using soft-DTW distance as loss.

Thus, incorporating DTW has not shown a great difference in the interpretability of the latent space, so it may be better used in conjunction with the rest of the proposed solutions (see Sections 4.1, 4.2.1).

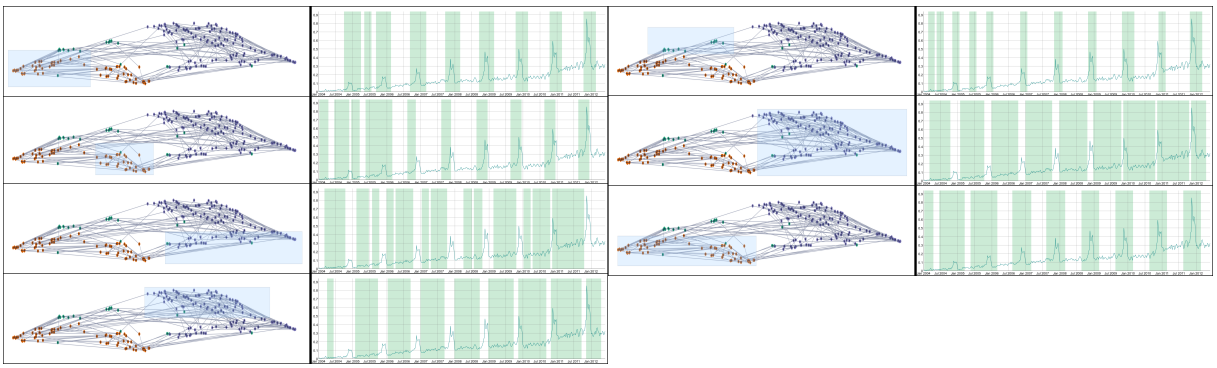

Figure 4.35: Kohl's execution for MOMENT-small fine-tuning through 19 epochs with 1 window, 30% dataset percent, 25 masked percent and soft-DTW as loss function.

Chapter 5

Discussion

¡Y p'arriba con las Mates,
admirables y sublimes!
Sin ellas el Universo
no sería inteligible

— Antonio S.

5.1 Relevant findings

The dissertation, its methodology and implementations are conducted to answer the questions in Section. 1.2. The first step resulted in a scalability analysis that showed how DeepVATS(DeepVATS) has an excellent performance for small datasets and medium datasets but had performance problems to process larger datasets (see Section. 3.4). To fix this problem, three solutions were proposed:

- The first focused on reshaping, redesigning, and improving the code. This has been accomplished with internal changes: using cached values, batching the embeddings obtainment, improving the application interactions, etc. It is not related to the main results in the rest of the dissertation.
- The second focused on the stability of the embedding projection plot. The projection plot used to have issues in its projection stability, resulting in unexpected situations when zooming in (e.g., white plots at zooming in). To fix this, as proposed, the option of executing PCA(Principal Components Analysis) followed by UMAP(Principal Components Analysis) was added, allowing a fluent and stable analysis in the rest of the sections, especially useful in Section. 4.2, where lots of plots needed to be analyzed in different zoom sizes.
- The third focused on the addition of other state-of-the-art tools. This has been the main methodology for the implementation within the thesis. Two different tools have been analyzed: the Distance Matrix Plot and the Time Series Foundation models.

The next sections show the results of following the path of the third solution as well as the

hypotheses and research questions analyzed within the thesis.

5.1.1 First research question

The first research question (**RQ1**) is "Can MPlot enhance the development of efficient and interpretable DVA tools?". The answer to this question is subdivided into two smaller ones (see Section 4.1: MPlot integration into DeepVATS).

The first question, "Are MPlot easy to interpret? Specifically, do they surpass DeepVATS' capabilities in behavior representation?" (**RQ1.1**), is related to efficiency. Table 4.1 shows how the use of the $10K$ version of MPlot can suppose a great difference in execution time, reducing from more than two hours to 7 seconds. This demonstrates that this version (which may lack some information) is faster than MTSAE.

The second question, "Are MPlot efficient enough in their $10K$ version? That is, do they improve DeepVATS' MTSAE effiency?" (**RQ1.2**), is related to the capabilities of MPlot to detect patterns and make them easy to detect in a visual inspection and to see if they improve the previous results obtained for MTSAE. To answer this question, the MPlot are added to DeepVATS and compared in both time and visual analysis against MTSAE. The results indicate that this tool is great for detecting both global and local trends but it less effective for detecting anomalies or patterns in the first view. As MTSAE has not shown good trend detection, it appears to be an easy improvement in trend visual analysis.

Together, these answers, suggest that Distance Matrix Plot can improve DeepVATS efficiency by allowing its execution in the meantime the model is still traning, thus supporting a positive answer to **RQ1** with the limitations to univariate time series and the fact that the analyst may not easily check the anomalies depending on the time series characteristics.

To solve those problems, two FL were initially proposed. The first one, "Checking the use of MP-derived features (such as the multidimensional MP) for better analysis." (**FL1**), has not yet been tested. However, the second one, "The inclusion of new TSFM to the DL module." (**FL2**) provides a pathway to the second part of the methodology and its analysis is concluded in the next section.

5.1.2 Second research question

The second research question (**RQ2**) is "Can TSFMs enhance the development of efficient and interpretable DVA tools?". This section shows the insights obtained for this question within the research conducted.

The Time Series Foundation models MOMENT family has been shown to capture TSs behavior within the embedding space [1]. However, the experiments conducted within DeepVATS's benchmark did not reproduce these outcomes: the latent space representation generated by MOMENT failed to capture TSs dynamics as effectively as DeepVATS' MTSAE in its three zero-shot variants. As a result of this discrepancy, three more specific questions were formulated (see Fig. 1.1):

- **RQ2.1** "How well do TSFMs' embedding space capture the behaviors of TS? Are they easier or, at least, similar to interpret than DeepVATS' MTSAE?"

- **RQ2.2** "Do quantitative improvements in the classical tasks result in embeddings space interpretability enhancement?"

- **RQ2.3** "Does fine-tune enhance the latent space interpretability? How much tuning is required to obtain high quality descriptive clusters for a specific time series?"

The three questions are related to the interpretability of MOMENT, assuming their fast use as they are foundation models (the analyst can directly use them on new datasets without training). The experiments show that MOMENT has not reached good interpretability in any of the tasks where DeepVATS do, nor in trend detection. This leads to a negative response to **RQ2.1**.

The loss was greatly improved (with percentages of up to 20% - see Fig. 4.12) but the embedding spaces were not easy to interpret in any of the configurations; thus, it's not possible to conclude that the improvement in terms of loss is directly related to the embedding space interpretability, as they seem to be independent (**RQ2.2**). Also, since this fine-tuning with different sizes of the model did not result in a visual enhancement, **RQ2.3** cannot be answered.

The absence of interpretable plots gives the impression that it is not worth introducing TSFM into DeepVATS as a visualization tool, but the results are strange compared to the state-of-the-art. Also, those results are not dependent on the number of samples in the training shots within the experiment, but it remains to try to train the model with the full dataset or even make it forbid its previous knowledge by initializing with random weights. Thus, they need another opportunity, explicitly stated in the previously proposed FL:

- Modify the loss distance function (**FL3**). Using other distance loss should not have real implications in terms of how well the model fits a TS, but does it change the configuration of the PP? Does the selection of a more global distance result in better embedding topology? If so, using a TSFM with few-shot with an specific distance for each task would be a great option for visual analysis of a TS.

- **Using other projection techniques (FL4).** The projection technique used within the experiment was PCA followed by UMAP to enable the extension of the results in [7]. However, the original article of MOMENT [1] uses PCA and t-SNE. The latent space can be checked to be easier to interpret when these techniques are used within our benchmark.

- **Data preprocessing (FL5)**. This option should clearly help in any case, as it is always a good idea to preprocess the data before using any task. However, the goal is not just to preprocess it and then analyze all its aspects but making preprocessed copies for the fine-tuning and later using the original dataset to infer new latent space and check if it has an easier-to-interpret topology.

- **Freezing layers (FL6)**. The art of fine-tuning models contains different techniques. One of the most extended is to freeze the parameters of specific layers so that the model can be improved focusing on just some parts of it. Now, the next question is: is it possible, even with greater losses, to get more interpretable latent space projections by freezing the final layer parameters?

5.2 Critical analysis of the methodology

The integration of MPlot and MOMENT into DeepVATS has provided improvements in terms of interpretability and efficacy. However, both paths in the methodology have some limitations that warrant further analysis.

5.2.1 Integration of MPlot into DeepVATS

The integration of MPlot into DeepVATS has shown significant improvements in terms of interpretability and computational efficiency compared to the use of MTSAE as the unique backbone model. However, they still have two fundamental limitations: the dependence on a fixed subsequence length (m) and their specification for univariate TSs.

The dependence on a predefined subsequence length (m) directly influences the generated similarity matrix. The choice of m is non-trivial even if using the Fourier-based proposed window sizes:

- If m is too small, short-term variations dominate, obscuring long-term patterns.
- If m is too large, the structural details are averaged, reducing the ability to capture localized anomalies or motifs.

This rigid dependence on m makes it difficult to analyze TSs with multiple timescales, requiring manual tuning for different datasets. This can be solved by translating multifocal MPlots [11] from `MATLAB` to `Python` and introducing them into DeepVATS. This method eliminates the need to manually select m by generating multiple views on different scales. Currently, this approach is not implemented in DeepVATS due to its complexity, limiting its ability to automatically adjust to different TSs characteristics (even though the window lengths proposed by Fourier Transform should be enough for detecting the main patterns and anomalies). Future iterations should integrate multifocal MPlot, allowing for dynamic, multiscale visualization without user intervention. This would also be an step nearer to Deep Learning field's view of "letting the information rise from the data rather than building it".

5.2.2 Analysis of MOMENT integration

The integration of TSFMs into DeepVATS to evaluate their effectiveness versus DeepVATS original backbone has two crucial points.

The first point is the dataset selection. MOMENT was trained and evaluated on the Time Series Pile[1], a large collection of task-specific datasets compiled from multiple public repositories [1]. This data set includes thousands of time series used for the analysis of MOMEMNT's performance. In contrast, DeepVATS has primarily been tested on a narrower dataset selection, restricting its ability to generalize across diverse real-world applications. Although it is impossible to check thousands of datasets in a visual way as no automatic evaluation is applicable, incorporating datasets from the Time Series Pile into DeepVATS evaluation could provide a more comprehensive performance assessment.

The second point is the zero-shot learning and the need for fine-tuning in MOMENT. MOMENT is designed as a family of pretrained foundation models, capable of performing zero-shot forecast-

[1] Datasets available at `https://huggingface.co/datasets/AutonLab/Timeseries-PILE`.

ing, classification, and anomaly detection. However, when applied in DeepVATS, the zero-shot performance resulted in highly entangled clusters that made embeddings difficult to interpret, which required fine-tuning to improve their definition. However, this fine-tuning was not directly related to an enhancement within the embeddings interpretability, suggesting the tuning technique should be improved.

5.2.3 Research limitations

Despite the promising results, the application still presents some constraints:

- Scalability limitations. Although MPlot has shown to aid on previewing information and MVP has been enhanced to make it possible to analyze large datasets, the application still has the issue of MOMENT models scalability.
- Embedding precision. While TSFMs enhanced computation time, it still lacks precision within the latent space, the fine-tuning process should be enhanced to ensure easy-to-interpret latent space projections.
- Multivariate MPlot analysis. To guarantee the joining between DL and DM, multidimensional MPlot can be integrated into the application, so both univariate and multivariate TSs can be analyzed while training DL models.

5.3 Practical implications and potential applications

As exposed in Chapter 2, time series modelation can be useful in the detection and prevention of rare events preventing detrimental financial applications [21], optimization of resources in natural disasters or behavior analysis [23, 27] or the prediction of orbits in space objects [23] or the early classification of space objects based on astronometric TSs data [322].

Recent advances in VA tools for TSs in both DL and DM fields have significantly expanded its applications in multiple domains, enabling NRT monitoring [323, 324, 325], predictive analytics [326], and anomaly detection [248, 327]. These developments show how different methodologies can be leveraged to solve domain-specific challenges, improving decision making in areas like healthcare, finance, industrial automation, environmental monitoring, or smart cities.

One of the more impactful applications is healthcare care, where TSs analysis supports continuous patient monitoring and predictive analytics. NRT patient data can be analyzed to detect irregularities in physiological signals, such as fluctuations in heart rate or abnormal oxygen levels, allowing the early detection of medical conditions such as Pulsus Paradoxus [11]. For example, weareable devices now incorporate DL powered TSs analysis provide NRT alerts to healthcare providers, allowing them to anticipate emergencies and optimize treatments in fields such as oncology and cardiology [328, 329, 330, 331, 332]. One recent innovation in this field is the application of spectogram-based models to medical TSs. The study provided by Zeng et al. [333] shows how spectograms are a good visual representation to treat time series as images and applying visual DL models to TSs. This technique is currently employed for the detection of latent brain states for spontaneous neural activity in the amygdala [334]. This type of vision-based algorithms and data preprocessing can improve DeepVATS by adding different valuable backbone models that generate the embeddings based on different information from the TSs rather than the input TSs, allowing, for example, the execution of different models to build other perspectives of the TSs when MTSAE may not capture the target property. This raises the question:

Do spectogram-based embeddings result in more interpretable latent spaces or not? This is a new open research line for the future. Additionaly, acsTSFM are proving effective in tracking infectious disease epidemics [335] which makes them even more valuable for their integration into DeepVATS.

In finance, TSs forecasting is fundamental for predicting market trends, optimizing investment strategies and improving risk assessment models. The rise of AI-driven investment strategies for timeseries forecasting has become a cornerstone of risk assesment and stock market tools optimization [41, 42, 43], influence the building of automatic AI-based investor techniques and tools [44].

Industrial applications, particularly in manufacturing and automation, have also benefited from DL-driven TSs analysis. Predictive maintenance and fault analysis in manufacturing and industrial automation rely on analyzing machine sensor data to prevent failures [336] and predict the remaining useful life [337].

In the field of environmental monitoring, TSs visualization plays a key role in managing water resources and assessing risks of climate change. Predicting key environmental variables in unmonitored locations is challenging due to insufficient observational data. The need for accurate predictions has become more urgent with increasing climate variability and extreme weather events, which affect global water resources [29, 338, 339]. Machine learning models improve hydrological forecasting, water quality assessment, and environmental risk management. Large-scale ML frameworks integrate site-specific features for better flood predictions, groundwater level estimation, and pollution monitoring. They also integrate hydrological, geomorphological, and other related exogeneous features, allowing better generalization to unmonitored regions [340, 341, 342]. These methods reduce the costs associated with large-scale monitoring. As traditional observation networks are prohibitely expensive [343], DL models provide a cost-effective alternative by leveraging existing data for extrapolation. Transfer learning and knowledge-guided AI approaches allow models trained in data-rich regions to make reliable predictions in unmonitored areas [344, 345, 346, 347], offering interesting tools for decision-making in locations where observational data may be incomplete or sparse.

In the context of smart cities, ML models are becoming essential to monitor traffic, detect unusual events, and improve security. These models are used to analyze road conditions in NRT, enhancing traffic management and ensure smoother mobility [348]. At the same time, they contribute to cybesecurity, protecting smart city networks from attacks in highly connected urban environments [349]. Beyond security and mobility, these models are useful in improving the reliability of urban infrastructure. By predicting pedestrian behavior, AI-driven systems make streets safer for both people and autonomous vehicles [350], while anomaly detection in Internet of Things (IoT) networks helps identify irregular patterns in energy consumption, transport demand, and infrastructure performance, ensuring efficient resource allocation and reliable urban services [351]. Furthermore, traffic prediction helps to improve plan transport systems, allowing authorities to optimize them and minimize delays [352]. However, traditional models often struggle to generalize to unpredictable urban dynamics because of their reliance on predefined datasets. Foundation models, such as those explored in Zhao et al. [353] facilitate effective operation in varying urban conditions. These models integrate generative architectures that enable real-time adaptation, anomaly detection, and decision making in complex urban environments. Their ability to generalize without extensive retraining then makes them a good solution for traffic optimization, infrastructure monitoring, and emergency response in smart cities.

Combined with advances in TSs forecasting and anomaly detection, DeepVATS provides an interactive Visual Analytics tool that enhances the interpretability of ML models applied to TSs data. Its contribution to interactive visual analysis of latent spaces resulting from the application of these models allows practitioners to gain deeper insights into the decision-making process by offering a complementary perspective. This added layer of interpretability is particularly valuable in critical fields such as healthcare, finance, environmental monitoring, and smart cities, where understanding the behavior of models can lead to more informed and reliable decisions. Using DeepVATS in these domains, users can not only validate predictions, but also explore patterns and trends that could otherwise remain hidden, contributing to improved safety, resource management, and the general well-being of society.

Chapter 6

Conclusions

Now this is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning.

— Winston Churchill

DeepVATS is a powerful tool designed for the easy visual analysis of both univariate and multivariate TSs. It allows users to interact with the embeddings projection plot and the original TS data plot, facilitating the detection of patterns within the data. The scalability analysis in Section 3.4 has demonstrated excellent performance for small datasets (ranging from 49.3K to 98.6K elements) and medium-sized datasets (up to 493.1K elements). However, performance issues appeared when processing larger datasets, with noticeable degradation at 3.7 million elements and application crashes at 7.4 million elements.

To enhance the application's usability for large datasets, three development work lines were proposed. First, to eliminate redundant processes, the `reactive` variables should be checked to determine if they can be converted to `reactiveVal` to ensure the use of cached values. This line has already been undertaken, facilitating the posterior analysis within the present research. Second, to ensure high-quality dimensionality reduction, two strategies are suggested: adopting an alternative UMAP implementation [2], and exploring the application of PCA followed by UMAP, rather than UMAP alone [3]. In this case, the second path is selected, allowing for better analysis as the stability enhancing the zoom operations for local and global analysis of the embedding spaces.

These modifications give the application a improvement in performance (e.g., the removal of an additional 28-second delay required to compute projections for the 4s frequency dataset) and improve stability in the GPU embeddings projection, making DeepVATS a more robust tool for the visual and interactive analysis of TSs.

After including these modifications, the research continued by integrating MPlot into the application. This integration, using a more efficient and flexible version of MPlot (including the specific function to measure trends), everages DeepVATS in univariate TSs. Also, the integration with MPlot allows the user to have an easy interactive app that allows the interaction with the

MPlot to visualize the relation with both the TS and the MatrixProfile. It also gives powerful new functionality for a fast preview of the behavior of the TS, resulting in a plot obtained in less than 10 seconds (see Table 4.1), enhancing the interactiveness of DeepVATS. Although MPlot cannot be used for multivariate TSs; the MatrixProfile has an analog definition for that case. The addition of multidimensional MatrixProfile is future work. Adding MatrixProfile-based features to the TS as another variable for model training could be interesting for concise analysis.

Furthermore, some new foundation models [1, 4, 5] can be useful to detect trends and could enhance the analysis within DeepVATS. Therefore, and to improve the development and functionalities of the DeepVATS tool, the new foundational models are included into the DL module. This integration was expected to enable the inclusion of new backbone models into DeepVATS that require no time-expensive training to perform a faster yet effective analysis of TSs. However, after checking the embeddings projection space through PCA followed by UMAP, they were not easy to interpret. Aa first fine-tune (similar to the training technique used for MTSAE); based on those results, the research concluded that the improvement in terms of loss is not directly related to the interpretability of the projections plot, making it difficult to enhance it in this way. However, the problem can be related to the way the inclusion has been done as it has been included in its naive version, so it is necessary to try different perspectives as proposed: modifying the distance metric in the loss function and evaluating their impact on the interpretability of the projections plot, the use of other projection techniques to ensure no dependence from the way the vectors are projected, data preprocessing for making more specific tasks, and improving the fine-tuning process by freezing some layers and checking the final projections plot (even in detriment of the loss improvement).

The integration of these tools and the new visual analytics tools introduced in Chapter 5 supposes a great join of the Deep Learning and Data Mining fields for improving the visual analysis of TSs. The next section closes this thesis by summarizing the future research lines detected for further enhancement of the interpretability and efficiency of DVA tools.

6.1 Future work

Once completed, the dissertation analyzed and enhanced the capabilities and limitations of VA tools (particularly DeepVATS for TSs analysis), identifying several promising avenues for future research that could extend the present work:

- **FL1**. Using MPlot-derived features as exogeneous variates may result in a better analysis as more distance-based information is condensed in the inputs of the model.
- **FL2**. The integration of new TSFM to the DL module. This path has been started with the addition of MOMENT into DeepVATS, but new multitask models that could appear or, even, any of the foundation models mentioned in Section 3.6 can also be added.
- **FL3**. Modify the loss distance function, with seem promising after the first trial with soft-DTW [6] in Section 4.2.8.
- **FL4**. Applying data preprocessing not as exogenous variates, but as an option for the fine-tuning of a TSFM (using the exogeneous variates to get the embeddings instead of the original TS data) This line would help improve precision within the embedding space for specific tasks.

- **FL5**. Using other projection techniques, even depending on the task, can help in the interpretability of the PP, highlighting different information captured by the embedding space.

- **FL6**. Freezing layers can aid in the precision of the latent space.

- **FL7. Enhancing the code** (first solution in Section 3.4). Taking a look into MOMENT's embedding obtainment and batching the computation. This should result in better performance within large datasets.

- **FL8. New visual analysis techniques from other fields.** As mentioned in the dissertation, the use of VA tools is on the rise, driven by the need to make processes more explainable and interpretable. Each tool employs its own set of techniques. Conducting a thorough review of the state of the art (not only covering current methods for time series analysis but also including techniques used in other machine learning models and latent space analysis) could prove highly valuable. This approach would help expand the capabilities of DeepVATS by integrating different perspectives into a single open-source application, ultimately improving both the understanding of the analyzed data and the analysis process itself.

- **FL9. Modification of the MTSAE model**. Enhancing TSs long patterns such as trends can involve modifying the MTSAE model, for example, by adding layer-wise cumulative dilations for spreader convolutions or adding larger kernels. In this way, larger patterns of the TSs could be captured.

All the directions would be beneficial by comparing the improvement achieved with respect to the computational resources and execution time used. Combined, these directions point to a future in which VA tools will become increasingly accurate, efficient, and interpretable, laying the basis for more insightful and interactive TSs analysis in fields that use HOTL-related tools and potentially transforming broader analytical practices.

Appendix

Appendix A

Additional images

This appendix contains supplementary figures that support the research but were not included in the main body of the text due to their size.

A.1 Additional plots for section 4.1.2

Figure A.1 shows the MPlot for the $T2$ feature of `M-Toy` with no clear results, while Figure A.2 displays the associated embedding space analysis using MTSAE, showing how two injected anomalies are well separated from the rest of the embedding trajectories.

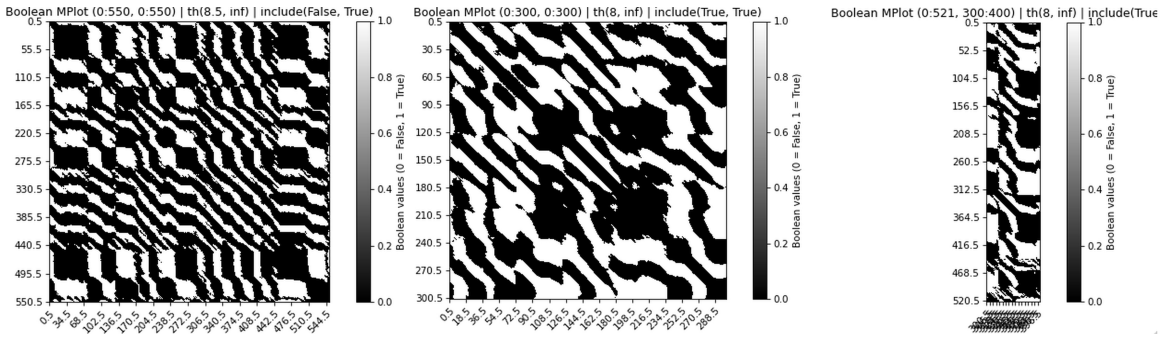

Figure A.1: MPlot for the `M-Toy` dataset $T2$ variable. The first row shows complete MPlot while the second row displays different zoom levels.

Figure A.3 illustrates the representation learned by MTSAE for `PulsusParadoxus`. The embedding projection reveals the different patterns in the cardiac signal.

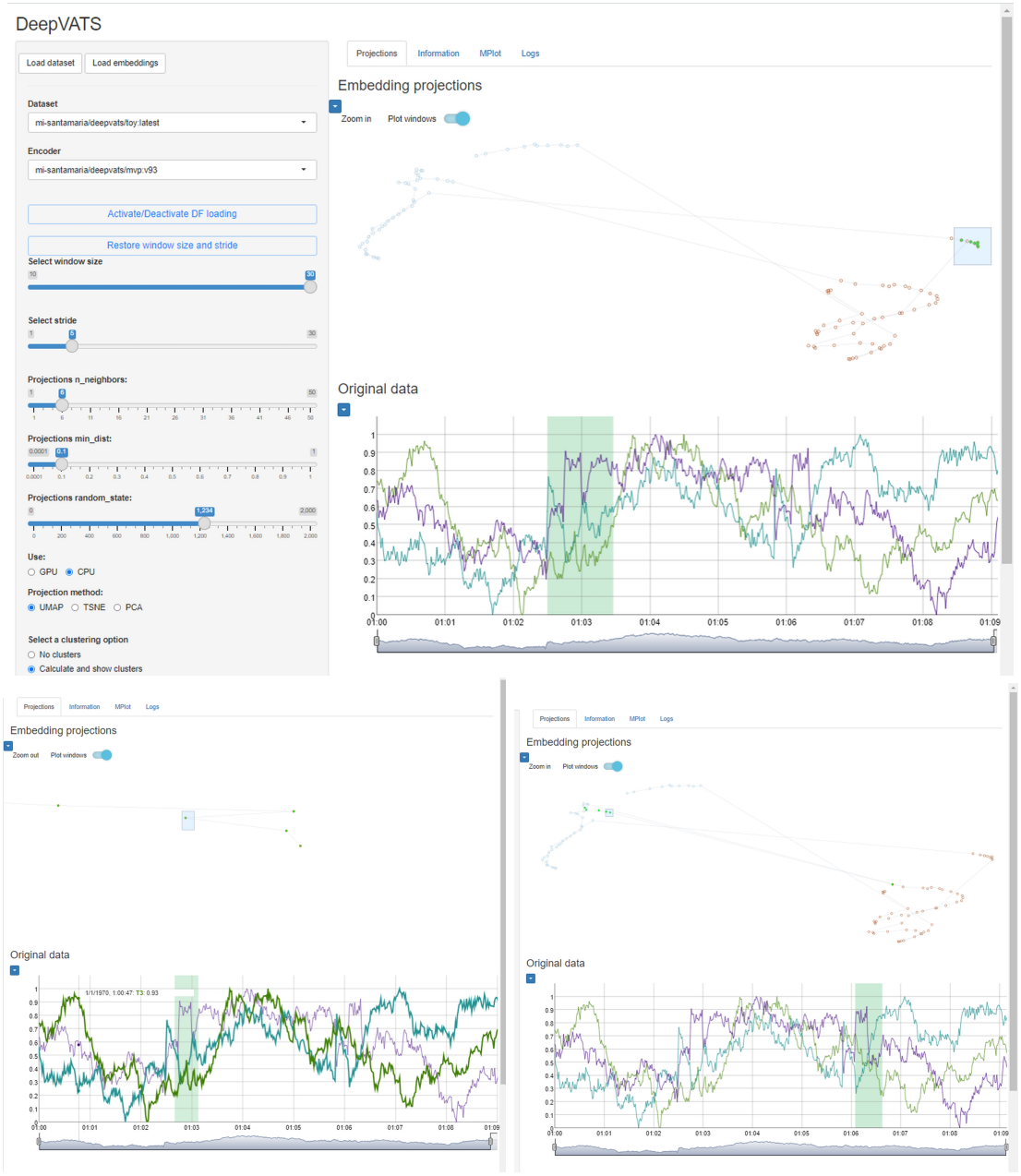

Figure A.2: Embedding space analysis for the two anomalies of the M-Toy time series.

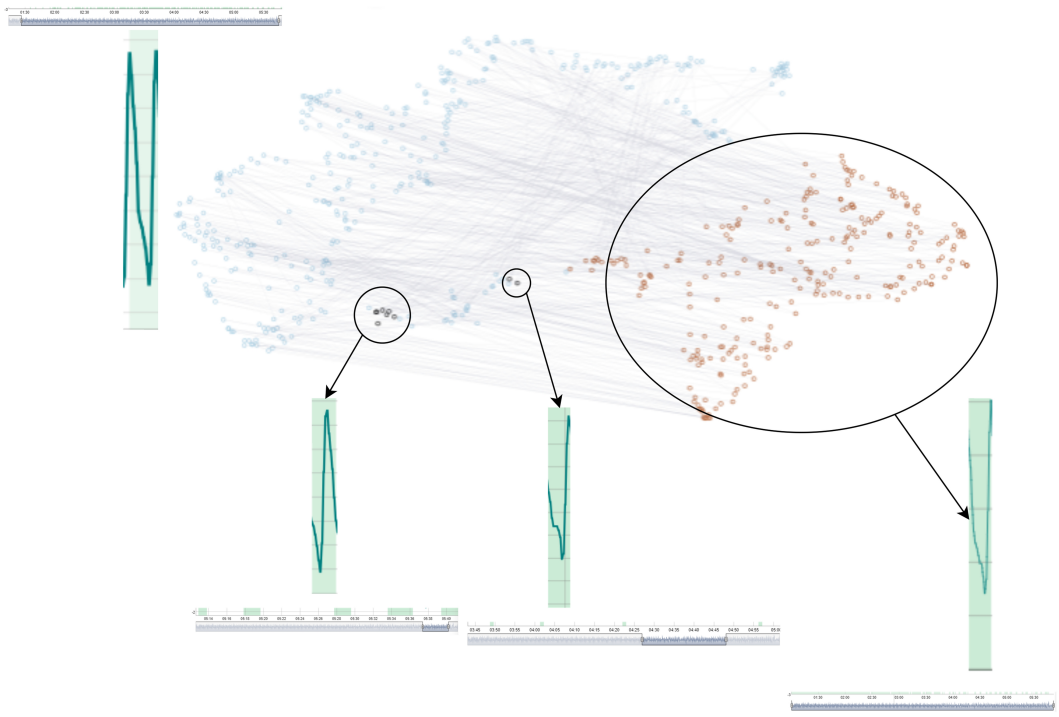

Figure A.3: Pulsus Paradoxus embedding space: checking patterns and anomalies.

A.2 Additional images for Section 4.2.2

Figures A.4, A.5, A.6, A.7, and A.8 show the analysis of the different clusters of the embedding space inferred by MOMENT-small (zero-shot version) for S1 taking the mean in batches of 20 minutes. The execution is performed for window length 54 (up) and stride 2. The parameters for dimensionality reduction (using PCA followed by UMAP with GPU) are the default values. The parameters for cluster computation are `metric=euclidean`, `min_size`= 40, `min_samples`= 15,`epsilon`= 0.08.

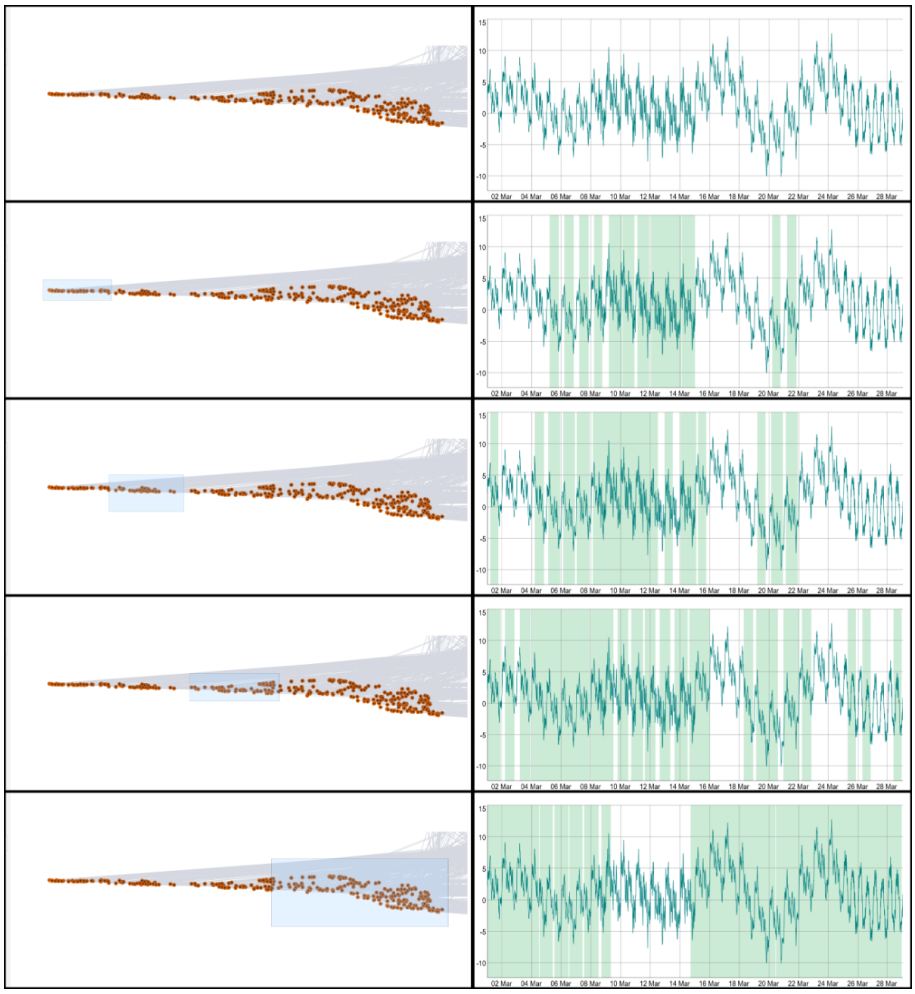

Figure A.4: Cluster I. Execution of MOMENT-small for S1. In the last row, all segments are highlighted except df_2, which is near to be detected in the second row.

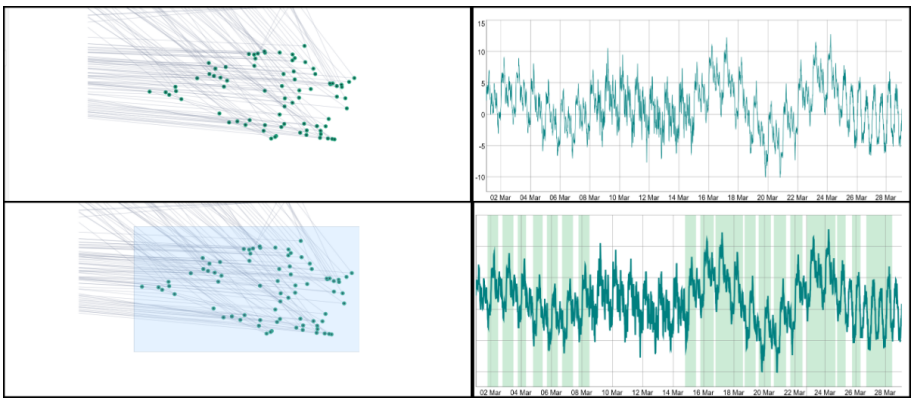

Figure A.5: Cluster II. Execution of MOMENT-small for S1.

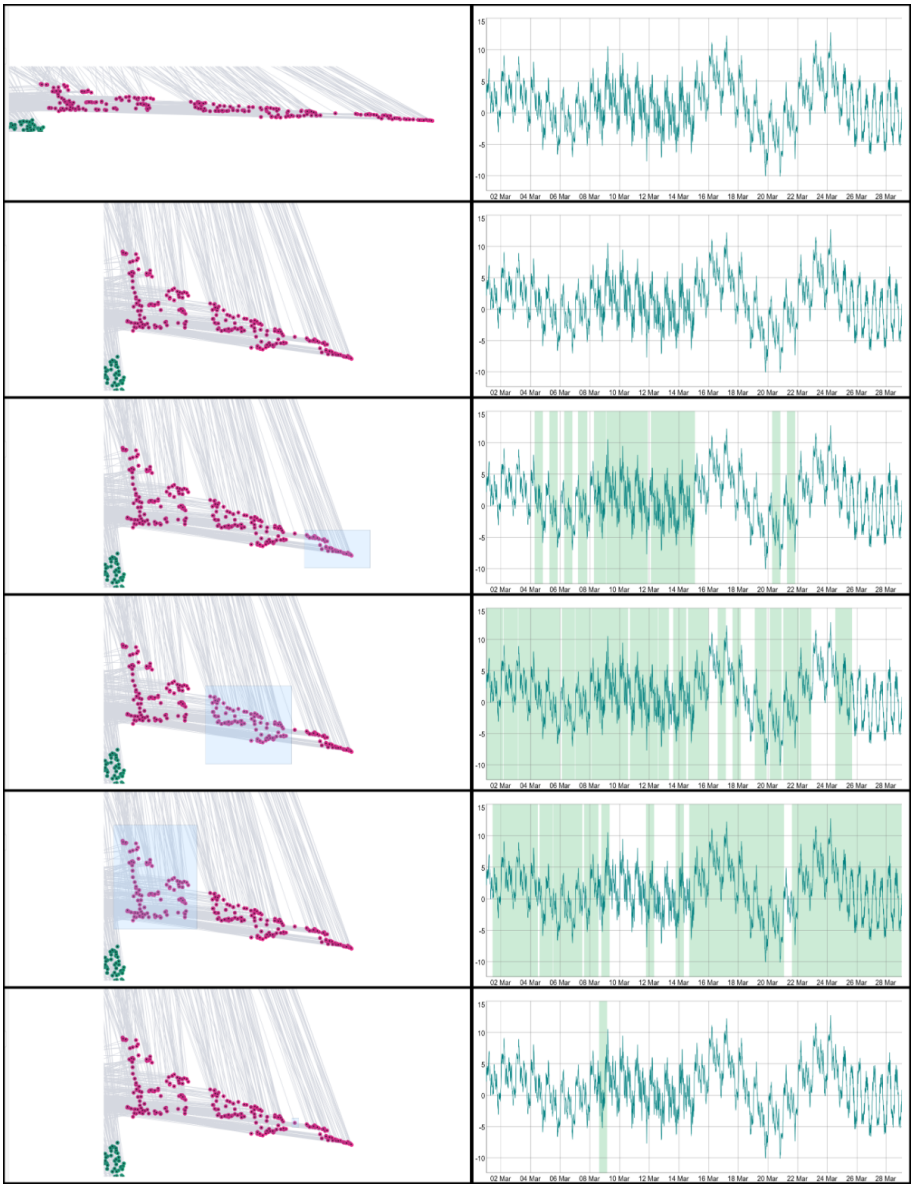

Figure A.6: Cluster III. Execution of MOMENT-small for S1.

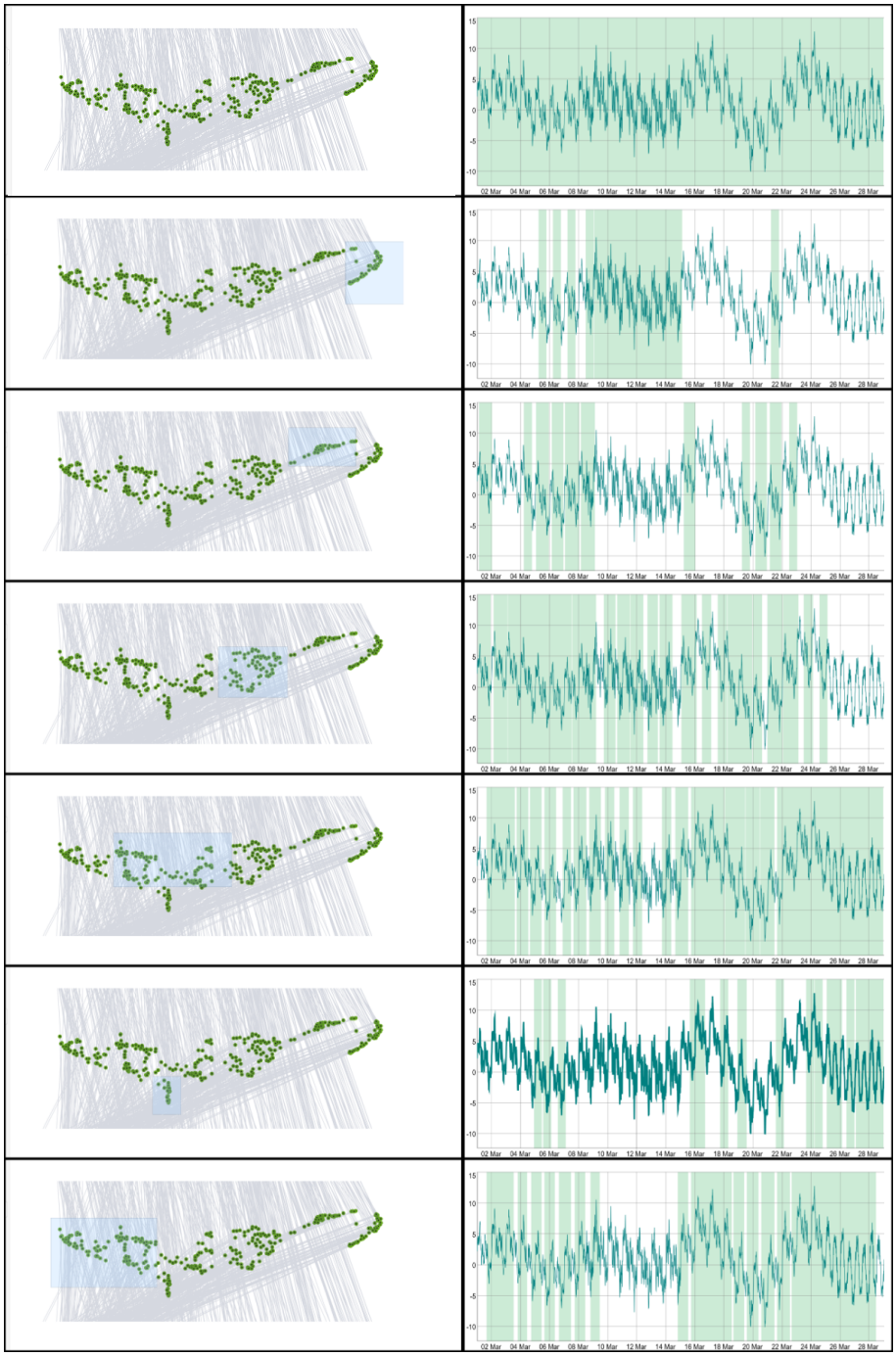

Figure A.7: Cluster IV. Execution of MOMENT-small for S1. Second and last row are about to detect the second segment.

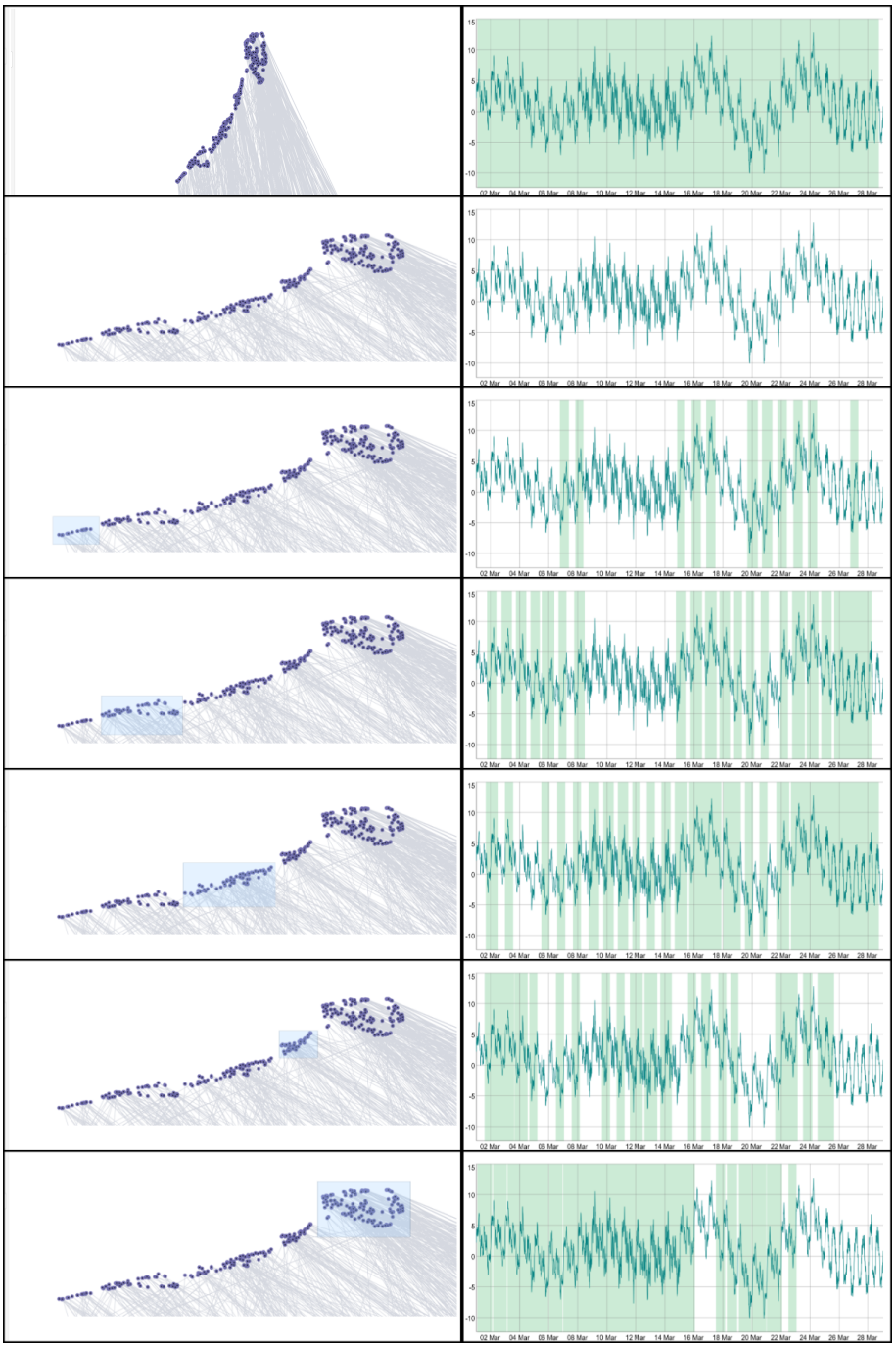

Figure A.8: Cluster V. Execution of moment-SMALL for S1.

Figure A.9 shows the global view of the zero-shot version of MOMENT-base applied to S2. The points highlighted are the two anomalies. Figure A.10 shows the global view of the embeddings projection of MOMENT-large (zero-shot) for S2.

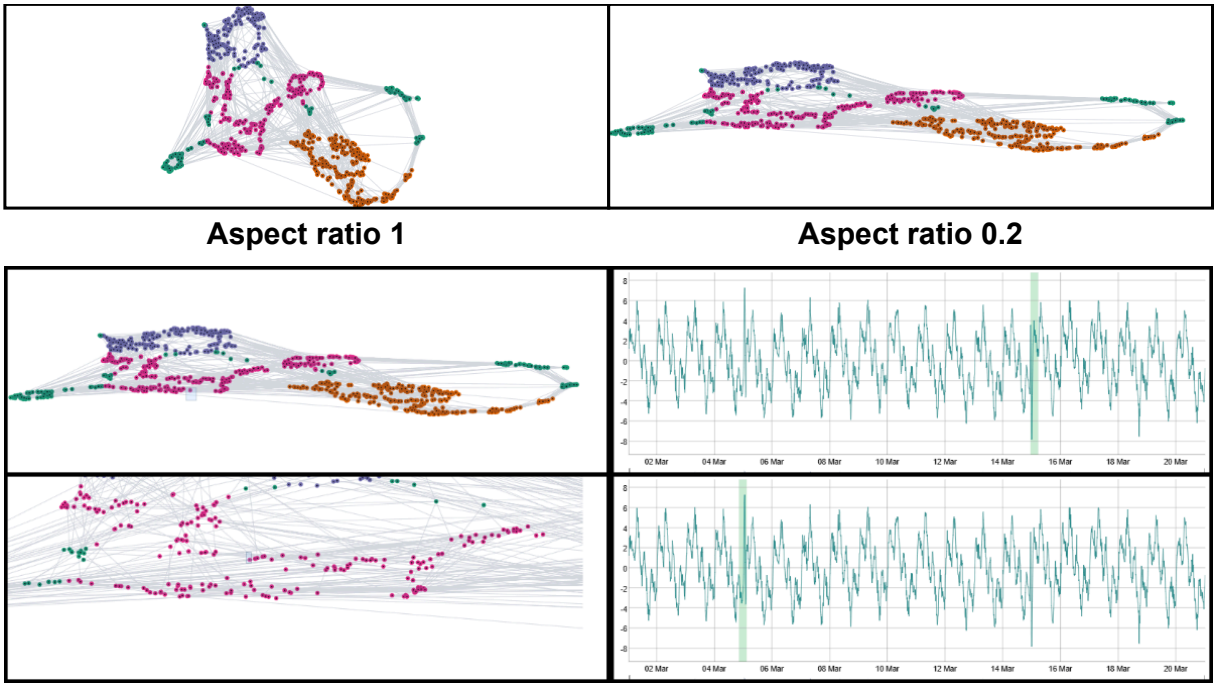

Figure A.9: Global view of the embeddings projections of the zero-shot version of MOMENT-base applied to S2.

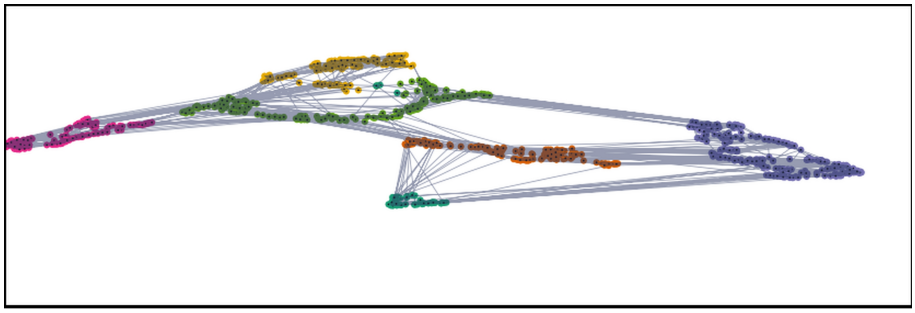

Figure A.10: Global view of the embeddings projections of the zero-shot version of MOMENT-large applied to S2 (stretched in the vertical axis using a 0.2 ratio).

Figures A.14 and A.15 show the zoomed view of each cluster in the latent representation of the zero-shot version of MOMENT-small.