

IDLat: An Importance-Driven Latent Generation Method for Scientific Data

Jingyi Shen, Haoyu Li, Jiayi Xu, Ayan Biswas, and Han-Wei Shen, *Member, IEEE*

Abstract—Deep learning based latent representations have been widely used for numerous scientific visualization applications such as isosurface similarity analysis, volume rendering, flow field synthesis, and data reduction, just to name a few. However, existing latent representations are mostly generated from raw data in an unsupervised manner, which makes it difficult to incorporate domain interest to control the size of the latent representations and the quality of the reconstructed data. In this paper, we present a novel importance-driven latent representation to facilitate domain-interest-guided scientific data visualization and analysis. We utilize spatial importance maps to represent various scientific interests and take them as the input to a feature transformation network to guide latent generation. We further reduced the latent size by a lossless entropy encoding algorithm trained together with the autoencoder, improving the storage and memory efficiency. We qualitatively and quantitatively evaluate the effectiveness and efficiency of latent representations generated by our method with data from multiple scientific visualization applications.

Index Terms—Latent space, scientific data representation, deep Learning

1 INTRODUCTION

As machine learning techniques become increasingly more ubiquitous for scientific visualization and analysis, latent representations generated by autoencoders have attracted great attentions of researchers in recent years. Latent representations have been successfully demonstrated to retain essential information in the original data, and can be used for similarity analysis [11, 12, 18, 25, 28], generation of visualizations [6], synthesis of simulations [22, 42, 43], data reductions [26, 44], and have been applied to multivariate volumetric data [28], streamlines and stream surfaces [18], isosurfaces [12], and particles [25].

Although latent representations for large-scale scientific data have been used extensively, there are still several challenges. First, domain scientists have diverse interests in different data portions, but latent representations trained using unsupervised approaches have limited support for incorporating such domain interests. Given that scientific data complexity varies across space and time [14], domain scientists' interests should be taken into account during latent generation so that it is possible to perform importance-driven scientific data explorations as well as to reduce data that are not deemed important. To the best of our knowledge, related works only support generating latent representations associated with simulation parameters [22, 43], time [42], and aggregated queries [41]. Second, how to represent diverse domain interests in a unified way for latent generation is non-trivial. Domain interest in scientific visualization can be defined in many ways, either mathematically related to physical attributes or spatially/temporally related to particular ranges [16, 32–34, 36]. A generalized representation is required to incorporate different types of scientific interests. Third, the costs of importance-driven latent generation can be high. Previous latent representations are tightly coupled with specific scientific visualization applications [12, 18, 28]. If scientists change their interests during exploration, re-training the model will be needed but can be prohibitively expensive. Also, current latent generation methods cannot adapt the size of the latent to the domain interest once neural network architecture is determined, leading to high storage and I/O costs.

In this paper, to generate latent representations guided by scientific

interests, we propose an **Importance-Driven Latent** generation method (**IDLat**) based on a convolutional autoencoder to combine the power of the convolution operations for extracting local features and the autoencoder for representation learning [5]. First, to incorporate domain interests into latent representations, we extend the basic autoencoder with a feature transformation network that takes domain interest as an input to guide the mapping from scientific data to latent representations. Second, based on the proposed network, we represent various types of domain interests with discretized spatial importance maps. Every element in the importance map is a real value indicating how vital this spatial location is when generating the latent representation. The importance values can be derived mathematically based on the domain or heuristically based on distances, distributions, locations, etc., depending on the underlying scientific applications. With the location-wise control of spatial importance, we can flexibly represent various types of scientific interests and use them to guide latent generation. Third, our model only needs to be trained once for each dataset, and used even when scientists change the definition of importance. The produced latents are optimized in storage size with the help of feature transformation networks and a lossless entropy encoding module. The motivation for jointly pursuing importance-driven latent and compression can be summarized into two aspects. The first is to further reduce the storage cost of scientific data based on its importance. Latent representations are compact, but their sizes are determined by the network architecture, not the amount of information according to domain interest. To optimize the usage of storage, we quantize and compress latent with importance taken into account, i.e., reduce the size of latent for unimportant data. The second is to improve the effectiveness of latent in representing scientific features. The original data may contain unimportant information such as noise or non-feature regions which compromise latent's ability to represent features. However, with importance control and entropy constraint in the latent space, the model will optimize the utilization of limited latent dimensions by preserving more important information and sacrificing the unimportant information. As a result, each latent is instructed to encode important information effectively.

Our latent generation workflow is as follows. First, spatial importance maps are generated based on scientific interests. Second, both the original scientific data and importance maps are taken as input to our model, which produces latent representations controlled by the importance map. Third, we quantize the generated latents into discrete symbols. Fourth, given that the entropy of discrete latents will be different under different importance settings, we apply lossless entropy encoding on the discrete latent vectors to further reduce the latent size. After the model is trained, we support visualization and analysis in both latent space and data space. In latent space, the discrete latent representations are losslessly recovered through entropy decoding for scientific analysis such as similarity comparison and feature exploration. In data

Jingyi Shen, Haoyu Li, Jiayi Xu and Han-Wei Shen are with the Department of Computer Science and Engineering, The Ohio State University. E-mail: {shen.1250, li.8460, xu.2205 and shen.94}@osu.edu.

Ayan Biswas is with Los Alamos National Laboratory. E-mail: ayan@lanl.gov.

Manuscript received 31 March 2022; revised 1 July 2022; accepted 8 August 2022.
Date of publication 27 September 2022; date of current version 2 December 2022.
This article has supplementary downloadable material available at <https://doi.org/10.1109/TVCG.2022.3209419>, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2022.3209419

space, the discrete symbols are further decoded to obtain reconstructed data through the autoencoder's decoder.

Our latent representation is useful for scientific visualization and analysis due to its compactness and effectiveness in preserving domain interests. Each latent representation is forced to focus on representing data of interest instead of all details of the raw data, which amplifies the more salient information and reduces the effect of noise, resulting in more salient and robust data representations. By transforming data into compact latent representations, similarity comparison or distance computation between data becomes efficient and robust. Also, it reduces the storage cost by only saving the compressed latent representations that can later be used for downstream scientific analysis tasks such as projection, retrieval, feature exploration, clustering, query, etc.

We qualitatively and quantitatively evaluate the usefulness and effectiveness of our importance-driven latent representations through data reconstruction and latent space exploration tasks on three scientific datasets. In summary, the contributions of our work are threefold:

- First, we present a novel and flexible pipeline for generating importance-driven scientific data representation with an autoencoder model.
- Second, we utilize a location-based importance map to incorporate domain interests into the generation of latent representation.
- Third, we further reduce the size of latent representation through entropy encoding to reduce the I/O and storage costs.

2 RELATED WORKS

Our study makes use of a deep learning based latent representation for importance-driven data visualization and analysis. We summarize the related works of these two fields.

2.1 Latent Representations in Scientific Visualization

In scientific visualization, there are three main usages of autoencoders. The first, and the most related one to our work, is to use autoencoders for user-controlled data synthesis. Wiewel et al. [43] converted raw volume data into latent representations and controlled data properties such as velocity or density through different latent dimensions. Berger et al. [6] proposed to learn the mapping from transfer functions to rendered volumes with an encoder-decoder architecture. By traversing the latent space and generating rendered images of the volume under various viewpoints and transfer functions, scientists can get a better understanding of the volume features efficiently. Kim et al. [22] proposed a latent space integration network to learn the mapping of latent representations from the current time step to the next time step. Second, latent representations are also used as feature descriptors of the raw data to select representatives. FlowNet [18] proposed to identify representative flow lines or surfaces in the lower dimensional latent space by applying density-based clustering on latent representations. To select representative time steps for volumetric time-varying data, instead of using handcrafted features, Porter et al. [28] adopted autoencoders to learn a representation for each volume and selected representations in the t-SNE projection. The third usage of autoencoders is data reduction. AE-SZ [26] and multi-branch decoder network [44] demonstrate the effectiveness of autoencoders for scientific data reduction.

However, existing autoencoder-based works assume every data element is equally important without considering scientists' interest when generating the latent. Also, from a data reduction point of view, knowing which region scientists have low interests and thus can afford to have a lower quality will help achieve a better trade-off between the size and the quality of the latent representation. Therefore, we extend the basic autoencoder into one conditioned on user interest.

2.2 Importance-Driven Visualization and Analysis

For different scientific applications, it is well advised to consider the varying importance throughout the dataset during visualization and analysis. Importance-driven techniques can be classified into two categories: with and without direct user interaction.

Studies that involve user interaction usually require users to decide the importance. Driven by the visualization goal, Peng et al. [27] proposed to define mesh importance using transfer functions for interactive

isosurface rendering. Burger et al. [9] proposed to control the shape and density of particles so that scientists can focus on the important regions where the region of interest is either user-defined or feature-based. Viola et al. [32] defined the object of interest through user selection and smoothly modifies viewpoint and visual parameters when changing the object of interest. Viola et al. [34] proposed importance-driven volume rendering by manually assigning different importance to the pre-segmented objects in the data to maximize the visual information in the rendered results. Wang et al. [40] proposed a feature-preserving data reduction method that allows users to magnify regions according to the degree of interest for focus+context visualization.

Importance-driven visualization without user interaction has pre-defined importance based on the domain knowledge or is totally data-driven. Wang et al. [36] incorporated domain knowledge, e.g., salient isosurface and defined the importance of data based on the inverse distance to the surface of interest. To reduce massive visual information during particle tracing, Viola et al. [33] utilized the object importance to define the sparseness level of each feature for controlling opacity values and rendering styles of the feature. Other works define data importance based on statistical models. For example, Wang et al. [35] defined importance through conditional entropy by measuring the amount of entropy one block remains given blocks of neighboring time steps. Gosink et al. [16] introduced a statistical framework to explore variable trends and identify important variables for different regions.

Our work is related to importance-driven visualization and analysis. The difference is that we use the importance to generate a controlled latent representation.

3 BACKGROUND

Our importance-driven latent generation framework is based on an autoencoder with a quantizer in the latent space. In this section, we introduce this model.

3.1 Non-linear Transform Coding using Autoencoders

A recent work [1] indicates that compared with linear transform coding, nonlinear transform coding is more flexible and can better adapt to the source signal distribution. In our work, we utilize nonlinear transform coding via a convolutional autoencoder. The autoencoder contains two parts, an encoder f which converts the raw data x into a latent representation y and a decoder g which decodes the latent y and gets a reconstruction $\hat{x}$ of x . The latent size is often smaller than the raw data, which forms a bottleneck to restrict the information flow from the encoder to the decoder. For example, as shown in Fig. 2, after several convolutional layers, the original data are converted to a latent of size $K \times 3 \times 3 \times 3$, where K is the number of filters in the last convolutional layer, also known as channel size of the latent. The bottleneck forces the latent to preserve only the most vital information in the data. Thus, the autoencoder is suitable to generate compact data representations.

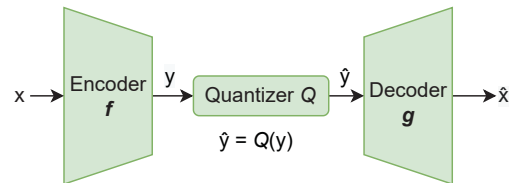


Fig. 1. Autoencoder with a quantizer Q in the latent space.

3.2 Quantized Latent Space for Data Reduction

To get a better data reduction performance, unlike a basic autoencoder which only minimizes the reconstruction loss, Ballé et al. [3] further quantize the latent representation by a quantizer Q , as shown in Fig. 1. Then, the discrete symbols themselves are losslessly compressed through entropy coding. The optimization goal of this autoencoder is to minimize both the reconstruction loss and the entropy of the quantized latent representations, formulated as:

$$\underbrace{\mathbb{E}_x[-\log_2 p_{\hat{y}}(\hat{y})]}_R + \lambda \underbrace{\mathbb{E}_x[\|x - \hat{x}\|_2^2]}_D \quad (1)$$

where the quantized latent $\hat{y} = Q(f(x))$, the reconstructed data $\hat{x} = g(\hat{y})$, and Q is the quantizer. Equation 1 is aligned with the rate-distortion theory. R is the rate that determines the number of bits per symbol for data reduction; in our case, it is the latent entropy. D is the distortion between the original and the reconstructed data, i.e., reconstruction loss. λ is a tradeoff parameter. A larger λ will focus more on reducing the distortion D during optimization. As a result, more bits are required to maintain the reconstruction quality, and we will have a larger rate R .

However, the quantization is not differentiable. To make quantization differentiable and incorporate the quantization error during training, Ballé et al. [2] replace the quantizer with additive uniform noise. Now instead of the quantized representation $\hat{y} = Q(f(x))$, we have a “noisy” representation $\tilde{y} = f(x) + \Delta y$, where $\Delta y \sim U(-\frac{1}{2}, \frac{1}{2})$. The optimization goal changes into [3]:

$$\underbrace{\mathbb{E}_{x, \Delta y} [-\log_2 p_{\tilde{y}}(\tilde{y})]}_R + \lambda \underbrace{\mathbb{E}_{x, \Delta y} [\|x - \tilde{x}\|_2^2]}_D \quad (2)$$

where $\tilde{x} = g(\tilde{y})$ is the reconstruction. To remove the constraint on the input size, a non-parametric distribution is used to model the probability density for channels of latent $p_{\tilde{y}}$, as shown in Fig. 2 (left). During testing, the actual quantization, such as rounding is applied. After that, a lossless entropy encoding on the quantized latents is applied to convert latents into bitstreams. More frequent data will be represented by shorter bits than less frequent data.

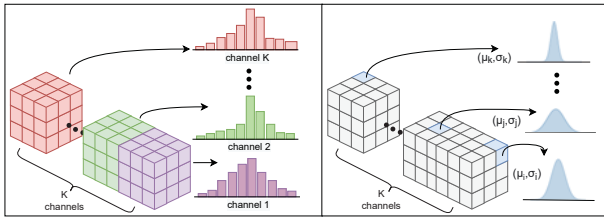


Fig. 2. The probability estimation for every latent channel (left), and probability estimation for every latent dimension (right).

One limitation of the above method is that the entropy estimation of the latent representation is not accurate. Entropy encoding relies on the probabilistic distributions of discrete latents to decide which codeword will represent which quantized symbol so that the average bit length is minimal. The better the probabilities are modeled, the closer the bit rate approaches the optimal lower bound. However, the above method does not consider spatial and raw data dependencies when estimating the probability [4], due to the reason that it only models a channel-wise latent distribution for an ensemble of input.

To improve entropy estimation, one follow-up work [4] introduces a hyperprior network to extract side information to assist latent probability estimation. The hyperprior network takes side information as input to predict a prior on the parameters of latent’s probability distribution. As shown in Fig. 2 (right), each latent dimension is modeled as a Gaussian where the scale and mean of each Gaussian are predicted by the side information [4].

With the quantized latent space and improved entropy estimation, we can achieve a better data reduction performance. In the next section, we will present our latent generation method based on a quantized autoencoder which achieves the data reduction goal, and more importantly, takes domain interest into consideration during latent generation.

4 OVERVIEW AND ALGORITHM REQUIREMENTS

Algorithm Requirements: We summarize three algorithm requirements to generate domain-interest-guided latent representations.

- The generated latent representations need to respond to different domain interests.
- The algorithm needs to be adaptive to different types of domain interests such that scientists do not need to train multiple neural network models when they vary their interests.
- The algorithm needs to generate compact latent representations whose size depends on the domain interest, i.e., low domain interest means a more compact latent representation.

Overview To generate latent representations for scientific data guided by scientists’ interests, we propose an importance-driven latent generation algorithm. An overview of the proposed method is shown in Fig. 3. The first stage of our method is to properly represent various scientific interests with spatial importance maps, which can be interpreted as, for each spatial location, how much information scientists want to preserve when generating the compact latent representation. Then the second stage is to generate scientific interest-guided latent representations through our autoencoder model. We take a block-wise processing strategy. Volumetric data and corresponding importance maps are divided into blocks and then processed by the model. Conditioned on the importance map, the data blocks are non-linearly encoded and transformed by the autoencoder’s encoder into compact latent representations. The third stage is a lossless data reduction component in the latent space. Inspired by autoencoders used for image compression [3, 4] as discussed in Sect. 3.2, our latent representations are further quantized into discrete symbols. After that, an entropy encoding algorithm, e.g., Asymmetric Numeral Systems (ANS), is adopted to losslessly compress quantized latents into bitstreams for saving.

Analyses can be done in either latent space or the original physical space. Importance-driven latent space has a simpler structure, and therefore tasks like feature extraction can be easily performed in this space. When the precise visualization of the dataset is needed, latent representations can be decoded back into the physical space for various visualization tasks.

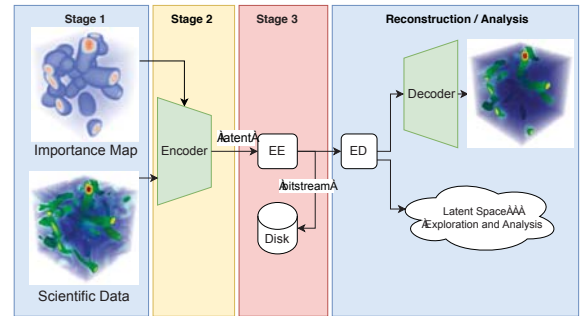


Fig. 3. Our proposed method to generate importance-driven latent representations conditioned on the input importance maps. The resulting latent representation will be entropy encoded into bitstreams for saving.

5 METHOD

There are three main issues to address when generating domain-interest-guided latent representations: (1) how to define a unified representation to incorporate various domain knowledge and avoid training multiple models when scientists vary their interests, (2) how to fuse domain knowledge during latent generation, and (3) how to control the latent size and make trade-offs between latent size and latent quality. In this section, we discuss how we address these issues.

5.1 Importance Driven Latent Generation

5.1.1 Domain Interest Representation

This section discusses how to represent diverse domain interests with a unified representation. Our latent generation method is domain-interest-guided, meaning not all data are treated equally important during latent generation. This strategy is commonly used for many scientific data analysis applications. Given a massive amount of scientific data, scientists often try to identify features of interest by narrowing down their search space, which is defined by how data are relevant to the important features. Generating latent representations for scientific data according to the data importance can not only reduce the size of data, but also allow scientists to focus on the most salient portion of the data.

In our method, we utilize data importance to assist the process of latent generation. To create a unified representation of importance that can be taken by our autoencoder for a variety of needs, we summarize commonly used importance definitions in scientific visualization literature as below:

- **Location-based:** Scientists assign an importance value for every spatial location based on whether it is in the pre-selected region of interest [32–34]. For example, hurricane eye regions are of high interest for hurricane and tropical cyclone research scientists, and as a result, those regions will have higher importance values than non-hurricane eye regions [33].
- **Distance-based:** To enhance the understanding of features of interest, scientists define the importance of each data element based on its distance to the feature of interest. For example, based on the distance to object of interest [33] or salient isosurfaces [36].
- **Value-based:** Scientists define the importance based on the differences between a pre-selected reference value and data values or based on transfer functions [27, 40]. If the value of a data element is close to the values of interest, high importance will be assigned. We note that here the value can take a variety of forms: scalars, vectors, and tensors, to name a few.
- **Time-based:** To effectively visualize and analyze large-scale time-varying data, scientists assign different importance values for time steps. For example, based on the relative information a time step contains about its temporal neighbours [35] or assign salient time steps with high importance values.
- **Multivariate-based:** For a multivariate dataset, the importance can be derived from the joint distribution of variables [16]. For instance, to locate interesting regions for turbulent combustion data, multiple variables such as Mixture Fraction (MIX), Mass Fraction of the Hydroxyl Radical (OH), and Heat-Release Rate (HR) are jointly considered.
- **Statistical-based:** Importance can be defined based on statistical properties of data such as conditional entropy [35], correlation, or value histogram [7].

In the core of our method, we define a unified representation, i.e., a real-valued spatial importance map, to incorporate various importance definitions. The map is defined in the same domain as the data, and each value in the importance map indicates the scientific interest at that spatial location. Importance maps are taken as an additional input to the neural network model to control the latent representation generation.

There are two obvious advantages in using spatial importance maps. First, spatial importance maps can inform the neural network which regions are more important so that their information needs to be better preserved in the latent space, and for those regions with low importance, their latents can be simplified or smoothed out during encoding. Second, because we are using a unified representation for various domain interests and our latent generation is conditioned on the input importance map, as a result, we do not need to retrain different neural network models when scientists change their definition of spatial importance. In our paper, the importance value at every spatial location is calculated through a scientist-specified importance mapping function Ψ , mathematically defined as:

$$I_p := \Psi(p, F(p)) \quad (3)$$

where $\Psi: \mathbb{R}^3 \rightarrow \mathbb{R}$ is a mapping from spatial location $p \in \mathbb{R}^3$ to an importance value I_p given the location and its data value $F(p)$. $F(p) \in \mathbb{R}$ if it is a univariate data, and $F(p) \in \mathbb{R}^n$ if it is a multivariate data with n variables. We evaluate I_p on all voxel locations to obtain an importance map I .

During training, we randomly generate importance maps with different spatial variations such as distance ramps, Gaussian distributions with various centers, data gradients, and random uniform maps. During testing, the trained model is applied to various importance maps derived from different scientific interests. In our evaluation in Sect. 6, we demonstrate that these predefined importance maps are effective to train a generalized model which does not constrain a scientist's importance map specification during testing.

5.1.2 Autoencoder with Condition Network

To generate reduced data representations, we utilize a convolutional autoencoder model which converts input data into a latent representation through an encoder and decodes the latent back to data through a

decoder. We utilize autoencoder for the reason that, as also discussed in Sect. 3.1, compared with linear coding methods such as discrete cosine transform (DCT), the non-linear coding ability of autoencoders makes them suitable and powerful to represent data.

To properly fuse domain knowledge into latent representations, we utilize Spatial Feature Transform (SFT) layers which are widely used in computer vision for image super-resolution [10, 17, 39], conditional generation [37, 38], compression [30], and segmentation [24]. In these works, SFT layers are used to incorporate conditional knowledge by generating affine transformation parameters for feature modulation. We adopt a similar method as Song et al. [30] which performs image compression given a classification or text-preserving task.

In our work, we utilize SFT layers to fuse domain knowledge into latent representations. The reason for using SFT layers is that they can capture rich spatial prior information from prior knowledge, e.g., regions of interest, to modify the intermediate feature maps of the data in the autoencoder. For example, smoothing out details in regions where scientists have low interest. More specifically, we adopt the autoencoder model by connecting it with two SFT-layer-based feature transformation networks. The transformation network connected to the encoder is shown in Fig. 4 who takes domain interest (i.e., a spatial importance map I) as input to extract conditions of different resolutions to have a layer-by-layer control of the encoder. Each condition Ω produces two sets of affine transformation parameters (i.e., α for scaling and β for shifting) for each encoder layer, formulated as:

$$\Omega = \text{conv}(I) \quad (4)$$

$$\Phi(\Omega) = (\alpha, \beta) \quad (5)$$

where conv are the convolutional layers. Φ is a mapping function from condition Ω to the scaling parameter α and the shifting parameter β . α and β are used to transform the intermediate feature map F generated by the autoencoder:

$$F' = \text{SFT}(F|\alpha, \beta) = F \odot \alpha + \beta \quad (6)$$

Then, the transformed feature map F' is taken as input to the next encoding layer. α and β are of the same size as the feature map F , and $\odot$ denotes element-wise multiplication. Applying multiplication and addition on feature maps is a simple and effective way to gradually fuse two sources of information (i.e., importance and data) from different levels. Scaling the feature map is like gating so that information in regions with high importance are preserved and others are suppressed. Shifting the feature map has a similar effect. Combining these two, we have the flexibility to leverage importance maps for latent generation.

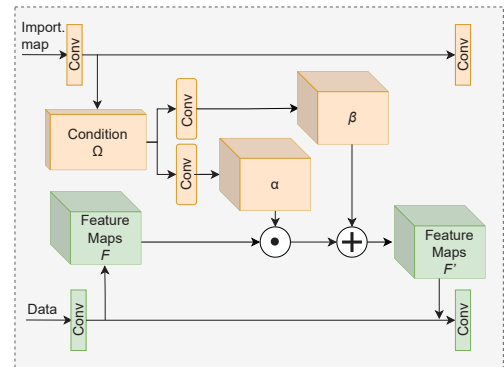


Fig. 4. Spatial Feature Transform (SFT) [39] layers take the condition Ω generated from the importance map as input and output affine transformation parameters to scale (α) and shift (β) feature map F of data.

5.2 Entropy Encoding in Latent Space

To optimize the size of the latent representation, we apply lossless entropy encoding to the latent vectors generated from the autoencoder. We adopt the entropy model widely used for neural-network-based image compression [4], as discussed in Sect. 3.2.

Our proposed importance-driven latent generation method is shown in Fig. 5. After layers of non-linear coding using autoencoder's encoder

f , the input data x is converted into a latent representation y , controlled by a scientist-specified input importance map I , so we have:

$$y = f(x, T_f(I)) \quad (7)$$

where T_f is the transformation network connected to the encoder f . Following the technique in Sect. 3.2, we quantize the latent vector y and apply the entropy encoding algorithm, Asymmetric Numeral Systems (ANS) [13], on latent vectors. The resulting bitstreams are saved into the disk. During decoding, the saved bitstreams are entropy decoded into discrete latent $\tilde{y}$ and sent to the decoder to get the reconstruction $\tilde{x}$. As shown in Fig. 5, to save storage and mainly reduce unimportant information in the latent representation, we do not use importance maps to modulate information during decoding. The decoder only takes latent $\tilde{y}$ as input to the transformation network connected to the decoder to incorporate conditions during reconstruction, formulated as:

$$\tilde{x} = g(\tilde{y}, T_g(\tilde{y})) \quad (8)$$

where T_g is the transformation network connected to the decoder g .

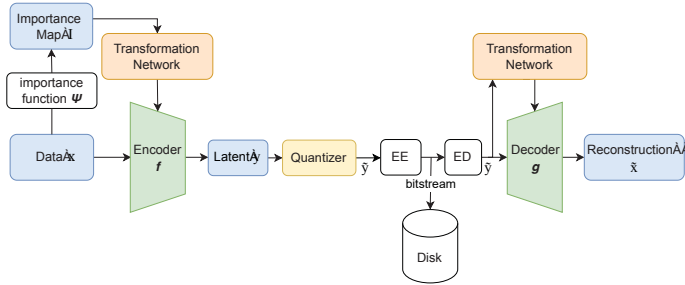


Fig. 5. Our proposed importance-driven latent generation method. Our model is a combination of an autoencoder and entropy encoding model.

5.3 Loss Functions

To make trade-offs between latent size and quality based on domain interests, we formulate our importance-driven latent generation as an optimization problem of minimizing the rate-distortion Lagrangian as discussed in Sect. 3.2, formulated as:

$$\mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_D \quad (9)$$

where $\mathcal{L}_R$ is the quantized latent entropy loss, $\mathcal{L}_D$ is the reconstruction loss, and λ is the Lagrangian multiplier, a hyperparameter to balance between reconstruction quality and latent size.

To force the generated latent representation to be guided by scientists' spatial interests, instead of a reconstruction loss with equal importance on every spatial location like a basic autoencoder, we use an adaptive reconstruction loss between the input data x and reconstructed data $\tilde{x}$:

$$\mathcal{L}_D = \mathbb{E}_{x, \Delta y} \left[\sum_{i=1}^N w_i (x_i - \tilde{x}_i)^2 \right] \quad (10)$$

where w_i is the distortion weight at index i derived from the importance map I by an exponential function with a hyperparameter a , i.e., $w_i = e^{aI_i}$. The goal of exponential weighting is to have a finer distortion control across different spatial locations such that different importance values will have significantly different contributions to $\mathcal{L}_D$. N is the number of data elements in x . Δy is the additive uniform noise to relax quantization.

The entropy loss for discrete latent $\tilde{y}$ is:

$$\mathcal{L}_R = \mathbb{E}_{x, \Delta y} [-\log_2 p_{\tilde{y}}(\tilde{y})] \quad (11)$$

where $p_{\tilde{y}}$ is the latent probability distribution for entropy coding.

5.4 Latent Space Analysis

In this section, we discuss how we use block-wise latent representation for feature-related analysis and briefly introduce the visual exploration tool for latent spaces. Feature-related analysis in the latent space originated from the observation that similarity defined in the latent space can better represent the similarity of higher-level features than the similarity of the raw data [5]. Our latent vector generation method

considers domain knowledge provided as an importance map to neural networks. As a result, latent space distribution is conditioned on the provided importance map. Even though the structure of the full latent space of the dataset is complicated and requires a sophisticated tool [25] to explore, importance-driven latent space is easier to understand and to be made into use for feature analysis.

On the saved latent representations for each data block, we perform a hierarchical clustering algorithm to identify blocks that are similar to each other. These clusters can be either used to extract a subset of data for further analysis or for feature-driven visualization like the one presented by Cheng et al. [11]. It is worth noting that the data block size is a hyperparameter we should choose based on the feature size, data complexity, and desired representation storage size.

Our latent space visual analysis tool is based on the one designed by Li and Shen [25], where clustering is performed on block-wise latent vectors and cluster results are visualized in latent, and physical space for feature-related analysis. There are three main views in our tool, *hierarchical clustering view*, *latent space view*, and *physical space view*, demonstrated in Fig. 9. *Hierarchical clustering view* presents each cluster as a node in a tree graph, where clusters can be modified by interacting with the nodes. *Latent space view* shows projected latent vectors into 2D using t-distributed stochastic neighbor embedding (t-SNE) projection [31]. This view is updated when the clustering result is modified. Finally, *physical space view* visualizes data in the selected cluster. Spectral clustering is used in our latent analysis approach since it adapts well to complex spaces with unknown cluster shapes [29], which is usually the case for latent spaces generated by neural networks.

5.5 Implementation

5.5.1 Block-based Processing

Processing large-scale scientific data requires a big convolutional neural network model which has high computational cost and memory consumption. Another problem is that we need a large collection of data for training, but building such training data is prohibitive for scientific simulations due to the high cost of generating and saving large-scale data. To meet GPU memory constraints, some preprocessing steps such as downsampling or cropping [18, 19] are applied to data. However, the drawback of downsampling is that it inevitably introduces errors and uncertainties in the downsampled data.

To address the above problems, we adopt a block-based processing strategy, i.e., volumetric data are divided into blocks for the neural network model to process. Data blocks can be processed in parallel with a large batch size for speedup. To reduce the reconstruction error at the block boundary introduced by zero-padding or reflection-padding, we pad each block with the actual data for the network to process. During reconstruction, we crop the reconstructed data, and only the central data regions are attached to reconstruct the whole volume. For instance, if the block size is 24^3 and we have a padding size equal to 4, then the data size each latent represents is 16^3 .

5.5.2 Training Data Sampling

The most intuitive way to build the training data is to randomly sample large amounts of data blocks to ensure a good coverage of different patterns for training. However, this data size will grow proportionally to the size of the original data and make the training extremely ineffective. To solve this and to force the model to learn complex patterns, we adopt a complexity-aware training data sampling strategy, i.e., the training dataset is designed to include more complex (high entropy) blocks and less homogeneous (low entropy) blocks. For the Hurricane Isabel dataset, the sampling ratio between high and low entropy blocks is 10:1 to ensure the complex data regions are covered in the training data given that a large portion of the original data is homogeneous.

6 RESULTS

In this section, we evaluate the **Importance-Driven Latent** generation method (**IDLat**) both quantitatively and qualitatively from four different perspectives: (1) the quality of latent representations; (2) the size of latent representations; (3) the influence of different important maps; and (4) the use of latent representations for latent space analysis.

Table 1. Dataset name, variable name, data resolution, training epochs and time, training data size (number of blocks), encoding and decoding time on each volume (seconds). Encoding and decoding time do not induce the time used for writing/reading bitstreams files into/from disk.

Dataset	Variable	Size	Epochs	Training time	# Training blocks	Enc. time	Dec. time
Vortex	vorticity magnitude	$128 \times 128 \times 128$	600	4h 20m	1000	0.0232s	0.0105s
Nyx	log density	$256 \times 256 \times 256$	100	3h 14m	5000	0.2272s	0.0794s
Isabel	pressure	$512 \times 512 \times 96$	200	5h 30m	5500	0.2000s	0.1191s

6.1 Dataset and Training Parameters

We evaluated our importance-driven latent generation method using three scientific datasets for multiple scientific applications.

Vortex is a simulation of vortex structures with spatial resolution $128 \times 128 \times 128$ across 30 time steps. We used the vorticity magnitude scalar field for experiments. We randomly sampled 1000 data blocks from 5 time steps as the training data. **Nyx** is a cosmological simulation produced by Lawrence Berkeley National Laboratory. We used the log density field with resolution $256 \times 256 \times 256$. 5000 data blocks from 5 ensemble members were randomly sampled for training. **Hurricane Isabel** is a simulation of Hurricane Isabel, produced by the Weather Research and Forecast (WRF) model, courtesy of NCAR and the U.S. National Science Foundation (NSF). The data were sliced along the z dimension to remove the special value 1×10^{35} representing "no data" (the land region). In our experiment, the resolution of data is $512 \times 512 \times 96$ with 48 time steps. We chose the pressure field for evaluation. Training data contain 5500 data blocks from 5 time steps.

Our work consists of two main components: the IDLat model and a latent space visual analysis tool. The IDLat model is implemented based on PyTorch¹ and trained with a single NVIDIA Tesla P100 GPU. For all datasets, we use Adam optimizer [23]. The learning rate for the autoencoder model and the entropy model is 10^{-4} and 10^{-3} , respectively. Total training time for each dataset is listed in Table 1. Based on a fully convolutional model with block-based training and inference strategy, we can apply IDLat on data of any resolution. The hierarchical clustering view and latent space projection view from the latent space exploration tool are implemented with Vue.js² as the front-end framework and Flask³ as the back-end framework. VTK APIs⁴ are used to visualize the extracted blocks.

6.2 Quantitative Evaluation

In this section, we quantitatively evaluate the size and the quality of latent representations generated by IDLat.

6.2.1 Evaluation Metric

To evaluate the size of latents, we use the ratio between the original data size and the saved bitstream file size, i.e., latent size ratio (LSR):

$$LSR = \frac{\text{original data size}}{\text{bitstream file size}} \quad (12)$$

To evaluate the quality of importance-driven latent representations, we analyze how well the important regions are preserved during reconstruction under various importance maps. We compute the error between the decoder's reconstruction and the raw data. Because we focus more on the quality of important regions, we utilize a weighted Mean Squared Error (wMSE) defined as:

$$\text{wMSE}(x, \tilde{x}) = \frac{1}{\sum_{i=1}^N I_i} \sum_{i=1}^N I_i (x_i - \tilde{x}_i)^2 \quad (13)$$

where x_i , $\tilde{x}_i$ are the original and the reconstructed data at position i , respectively. N is the number of data elements in x . I_i is the importance at position i in range $[0, 1]$ defined by Equation 3 for different applications. Locations with a larger importance value will have higher weights in the error estimate. The peak signal-to-noise ratio (PSNR) is defined based on wMSE:

$$\text{PSNR}(x, \tilde{x}) = 10 \log_{10} \frac{v^2}{\text{wMSE}(x, \tilde{x})} \quad (14)$$

where v denotes the value range in the original data.

6.2.2 Evaluation on Different Importance Maps

As discussed in Sect. 5.1.1, there are various criteria focusing on representing regions of interest for importance-driven visualization. In this section, we evaluate IDLat's quality and quantity under different importance definitions.

To evaluate the effectiveness of the entropy encoding module in IDLat, especially its ability to generate the latent representation of optimal size with the presence of an importance map, we compare IDLat with a baseline method, i.e., a basic autoencoder without the importance map, quantization, entropy module, and the entropy loss. We train this baseline model with the same training data and parameter setting as IDLat, but only with the reconstruction loss.

Distance-based importance maps: We evaluate importance-driven latent's quality and quantity conditioned on distance-based importance maps through the Vortex dataset. Vortex data contains vortex structures that have been widely used for isosurface tracking to analyze vortex core regions over time [20, 21]. In this evaluation, the importance maps are defined based on distances to the selected isosurfaces where the importance value I_p for each spatial location p defined in Equation 3 can be specified as:

$$I_p := \Psi_{\text{Vor}}(p) = e^{-0.2|SDF(p, S)|} \quad (15)$$

$SDF(p, S)$ represents the signed distance from location p to surface S . We use a negative exponential function to convert absolute SDF distances into importance values in $[0, 1]$ for the model to process. The importance will decrease exponentially as p becomes far from the surface. 0.2 is a parameter that we choose to have a proper slope of the exponential curve. We can also use other functions to convert distances into weights, e.g., the inverse distance function.

To evaluate the influence of different distance-based important maps, we chose a time step in Vortex data and pre-selected several vorticity magnitude values (e.g., 5.0, 6.0, 7.0, and 8.0) as salient isovalues to be preserved in the latent representations. We used Ψ_{Vor} in Equation 15 to convert the volumetric data into importance maps for IDLat to generate importance-driven latent representations.

We report the quantitative results, i.e., PSNR and latent size ratio (LSR), of applying different distance-based importance maps on Vortex data in Table 2. From the table, we found that IDLat indeed can generate importance-driven latent representations for Vortex. Given the same data but different importance maps, IDLat generates latents with different quality and size, as shown in the PSNR (IDLat) and LSR (IDLat) columns. As for the baseline model, because the latent size is determined by the input data size, the latent size ratio for the baseline model is fixed at a very small number (7.3143) as shown in LSR (Base). It is clear in the table that the proposed IDLat can largely decrease the latent size (i.e., increase LSR) based on different importance maps without losing much quality. We also found that IDLat can achieve increased latent size ratios as we increase the target isovalue. This is because IDLat is designed to have higher quality on regions with high interests, and as the isovalue increase, the important regions are getting smaller for this dataset, and as a result, LSR is getting larger.

Value-based importance maps: Given value-based importance maps, we evaluate the size and quality of latents generated by IDLat on Nyx data. One important post-hoc analysis task for Nyx simulation is to find dark matter halos which are related to the high-density field in the data [15]. In our experiment, the importance value I_p of each spatial location p is defined based on the log density value $F(p)$ and a reference log density value F_{ref} as follows:

¹<https://pytorch.org>

²<https://vuejs.org>

³<https://flask.palletsprojects.com>

⁴<https://vtk.org>

Table 2. Reconstruction PSNR and latent size ratio (LSR) for IDLat with different importance maps and for the baseline model.

Data	Imp. Def.	PSNR (IDLat)	PSNR (Base)	LSR (IDLat)	LSR (Base)
Vortex	iso 5.0	34.3987	35.4927	107.3825	7.3143
	iso 6.0	33.9121	35.2154	109.8901	
	iso 7.0	33.2551	34.5266	111.849	
	iso 8.0	33.9757	34.2463	113.6767	
Nyx (m1)	log den > 9.9	33.8422	35.2382	209.4241	7.3143
	log den > 10.2	30.9899	31.0386	218.8783	
	log den > 10.5	28.2268	27.1545	225.6700	
Nyx (m2)	log den > 9.9	31.3774	32.2409	172.5998	7.3143
	log den > 10.2	29.0695	29.2020	179.3722	
	log den > 10.5	26.761	26.1568	188.2353	
Isabel	hurricane eye	44.9749	44.6605	199.1288	

$$I_p := \Psi_{Nyx}(p, F(p)) = \begin{cases} 1 & \text{if } F(p) > F_{ref} \\ 0 & \text{else} \end{cases} \quad (16)$$

To evaluate the effect of different value-based importance maps, driven by the domain interests of the Nyx data discussed above, we conduct experiments on two ensemble members of Nyx with different reference values. In our experiment, we select 9.9, 10.2, and 10.5 as log density reference values and utilize Ψ_{Nyx} defined in Equation 16 to compute value-based importance maps.

We report the quantitative results in the second and the third blocks of rows of Table 2. The two blocks of rows represent experiments on two different ensemble members, denoted as Nyx (m1) and Nyx (m2). As we can see, for each ensemble member, the latent size ratio (LSR (IDLat)) increases as we increase the log density reference value (i.e., reduce the number of important voxels), but the quality (PSNR (IDLat)) is kept comparable to the baseline (PSNR (Base)). Another thing we notice is that PSNR drops as we increase the log density reference value for both baseline and IDLat models. A possible reason can be that these high-value regions are harder to model due to high data complexity.

Location-based importance maps: We also evaluate the latent representations' size and quality given location-based importance maps using the Isabel dataset. For this dataset, one task that scientists are interested in is to identify and analyze the hurricane eye region. So the importance maps are built based on the hurricane eye locations where voxels inside the interested region C will have high importance values. The importance value for each spatial location p is defined as:

$$I_p := \Psi_{Isa}(p) = \begin{cases} 1 & \text{if } p \in C \\ 0 & \text{else} \end{cases} \quad (17)$$

We use the first time step of Isabel and draw a bounding box of the hurricane eye as the region of interest C . Based on the importance mapping function in Equation 17, we compute the importance map.

Table 2 shows the quantitative results of Isabel data given a location-based importance map using IDLat and the baseline model. Compared with the baseline, the latent representation generated by IDLat with a location-based importance map is more compact (higher LSR) with slightly higher quality (higher PSNR) than the baseline.

Essentially, the baseline model is a special case of our IDLat with $\lambda \rightarrow \infty$ in Equation 9, which achieves an upper bound for reconstruction error and a lower bound for the latent size ratio. Instead of a static model with a fixed latent size, the proposed IDLat can achieve various quality and latent vector sizes based on target scientific applications.

6.3 Qualitative Evaluation

We qualitatively evaluate the proposed IDLat by visualizing the reconstructions with volume and isosurface rendering.

In Fig. 6, we show isosurface rendering of the reconstructed Vortex data generated by the baseline and by IDLat with isosurface-distance-based importance maps. In Fig. 6, in each row from left to right are isosurface rendering for isovalue 5, 6, 7, and 8. Comparing isosurface rendering of IDLat's reconstruction (third row) with the ground truth (first row) and the baseline (second row), we found that our latent representations can capture the structure and also the details in the ground truth, even though they are much smaller in size as discussed

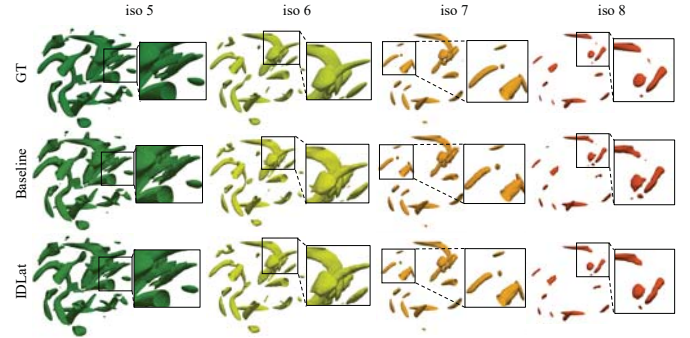


Fig. 6. Comparison of isosurface rendering results of Vortex at time step 6 between ground truth (top row), baseline (middle row) and IDLat's reconstruction (bottom row). The four importance maps for IDLat are defined based on distances to isosurfaces 5, 6, 7 and 8.

in Sect. 6.2. In some regions, IDLat can have slightly better reconstruction quality compared to the baseline, as shown in the zoom-in regions in Fig. 6. These results demonstrate that our importance-driven latent representations can capture spatial importance information and have high reconstruction quality in important regions.

Fig. 7 displays volume rendering images of ground truth (first column), baseline, and IDLat's reconstruction based on value-based importance maps for two ensemble members of the Nyx dataset. The second and the third columns are volume rendering images of the baseline's reconstruction and difference maps between the original and the reconstructed data. Other columns are IDLat's results based on three different value-based importance maps. The reference values are 9.9, 10.2, and 10.5, respectively. In the first two rows, each row shows the result for one ensemble member. We use the same transfer function for each row to ensure the volume rendering difference is caused by the reconstructed data but not the transfer function difference. The transfer function for volume rendering of the reconstructed and ground truth data is the vertical colorbar and for difference map is the horizontal colorbar. From volume rendering images, we found the latent representation has high reconstruction quality on every importance map. From left to right, we can see the differences in the difference map spreads out more when we increase the reference value, which matches the observation that as the reference value increases, the important regions in the dataset are getting smaller and unimportant regions are enlarged so that the difference at unimportant locations becomes more obvious. The third row of Fig. 7 shows zoom-in of ground truth (A), baseline (B) and IDLat's reconstruction (C and D). As shown in the figure, when the reference value increases from 9.9 in Fig. 7 (C) to 10.5 in Fig. 7 (D), the reconstruction loses more details of the unimportant data compared to the ground truth and the baseline, e.g., the red dashed circled regions are more smoothed out in Fig. 7 (D). The zoom-in of difference maps also shows more obvious spread-out differences in the unimportant regions. The pattern of difference maps and high-quality volume rendering results demonstrate that IDLat is under the guidance of spatial importance when generating latent representations.

Fig. 8 shows volume rendering images of Isabel dataset. From left to right, they are ground truth, baseline's, and IDLat's reconstruction. IDLat utilizes a location-based importance map where the hurricane eye is the region of interest, as shown in the selected bounding box. Comparing the ground truth and baseline with IDLat's reconstruction, we can see the quality of the hurricane eye is highly preserved, although IDLat has a smaller latent size as discussed in Sect. 6.2.

The above quantitative and qualitative results validate that IDLat is under spatial importance guidance when generating compact latent representations with high quality in important spatial regions.

6.4 Latent Space Exploration and Analysis

As discussed in Sect. 5.4, our method produces latent representations that can be used for feature-related exploration and analysis. To show that the proposed importance-driven latent representations are succinct and suitable for representing features of interest, we perform latent space exploration and analysis on Vortex data with two case studies.

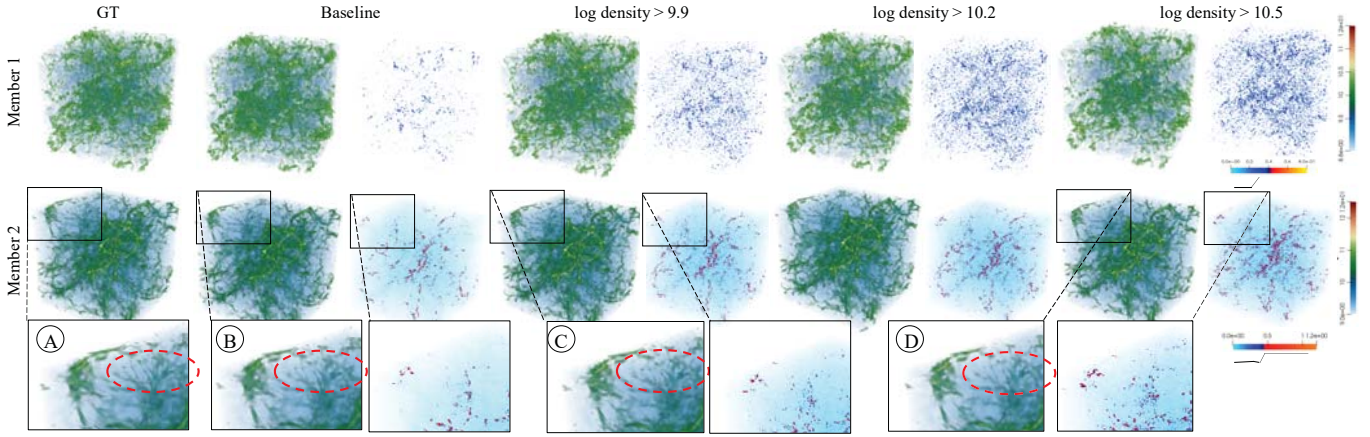


Fig. 7. Volume rendering of ground truth, baseline and IDLat's reconstructions on two ensemble members of Nyx data with value-based importance maps. Reference values are 9.9, 10.2 and 10.5. The difference map for each reconstruction is shown on its right. From left to right, in the difference map, the difference spreads out more when we increase the reference value, which matches the fact that as the reference value increases, the important regions are getting smaller and unimportant regions are enlarged so that the difference at unimportant regions becomes more obvious.

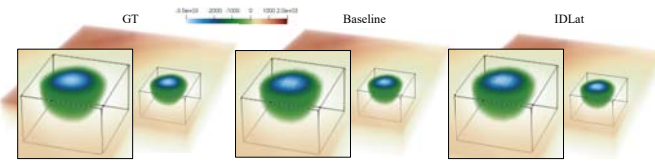


Fig. 8. Volume rendering of ground truth, baseline and IDLat's reconstruction on Isabel data with a location-based importance map where the hurricane eye in the bounding box is the region of interest.

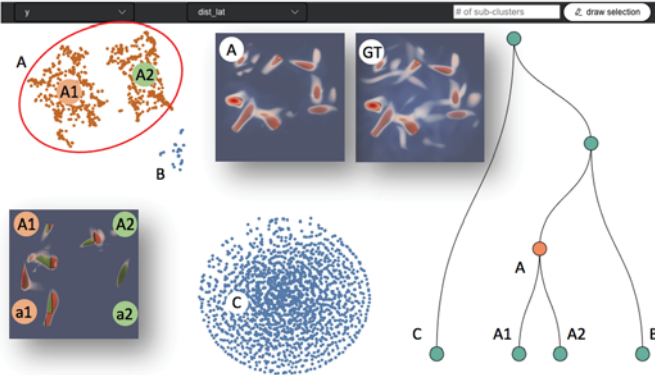


Fig. 9. Latent space projection and hierarchical clustering results given importance map based on distances to isosurface 8.0. Volume rendering and isosurface (*isovalue* = 8) for the ground truth data (GT), for blocks from cluster A (A) and for child clusters of cluster A (bottom left), where isosurfaces are shown in red for cluster A1 and green for cluster A2.

6.4.1 Case Study 1: Obtain Insight of Features in Latent Space

In the first case study, we show the effectiveness of importance-driven latents in identifying features by exploring the latent space. We compare block-wise latent representations generated by a uniform importance map and by a distance-based importance map in Sect. 6.2.2.

First, we show the exploration results for the importance-driven latent space. The importance map is defined based on distances to the isosurface with *isovalue* = 8, which reveals the interest features of the vortex cores. In Fig. 9, we can easily identify four distinct clusters in the t-SNE projection of the latent vectors. The users can also modify the hierarchical clustering view to investigate the detail related to each cluster. On the right side of Fig. 9, we show the hierarchical clustering results of latent vectors. Cluster A consists of the blocks that contain the vortex cores and can be further separated into two child clusters (A1, A2) based on the separation in the t-SNE projection. The difference between cluster A1 and cluster A2 will be discussed later. Cluster C

consists of blocks that do not intersect with the isosurface of interest. Cluster B contains some boundary blocks (e.g., blocks on the edges and in the corners).

To further investigate cluster A, we show images of volume rendering and isosurfaces for the ground truth data in Fig. 9 (GT) and the blocks from cluster A identified in latent space in Fig. 9 (A). It can be observed that Fig. 9 (A) and Fig. 9 (GT) reveal the same isosurfaces (ignoring the fuzzy region in the ground truth image, which is not part of the isosurface), indicating that our latent representations allow us to preserve the important regions with good quality and in the meantime enable us to visualize and separate the features easily in the latent space.

In the t-SNE projection, cluster A consists of two child clusters A1 and A2. We show the isosurfaces corresponding to these two clusters in Fig. 9 (A1, A2). Cluster A1's isosurface is in red and cluster A2 in green. We found that except for two boundary vortices (a1, a2), all other vortices are split into two clusters (red and green). The splitting reveals the internal structures of the vortices. Since the splitting happens along one axis, we suspect one possible reason for this is the differences of the scalar values in the block along this direction. For this dataset, in the core of vortices, it has high scalar values and the value is decreasing and the isosurface is getting enlarged from inside vortex core to outside. They are classified into different clusters probably due to the opposite direction of value decreasing on this axis, which is related to the gradient of the values. To validate this hypothesis, we calculate the average gradient distribution along the x-axis using Gaussian kernel density estimation for each data block from these two clusters as shown in Fig. 10, where we can identify the apparent gradient distribution difference among these two clusters. The further separation of the feature clusters helps visualize and understand the internal structures of the data of interest.

We also perform latent space exploration for latent vectors generated using a uniform importance map. The t-SNE projection of their latent space is shown on the left side of Fig. 11. We did not find any visual clusters in the t-SNE projection, and the clustering of the latent vectors splits all blocks into clusters of high and low average values, which is not helpful in feature-related analysis. The process of further splitting of the clusters is tedious and did not bring us anything interesting.

By comparing the structures of importance-driven and uniform latent spaces, we found that the importance-driven latent space is highly related to the features of interest and is easier to explore. From the clustering result, scientists can reduce the effort of similarity comparison between blocks by quickly filtering out unimportant regions, resulting in fast and scalable data analysis.

6.4.2 Case Study 2: Representative Isosurface Selection

In the second case study, we show the usefulness of the importance-driven latent for representative isosurface selection.

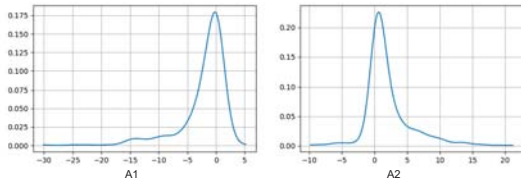


Fig. 10. Gradient distribution for data blocks from two clusters.

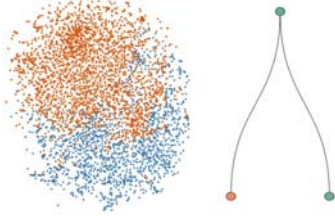


Fig. 11. Latent exploration of latent vectors generated using uniform importance map. No visual clusters are identified in the t-SNE projection.

To quantify and visualize similarities between isosurfaces, Bruckner and Möller [8] proposed the isosurface similarity map. Each element in this map is the similarity value between two isosurfaces. They use distance fields, i.e., the minimal distance of each point to the surface, to represent isosurfaces and use mutual information between two distance fields as the similarity measure. However, one limitation of this method is the high computation cost. First, representing isosurfaces as distance fields is expensive without acceleration such as approximations. Second, generating the isosurface similarity map needs to calculate mutual information between every pair of isosurfaces, which requires building a joint histogram of every two distance fields. A more effective surface representation and efficient similarity computation is desired.

To solve this, we utilize IDLat with value-based importance maps to generate isosurface representations. We use value-based importance maps for two reasons. First, the spatial information of each voxel inside each block is encoded in the latent representation, so we do not need implicit distance fields to indicate surface locations. Second, when generating importance-driven latent representations, voxels with higher importance values will have a higher contribution, and voxels with low importance will be suppressed, which helps encode the surface information and zero out non-surface information. These two properties eliminate the heavy computation of distance fields. After we have latent representations for all blocks, we concatenate them into a single latent to represent the whole isosurface. We note that compared to the encoding time reported in Table 1, we have 8 times more blocks due to smaller block size (8^3 instead of 16^3 as in Table 1), so representation generation is about 8 times slower. By changing the value-based importance maps, we can generate compact isosurface representations use different isovalues as the importance measure. Then, isosurface similarities are efficiently computed through cosine similarities between the isosurface-drive latent representations.

In the isosurface similarity map, we can find clusters of isovalues to select representative isovalues. Given surface similarities, we use the same isosurface selection algorithm as Bruckner and Möller's [8] to automatically identify representative isovalues. Fig. 12 shows isosurface similarity maps and selected isosurfaces computed by Bruckner and Möller's method [8] and by ours. Compared to Bruckner and Möller's, our method can generate better results of the top four representative isosurfaces to reveal the structure of Vortex data. The selected isosurface (number 3) in our result is missing in theirs. They may identify it later but need to increase the number of selections. In Table 3, we show the performance of these two methods. Compared to Bruckner and Möller's [8], our method is much more efficient in both representation generation stage (rep) and similarity computation stage (sim).

Table 3. Selected isovalues, time (seconds) for all isosurface representation generation and for computing the isosurface similarity map using Bruckner and Möller's [8] and importance-driven latent representations.

Method	Selected Isovalues	Time (rep)	Time (sim)
Bruckner and Möller's [8]	4.1, 5.5, 8.5, 9.7	523.0288	824.9641
IDLat	3.7, 5.7, 7.0, 9.1	183.8828	0.4095

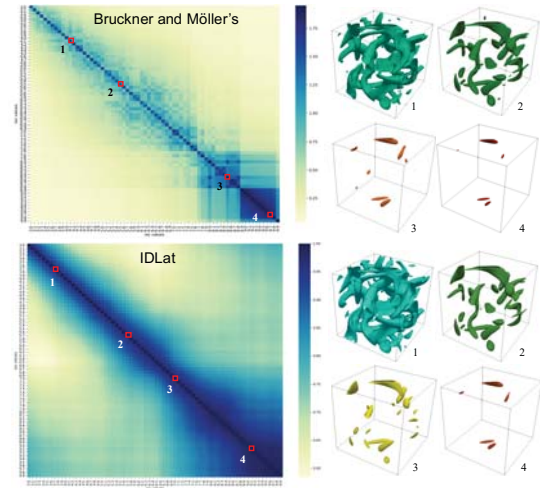


Fig. 12. **Top Row:** Bruckner and Möller's [8] isosurface similarity map (left) and four selected isosurfaces (right). **Bottom Row:** Our results of isosurface similarity map (left) and four selected isosurfaces (right).

7 DISCUSSION AND FUTURE WORK

Even though we have demonstrated that the proposed IDLat can generate latent representations of compact size and correspond well to the importance definition, there are still several limitations to our work.

First, the generalizability of our method has not been fully investigated. Our evaluation results demonstrate that a model trained using data blocks from several time steps can generalize well to other time steps and is sensitive to different importance maps. However, to what extent our model can generalize is not fully understood.

Secondly, in the practical use of our method, the importance definition and dataset itself may not always be in the same resolution. For example, in our isosurface-based importance definition, importance is continuously defined, while scalar data are only defined in the grid points, which forces us to sample the importance field to match the data resolution. How we interpolate and sample the importance field or the dataset can largely influence the latent representation quality.

Finally, the full potential of latent representation for scientific data analysis has not been extensively studied in this work. For example, the usage of importance-driven latent vectors on time-varying data analysis and feature tracking is one of our future studies.

8 CONCLUSION

In this paper, we present an importance-driven latent generation method (IDLat) based on an autoencoder model which tightly relates latent representations to specific data of interest, such as salient regions or features of interest. We represent data of interest by spatial importance maps and utilize the location-wise importance information to guide latent generation. With a trained model, scientists can flexibly define various importance criteria and obtain different latent representations. We further reduce the latent size through a lossless entropy coding model. In addition, we develop a visual exploration tool for latent space analysis and demonstrate the efficiency of identifying and analyzing feature regions with importance-driven latent representations. Through quantitative and qualitative evaluations, we validate the effectiveness of our importance-driven latent generation method in representing data under domain interests control.

ACKNOWLEDGMENTS

This work is supported in part by US Department of Energy SciDAC program DE-SC0021360, National Science Foundation Division of Information and Intelligent Systems IIS-1955764, and National Science Foundation Office of Advanced Cyberinfrastructure OAC-2112606. This research was also supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20200065DR (LA-UR-22-23024).

REFERENCES

- [1] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici. Nonlinear Transform Coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.
- [2] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-End Optimization of Nonlinear Transform Codes for Perceptual Quality. In *2016 Picture Coding Symposium (PCS)*, pp. 1–5. IEEE, 2016.
- [3] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-End Optimized Image Compression. *arXiv preprint arXiv:1611.01704*, 2016.
- [4] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational Image Compression With a Scale Hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.
- [5] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.
- [6] M. Berger, J. Li, and J. A. Levine. A Generative Model for Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 25(4):1636–1650, 2018.
- [7] A. Biswas, S. Dutta, E. Lawrence, J. Patchett, J. C. Calhoun, and J. Ahrens. Probabilistic Data-Driven Sampling via Multi-Criteria Importance Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 27(12):4439–4454, 2020.
- [8] S. Bruckner and T. Möller. Isosurface Similarity Maps. In *Computer Graphics Forum*, vol. 29, pp. 773–782. Wiley Online Library, 2010.
- [9] K. Burger, P. Konradtveit, J. Kruger, and R. Westermann. Importance-Driven Particle Techniques for Flow Visualization. In *2008 IEEE Pacific Visualization Symposium*, pp. 71–78. IEEE, 2008.
- [10] C. Chen, D. Gong, H. Wang, Z. Li, and K.-Y. K. Wong. Learning Spatial Attention for Face Super-Resolution. *IEEE Transactions on Image Processing*, 30:1219–1231, 2021. doi: 10.1109/TIP.2020.3043093
- [11] H.-C. Cheng, A. Cardone, S. Jain, E. Krokos, K. Narayan, S. Subramaniam, and A. Varshney. Deep-Learning-Assisted Volume Visualization. *IEEE transactions on visualization and computer graphics*, 25(2):1378–1391, 2018.
- [12] H. Dai, Y. Tao, X. He, and H. Lin. IsoExplorer: An Isosurface-Driven Framework for 3D Shape Analysis of Biomedical Volume Data. *Journal of Visualization*, 24:1253 – 1266, 2021.
- [13] J. Duda, K. Tahboub, N. J. Gadgil, and E. J. Delp. The Use of Asymmetric Numeral Systems as An Accurate Replacement for Huffman Coding. In *2015 Picture Coding Symposium (PCS)*, pp. 65–69, 2015. doi: 10.1109/PCS.2015.7170048
- [14] W. Fox, M. Wolf, J. Logan, J. Y. Choi, S. Klasky, and T. Kurc. Feature-Relevant Data Reduction for In Situ Workflows. In *The 4th International Workshop on Data Reduction for Big Scientific Data (DRBSD-4)*, 2018.
- [15] B. Friesen, A. Almgren, Z. Lukić, G. Weber, D. Morozov, V. Beckner, and M. Day. In Situ and In-Transit Analysis of Cosmological Simulations. *Computational Astrophysics and Cosmology*, 3(1):1–18, 2016.
- [16] L. J. Gosink, C. Garth, J. C. Anderson, E. W. Bethel, and K. I. Joy. An Application of Multivariate Statistical Analysis for Query-Driven Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):264–275, 2010.
- [17] J. Gu, H. Lu, W. Zuo, and C. Dong. Blind Super-Resolution with Iterative Kernel Correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1604–1613, 2019.
- [18] 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*, 26(4):1732–1744, 2020. doi: 10.1109/TVCG.2018.2880207
- [19] J. Han, H. Zheng, Y. Xing, D. Z. Chen, and C. Wang. V2V: A Deep Learning Approach to Variable-to-Variable Selection and Translation for Multivariate Time-Varying Data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1290–1300, 2021. doi: 10.1109/TVCG.2020.3030346
- [20] G. Ji and H.-W. Shen. Feature Tracking using Earth Mover’s Distance and Global Optimization. In *Pacific graphics*, vol. 2. Citeseer, 2006.
- [21] G. Ji, H.-W. Shen, and R. Wenger. Volume Tracking Using Higher Dimensional Isosurfacing. In *IEEE Visualization, 2003. VIS 2003.*, pp. 209–216, 2003. doi: 10.1109/VISUAL.2003.1250374
- [22] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. In *Computer Graphics Forum*, vol. 38, pp. 59–70. Wiley Online Library, 2019.
- [23] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In Y. Bengio and Y. LeCun, eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [24] C. Li, W. Xia, Y. Yan, B. Luo, and J. Tang. Segmenting Objects in Day and Night: Edge-Conditioned CNN for Thermal Image Semantic Segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3069–3082, 2021. doi: 10.1109/TNNLS.2020.3009373
- [25] H. Li and H.-W. Shen. Local Latent Representation based on Geometric Convolution for Particle Data Feature Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 2022.
- [26] J. Liu, S. Di, K. Zhao, S. Jin, D. Tao, X. Liang, Z. Chen, and F. Cappello. Exploring Autoencoder-Based Error-Bounded Compression for Scientific Data. In *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 294–306. IEEE, 2021.
- [27] Y. Peng, L. Chen, and J.-H. Yong. Importance-Driven Isosurface Decimation for Visualization of Large Simulation Data Based on OpenCL. *Computing in Science & Engineering*, 16(1):24–32, 2013.
- [28] W. P. Porter, Y. Xing, B. R. von Ohlen, J. Han, and C. Wang. A Deep Learning Approach to Selecting Representative Time Steps for Time-Varying Multivariate Data. In *2019 IEEE Visualization Conference (VIS)*, pp. 1–5, 2019. doi: 10.1109/VISUAL.2019.8933759
- [29] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [30] M. Song, J. Choi, and B. Han. Variable-Rate Deep Image Compression through Spatially-Adaptive Feature Transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2380–2389, 2021.
- [31] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.
- [32] I. Viola, M. Feixas, M. Sbert, and M. E. Groller. Importance-Driven Focus of Attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, 2006. doi: 10.1109/TVCG.2006.152
- [33] I. Viola, A. Kanitsar, and M. Groller. Importance-Driven Feature Enhancement in Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005. doi: 10.1109/TVCG.2005.62
- [34] I. Viola, A. Kanitsar, and M. E. Groller. Importance-Driven Volume Rendering. In *IEEE visualization 2004*, pp. 139–145. IEEE, 2004.
- [35] 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. doi: 10.1109/TVCG.2008.140
- [36] C. Wang, H. Yu, and K.-L. Ma. Application-Driven Compression for Visualizing Large-Scale Time-Varying Data. *IEEE Computer Graphics and Applications*, 30(1):59–69, 2009.
- [37] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy. Edvr: Video Restoration with Enhanced Deformable Convolutional Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.
- [38] X. Wang, Y. Li, H. Zhang, and Y. Shan. Towards Real-World Blind Face Restoration with Generative Facial Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9168–9178, 2021.
- [39] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering Realistic Texture in Image Super-Resolution by Deep Spatial Feature Transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 606–615, 2018.
- [40] Y.-S. Wang, C. Wang, T.-Y. Lee, and K.-L. Ma. Feature-Preserving Volume Data Reduction and Focus+Context Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):171–181, 2011. doi: 10.1109/TVCG.2010.34
- [41] Z. Wang, D. Cashman, M. Li, J. Li, M. Berger, J. A. Levine, R. Chang, and C. E. Scheidegger. NeuralCubes: Deep Representations for Visual Data Exploration. *2021 IEEE International Conference on Big Data (Big Data)*, pp. 550–561, 2021.
- [42] S. Wiewel, M. Becher, and N. Thuerey. Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow. In *Computer graphics forum*, vol. 38, pp. 71–82. Wiley Online Library, 2019.
- [43] S. Wiewel, B. Kim, V. C. Azevedo, B. Solenthaler, and N. Thuerey. Latent Space Subdivision: Stable and Controllable Time Predictions for Fluid Flow. In *Computer Graphics Forum*, vol. 39, pp. 15–25. Wiley Online Library, 2020.
- [44] Y. Zhang, H. Guo, L. Shang, D. Wang, and T. Peterka. A Multi-branch

Decoder Network Approach to Adaptive Temporal Data Selection and Reconstruction for Big Scientific Simulation Data. *IEEE Transactions on Big Data*, pp. 1–1, 2021. doi: 10.1109/TBDATA.2021.3092174