A Deep Learning Approach to Selecting Representative Time Steps for Time-Varying Multivariate Data

William P. Porter* Univ. of Notre Dame
Yunhao Xing† Sichuan Univ.
Blaise R. von Ohlen‡ Univ. of Notre Dame
Jun Han§ Univ. of Notre Dame
Chaoli Wang¶ Univ. of Notre Dame

ABSTRACT

We present a deep learning approach that selects representative time steps from a given time-varying multivariate data set. Our solution leverages an autoencoder that implicitly learns feature descriptors of each individual volume in a latent space. These feature descriptors are used to reconstruct respective volumes for error estimation during network training. We then perform dimensionality reduction of these feature descriptors and select representative time steps in the projected space. Unlike previous approaches, our solution can handle time-varying multivariate data sets where the multivariate features can be learned using a multichannel input to the autoencoder. We demonstrate the effectiveness of our approach using several time-varying multivariate data sets and compare our selection results with those generated using an information-theoretic approach.

1 INTRODUCTION

Time-varying multivariate data analysis and visualization has been an important research topic in scientific visualization. Along this topic, a key question researchers have studied is how to select representative time steps from a series of volumes. The selected time steps can be treated as a summarization of the entire time series for subsequent analysis and visualization in a cost-effective manner. Existing works for time step selection are mainly based on information-theoretic methods [12, 15], dynamic programming techniques such as dynamic time warping [10], or a combination of both [15]. In addition, a solution based on a minimum-cost flow-based technique was proposed [1] for adaptive time step selection.

Although effective, the aforementioned solutions rely on hand-crafted data features such as histograms, distributions, or isosurfaces to evaluate the similarity or difference of the corresponding time steps. Inspired by recent work on feature learning from streamlines or stream surfaces [4], we advocate a machine learning approach that automatically "learns" implicit feature descriptors of volumes at individual time steps in a latent space. This can be realized using an encoder-decoder framework in an unsupervised manner. Once learned, the feature descriptors can well represent the underlying volumetric data and can be used for time step selection. We achieve this through dimensionality reduction and selection of representative time steps in the 2D projected space.

The contributions of our work are the following. First, our work is the first that applies deep learning techniques for time step selection. The deep learning approach enables implicit feature learning, eliminating the need for explicit feature engineering. Second, we integrate feature learning, projection, and exploration into a single framework for time step selection and compare our work against existing work. Third, unlike all previous works which only address the time step selection problem for a single variable [1, 10, 12, 15], our work can naturally handle multivariate data sets by selecting representative time steps based on multivariate temporal features.

*e-mail: wporter2@nd.edu
†e-mail: yhxing98@gmail.com
‡e-mail: bvonohle@nd.edu
§e-mail: jhan5@nd.edu
¶e-mail: chaoli.wang@nd.edu

IEEE VIS 2019: Short Papers
20-25 October, Vancouver, BC, Canada
978-1-7281-4941-7/19/$31.00 ©2019 IEEE

2 RELATED WORK

Deep learning has achieved impressive results in video summarization. Zhang et al. [14] built a convolutional neural network (CNN) with long short-term memory (LSTM) to capture temporal dependency among video frames though annotated videos. Gong et al. [3] proposed sequential determinantal point process (SeqDPP) for selecting informative and diverse video frames to meet human-perceived evaluation metrics in a supervised way. Zhang et al. [13] established a subset selection technique that utilizes CNN though human-created summaries to perform automatic keyframe-based video summarization. Our work differs from the above works in the following two aspects. First, the above works require a large human labeled data set but our deep learning framework can automatically select key volumes without annotation. Second, due to the difficulty of training LSTM, we only utilize CNN to extract volumetric features and then select the key time steps based on the extracted features.

For feature learning using neural nets, Girdhar et al. [2] proposed an encoder-decoder to learn object features and applied these features for 3D object classification. Liu et al. [8] utilized a generative adversarial network (GAN) to automatically learn object features and recombined these features to synthesize unseen 3D objects. Our work is similar to Han et al. [4] which establishes an autoencoder to learn streamline or stream surface features from their respective binary volume representations and utilizes these features to select representatives. The difference is that instead of selecting streamlines or stream surfaces, we aim to select representative time steps for time-varying multivariate data sets.

3 APPROACH

Our approach consists of two phases: *feature learning* and *time step selection*, as sketched in Figure 1 (a). At the first phase, our network accepts the volume at each time step as input, generates its feature descriptor, and then utilizes the feature to reconstruct the corresponding volume for network training. At the second phase, we project the feature descriptors to a 2D space and select representative time steps in the projected space.

Our network contains an encoder and a decoder. The encoder takes a $C \times L \times W \times H$ volume as input and generates a feature descriptor while the decoder accepts the feature descriptor as input and outputs a reconstructed volume. L, H, and W denote the dimension of the volume and C denotes the number of channels of this volume. If $C = 1$, it is for single variable while if $C \geq 2$, it is for multiple variables. The encoder (decoder) consists of four learning blocks where a learning block includes a convolutional (deconvolutional) layer, a rectified linear units (ReLU) layer [9], and a residual block [6], as sketched in Figure 1 (b). After each learning block, the resolution is halved (doubled) in the encoder (decoder). Following the four learning blocks in the encoder are one convolutional (Conv) layer and one ReLU layer. The encoder generates a 1024-dimension feature descriptor as output. Similar to the encoder, for the decoder, we add one deconvolutional (DeConv) layer after four learning blocks to produce the reconstructed volume. Note that $\tanh(\cdot)$ is applied after

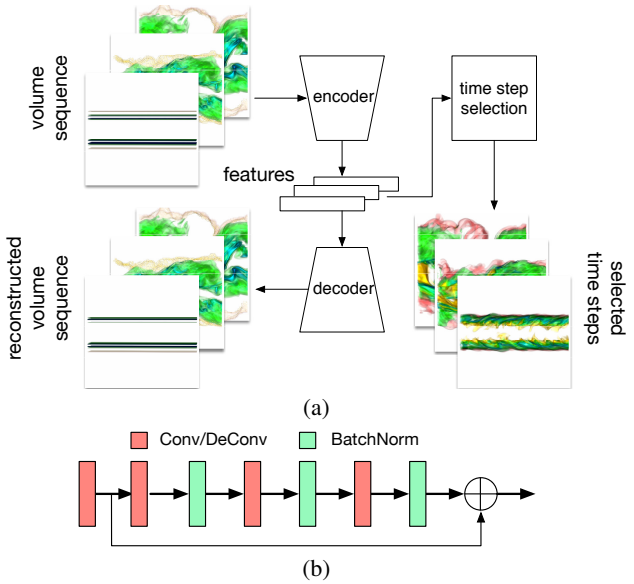


Figure 1: (a) Our approach learns a feature descriptor for each time step (single or multiple variables) and selects representatives in the projected space. We use downsampled volumes as input due to the GPU memory limitation. (b) The diagram of a learning block.

the last DeConv layer. For network optimization, we use the mean-square error (MSE) to calculate the loss between the ground-truth and reconstructed volumes

$$\mathscr{L} = \frac{1}{L \times W \times H} \sum_{k=1}^{N} ||V_k^{'} - V_k||_2, \tag{1}$$

where N is the number of training samples, $V_k^{'}$ and V_k are the reconstructed and ground-truth volumes at time step k, respectively, and $||\cdot||_2$ denotes the L_2 norm.

After collecting the feature descriptors of all the time steps, we apply t-SNE [11] to reduce the feature dimension from 1024 to 2. As a neighborhood-preserving method, t-SNE has been shown to perform better than other dimensionality reduction techniques based on distance-preserving methods such as MDS and Isomap [4]. Then in the 2D projected space, each point represents a time step. We connect the neighboring time steps to form a "path" and select representatives in the following ways:

- *Arclength-based selection.* Compute the Euclidean distance between the neighboring points and select representatives as path resampling based on the arclength. Given a distance threshold ε, different numbers of representatives can be selected.
- *Angle-based selection.* Compute the angle formed among consecutive neighboring points and select representatives as path simplification based on the accumulated angle. Given an angle threshold θ, different numbers of representatives can be selected.
- *Mixed selection.* Combine arclength-based and angle-based selections linearly. A threshold $\alpha \in [0,1]$ is used to control the importance between arclength and angle. By default, $\alpha = 0.5$.

Note that the representatives are selected in the 2D projected space instead of the original feature space. This is because in the projected space, we can visualize features distribution and observe their pattern. Moreover, we can intuitively explain why certain time steps are selected from both quantitative and qualitative perspectives. If we select the representatives in the feature space, the features may not form a curve pattern and traditional measures such as the Euclidean distance become inapplicable [4]. Our approach can also select representative time steps for multivariate data sets. This is achieved by taking the multivariate volumes at each time step as the multichannel input to the encoder and generating the corresponding multivariate volumes as the output of the decoder. The resulting feature descriptor thus captures the multivariate features of the underlying data for time step selection.

Table 1: The dimensions of each data set.

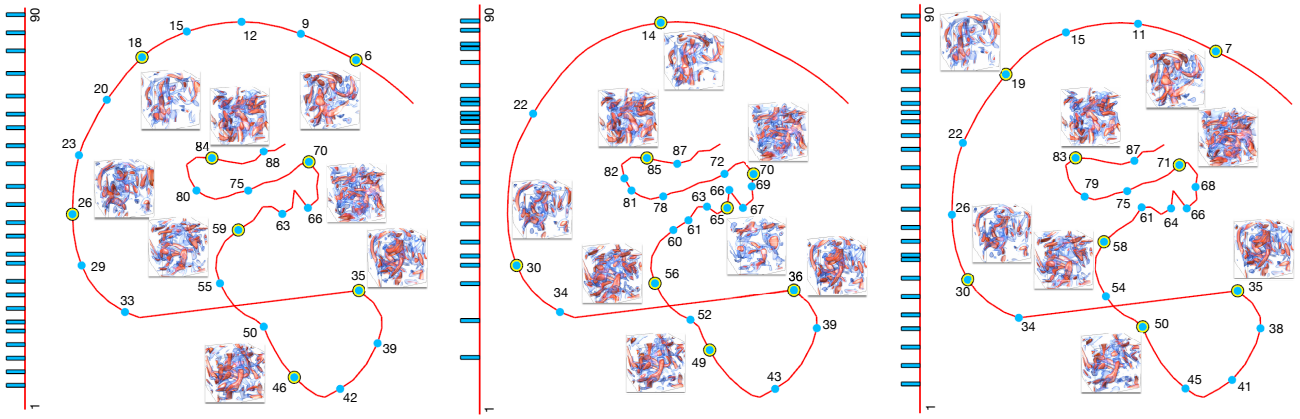
data set (variable)	ori. dimension ($x \times y \times z \times t$)	downsampled ($x \times y \times z$)
climate (temperature)	$360 \times 66 \times 27 \times 60$	$360 \times 66 \times 27$
earthquake (amplitude)	$256 \times 256 \times 96 \times 598$	$96 \times 96 \times 24$
combustion (CHI)	$480 \times 720 \times 120 \times 122$	$120 \times 180 \times 30$
combustion (HR)	$480 \times 720 \times 120 \times 122$	$120 \times 180 \times 30$
combustion (MF)	$480 \times 720 \times 120 \times 122$	$120 \times 180 \times 30$
ionization (He+)	$600 \times 248 \times 248 \times 100$	$150 \times 62 \times 62$
vortex (vorticity)	$128 \times 128 \times 128 \times 90$	$64 \times 64 \times 64$

4 Results

Data sets. We experimented with our approach using the data sets listed in Table 1. The climate data set is from a simulation of salinity and temperature in the equatorial region from 20°S to 20°N for a period of 100 years. This data set has 1200 time steps (one month per time step) and we used the first 60 of them and only the temperature variable. The earthquake simulation models the 3D seismic wave propagation of the 1994 Northridge earthquake. We used the amplitude scalar variable. The combustion data set comes from direct numerical simulation of temporally evolving turbulent non-premixed flames where combustion reactions occur within the two layers. These layers are initially thin planar layers and then evolve into complex structures as they interact with the surrounding turbulence. The simulation generates multiple variables and we used three of them: scalar dissipation rate (CHI), heat release (HR), and stoichiometric mixture fraction (MF). The ionization data set is made available through the IEEE Visualization 2008 Contest. The simulation is concerned with 3D radiation hydrodynamical calculations of ionization front instabilities for studying a variety of phenomena in interstellar medium such as the formation of stars. The simulation generates multiple variables and we used He+ mass abundance (He+). Finally, the vortex data set has been widely used in feature extraction and tracking. The data set comes from a pseudo-spectral simulation of vortex structures. We used the vorticity magnitude scalar variable.

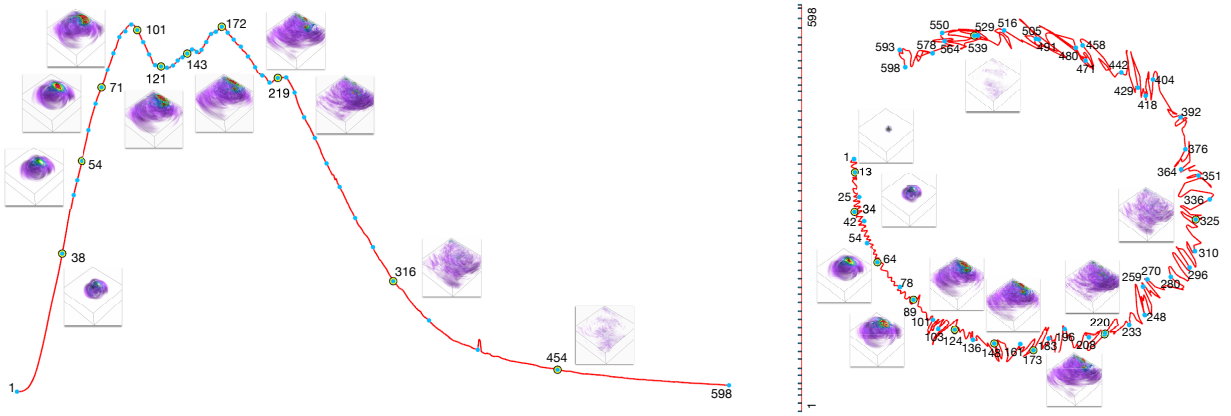
Training details. A single NVIDIA TITAN Xp 1080 GPU was used for network training. For data preprocessing, we use bicubic interpolation to downscale the volumes. This process can reduce the GPU memory requirement and speed up the training. we scaled the range of each downsampled volume to $[-1,1]$ and that of the output volume to $[-1,1]$. This is because the value range for the output of the final activation function $\tanh(\cdot)$ is $[-1,1]$. We used 80% of data for training. For optimization, we followed He et al. [5] to initialize parameters and applied the Adam optimizer [7] for parameter updates. We set one training sample per minibatch and trained the network for 100 epochs. It took anywhere from 1 (vortex) to 6 hours (earthquake) to train one data set. The training time is mainly determined by the number of time steps and the volume resolution. Using multiple variables does not significantly increase the training time. We run $2,000$ iterations when generating the t-SNE projection.

Comparison of different selections. In Figure 2, we compare three different time step selections: arclength-based selection, angle-based selection, and mixed selection. We can observe that the result of arclength-based selection is similar to that of uniform selection (i.e., selecting every ith time step), but it selects those time steps



(a) arclength-based, $\varepsilon = 0.05$ (b) shape-based, $\theta = \pi/4$ (c) mixed, $(\varepsilon, \theta, \alpha) = (0.04, \pi/5, 0.5)$

Figure 2: Comparing different ways of time step selection using the vortex data set. All select 24 time steps from 90 time steps. In the t-SNE projection, selected time steps are marked with blue dots and labeled with time step IDs, and those shown along with thumbnails are highlighted. We also indicate selected time steps along the linear vertical timeline.



(a) information-theoretic approach [12] (b) our approach, $(\varepsilon, \theta, \alpha) = (0.015, \pi/4, 0.6)$

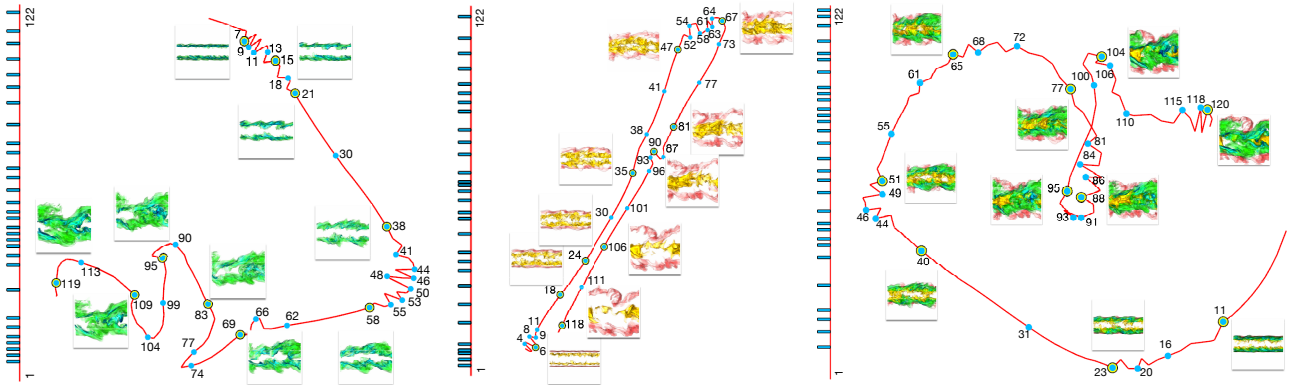
Figure 3: Comparing selected time steps using the earthquake data set. Both select 50 time steps from 598 time steps. In (a), the horizontal and vertical directions represent time step and importance, respectively.

that are close yet their distances in the t-SNE projection are large (for example, time steps 33 and 35). As for shape-based selection, it selects the representatives based on the change of accumulated angles. We can observe that when the change of accumulated angles is small, it will only select a few time steps (such as at the beginning of the sequence). However, when the change gets large, it will select time steps densely (such as the time period from time steps 60 to 70). Due to the drawbacks of these two selections, where the arclength-based selection fails to select time steps whose accumulated angles change rapidly and the angle-based selection samples time steps sparsely whose accumulated angles change slowly, we combine these two selections and present a mixed selection, as shown in Figure 2 (c). We can find that the mixed selection samples more densely at the beginning of the sequence compared to shape-based selection and it can also detect the time steps whose accumulated angles change rapidly, such as time steps 61, 64, and 66. Therefore, we opt to use the mixed selection to show our representative time steps in the following results.

Qualitative and quantitative analysis. To demonstrate the effectiveness of our approach, we show qualitative results and compare our method against the information-theoretic approach [12] using the earthquake data set. In Figure 3, we compare the representative time steps selected by our approach and information-theoretic approach. In Figure 3 (a), the importance score of each time step is plotted and 50 time steps are selected. As we can see, around 60% of the selected time steps are from time steps 70 to 220, since the conditional entropy peaks among these time steps. In Figure 3 (b), the t-SNE projection is shown and we also select 50 time steps from the sequence. We can observe that our approach selects the representatives more balanced over the sequence. In the t-SNE projection, the small amplitude of fluctuation at the early time steps is due to the steady increase of meaningful visual content as the

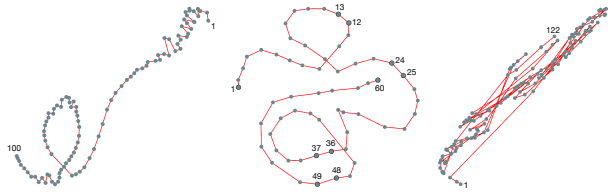
Table 2: Comparison of the average PSNR (in dB) and RMSE values. The best ones are highlighted in bold.

earthquake				combustion (HR)			
# rep.	approach	PSNR	RMSE	# rep.	approach	PSNR	RMSE
25	ours	**40.17**	0.00760	15	ours	**25.01**	0.145
	[12]	39.18	**0.00721**		[12]	24.91	**0.123**
	uniform	40.14	0.00778		uniform	24.97	0.136
50	ours	**42.11**	0.00589	30	ours	**27.91**	0.101
	[12]	41.10	**0.00489**		[12]	27.42	**0.093**
	uniform	42.05	0.00592		uniform	27.73	0.097
75	ours	**44.15**	0.00424	42	ours	**29.80**	**0.074**
	[12]	42.33	**0.00383**		[12]	29.01	0.075
	uniform	44.00	0.00483		uniform	29.65	0.077
100	ours	**46.40**	**0.00313**	62	ours	32.13	0.058
	[12]	43.21	0.00315		[12]	32.06	0.059
	uniform	46.10	0.00391		uniform	**32.42**	**0.056**



(a) HR, $(\varepsilon, \theta, \alpha) = (0.033, \pi/3, 0.6)$ (b) MF, $(\varepsilon, \theta, \alpha) = (0.033, \pi/5, 0.6)$ (c) HR+MF, $(\varepsilon, \theta, \alpha) = (0.035, \pi/4, 0.6)$

Figure 4: Comparing t-SNE projection and time step selection results under single and multiple variables using the combustion data set. All select 30 time steps from 122 time steps.



(a) (b) (c)

Figure 5: Comparing t-SNE projections using different types of time-varying data sets. (a) ionization, (b) climate, and (c) combustion (CHI). (a) to (c) are for regular, periodic, and turbulent types of data, respectively. Time steps are marked with black dots. In (b), both ends of the periods are highlighted (each period consists of 12 time steps).

earthquake shockwave quickly build ups. The first 200 time steps of the sequence have slightly more time steps selected than the later part of the sequence.

Furthermore, to quantitatively compare our approach against the information-theoretic approach [12] and uniform selection, we use the representative time steps to linearly interpolate the intermediate time steps and utilize the peak signal-to-noise ratio (PSNR) and root-mean-square error (RMSE) to evaluate the quality of the interpolated time steps. PSNR measures the peak error, whereas RMSE represents the cumulative error between the reconstructed and original volumes. The results with different numbers of representatives are reported in Table 2. For PSNR (RMSE), the higher (lower) the value, the better the quality. For the earthquake data set, we can see that our deep learning approach is the best in terms of PSNR across all cases reported here, while the information-theoretic approach is the best in terms of RMSE for three out of four cases. Our approach is the best in terms of both PSNR and RMSE when 100 time steps are selected. For the combustion (HR) data set, we can observe that our approach achieves the highest PSNR for three out of four cases, while the information-theoretic approach gets the lowest RMSE for two out of four cases. Our approach is the best in terms of both PSNR and RMSE when 42 time steps are selected.

In Figure 4, we show the representatives of single and multiple variables using the combustion data set. For the HR variable, we can observe that the distance between the beginning and subsequent time steps increases in the t-SNE projection, as shown in Figure 4 (a). This is because the two initially parallel layers get increasingly turbulent as the simulation goes. For the MF variable, there is a clear "U-turn" as shown in Figure 4 (b). The distance between beginning time steps and time steps before the U-turn keeps increasing while that between beginning time steps and time steps after the U-turn keeps decreasing. A likely explanation is that the yellow parts in the rendering get merged and then vanish while the red parts gradually move apart. In Figure 4 (c), we can see that the t-SNE projection of HR and MF variables preserves some patterns of individual variables. For example, the fluctuation from time steps 44 to 51 is similar to that from time steps 41 to 55 in the HR variable, while the pattern of distance change between the beginning and subsequent time steps is similar to that in the MF variable. This indicates that the projection of multiple variables "assimilates" the information from each individual variable, which confirms the meaningfulness of our approach for representative time step selection from time-varying multivariate data sets. We point out that we only show results with two variables here due to the limitation of volume rendering (we want to see all variables clearly in the rendering). The deep learning framework itself can handle three or more variables.

Further verification. In Figure 5, we compare t-SNE projections using different types (i.e., regular, periodic, and turbulent) of volumetric data to further verify the effectiveness of our approach. We point out the periodic (circular) pattern exhibited in the climate data set (Figure 5 (b)) and the turbulent (zigzag) pattern exhibited in the combustion (CHI) data set (Figure 5 (c)). The regular pattern exhibited by the ionization data set (Figure 5 (a)) is similar to what we observe in Figure 4 (c) for the HR and MF variables of the combustion data set. However, the CHI variable of the combustion data set (Figure 5 (c)) reveals more the truly turbulent nature of the data.

5 Conclusions and Future Work

We have presented a deep learning approach for selecting representative time steps from time-varying multivariate data sets. Using an autoencoder, our approach can automatically learn feature descriptors from volumetric data and across multiple variables in a latent space, although this process takes longer time than previous approaches. The learned features are used to guide the selection of representatives in the 2D projected space. We demonstrate the effectiveness of our approach and compare our time step selection result against those generated using an information-theoretic approach and uniform selection. In the future, we would consider the visual quality of interpolated intermediate time steps as a constraint for selecting representative time steps using deep reinforcement learning.

Acknowledgments

This research was supported in part by NSF grants IIS-1455886, CNS-1629914, DUE-1833129, and the NVIDIA GPU Grant Program. The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] S. Frey and T. Ertl. Flow-based temporal selection for interactive volume visualization. *Computer Graphics Forum*, 36(8):153–165, 2017.

[2] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of European Conference on Computer Vision*, pp. 484–499, 2016.

[3] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Proceedings o Advances in Neural Information Processing Systems*, pp. 2069–2077, 2014.

[4] J. Han, J. Tao, and C. Wang. FlowNet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2019. Accepted.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[7] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference for Learning Representations*, 2015.

[8] J. Liu, F. Yu, and T. Funkhouser. Interactive 3D modeling with a generative adversarial network. In *Proceedings of International Conference on 3D Vision*, pp. 126–134, 2017.

[9] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of International Conference on Machine Learning*, pp. 807–814, 2010.

[10] X. Tong, T. Lee, and H. Shen. Salient time steps selection from large scale time-varying data sets with dynamic time warping. In *Proceedings of IEEE Symposium on Large Data Analysis and Visualization*, pp. 49–56, 2012.

[11] L. J. P. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[12] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.

[13] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1059–1067, 2016.

[14] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *Proceedings of European Conference on Computer Vision*, pp. 766–782, 2016.

[15] B. Zhou and Y.-J. Chiang. Key time steps selection for large-scale time-varying volume datasets using an information-theoretic storyboard. *Computer Graphics Forum*, 37(3):37–49, 2018.