Refining Temporal Visualizations Using the Directional Coherence Loss

Pavlin G. Poličar[1(✉)] and Blaž Zupan[1,2]

[1] Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
pavlin.policar@fri.uni-lj.si

[2] Department Education, Innovation and Technology, Baylor College of Medicine, Houston, TX, USA

Abstract. Many real-world data sets contain a temporal component or include transitions from state to state. For exploratory data analysis, we can present these high-dimensional data sets in two-dimensional maps, using embeddings of data objects under exploration and representing their temporal relations with directed edges. Most existing dimensionality reduction techniques, such as t-SNE and UMAP, disregard the temporal or relational nature of the data during embedding construction, leading to cluttered visualizations obscuring potentially interesting temporal patterns. To address this issue, we introduce Directional Coherence Loss (DCL), a differentiable loss function that we can incorporate into existing dimensionality reduction techniques. We have designed DCL to highlight the temporal aspects of the data, revealing temporal patterns that might otherwise remain unnoticed. By encouraging local directional coherence of the directed edges, the DCL produces more temporally-meaningful and less-cluttered visualizations. We demonstrate the effectiveness of our approach on a real-world multivariate time-series data set tracking the progression of the COVID-19 pandemic in Slovenia. We show that incorporating the DCL into the t-SNE algorithm elucidates the time progression of the pandemic in the embedding and reveals interesting cyclical patterns otherwise hidden in standard embeddings.

Keywords: Temporal-data visualization · Dimensionality reduction · Data visualization

1 Introduction

A common method for analyzing the structure of high-dimensional data involves representing it in two-dimensional, point-based visualizations. We can use dimensionality reduction approaches such as principal component analysis, multi-dimensional scaling, or t-SNE to obtain such data maps. Additionally, we can overlay these data maps with arrows indicating temporal dependence between data points to present temporal relations between data points. This approach has been used extensively for the visualization of dynamic graphs [4], multi-variate time-series [1], and gene-expression data [5].

A. Bifet et al. (Eds.): DS 2023, LNAI 14276, pp. 204–215, 2023.
https://doi.org/10.1007/978-3-031-45275-8_14

Existing approaches for visualizing temporal data via two-dimensional embeddings rely primarily on off-the-shelf embedding techniques, which do not incorporate the temporal aspects of the data. Commonly-used dimensionality reduction approaches, however, may be semantically constrained. Principal component analysis (PCA) [10], for example, relies on the linear transformation of attribute space and may fail to reveal complex patterns with non-linear interactions of input features. Non-linear data embedding techniques, such as multi-dimensional scaling [3], t-SNE [6], and UMAP [7], may overcome this limitation and introduce distortions into the embedding. None of these techniques, however, explicitly incorporates the available temporal information into the embedding construction process, resulting in embeddings that fail to reflect or even obscure the temporal patterns in the underlying data.

This report introduces the *directional coherence loss* (DCL), which integrates available temporal information into the embedding construction process. The result of DCL is embeddings designed to facilitate the discovery of temporal patterns in the two-dimensional embedding space. The DCL is differentiable, and we can incorporate it into existing dimensionality reduction techniques. Adding the DCL to the existing data embedding approach reveals temporal patterns in the resulting embeddings, aiding in discovering temporal patterns in the data.

2 Related Work

There are a plethora of approaches that we can use for the visualization of high-dimensional, temporal data. Rauber et al. [11] developed Dynamic t-SNE, which constructs a series of t-SNE embeddings and stacks them stacked along a third dimension corresponding to time. A similar approach has been proposed for UMAP, termed AlignedUMAP [7].

Alternatively, van den Elzen et al. [4] portray the progression of time in two-dimensional embeddings by connecting data points with arrows. Their approach focuses on visualizing dynamic graphs. At each point in time, the graph adjacency matrix is treated as a high-dimensional data point. This high-dimensional collection of graph snapshots is subsequently embedded into a two-dimensional visualization using an off-the-shelf embedding technique. Ali et al. [1] apply a similar approach to multivariate time-series data, where each sliding time window is treated as a single high-dimensional data point. In this way, they embed temporal sequences into two dimensions, where arrows connect consecutive time points. Unlike dynamic t-SNE and AlignedUMAP, which construct a three-dimensional embedding by stacking multiple two-dimensional embeddings along a time dimension, these approaches illustrate the entire temporal progression into two dimensions and indicate dependence using arrows.

In bioinformatics, single-cell RNA velocity [5] may accompany more standard gene expression data and requires a different visualization approach. Each data point corresponds to the gene expression of a single cell, characterized by tens of thousands of genes. Then, for each cell, single-cell RNA velocity estimates the likely transitions between different cell states, for instance, during differentiation.

The resulting visualization typically consists of a two-dimensional embedding constructed using t-SNE or UMAP overlaid with arrows to indicate likely cell-to-cell transitions. This approach is conceptually similar to van den Elzen et al. [4] and Ali et al. [1], where we deal only with a single time-step for every cell.

Another notable approach, Time Curves [2], offers general guidelines for visualizing the temporal progression of a single entity. The framework may be viewed as a generalization of the work by Ali et al. [1], allowing for arbitrary time steps between snapshots.

3 Methods

Consider a high-dimensional data set $\mathbf{X} \in \mathbb{R}^{N \times d}$, where N is the number of data points and d is the dimensionality of each data point. Let G be a directed graph $G = (V, E)$, where V denotes the set of vertices v_i corresponding to individual data points $\mathbf{x}_i$. E is the set of edges e_{ij} representing the temporal connections between data points i and j. When visualizing high-dimensional data sets, our primary objective is to find a low-dimensional embedding $\mathbf{Y} \in \mathbb{R}^{N \times 2}$ that accurately reflects the topological features of $\mathbf{X}$. In two-dimensional visualizations, we represent the connections e_{ij} as directed line segments $\mathbf{p}_{ij}$ (depicted as arrows) linking two related data points i and j in the embedding space such that $\mathbf{p}_{ij} = [\mathbf{y}_i, \mathbf{y}_j]$.

3.1 t-SNE

t-distributed stochastic neighbor embedding (t-SNE) is a non-linear dimensionality reduction technique commonly used to visualize high-dimensional data [6]. t-SNE aims to find a low-dimensional representation $\mathbf{Y}$ such that if two data points are close in the high-dimensional space $\mathbf{X}$, then they are also close in the low-dimensional space $\mathbf{Y}$.

Formally, the t-SNE algorithm aims to find a low-dimensional representation $\mathbf{Y}^*$, such that the Kullback-Leibler (KL) divergence between similarities $\mathbf{P}$ between data points in the high-dimensional space $\mathbf{X}$ and the similarities $\mathbf{Q}$ between data points in the low-dimensional space $\mathbf{Y}$ is minimized, such that

$$\mathbf{Y}^* = \arg\min_{\mathbf{Y}} \mathrm{KL}(\mathbf{P} \mid\mid \mathbf{Q}). \tag{1}$$

The similarities $\mathbf{P} = [p_{ij}]$ between data points in $\mathbf{X}$ are obtained using the Gaussian kernel,

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \quad p_{j|i} = \frac{\exp\left(-\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)/2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\mathcal{D}(\mathbf{x}_i, \mathbf{x}_k)/2\sigma_i^2\right)}, \quad p_{i|i} = 0, \tag{2}$$

where $\mathcal{D}$ is some distance measure and the bandwidth of each Gaussian kernel σ_i is selected such that the perplexity u of each conditional distribution matches a user-specified parameter value,

$$\log(u) = -\sum_j p_{j|i} \log\left(p_{j|i}\right) \tag{3}$$

In the low-dimensional representation $\mathbf{Y}$, the similarities $\mathbf{Q} = [q_{ij}]$ are characterized by the t-distribution,

$$q_{ij} = \frac{\left(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2\right)^{-1}}{\sum_{k \neq l} \left(1 + ||\mathbf{y}_k - \mathbf{y}_l||^2\right)^{-1}}, \quad q_{ii} = 0. \tag{4}$$

3.2 Directional Coherence Loss (DCL)

The key idea behind the *directional coherence loss* (DCL) is that arrows close to one another in the embedding space should point in approximately the same direction. Since each arrow is defined as a line segment parameterized by points $\mathbf{y}_i$ and $\mathbf{y}_j$, we can achieve this directional coherence by adjusting the positions of points $\mathbf{y}_i$ and $\mathbf{y}_j$ accordingly.

Let $\mathbf{u}_{ij}$ be the unit vector corresponding to the line segment $\mathbf{p}_{ij} = [\mathbf{y}_i, \mathbf{y}_j]$,

$$\mathbf{u}_{ij} = \tilde{\mathbf{u}}_{ij} / ||\tilde{\mathbf{u}}_{ij}||, \quad \tilde{\mathbf{u}}_{ij} = \mathbf{y}_j - \mathbf{y}_i \tag{5}$$

Then, for each pair of edges e_{ij} and e_{kl} in E, we can determine the directional coherence of their corresponding arrows in the embedding by computing the dot product $\mathbf{u}_{ij} \cdot \mathbf{u}_{kl} = ||\mathbf{u}_{ij}||\,||\mathbf{u}_{kl}|| \cos\theta$, where θ denotes the angle between the two vectors. In our case $||\mathbf{u}_{ij}|| = ||\mathbf{u}_{kl}|| = 1$, so their dot product simplifies to $\mathbf{u}_{ij} \cdot \mathbf{u}_{kl} = \cos\theta$. When $\mathbf{u}_{ij}$ and $\mathbf{u}_{kl}$ point in the same direction, their dot product is 1. Conversely, when $\mathbf{u}_{ij}$ and $\mathbf{u}_{kl}$ point in opposite directions, their dot product is -1. Therefore, to achieve good directional coherence for any pair of arrows in E, we must maximize the dot product of their corresponding directional vectors.

To make directional coherence compatible with existing dimensionality reduction loss functions, we convert the directional coherence into a strictly positive minimization loss. To convert the maximization into a minimization objective, we multiply the equation with -1. To enforce strict-positivity and avoid negative penalties, we add a $+1$ term to the above formulation and shift the domain from $[-1, 1]$ to $[0, 2]$. Additionally, we have found it beneficial to square the resulting equation, leading to faster convergence and more visually appealing visualizations. The directional coherence loss between edges pair of edges e_{ij} and e_{kl} then becomes

$$\mathrm{DCL}(\mathbf{p}_{ij}, \mathbf{p}_{kl}) = \left(-\left(\mathbf{u}_{ij} \cdot \mathbf{u}_{kl}\right) + 1\right)^2 \tag{6}$$

We penalize only nearby arrow pairs in order to enforce the local penalization of the DLC. The distance between two line segments $\mathbf{p}_{ij} = [\mathbf{y}_i, \mathbf{y}_j]$ and $\mathbf{p}_{kl} = [\mathbf{y}_k, \mathbf{y}_l]$ is defined as

$$d(\mathbf{p}_{ij}, \mathbf{p}_{kl}) = \arg\min_{s,t} ||\left[s \cdot \mathbf{y}_i + (1-s) \cdot \mathbf{y}_j\right] - \left[t \cdot \mathbf{y}_k + (1-t) \cdot \mathbf{y}_l\right]||, \tag{7}$$

where $s, t \in [0, 1]$. Intuitively, their distance corresponds to the distance between the two closest points on these line segments. If the line segments intersect, then their distance is 0.

We penalize nearby arrow pairs using a Gaussian kernel on the obtained pairwise line-segment distances,

$$w(\mathbf{p}_{ij}, \mathbf{p}_{kl}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-d(\mathbf{p}_{ij}, \mathbf{p}_{kl})/2\sigma^2\right), \tag{8}$$

where σ^2 is the variance of the Gaussian distribution. The variance σ^2 determines the region around each arrow where we wish the arrows to point in the same direction. This parameter can greatly affect the final embedding, as a large value of σ^2 will enforce the DCL across the entire embedding. In contrast, small values of σ^2 will have a limited effect on the point positions. It is also worth noting that this parameter should depend on the scale of the embeddings, which can change during optimization and vary across different dimensionality reduction algorithms. In our experiments, we use $\sigma^2 = 1$.

Combining the directionality penalty from Eq. 6 and the weights from Eq. 8, we obtain the final directional coherence loss,

$$\mathcal{L}_{DCL} = \frac{1}{\binom{|E|}{2}} \sum_{e_{ij} \in E} \sum_{e_{kl} \in E} w(\mathbf{p}_{ij}, \mathbf{p}_{kl}) \left(-\left(\mathbf{u}_{ij} \cdot \mathbf{u}_{kl}\right) + 1\right)^2, \quad (i,j) \neq (k,l). \tag{9}$$

We can incorporate the DCL loss into various dimensionality reduction methods. In our case, we augment the t-SNE algorithm with the DCL loss,

$$\mathcal{L} = \mathcal{L}_{t\text{-}SNE} + \lambda \mathcal{L}_{DCL} \tag{10}$$

where λ is the trade-off parameter between the two loss functions. In our experiments, we used $\lambda = 10$.

4 Results and Discussion

Below, we demonstrate the conceptual idea and expected results of our approach using a toy example. Additionally, we include a real-world case study on the progression of the COVID-19 pandemic in Slovenia. We conclude this section with a discussion of the potential shortcomings and limitations of the proposed approach.

4.1 Toy Example

We first consider a toy example to demonstrate that adding the DCL to the t-SNE dimensionality reduction algorithm elucidates trajectories or transitions between different clusters. This synthetic data set consists of seven distinct, non-overlapping clusters at equal distances from one another, each containing 50 points sampled from unit-Gaussian distributions. To simulate transitions between clusters, we connect each point from a given cluster c to a randomly chosen point from the subsequent cluster $c+1$. The data points in the last cluster from a sequence of connected clusters are connected to the data points from the

first cluster. This toy example can be thought of as a cyclic process containing seven distinct states where transitions are only possible between adjacent states. This may correspond to, for instance, single-cell data containing gene expression profiles corresponding to four different cell-cycle states in cell division.

We optimize the embedding using batch gradient descent as implemented in `pytorch` [9] for 10,000 iterations using a learning rate of 10. We use `ReduceLROnPlateau` to reduce the learning rate once the loss has not improved for 3,000 iterations. We use a perplexity value of 30 in the t-SNE loss function.

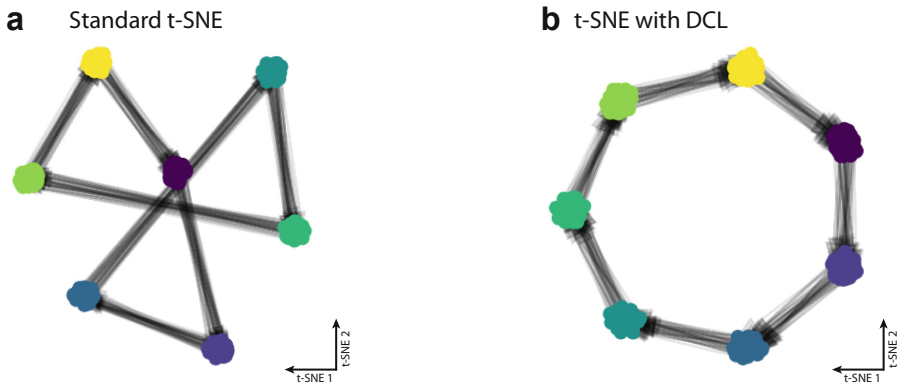

Fig. 1. The toy example demonstrates that incorporating the directional coherence loss (DCL) can help highlight the temporal transitions between data points. We construct a standard t-SNE embedding in **(a)**, which can recover the seven distinct clusters. However, the arrows between clusters cross over one another, making it challenging to observe the underlying cyclic pattern. Incorporating the DCL in **(b)** helps untangle the crossing arrows and highlights the cyclic pattern in the underlying data set while still recovering the seven clusters.

Figure 1a shows that while t-SNE can recover the seven distinct clusters from the high-dimensional space, overlaying the embedding with arrows clutters the visualization, concealing the cyclic pattern in the underlying data set. On the other hand, augmenting the standard t-SNE loss function with the DCL untangles the arrows and highlights the cyclic pattern as shown in Fig. 1b. Combining the t-SNE dimensionality reduction algorithm, which can identify the distinct clusters, with the DCL, which positions the clusters so that the transitions between the clusters are most apparent, considerably enhances the interpretability of the embedding and the underlying temporal pattern.

The t-SNE algorithm aims to preserve distances to a user-specified number of neighbors. However, accurately preserving distances obtained from high-dimensional data sets in a two-dimensional embedding is only possible in some of the most straightforward data sets. Using a perplexity value of 30, t-SNE does its best to preserve distances to each point's 30 nearest neighbors in the high-dimensional space. However, t-SNE also attempts to preserve distances to

other data points, albeit to a much lesser extent. In our synthetic data set, each cluster comprises 50 data points, meaning that, in addition to the points in the same cluster, t-SNE also attempts to preserve at least some distances from the other clusters. In Fig. 1a, the purple cluster is positioned centrally to other clusters, roughly at equal distances from the remaining clusters. Here, t-SNE can preserve the distances reasonably well. On the other hand, the top-left yellow cluster appears close to the central purple cluster and the light-green cluster below it, suggesting that these clusters are closer to one another than to, for instance, the right-most green cluster. However, by design, all seven clusters are at equal distances from one another in the high-dimensional space and cannot be accurately embedded in a two-dimensional plane. Consequently, the between-cluster distances in all nearest-neighbor-based two-dimensional embeddings are often meaningless and should never be taken at face value. This is a general limitation of dimensionality reduction techniques and has been documented in numerous reviews, e.g., by Nonato and Aupetit [8].

Note, however, that incorporating the DCL necessarily reduces the embedding quality regarding the t-SNE loss function. For instance, although the distances between clusters were poorly preserved in Fig. 1a, the between-cluster distance distortions were arguably less severe than in Fig. 1b, where each cluster is closest to its preceding and subsequent cluster, and progressively further from the remainder. This layout indicates that adjacent clusters are more similar than non-adjacent ones when, in reality, all clusters are at equal distances from one another. Nonetheless, despite this embedding being quantitatively worse at preserving distances between clusters, we argue that it provides a more informative visualization. When constructing embeddings for high-dimensional data sets, distances between clusters in the embedding should never be taken at face value, regardless of the dimensionality reduction technique. While the spatial relationships between clusters can aid in hypothesis generation, they should always be validated using alternative techniques.

Given that the spatial relationships between clusters lack informative value and can even mislead, it would be more sensible to position clusters in a temporally coherent manner. In this way, at least, the temporal relationships are more clearly highlighted, and the user is more directly aware of the limitations of interpreting spatial relationships, an often overlooked limitation of non-linear dimensionality methods. This way, the embedding algorithm can still recover well-defined clusters of data points in the high-dimensional space. Still, we explicitly decide that the spatial positions will reflect the temporal component of the embedding and not the spatial relationships between clusters.

4.2 COVID-19 Pandemic in Slovenia

We obtain Slovenian national data on the COVID-19 pandemic spanning from the beginning of March 2020 up until the end of March 2022[1]. Although the data

[1] National Slovenian data on the COVID-19 pandemic is available at https://covid-19.sledilnik.org/en/stats.

includes many variables, we limit our analysis to three time-series variables: the daily number of tests performed, the daily number of confirmed cases, and the daily number of hospital patients. We plot the individual time series in Fig. 2a. The line plots indicate the progression of the COVID-19 pandemic in Slovenia, with visible distinct phases of the pandemic.

To construct a two-dimensional visualization of the pandemic progression through time, we follow the approach from Ali et al. [1]. We first convert this multi-variate time series into a high-dimensional data set by constructing vectors from a sliding window with window size 7. Thus, the 160 21-dimensional data points represent one week of the pandemic. We connect data points corresponding to subsequent weeks with arrows.

We construct a t-SNE visualization of the high-dimensional data set in Fig. 2b. While the plot indicates a clear progression through time, the plot fails to reveal any underlying patterns in the data. Figure 2c depicts the results of our approach. While the embedding has not changed much structurally, the visualization reveals two clear cyclic patterns in the upper-right and lower regions of the embedding space.

We investigate the top-right cyclic pattern in Fig. 2c. Inspecting the two corresponding time spans highlighted in the original time series in Fig. 2a, it appears that this cyclic pattern coincides with high hospitalization rates, moderate levels of testing, and a moderate number of positive tests. Interestingly, both periods occurred during the spring season, one in 2021 and one in 2022. The first of these periods was substantially longer, lasting to the end of May, while the second lasted only a month and a half. It is also interesting to inspect which COVID-19 variants were prevalent in the country at that time[2]. During the first period in 2021, we were dealing with the initial 20A strain. The second period coincides with the transition from the Delta strain to the Omicron strain. The highlighted region in Fig. 2a corresponds to the final weeks of the Delta variant, which had higher mortality rates than the Omicron variant [12]. These strain prevalence and dynamics may explain the subsequent peak in the positive test cases and lower hospitalization rate following the highlighted region.

Finally, adding the DCL to the t-SNE algorithm elucidates the time progression of the time series. For instance, in Figs. 2e and 2e, we focus on a particular region of the embedding space, where it first appears as though the standard t-SNE embedding better highlights the temporal progression than with the addition of the DCL. Upon closer inspection, however, it is challenging to trace the arrows denoting the temporal progression of the pandemic as the arrow seems to veer off to the right, then cycle back, only to make another cycle back to the originating point. It is unclear which of these cycles occurred first and which second. With the addition of the DCL, it becomes easy to trace the temporal progression, as indicated by the red arrow drawn on top of the arrows to facilitate reading the embedding.

[2] The prevalence of the different COVID-19 variants in different countries is available at https://covariants.org/per-country.

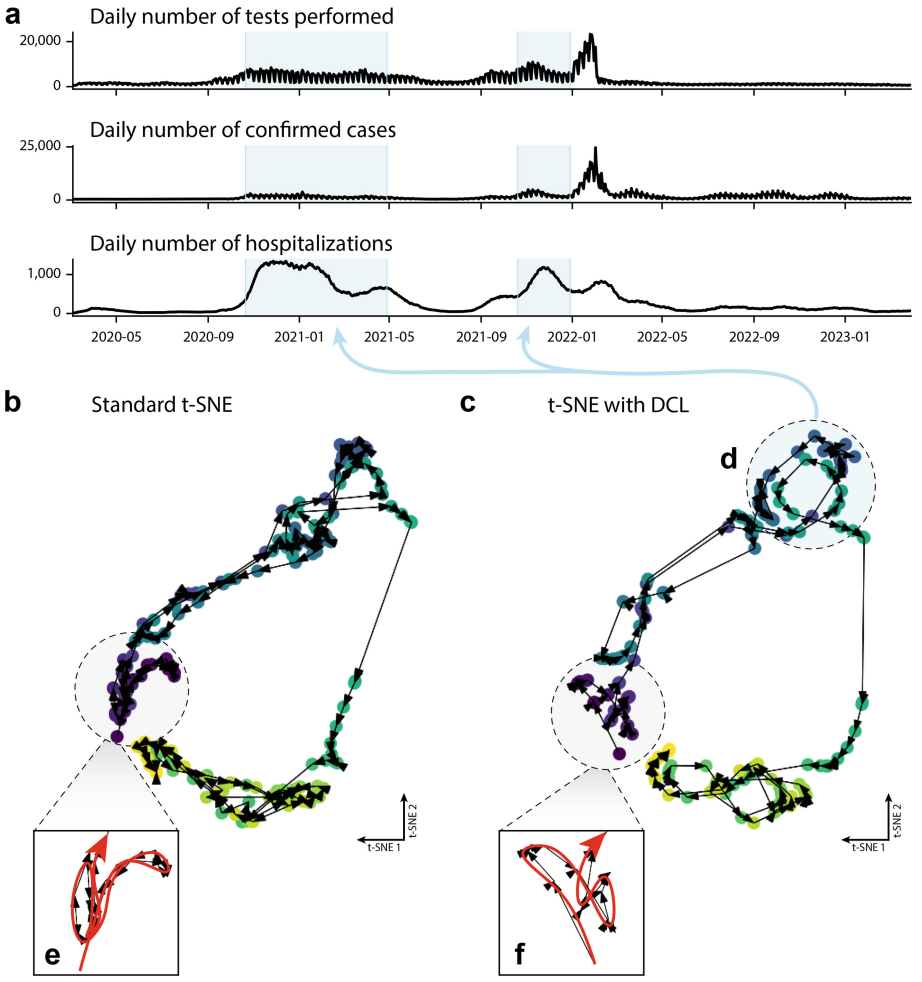

Fig. 2. We plot the progression of the COVID-19 pandemic in Slovenia from March 2020 to March 2022. **(a)** depicts individual line plots of the three variables under consideration. We construct a t-SNE embedding of the multivariate time series in **(b)** and augment the t-SNE loss function with our directional coherence loss in **(c)**. Individual points correspond to one week of the time series. We indicate the chronological progression by point colors where dark, purple colors correspond to the start of the pandemic, while lighter, yellow colors coincide with later stages of the pandemic. We connect consecutive weeks by arrows. Incorporating the directional coherence loss uncovers interesting temporal patterns in the visualization. We highlight one such cyclic region in **(d)** and mark the corresponding time spans in the original line plots. Panels **(e)** and **(f)** provide close-up views of regions of the original and augmented t-SNE embedding. We clarify the time progression by superimposing a red arrow onto the plot.

4.3 Hyperparameter and Evaluation Considerations

Incorporating the DCL into existing algorithms introduces two additional hyperparameters to the visualization procedure. The kernel bandwidth σ determines the radius in which the DCL is enforced. A larger bandwidth emphasizes global coherence, while a lower value results in more locally consistent arrows. Additionally, the parameter λ determines the trade-off between the visualization loss and the DCL. Placing a greater emphasis on temporal coherence highlights temporal progression, enabling a clearer visualization of temporal dependencies. However, this may obscure the underlying structure in the resulting visualization. Therefore, finding optimal parameter settings for the DCL is crucial to achieving a well-balanced visualization and likely varies from dataset to dataset.

We evaluate our approach using a toy dataset designed to illustrate the conceptual motivation behind our method. While this example supports the validity of our approach, real-world data may not display such straightforward temporal patterns. For a more comprehensive evaluation, we could create other synthetic datasets to test various scenarios and temporal patterns. While we could also apply our approach to real-world, multivariate time-series data, the interpretation of such study outcomes might be subjective. An ideal evaluation would involve an objective measure of visualization quality. However, devising such a quantitative metric is challenging even for non-temporal, two-dimensional embeddings. Adding temporal coherence to this metric introduces additional complexities and challenges.

5 Conclusion

The work presented here was motivated by the difficulties of identifying temporal patterns in presentations of multi-variate data in a low-dimensional, non-linear embedding. There, we may expose the temporal relations using arrows to indicate the transitions. These visualization elements often clutter the data presentations and obscure the underlying temporal patterns. Existing dimensionality reduction techniques do not account for the temporal nature of the data. To this end, we propose the *directional coherence loss* (DCL), which can be incorporated into existing dimensionality reduction techniques. Uniquely, the DCL explicitly integrates the temporal information into the embedding construction process and produces embeddings highlighting the temporal patterns in the underlying data more clearly.

This presented work opens up several avenues for future research. First, the DCL enforces directional coherence by affecting the positions of the data points in the two-dimensional embedding. While this approach is viable for simpler data sets, such an arrangement may be difficult to achieve in the presence of more complex patterns. Secondly, the DCL is applicable when the arrows begin at one data point and end at another. This is not the case in data such as those from bioinformatics that include RNA velocity, where arrows originate from data points but end in an average position of multiple data points. The DCL must be extended to make it applicable to this case. Thirdly, in its current

form, the DCL exhibits quadratic scaling in the number of connections between data points, making it unsuitable for visualizing large data sets. Due to the local nature of the DCL, approximation schemes could be developed which would only compute the interaction between nearby line segments. Lastly, we could find better optimization schemes leading to faster convergence.

Acknowledgements. This work was supported by the Slovenian Research Agency Program Grant P2-0209 and Project Grant V2-2272.

References

1. Ali, M., Jones, M., Xie, X., Williams, M.: Towards visual exploration of large temporal datasets. In: 2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA), pp. 1–9 (2018). https://doi.org/10.1109/BDVA.2018.8534025
2. Bach, B., Shi, C., Heulot, N., Madhyastha, T., Grabowski, T., Dragicevic, P.: Time curves: folding time to visualize patterns of temporal evolution in data. IEEE Trans. Visual Comput. Graphics **22**(1), 559–568 (2016). https://doi.org/10.1109/TVCG.2015.2467851
3. Borg, I., Groenen, P.J.: Modern Multidimensional Scaling: Theory and Applications. Springer, New York (2005). https://doi.org/10.1007/0-387-28981-X
4. Van den Elzen, S., Holten, D., Blaas, J., van Wijk, J.J.: Reducing snapshots to points: a visual analytics approach to dynamic network exploration. IEEE Trans. Visual Comput. Graphics **22**(1), 1–10 (2016). https://doi.org/10.1109/TVCG.2015.2468078
5. La Manno, G., et al.: RNA velocity of single cells. Nature **560**(7719), 494–498 (2018). https://doi.org/10.1038/s41586-018-0414-6
6. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
7. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. ArXiv e-prints (2018)
8. Nonato, L.G., Aupetit, M.: Multidimensional projection for visual analytics: linking techniques with distortions, tasks, and layout enrichment. IEEE Trans. Visual Comput. Graphics **25**(8), 2650–2673 (2019)
9. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019)
10. Pearson, K.: On lines and planes of closest fit to systems of points in space. Phil. Mag. **2**(11), 559–572 (1901). https://doi.org/10.1080/14786440109462720
11. Rauber, P.E., Falcão, A.X., Telea, A.C.: Visualizing time-dependent data using dynamic t-SNE. In: EuroVis 2016 - Short Papers, pp. 73–77. The Eurographics Association (2016). https://doi.org/10.2312/eurovisshort.20161164
12. Wrenn, J.O., et al.: COVID-19 severity from Omicron and Delta SARS-CoV-2 variants. Influenza Other Respir. Viruses **16**(5), 832–836 (2022). https://doi.org/10.1111/irv.12982