

Evaluation and Selection of Autoencoders for Expressive Dimensionality Reduction of Spatial Ensembles

Hamid Gadirov¹[0000–0001–6578–4342], Gleb Tkachev²[0000–0003–2515–5263],
Thomas Ertl²[0000–0003–4019–2505], and Steffen Frey^{1,2}[0000–0002–1872–6905]

¹ University of Groningen, 9712 CP Groningen, the Netherlands

² Universität Stuttgart, Keplerstraße 7, 70174 Stuttgart, Germany
{h.gadirov,s.d.frey}@rug.nl,
{gleb.tkachev,thomas.ertl}@vis.uni-stuttgart.de

Abstract. This paper evaluates how autoencoder variants with different architectures and parameter settings affect the quality of 2D projections for spatial ensembles, and proposes a guided selection approach based on partially labeled data. Extracting features with autoencoders prior to applying techniques like UMAP substantially enhances the projection results and better conveys spatial structures and spatio-temporal behavior. Our comprehensive study demonstrates substantial impact of different variants, and shows that it is highly data-dependent which ones yield the best possible projection results. We propose to guide the selection of an autoencoder configuration for a specific ensemble based on projection metrics. These metrics are based on labels, which are however prohibitively time-consuming to obtain for the full ensemble. Addressing this, we demonstrate that a small subset of labeled members suffices for choosing an autoencoder configuration. We discuss results featuring various types of autoencoders applied to two fundamentally different ensembles featuring thousands of members: channel structures in soil from Markov chain Monte Carlo and time-dependent experimental data on droplet-film interaction.

Keywords: Feature Learning · Machine Learning · Dimensionality Reduction · Clustering · Ensemble Visualization.

1 Introduction

Driven by technological advances, scientific ensembles of increasing size are obtained from simulations and experiments. They offer significant potential for new insights in various domains across engineering and natural sciences, but their analysis induces many challenges [37]. Dimensionality reduction (DR) techniques have been successfully applied for analyzing large sample collections (e.g., [17]), and especially 2D projections widely used to provide a visual impression of the data distribution [23]. However, when applied directly to spatial data, the expressiveness of projection techniques like Uniform Manifold Approximation and

Projection (UMAP) [24] is generally negatively impacted by the high dimensionality and hence sparsity of the data.

This work explores the usage of unsupervised feature learning techniques to produce more suitable data representations for DR of spatial data, and specifically, for 2D projections. We investigate standard and sparse *autoencoders* (AE), as well as more advanced versions such as *Sliced-Wasserstein* and β -*Variational* autoencoders (SWAE and β -VAE). Our study with two different scientific ensembles demonstrates that they improve expressiveness in comparison to directly projecting the spatial data via UMAP. However, it also shows that the performance of autoencoder variants is highly data-dependant, i.e., it is not clear a priori which one to choose to adequately capture what is of interest in the data. To address this, we propose to employ complementary metrics quantifying the quality of a projection and selecting a specific autoencoder variant based on Pareto efficiency. These metrics assess the quality of the projection based on labels (generally provided by an expert). While labels can be prohibitively expensive to obtain for the full ensemble, we demonstrate that a small subset of labeled members is already sufficient to yield expressive results.

We consider (1) the study of several autoencoder variants for dimensionality reduction with diverse scientific ensembles, (2) the evaluation of projection metric stability for small partial labelings, and (3) the Pareto-efficient selection of a variant on this basis to be the main contributions of this work.

2 Related Work

Ensemble visualization. The analysis of ensemble data generally is a challenging visualization task [26]. Potter et al. [29] as well as Sanyal et al. [33] proposed early approaches to study climate ensembles, while Waser et al. [38] described a system for the interactive steering of simulation ensembles. Kehrner et al. [17], Sedlmair et al. [34], and Wang et al. [37] provided detailed surveys of techniques in the area. Bruckner and Möller [2] employ squared differences to explore the visual effects simulation space, Hummel et al. [15] compute region similarity via joint variance, and Kumpf et al. [21] track statistically-coherent regions using optical flow. Hao et al. [9] calculate shape similarities for particle data using an octree structure, while He et al. [10] employ surface density estimates for distances between surfaces. Fofonov et al. [3] propose fast isocontour calculation for visual representation of ensembles.

For projection of high-dimensional data, Vernier et al. [36] propose spatial and temporal stability metrics to evaluate the quality of PCA, t-SNE, UMAP, and Autoencoders. Bertini et al. [1] presented a systematic analysis of quality metrics supporting exploration. We, however, study how autoencoders impact the 2D projection quality when combined with traditional DR techniques.

Autoencoder-based Feature Extraction. Several works confirm the ability of autoencoders to extract expressive features for ensemble data. Hinton et al. [14] first demonstrated that autoencoders can be utilized for DR and can be applied to large datasets. Plaut [28] performed principal component analysis using

a linear autoencoder. Han et al. [8] developed an autoencoder-based framework *FlowNet* to extract such features as streamlines and stream surfaces. Jain et al. [16] utilized deep convolutional autoencoder to obtain a compact representation of multivariate time-varying volumes by learning high-level features. Lekschas et al. [22] developed a convolutional autoencoder-based technique *PEAX* for interactive visual pattern search. Guo et al. [7] developed a deep convolutional autoencoder minimizing the reconstruction and clustering losses for end-to-end learning of embedded features for clustering. Ge et al. [4] achieved state-of-the-art clustering performance on MNIST via dual adversarial autoencoders. He et al. [11] proposed a deep learning approach for comparison of multiple ensembles. Guo et al. [6] developed a visual analytics system based on autoencoders for medical records. Way et al. [39] utilize variational autoencoders to extract biologically relevant features from gene expression data. We also employ deep convolutional autoencoders to enable visual exploration of scientific data, but focus on a study of autoencoder variants and model selection in a partially labeled scenario.

3 Study Setup, Metrics and Selection

Standard DR techniques, such as PCA, t-SNE, and UMAP, lose efficiency when applied directly to high dimensional (ensemble) data (e.g. [14], [39]). To address this, we first reduce the dimensionality of ensemble data with autoencoders and then construct a 2D projection. An autoencoder is a neural network for unsupervised learning of efficient data encodings. It consists of an encoder followed by a decoder, with the former compressing the input and the latter trying to reconstruct it as accurately as possible. Different architectures are employed below, and evaluated both visually and quantitatively with metrics. These metrics also provide the basis for Pareto efficient selection.

Scientific Ensemble Datasets. We consider two ensemble datasets in our study. The first dataset depicts channel structures in soil from Markov chain Monte Carlo, consisting of independent members generated during simulation [31]. The images are monochrome and have a resolution of 50×50 . In total, there are 95K images. The second dataset Drop Dynamics stems from a physical experiment to study the impact of a droplet with a film [5]. The captured experiment images, similarly to the previous dataset, are monochrome and in this case have a resolution of 160×224 . In total, there are 135K images from 1K members. Subsets of the members of both ensembles are exemplified in the grid views in Fig. 2. A subset from both datasets was manually labeled, as required by the metrics that we use to evaluate the projections. Labeling is based on different behavior types observed in ensembles and was performed by marking groups of images with similar behavior types as one class label. In the labeling process a number of randomly selected members was considered. MCMC consists of five categorical classes (● ● ● ● ●) which depict qualitatively different types of channel structures. For Drop Dynamics, there are eight classes in total: “bubble”, “bubble-splash”, “column”, “crown”, “crown-splash”, “splash”,

“drop”, and “none” (not fitting any other category). 2.5K labeled images are available in total for MCMC and 7.2K for Drop Dynamics. On both projection views (Fig. 1 and Fig. 3) gray points ($\bullet$) indicate unlabeled images.

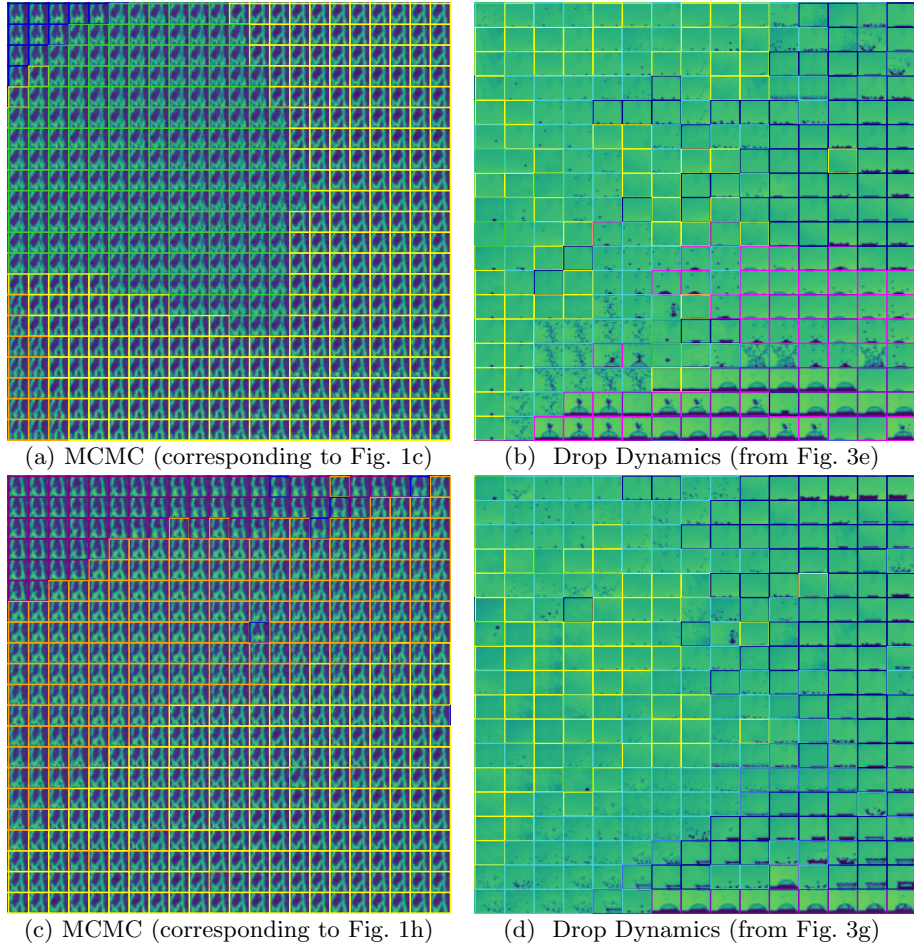


Fig. 2: Grid views with coloured frames around each image representing labels. The new position in the grid was found via linear assignment from a 2D projection to the grid [30].

Autoencoder-based Feature Extraction. Besides standard and sparse autoencoders (AE), we utilize variational autoencoders (VAE) [19], based on a variational inference and learning algorithm, as well as its constrained version (β -VAE) [12]. VAE is scalable for large datasets and its inherent regularization brings latent vectors closer together. SWAE [20] is based on Wasserstein Autoencoders [35], which share properties of VAE while achieving better reconstruction. All types of considered autoencoders (AE, SWAE, VAE and β -VAE) have a symmetric structure - the decoder is reversed to the encoder. Both parts

of autoencoders are represented via deep neural networks, containing several hidden *convolutional* and *fully connected* layers. The objective function used for training varies depending on the autoencoder type (see supplemental material for details). In our implementation of AEs and (β)-VAEs, we reduce the resolution of input images by half after each convolutional layer by using a *stride* of 2 in each dimension of spatial (2D) convolutions. For the spatio-temporal (3D) convolutions, a stride of 3 was used. The *kernel size* for convolution was set to 3 for each dimension, the *number of filters* of 64 in each hidden convolutional layer was used with zero *padding*. As an optimizer, *Adam* [18] with learning rate 0.0005 was employed throughout. The “ReLU” activation function was used in combination with random uniform weight initialization. We utilize L1 and L2 regularization for *Sparse Autoencoders*. We follow the implementation of SWAE from [20] utilizing “LReLU” activation and *average pooling*.

For MCMC dataset with no temporal information only 2D convolutional models were used, trained and validated on 20K unlabeled images. For the (spatio-temporal) Drop Dynamics, both 2D and 3D autoencoders were used, trained and validated on 15K unlabeled images. Note that we utilize unsupervised machine learning and projection techniques, labels are solely employed for the purpose of evaluation in this work (also see the discussion of metrics below).

Projection to 2D Space. After transforming the ensemble data from physical to feature space, we obtain a latent vector for each data sample. These feature vectors are subsequently projected to a 2D space using DR techniques. In general, we find that directly reducing the dimensionality of ensembles to 2D using autoencoders is inefficient for most of the models since the autoencoder cannot reasonably reconstruct the input which is an indication of poorly learned features (although there are exceptions, see discussion below). In the following, we restrict ourselves to UMAP projection, which outperformed other DR techniques such as t-SNE and PCA in preliminary experiments with the same subset of labels. UMAP is a non-linear technique and uses a smoothed version of k-nearest neighbors distance. We utilize UMAP with *min. distance = 1.0* parameter, deviating from the default. This value controls how close points are located on the newly created 2D map and produces visually less overlapping projections.

Metrics (and partially labeled data). We use two commonly used complementary metrics to capture the projection quality: *neighborhood hit* and *silhouette*. While neighborhood hit measures how well the clusters are separated, silhouette additionally provides the measure of tightness. The *neighborhood hit* metric [36] is based on k-nearest neighbors [27] ($k = 17$ in our setting, values ranging from 7 to 27 yield similar results), and since it computes the fraction of neighbors belonging to the same class for each labeled data point, it provides an accurate measure of the separation and was preferred to other similar metrics. The output is in the range of $[0, 1]$, where *higher values* represent a better separation of the clusters. The *silhouette* metric [32] is based on (Euclidean) distance and computes the distances between a data point and all data points in the same cluster. Since it takes into account distances to all data points in the nearest neighboring cluster as well, it still reaches a high score in the case of

well separated friend-of-friends clusters and therefore was preferred to other similar metrics. The output is in the range of $[-1, 1]$, where *higher values* represent a better matching of a data point to its own cluster rather than to neighboring clusters. Note that we also considered other metrics (Calinski-Harabasz and Davies-Bouldin) which, however, yielded similar results to the silhouette metric (see supplemental material).

Pareto efficient Selection. We identify the best performing feature extraction models based on multi-objective optimality criteria (Pareto frontier). For this, from our two considered metrics we construct a 2D plot, demonstrating the results of evaluating the clustering quality of final projections, with the axes of neighborhood hit and silhouette (maximum is better) metrics. After obtaining the Pareto frontier, it is possible to select one of the best shown models, check its corresponding visualization, and investigate the projection of ensemble data.

4 Evaluation

We now discuss 2D projection results of our ensembles, demonstrate Pareto-efficient selection, and evaluate metric stability with labeled subsets (see supplemental material for 2D projections of all variants).

MCMC (Fig. 1). In the baseline method that uses UMAP directly (a), we can observe well separated but dispersed clusters of each class (reflected by high neighborhood hit but low silhouette). The projection results improved for all classes with the 2D sparse autoencoder (b). Note that the number after the autoencoder type indicates the dimensionality of the latent space. However, classes partitioned into different clusters and yielding highly irregular shapes can still be observed. This is reflected by low values in the silhouette which position AE results between the baseline and β -VAEs in Fig. 1j. For SWAE results (c and d), we observe an improvement with all classes being more tightly clustered. Both metrics are improved because of the influence of the SWAE objective function. Interestingly, SWAE with direct projection to 2D (not using UMAP) also yields a good projection (c). 2D VAE (e) and 2D $\beta(2)$ -VAE (f) further improve the results. We see well-separated clusters and even higher neighborhood hit and silhouette values. Models like 2D $\beta(4)$ -, $\beta(8)$ -, and $\beta(10)$ -VAE (g, h, and i) create clusters in a shape of a Gaussian distributions, further improving metric scores. These are also on the Pareto frontier (Fig. 1j). Interestingly, $\beta(8)$ -VAE with a comparably low latent space dimensionality of 32 is able to extract features properly. Overall, we observe a significant benefit in performing feature extraction on the MCMC ensemble. Visualizations show that AE, SWAE, and (β)-VAE with different β values all outperform the baseline. In the final projection of almost all models, we see data points belonging to the same class located close to each other, forming clusters of the same classes. In the case of β -VAE, properly selected values of β (not too high) can improve the results and lead to visually pleasing projections.

Drop Dynamics (Fig. 3). The baseline result (a) successfully produces clusters of similar samples and thus achieves a relatively high neighborhood

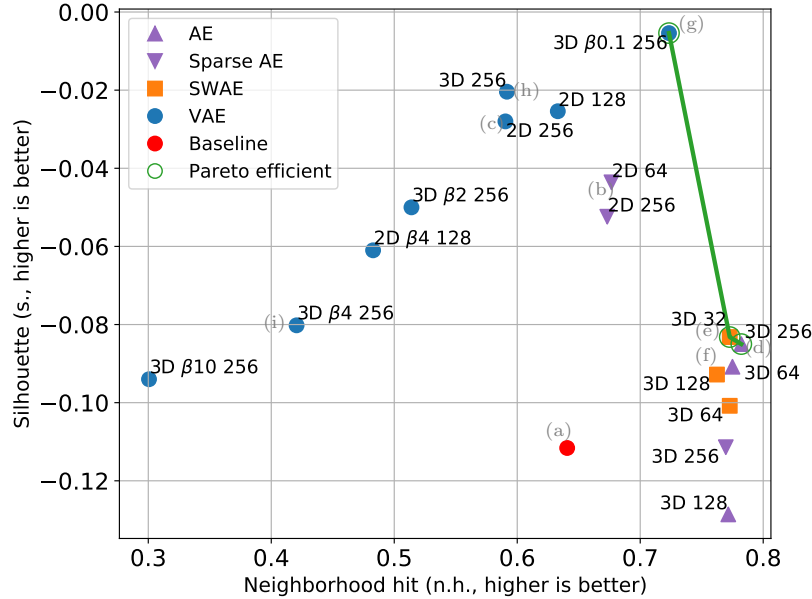
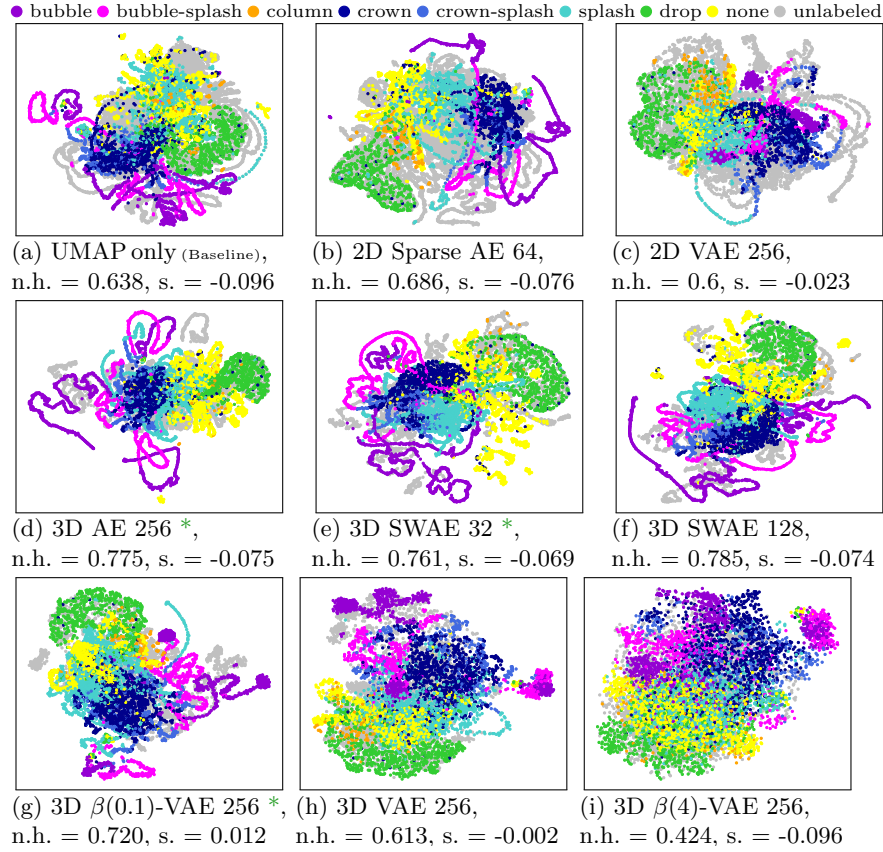


Fig. 3: (a–i) Projection views of the Drop Dynamics ensemble (* indicates the projection belonging to a set of Pareto efficient models in (j), see discussion in Sec. 4). (j) Metric values of projection and respective Pareto frontier (marked in green, links to projections in gray).

hit. This is clearly noticeable in the case of classes “bubble”, “bubble-splash”, “splash”, and time steps without any behavior category (“none”). However, it scatters the samples of the same type across many small clusters, which is highlighted by the poor silhouette values. In the case of simple 2D Sparse AE with latent dimensionality 64 (b), the model also cannot group all data points belonging to the same class into one cluster. We can see the “bubble”, “bubble-splash”, and “splash” type time steps present in different parts of the final projection, which is based on the latent space learned by the autoencoder. In (c), clusters start to appear in the form of Gaussian distributions (“bubble”), due to the KL divergence regularization term in the VAE loss calculation, improving the silhouette score. In (d), corresponding to the 3D AE, we can notice better connected clusters in the case of “crown” and “drop” classes, which is reflected in the neighborhood hit score. Elongated clusters still persist, since there is no KL regularization applied in this case. In (e) and (f) corresponding to the SWAEs, most clusters are separated from each other, but still have multiple subclusters. This is related to WAE loss calculation: different latent codes remain far away from each other. In (g) corresponding to the $\beta(0.1)$ -VAE with latent dimensionality of 256, we can notice better connected clusters in the case of “crown”. In the case of “bubble”, clusters are starting to appear in the form of a Gaussian distribution. This is reflected in the higher neighborhood hit and silhouette scores. In (h) corresponding to the VAE ($\beta = 1$), likewise decent projections can also be observed. Due to the KL divergence term in the VAE loss calculation, the 3D VAE model created clusters in the form of Gaussian distributions. It can be noticed e.g. in the case of “bubble” type time steps, no elongated clusters can be observed. In models with higher β values greater effect of KL loss can be observed. The influence of β trades off neighborhood hit for silhouette metric, helping to bring points of the same class closer, but also mixing some clusters. Such trade-offs are why we use Pareto optimality for our model selection (Fig. 3j). However, too high values of β (e.g. $\beta \geq 4$) mix the data points in the final visualization because the input has only a minor impact on the latent vector, which leads to poorly learned features (Fig. 3i).

Overall, we see that the autoencoder-based feature extraction can improve the baseline results regarding the metrics, which capture important characteristics under the presence of a large chunk of unlabeled (previously unseen) data in particular. We also note that most 3D convolutional models outperform models with 2D convolutions. 3D models were able to learn better features by using three time steps in the input instead of one, without the need to increase the dimension of the latent space.

Pareto efficient selection (Fig. 3j and Fig. 1j). As can be seen in Fig. 1j, corresponding to MCMC, a significant improvement was achieved with (β -)VAEs and SWAE over the baseline (•). The most efficient are β -VAE models with the values of β ranging from two to eight and SWAE with direct 2D projection. As can be seen in Fig. 3j, corresponding to Drop Dynamics, an improvement was achieved with 2D/3D AEs, 3D SWAEs, 3D VAE. The Pareto efficient models, connected with green line, are 3D AE, 3D SWAE, and 3D $\beta(0.1)$ -VAE.

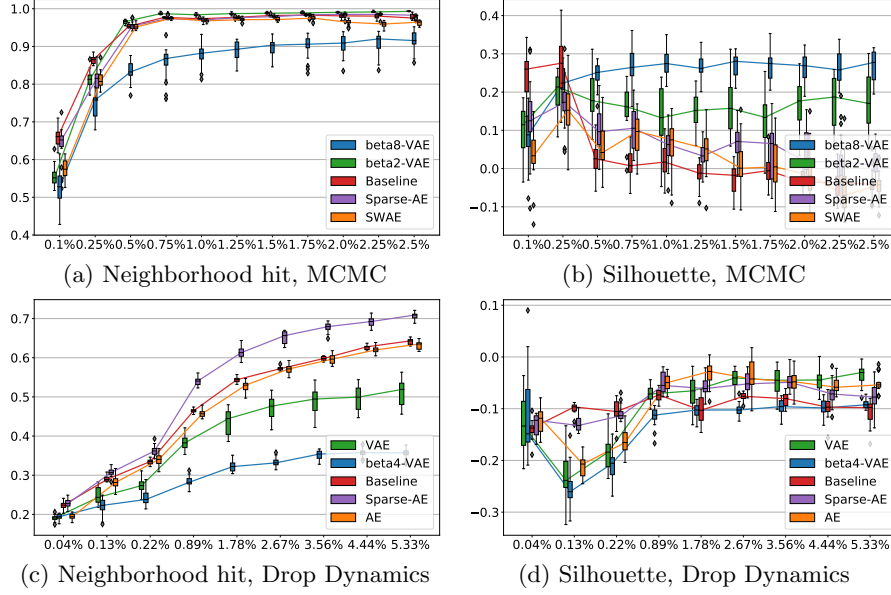


Fig. 4: Stability of metrics using labeled subsets, replicated on 20 runs.

Stability of Metrics using Labeled Subsets (Fig. 4). Since we utilize only a small subset of the labeled data which supports the selection of autoencoder variants, it is crucial to test the stability of the results. In Fig. 4, we demonstrate that the considered models produce stable metric scores for different numbers of labels. We use random label subsets of different sizes, ranging from 0.1% to 2.5% for MCMC and from 0.04% to 5.33% for Drop Dynamics. We can see that as we increase the amount of labels for both datasets, the metric values are converging with low uncertainty. In contrast, a small amount of labels leads to a high uncertainty in metric values. When the number of labels is critically small, (c), the neighborhood hit metrics breaks down and produces low values and very low uncertainty because of the label sparsity. In this case, the neighborhood includes points from far away and the metric converges to the value of $1/\text{number-of-classes}$ because it encounters points from all the classes. Overall, this stability study suggests that it is possible to achieve representative results utilizing a comparably small percentage of labeled data points.

5 Discussion and Outlook

The premise of this work is that unsupervised feature learning prior to dimensionality reduction with techniques like UMAP improves the results for spatial and spatiotemporal ensembles. The rationale is that higher-level features extracted via autoencoders from the field data yields a representation that better conveys characteristic differences between them, which are relevant for analyzing the ensemble. We demonstrate this by showing that all autoencoder variants

yield superior results when considering the manually assigned labels that correspond to the characteristic cases.

The properties of the autoencoders are directly reflected in the projection. Most prominently, the Kullback-Leibler divergence term of (β)-VAE causes the clusters to have a Gaussian distribution, with the influence depending on the Lagrangian multiplier β . We initially aimed to provide general suggestions on which autoencoder architecture is the most suitable, but our experiments showed that the results are highly data-dependent. While not evaluated in this paper, we also found that simple modifications of the datasets, such as normalization and cropping can have a significant influence. This is due to the fact that especially methods with an MSE reconstruction objective encode noisy, randomly varying, and unimportant features presented in the input. To distinguish which differences are meaningful, one requires human input, which we took in the form of labels. In this work, we propose to use suitable projection metrics and a Pareto optimality to guide the selection. Crucially, as scientific datasets typically come without labeling and manual annotation is time-intensive, we were able to show that basing the decision on a small subset already suffices for an informed selection. However, for a practical application to a novel ensemble without labeling, we still consider our analysis provided in this work useful to serve as an initial basis for model selection. According to our observations, if the images do not contain a high proportion of useful information (i.e., a large portion of the pixels relevant for feature extraction), models like AEs, SWAEs, or VAEs with small values of β (<1) are beneficial. Otherwise, VAEs with higher values of β (>1) can achieve a better clustering results.

For MCMC with a high proportion of relevant elements in the data, autoencoders achieve a more significant improvement over the baseline in comparison to the Drop Dynamics ensemble (where the majority of pixels just represent background). To further improve such cases in particular, in future work, we aim to investigate other unsupervised learning approaches besides autoencoders or generative models with a semi-supervised setup. The robustness and general performance of the proposed pipeline could further benefit from adequate prior preparation of the data, e.g., via noise reduction or segmentation. For β -VAEs it was challenging to find the best Lagrangian multiplier β , rather than to employ a fixed value, and so it might be better to gradually increase the β -weighted KL term during training in order to achieve both disentangled representation and high reconstruction quality. Finally, 2D projection is just one prominent example where prior feature learning is beneficial for visual analysis, and we aim to explore further scenarios in future work like clustering and search.

References

1. Bertini, E., Tatu, A., Keim, D.: Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics* **17**(12), 2203–2212 (2011)

2. Bruckner, S., Möller, T.: Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics* **16**(6), 1468–1476 (2010)
3. Fofonov, A., Molchanov, V., Linsen, L.: Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE transactions on visualization and computer graphics* **22**(8), 2037–2050 (2015)
4. Ge, P., Ren, C.X., Dai, D.Q., Feng, J., Yan, S.: Dual adversarial autoencoders for clustering. *IEEE transactions on neural networks and learning systems* **31**(4), 1417–1424 (2019)
5. Geppert, A., Chatzianagnostou, D., Meister, C., Gomaa, H., Lamanna, G., Weigand, B.: Classification of impact morphology and splashing/deposition limit for n-hexadecane. *Atomization and Sprays* **26**(10) (2016)
6. Guo, R., Fujiwara, T., Li, Y., Lima, K.M., Sen, S., Tran, N.K., Ma, K.L.: Comparative visual analytics for assessing medical records with sequence embedding. *Visual Informatics* **4**(2), 72–85 (2020)
7. Guo, X., Liu, X., Zhu, E., Yin, J.: Deep clustering with convolutional autoencoders. In: *International conference on neural information processing*. pp. 373–382. Springer (2017)
8. Han, J., Tao, J., Wang, C.: Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE transactions on visualization and computer graphics* (2018)
9. Hao, L., Healey, C.G., Bass, S.A.: Effective Visualization of Temporal Ensembles. *IEEE Transactions on Visualization and Computer Graphics* **22**(1), 787–796 (Jan 2016). <https://doi.org/10.1109/TVCG.2015.2468093>
10. He, W., Guo, H., Shen, H.W., Peterka, T.: efesta: Ensemble feature exploration with surface density estimates. *IEEE transactions on visualization and computer graphics* **26**(4), 1716–1731 (2018)
11. He, W., Wang, J., Guo, H., Shen, H.W., Peterka, T.: Cecav-dnn: Collective ensemble comparison and visualization using deep neural networks. *Visual Informatics* **4**(2), 109–121 (2020)
12. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework (2016)
13. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr* **2**(5), 6 (2017)
14. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *science* **313**(5786), 504–507 (2006)
15. Hummel, M., Obermaier, H., Garth, C., Joy, K.I.: Comparative visual analysis of lagrangian transport in cfd ensembles. *IEEE Transactions on Visualization and Computer Graphics* **19**(12), 2743–2752 (2013)
16. Jain, S., Griffin, W., Godil, A., Bullard, J.W., Terrill, J., Varshney, A.: Compressed volume rendering using deep learning (2017)
17. Kehrner, J., Hauser, H.: Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE transactions on visualization and computer graphics* **19**(3), 495–513 (Mar 2013). <https://doi.org/10.1109/TVCG.2012.110>
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)

20. Kolouri, S., Pope, P.E., Martin, C.E., Rohde, G.K.: Sliced-wasserstein autoencoder: An embarrassingly simple generative model. arXiv:1804.01947 (2018)
21. Kumpf, A., Rautenhaus, M., Riemer, M., Westermann, R.: Visual analysis of the temporal evolution of ensemble forecast sensitivities. *IEEE transactions on visualization and computer graphics* **25**(1), 98–108 (2018)
22. Lekschas, F., Peterson, B., Haehn, D., Ma, E., Gehlenborg, N., Pfister, H.: Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning. In: *Computer Graphics Forum*. vol. 39, pp. 167–179. Wiley Online Library (2020)
23. Liu, S., Maljovec, D., Wang, B., Bremer, P.T., Pascucci, V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE transactions on visualization and computer graphics* **23**(3), 1249–1268 (2016)
24. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
25. Ng, A., et al.: Sparse autoencoder. *CS294A Lecture notes* **72**(2011), 1–19 (2011)
26. Obermaier, H., Joy, K.I.: Future challenges for ensemble visualization. *IEEE Computer Graphics and Applications* **34**(3), 8–11 (2014)
27. Peterson, L.E.: K-nearest neighbor. *Scholarpedia* **4**(2), 1883 (2009)
28. Plaut, E.: From principal subspaces to principal components with linear autoencoders. arXiv preprint arXiv:1804.10253 (2018)
29. Potter, K., Wilson, A., Bremer, P.T., Williams, D., Doutriaux, C., Pascucci, V., Johnson, C.R.: Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data. In: *In Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*. pp. 233–240 (2009). <https://doi.org/10.1109/ICDMW.2009.55>
30. Quadrianto, N., Song, L., Smola, A.J.: Kernelized sorting. In: *Advances in neural information processing systems*. pp. 1289–1296 (2009)
31. Reuschen, S., Xu, T., Nowak, W.: Bayesian inversion of hierarchical geostatistical models using a parallel-tempering sequential gibbs mcmc. *Advances in Water Resources* **141**, 103614 (2020)
32. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* (1987)
33. Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., Moorhead, R.: Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty. *IEEE Transactions on Visualization and Computer Graphics* **16**(6), 1421–1430 (Nov 2010). <https://doi.org/10.1109/TVCG.2010.181>
34. Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., Möller, T.: Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics* **20**(12), 2161–2170 (2014)
35. Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558 (2017)
36. Vernier, E.F., Garcia, R., da Silva, I., Comba, J.L.D., Telea, A.C.: Quantitative evaluation of time-dependent multidimensional projection techniques. arXiv preprint arXiv:2002.07481 (2020)
37. Wang, J., Hazarika, S., Li, C., Shen, H.W.: Visualization and visual analysis of ensemble data: A survey. *IEEE transactions on visualization and computer graphics* **25**(9), 2853–2872 (2018)
38. Waser, J., Fuchs, R., Ribicic, H., Schindler, B., Blochl, G., Groller, E.: World Lines. *IEEE Transactions on Visualization and Computer Graphics* **16**(6), 1458–1467 (Nov 2010). <https://doi.org/10.1109/TVCG.2010.223>
39. Way, G.P., Greene, C.S.: Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *BioRxiv* p. 174474 (2017)