# Sparse $L^1$-Autoencoders for Scientific Data Compression

**Matthias Chung**[1], **Rick Archibald**[2], **Paul Atzberger**[3], **Jack Michael Solomon**[1]

[1]Department of Mathematics, Emory University, 400 Dowman Drive, Atlanta, GA, USA
[2]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
[3] Department of Mathematics, University of California Santa Barbara, Santa Barbara, CA 93106, USA

E-mail: matthias.chung@emory.edu

**Abstract.** Scientific datasets present unique challenges for machine learning-driven compression methods, including more stringent requirements on accuracy and mitigation of potential invalidating artifacts. Drawing on results from compressed sensing and rate-distortion theory, we introduce effective data compression methods by developing autoencoders using high dimensional latent spaces that are $L^1$-regularized to obtain sparse low dimensional representations. We show how these information-rich latent spaces can be used to mitigate blurring and other artifacts to obtain highly effective data compression methods for scientific data. We demonstrate our methods for short angle scattering (SAS) datasets showing they can achieve compression ratios around two orders of magnitude and in some cases better. Our compression methods show promise for use in addressing current bottlenecks in transmission, storage, and analysis in high-performance distributed computing environments. This is central to processing the large volume of scientific data, for instance for SAS data being generated at shared experimental facilities around the world to support scientific investigations. Our approaches provide general ways for obtaining specialized compression methods for targeted scientific datasets and is not limited to specific applications.

*Keywords*: autoencoders, compression, sparsity

Submitted to: *Machine Learning: Science and Technology*

## 1. Introduction & Background

Autoencoders are prominent and highly successful architectures for neural networks for extracting information from data sets [4, 23, 29]. They discover inherent structures within data by learning a parameterized encoder $e$ and decoder $d$ such that $x \approx d(e(x))$. Autoencoders also have strong connections to well-established mathematical concepts [2, 48] and are versatile when utilized as generative models [28]. Autoencoders have wide applicability in unsupervised learning environments ranging from denoising [59], anomaly detection [53], image and audio compression [31, 56], to general recommender systems [63].

Most commonly, autoencoders employ low dimensional latent spaces to serve as natural data-driven machine learning techniques for model reduction and data compression [12, 36]. Autoencoders' nonlinear characteristics make them a valuable complement to established techniques like principal component analysis and singular value decomposition [60], Fourier analysis [33], reduced order models [62], and dictionary learning approaches [55].

Mathematically, an autoencoder is a nonlinear parameterized mapping $a : \mathcal{X} \to \mathcal{X}$ with a functional composition $a = d \circ e$, into *encoder* $e : \mathcal{X} \times \Theta_e \to \mathcal{Z}$ and *decoder* $d : \mathcal{Z} \times \Theta_d \to \mathcal{X}$ with trainable parameters $\theta_e \in \Theta_e \subset \mathbb{R}^{n_e}$ and $\theta_d \in \Theta_d \subset \mathbb{R}^{n_d}$. The feature space $\mathcal{Z} \subset \mathbb{R}^\ell$ is referred to as *latent space* while $\mathcal{X} \subset \mathbb{R}^n$ is the *data space*.

While terminology varies, autoencoders are classified based on their model configuration. Standard autoencoders with small dimensional latent spaces $\ell < n$ are referred to as *undercomplete* and are widely utilized, while autoencoders with large dimensional latent spaces $\ell > n$ are referred to as *overcomplete* and are less common. A *shallow* autoencoder is characterized by having only one, while a *deep* autoencoder has more than one hidden layer. Autoencoders typically maintain network symmetry, i.e., the structure of $e$ and $d$ are mirrored such that the decoder transformations mimic the encoder in reverse order. Its general structure makes autoencoders flexible, for instance, giving up the mapping back into the input space has led to encoder-decoder networks which are widely used as likelihood-free surrogate models for physical forward propagation [64] and even inverse modeling [1, 15].

Despite its versatility, various drawbacks associated with autoencoders have been identified. For instance, the undercomplete "hour-glass" autoencoder shape compresses input signals $x \in \mathcal{X}$ into the low-dimensional latent space $z = e(x; \theta_e) \in \mathcal{Z}$ and may lead to corrupted reconstructions, e.g., blurring artifacts in the reconstructed images [42]. This drawback has been highlighted in the generative process of autoencoders, i.e., variational autoencoders. Autoencoders with a large number of trainable network parameters, such as overcomplete autoencoder, tend to overfit, resulting in "identity mappings" countering one of the main purposes of autoencoders: removing unwanted artifacts from the input $x$ [27]. A further drawback includes incorporating scientific and physical features into the network remains a major hurdle, where some initial research is making strides towards this goal [3, 11, 34].

To mitigate the challenges outlined above, we break with one common assumption of autoencoders, that is, we consider utilizing an overcomplete autoencoder framework with sparsity promoting mappings of the latent variable, illustrated in Figure 1. Overcomplete autoencoders are prone to severely overfit without taking mitigating measures. Hence, imposing sparsity onto the latent space variable is a regularizing measure and various strategies have been proposed. Despite being tremendously successful, we recognize that overcomplete autoencoders with sparsity-promoting features in the latent space are severely underutilized.

*Sparse autoencoders* have first been introduced in the 2010s with pioneering work from various research groups including [26, 39, 40, 45]. Here, various strategies have been entertained to promote sparsity of the latent variable. For instance, by selecting a fixed amount of $k$ nonzero elements of the latent variable $z$ with maximal reconstruction features [40]. Another approach limits the number of active latent components by utilizing a binary Bernoulli random variable model realized through a Kullback-Leibner divergence penalty [45]. A third approach, which we will follow here, is to use the compressed sensing framework via $L^1$-regularization [26]. Recent years have brought advances, various extensions of sparse autoencoders have been developed, and scientific applications considered [5, 27, 35, 38, 41, 46, 54], however, despite its successes sparse autoencoder have yet to find its way into mainstream applications. We like to point out that the term sparse autoencoders may not refer to sparsity induced onto the the latent variable, but sparsity imposed on the network parameters $\theta$, e.g., see [38, 54].

While research areas such as dictionary learning, manifold learning, and applications like speech recognition have been explored [5, 27, 35, 38, 41, 46, 54], to the best of our knowledge, the core application of sparse autoencoders for compressive representation has not yet been fully addressed. Hence our proposed method utilizes sparse latent space signals to efficiently store input signals. Furthermore, in contrast to prior work on sparse autoencoders, we not only consider sparsity on the latent variable $z$ via $L^1$-regularization, but also promote sparsity on a transformed signal $f(z)$, incorporated through regularization during training. The functional $f$ provides the possibility to promote structure within the latent variable $z$, e.g., through a total variation type (generalized lasso). Note that sparse autoencoders, Figure 1, share limitations of the general class of autoencoder, that is, the ability to effectively generalize to novel data instances, particularly when the training dataset does not accurately reflect the characteristics of the testing dataset.

Our work is structured as follows, in Section 2 we introduce our proposed sparsity promoting autoencoder for data compression tasks, discuss its numerical applications in Section 3, and provide concluding remarks and discuss future work in Section 4.

## 2. Sparse Autoencoders for Scientific Data Compression

Machine learning-based compression algorithms encounter distinct challenges when applied to scientific datasets. These datasets frequently necessitate stringent criteria for individual sample reconstruction accuracy, minimization of artifacts such as blurring, and preservation of critical data properties. To help address these challenges, we use large dimensional information-rich latent spaces and sparsity regularizations. This allows for developing representations amenable to further reductions and compression. Historically, embedding sparse signals into large dimensional vector spaces has had a major impact on signal processing starting in the 1990s with the *compressed sensing framework* [7, 8, 19, 57]. Here, we develop a strategy for autoencoders leveraging latent space dimensions larger than the feature space of the data, e.g., $\ell > n$ while imposing sparsity on features of the latent variable $z$ generating a well-posed compression problem. Let $\theta_e$ and $\theta_d$ be the trainable network parameter of the encoder and decoder, respectively, ideally, we may formulate the network training as

$$\min_{(\theta_e,\theta_d)\in\Theta_e\times\Theta_d} \quad \mathbb{E}\,\|f(e(x;\theta_e))\|_0 \quad \text{subject to} \quad \|d(e(x;\theta_e);\theta_d)-x\|_2 \leq \delta, \tag{1}$$

where $\mathbb{E}$ denotes the expectation over the data $x$, $\|\cdot\|_2$ the $L^2$-norm, $\|f(e(x;\theta_e))\|_0$ is defined as the cardinality of nonzero elements in $f(e(x;\theta_e))$, and $\delta > 0$ represents a desired reconstruction quality. We let $f : \mathcal{Z} \to \mathcal{F}$ be a predefined operator we refer to as the *sparse* or *geometric structure selector*. Solving (1) is NP-hard and approximations need to be developed to obtain efficient methods, [21].

### *2.1. Lossy Compression with $L^1$-Regularizations and $L^2$-Reconstructions*

Under the restricted isometry properties [8], we may reformulate using an $L^1$ convex relaxation of (1) which leads to the generalized lasso problem

$$\min_{(\theta_e,\theta_d)\in\Theta_e\times\Theta_d} \quad \mathbb{E}\,\|d(e(x;\theta_e);\theta_d)-x\|_2^2 + \lambda\,\|f(e(x;\theta_e))\|_1\,. \tag{2}$$

Comparable sparsity can be achieved with a suitable choice of $\lambda > 0$, see [7, 14] for details. This can be viewed as a rate-distortion objective, where $\lambda\,\|\cdot\|_1$ serves as a measure of the compression rate and $\|\cdot\|_2$ for the reconstruction distortion [17].

In our framework we propose to utilize an operator $f$ with the benefits that allow for going beyond the standard approach to sparsity based on the latent variable $z$ with

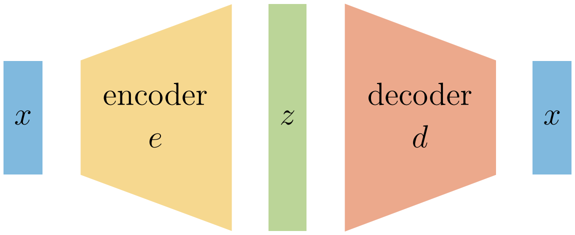

**Figure 1.** Overcomplete autoencoder architecture, where the latent space dimension $\ell$ is bigger than the input dimension $n$. Given the ill-posed nature of neural network training, overcomplete autoencoders require additional regularization, such as sparsity-promoting $L^1$-regularization. In our case, this regularization is applied to a function $f$ of the latent variable $z$.

a $L^1$-regularization e.g., $\|e(x;\theta_e)\|_1$. The standard lasso approach restricts sparsity to be enforced only on each component of $z$ individually, hence the latent variable carries only minimal interpretability or structure [41]. However, when using a mapping $f$ we may enforce further structure to the latent space variable $z$. The function $f$ effectively provides alternative features for the latent variable $z$ providing further representations on which we can impose the sparsity conditions. This can be used to remedy disadvantages of purely $L^1$-regularizations [52] and allows in some cases for representations with additional geometric interpretability [47]. For simplicity, we use here a total variation regularization approach $f(\,\cdot\,) = \nabla(\,\cdot\,)$, where $\nabla$ is the gradient operation. In practice, for finite dimensional spaces $\mathcal{Z}$, we approximate this operation by the finite difference operator $(f(z))_i = (z_{i+1} - z_i)/h$, e.g., with $h = 1$. This is also used to help in clustering of information in the latent space [47].

As a consequence of our autoencoder having a larger dimensional latent space and associated parameter space, our approaches have capacities enabling for discovering collective features allowing for going beyond just encoding and decoding training image individually within the network. Our numerical results in Section 3 clearly show that our approach generalizes well to testing data. Sparse autoencoders reduce overall storage by leveraging an overcomplete latent space in which only a small subset of components are active, yielding compact and efficient representations of the input data. During training, this sparsity is promoted through $L^1$ regularization applied to $f(z)$, a function of the latent variable, encouraging concise encodings that retain the essential structure of the signal. In our methods, the regularization parameter $\lambda$ balances potential over- and underfitting. Small $\lambda$ values may generate autoencoder with identity mappings for training data but may not generate any sparsity within the sparse structure selector. On the other hand large $\lambda$ may produce significant sparsity while missing to reconstruct the input signal $x$. Cross-validation techniques are readily available for calibrating the hyperparameter $\lambda$, however they are computationally expensive, requiring the training of $NK$ autoencoders when performing $K$-fold validation over $N$ candidate values—highlighting the need for more efficient calibration strategies and warranting future investigations.

### *2.2. Sparsity Promoting Linear Autoencoder*

To illustrate the advantages of sparsity-promoting autoencoders, we first consider a linear autoencoder $A \in \mathbb{R}^{n\times n}$. Assuming we have a $\ell$ dimensional latent space we may compute a generic optimal autoencoder by minimizing the Bayes risk, i.e.,

$$\min_{\operatorname{rank}(A)\leq\ell} \ \mathbb{E}\,\|(A-I)x\|_2^2\,, \tag{3}$$

where $I$ denotes the identity mapping. Assuming the random variable $x$ has symmetric positive definite second moment $\mathbb{E}\ xx^\top = \Gamma$ with Cholesky decomposition $\Gamma = BB^\top$, then

$$\mathbb{E}\ \|(A-I)x\|_2^2 = \operatorname{tr}\big((A-I)\Gamma(A^\top - I)\big) = \|AB - B\|_{\mathrm{F}}^2 \tag{4}$$

and Equation (3) is equivalent to

$$\min_{\operatorname{rank}(A)\leq \ell} \|AB - B\|_{\mathrm{F}}^2 , \tag{5}$$

where $\|\cdot\|_{\mathrm{F}}$ deontes the Frobenius norm.

Without any rank constraints $\ell = n$ we can immediately follow that $A = I$ is a solution. However, for rank constraint problems, we require a generalization of the Eckart–Young–Mirsky theorem. Which we here state as follows.

**Theorem 2.1** ([13, 22])**.** *Let matrices $B \in \mathbb{R}^{m\times n}$ and $L \in \mathbb{R}^{q\times n}$ with $k = \operatorname{rank}(L)$ be given. Further, let $L = U\Sigma V^\top$ denote the SVD and $U_k\Sigma_k V_k^\top$ the rank $k$ truncated skinny SVD of $L$. Then*

$$\widehat{A} = [BV_kV_k^\top]_\ell L^\dagger$$

*is a solution to the minimization problem*

$$\min_{\operatorname{rank}(A)\leq \ell} \|AL - B\|_{\mathrm{F}}^2 ,$$

*having a minimal $\|A\|_{\mathrm{F}}$. This solution is unique if and only if either*

$$\ell \geq \operatorname{rank}\left(BV_kV_k^\top\right)$$

*or $1 \leq \ell < \operatorname{rank}\left(BV_kV_k^\top\right)$ and $\sigma_\ell(BV_k^\top) > \sigma_{\ell+1}(BV_k^\top)$.*

The $[\,\cdot\,]_\ell$ refers to the rank $\ell$ approximation of a matrix. Let $U = [u_1, \dots, u_\ell, u_{\ell+1}, \dots, u_n]$ be the left singular vectors, $V = [v_1, \dots, v_\ell, v_{\ell+1}, \dots, v_n]$ be the right singular vectors, and $\Sigma = \begin{bmatrix} \Sigma_\ell & 0 \\ 0 & \widetilde{\Sigma}_\ell \end{bmatrix}$, be the diagonal matrix with sorted singular values $\sigma_j \geq \sigma_{j+1}$ and singular vectors, i.e., $U = \begin{bmatrix} U_\ell & \widetilde{U}_\ell \end{bmatrix}$.

*Proof.* See [13, 22] □

Using Theorem 2.1 for Equation (5) we obtain the following result.

**Theorem 2.2.** *Let matrix $B \in \mathbb{R}^{n\times n}$ have full row rank with SVD given by $B = U\Sigma V^\top$. Then*

$$\widehat{A} = U_\ell U_\ell^\top$$

*is a solution to the minimization problem*

$$\min_{\operatorname{rank}(A)\leq \ell} \|AB - B\|_{\mathrm{F}}^2 ,$$

*having a minimal Frobenius norm $\left\|\widehat{A}\right\|_{\mathrm{F}} = \sqrt{\ell}$ and $\left\|\widehat{A}B - B\right\|_{\mathrm{F}}^2 = \sum_{k=\ell+1}^n \sigma_k(B)$. This solution is unique if and only if either $\ell = n$ or $1 \leq \ell < n$ and $\sigma_\ell(B) > \sigma_{\ell+1}(B)$.*

Following this result, the obvious choice for the autoencoder $\widehat{A} = \widehat{D}\widehat{E}$, with encoder and decoder being $\widehat{E} = U_\ell^\top$ and $\widehat{D} = U_\ell$, respectively. Note that this decomposition is not unique, e.g., let $K$ be any $n \times n$ invertible matrix then $\widehat{E} = U_\ell^\top K$ and $\widehat{D} = K^{-1}U_\ell$, are valid choices.

*Proof.* Using Theorem 2.1, the solution to

$$\min_{\mathrm{rank}(A)\leq \ell} \|AB - B\|_{\mathrm{F}}^2$$

is given by

$$\widehat{A} = [BVV^\top]_\ell B^{-1} = [U\Sigma V^\top]_\ell B^{-1} = U_\ell \Sigma_\ell \begin{bmatrix} I \\ 0 \end{bmatrix} \Sigma^{-1} U^\top = U_\ell \Sigma_\ell \begin{bmatrix} I \\ 0 \end{bmatrix} \Sigma^{-1} U^\top = U_\ell U_\ell^\top,$$

where $U_\ell$ contains the first $\ell$ columns of orthogonal matrix $U$. Note that

$$\begin{aligned} \left\|U_\ell U_\ell^\top U\Sigma V^\top - U\Sigma V^\top\right\|_{\mathrm{F}}^2 &= \left\|U^\top U_\ell U_\ell^\top U\Sigma - U^\top U\Sigma\right\|_{\mathrm{F}}^2 && (6)\\ &= \left\|U^\top U_\ell U_\ell^\top U\Sigma - \Sigma\right\|_{\mathrm{F}}^2. && (7)\end{aligned}$$

Hence, with its definition $U = \begin{bmatrix} U_\ell & \widetilde{U}_\ell \end{bmatrix}$ we have

$$U^\top U_\ell = \begin{bmatrix} U_\ell^\top \\ \widetilde{U}_\ell^\top \end{bmatrix} U_\ell = \begin{bmatrix} U_\ell^\top U_\ell \\ \widetilde{U}_\ell^\top U_\ell \end{bmatrix} = \begin{bmatrix} I_\ell \\ 0 \end{bmatrix} \quad \text{and} \quad U^\top U_\ell U_\ell^\top U = \begin{bmatrix} I_\ell \\ 0 \end{bmatrix} \begin{bmatrix} I_\ell & 0 \end{bmatrix} = \begin{bmatrix} I_\ell & 0 \\ 0 & 0 \end{bmatrix}.$$

Resulting in

$$\left\|\widehat{A}B - B\right\|_{\mathrm{F}}^2 = \left\|U^\top U_\ell U_\ell^\top U\Sigma - \Sigma\right\|_{\mathrm{F}}^2 = \left\|\begin{bmatrix} \Sigma_\ell & 0 \\ 0 & 0 \end{bmatrix} - \Sigma\right\|_{\mathrm{F}}^2 = \left\|\begin{bmatrix} 0 & 0 \\ 0 & \widetilde{\Sigma}_\ell \end{bmatrix}\right\|_{\mathrm{F}}^2 = \sum_{k=\ell+1}^{n} (\sigma_k(B))^2. \tag{8}$$

□

The theorem above provides us with a squared error bound $e_{\mathrm{ls}} = \sum_{k=\ell+1}^n (\sigma_k(B))^2$ we can expect a linear autoencoder may achieve. Let us now consider the problem of finding an optimal linear autoencoder $A$ with the decomposition $A = DE$ into encoder $E \in \mathbb{R}^{\ell\times n}$ and $D \in \mathbb{R}^{n\times\ell}$ where $\ell > n$ by minimizing $L^1$-regularized optimization problem

$$\min_{D\in\mathbb{R}^{n\times\ell}, E\in\mathbb{R}^{\ell\times n}} \mathbb{E}\left\|(DE - I)x\right\|_2^2 + \lambda \left\|Ex\right\|_1 \tag{9}$$

with $\lambda > 0$. For simplicity of illustration, we chose $f = I$. Note under the assumptions stated above, given an encoder $\widetilde{E}$, a corresponding optimal decoder $\widetilde{D}$, can be computed using Theorem 2.2 resulting in $\widetilde{D} = (\widetilde{E}BB^\top\widetilde{E}^\top)^\dagger B\widetilde{E}B$, where † denotes the Moore-Penrose pseudoinverse. Can we achieve a squared error bound of similar magnitude as $e_{\mathrm{ls}}$ using while promoting sparsity in $Ex$? Recall, for any $\ell > n$ we may always find encoder and decoders $E$ and $D$ such that $\mathbb{E}\left\|(DE-I)x\right\|_2 = 0$. To investigate sparsity, let us consider the decompositions $D = \begin{bmatrix} D_1 & D_2 \end{bmatrix}$ and $E = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$ with $D_1, E_1 \in \mathbb{R}^{n\times n}$ then $D_2 \in \mathbb{R}^{n\times(\ell-n)}$, and $E_2 \in \mathbb{R}^{(\ell-n)\times n}$. We further assume that $D_1E_1 = I$ and $D_2E_2 = 0$. For arbitrary invertible $K \in \mathbb{R}^{n\times n}$ we let $D_1 = \alpha K^{-1}$ and $E_1 = \frac{1}{\alpha}K$, $\alpha > 0$ and the condition $\mathbb{E}\left\|(DE-I)x\right\|_2 = 0$ is trivially fulfilled and the optimization reads

$$\min_{D_2\in\mathbb{R}^{n\times r}, E_2\in\mathbb{R}^{\ell-n\times n}} \mathbb{E}\left\|D_2E_2x\right\|_2^2 + \lambda\left\|\begin{bmatrix}\frac{1}{\alpha}K \\ E_2\end{bmatrix}x\right\|_1 = \mathbb{E}\left\|D_2E_2x\right\|_2^2 + \tfrac{\lambda}{\alpha}\left\|Kx\right\|_1 + \lambda\left\|E_2x\right\|_1.$$

Hence for any $\epsilon > 0$ there exists an $\alpha > 0$ such that $\frac{\lambda}{\alpha}\left\|Kx\right\|_1 < \epsilon$ which is achieved by the numerical sparsity vector $(1/\alpha K)x$. As for the remaining $\ell - n$ elements we may gain additional sparsity with the selection of $E_2$ and $D_2$ with $\mathbb{E}\left\|D_2E_2x\right\|^2 \approx e_{\mathrm{ls}}$. Consequently, the generalized lasso approach may generate sparse vectors $Ex$ while maintaining the same expected squared error as an undercomplete linear autoencoder in Equation (3).

### *2.3. Neural network Training*

Given a representative set of (unsupervised) training samples $\{x_j\}_{j=1}^{J}$ we minimize the empirical generalized lasso

$$\min_{\theta_e,\theta_d\in\Theta} \quad \tfrac{1}{J}\sum_{j=1}^{J}\left\|d(e(x_j;\theta_e);\theta_d) - x_j\right\|_2^2 + \lambda\left\|f(e(x_j;\theta_e))\right\|_1, \tag{10}$$

Our developed methods leverage results in compressed sensing showing promise for having a significant impact on scientific compression techniques. Under mild assumptions, compressed sensing has shown high compression rates, far below theoretical Nyquist rates [9, 10]. Benefiting from its advantages in a trainable deep neural networks provides significant compression rates while maintaining high accuracy of the signal itself in Section 3. Theoretically our methods have also connections to dictionary learning frameworks. Sparse dictionary learning provides good reconstruction of a sparse selection of dictionary atoms [20, 30, 44, 58]. Here, since linear, zeros in its dictionary representation does not carry any information. However, utilizing sparsifying autoencoders allows for nonlinear transformations and therefore enriches information carried by the latent variable. Consequently, even "zero" elements in the latent variable $z$ carry information of the underlying signal. The autoencoder architecture for our numerical examples is given in Figure 2. The network is a fully connected five layer symmetric neural network with hidden layer size $(m, \ell, m)$, ReLU activation function between each layer, and input/output size $n$. Furthermore, the generic design of the neural network architecture provides an additional level of flexibility toward efficient encoding and decoding of the underlying signals $x$.

Compressed sensing has shown high compression rates. Leveraging and combining these methods based on a trainable and sparsity promoting deep neural network provides approaches for significant compression rates while maintaining a high level of accuracy of the signal. We note that ratio of encoder parameters size to image size is given as $r_e = \frac{\ell * m + m * n + m + n}{n} = 1 + \frac{m}{n} * (\ell + n + 1)$. This ratio exceeds one for any decoder architecture, indicating that the size of the decoder will always be larger than the image size. When employing autoencoders for compression, it is generally assumed that the one-time cost of transmitting the decoder to the receiver will be amortized over multiple image transfers, making it negligible in this context. To illustrate the difference in the number of images that need to be transferred to amortize the fixed cost of decoder transfer, we will compare the various decoders used in this study. Here, assuming an image size of $n = 64 \times 64$, the smallest autoencoder analyzed in this study, which is emblematic of traditional autoencoders, has a ratio $r_e = 1153.25$. In contrast, the largest autoencoder, which features latent spaces larger than the image size, has a ratio $r_e = 4610$. This demonstrates that the difference in amortizing the sizes of the smallest and largest autoencoders is not significantly radical..

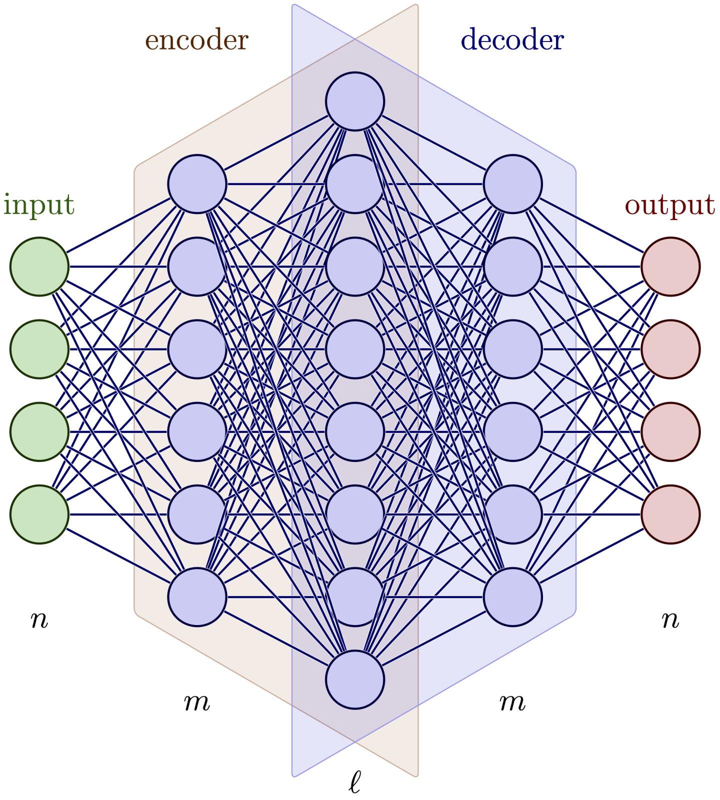

**Figure 2.** The autoencoder architecture for our numerical examples.

### *2.4. Lossless Arithmetic Compression Methods*

To further compress our representations $z$, we develop methods by combining our approaches with lossy quantization and lossless entropy encoding [17, 24]. We represent the information in $z$ as a list of the $M$ indices of nonzero entries $i_1, \ldots, i_M$ and the weights at these locations $w_k = z_{i_k}$. The indices are sorted from smallest to largest. We represent this by storing the distances between successive indices $\delta_k = i_{k+1} - i_k$ along with a termination symbol $\iota$ to obtain the sequence $\delta_1, \ldots, \delta_{M-1}, \iota$.

We expect in practice for most datasets that the probability distribution $\rho(\delta_i)$ over the differences $\delta_i$ will tend to skew toward the smaller values, such as having $\delta_i < \ell/2$ for most entries. By modeling this distribution we can obtain gains in the compression using an entropy encoder. In entropy encoding, from results in information theory [17] the optimal encoder would compress the data by assign to data $\delta$ a code-word of length ideally $\ell(\delta) = -\log(p(\delta))$. This would give the optimal average code-word length $\bar{\ell} = \mathbb{E}[\ell(\delta)] = \sum_\delta -p(\delta)\log(p(\delta))$ which is the entropy of the distribution $p(\delta)$, hence the name entropy encoding. In practice, since the $-\log(p(\delta))$ will seldom be an integer we can only obtain an encoding approximating this optimal average code-word length [17]. To obtain a lossless entropy encoding for our model probability distribution $\tilde{\rho}(\delta)$, we develop a lossless Arithmetic Coding compression method $\mathcal{A}$ to obtain $c = \mathcal{A}(\delta; \tilde{\rho}(\cdot))$, [32, 51, 61], see Figure 3.

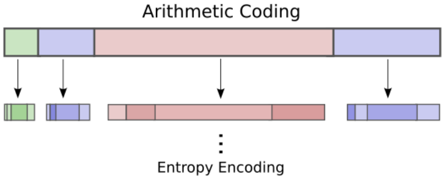

**Figure 3.** Arithmetic Coding is used to obtain an entropy encoder using our given model probability distribution for symbols. This is accomplished by encoding a description of the sequence of symbols in terms of successively divided intervals that have sizes at each stage proportional to our given model probability distribution.

In Arithmetic Coding, data is represented as the leading binary digits of a real-number that is represented by a binary sequence that specifies the number's location on the real-line by successive interval partitions that are proportional to the probability of a symbol's occurrences. The compression exploits the contrast with standard binary real-number encoding which uses equal-sized partitions to successively halve the interval giving similar encoding lengths for numbers within each interval. We show an illustration of the non-uniformity of the partitions that can arise, and even vary depending on the particular entry location within the sequence, see Figure 3. Arithmetic Coding can achieve near optimal encoding with average code-word lengths approaching the sequence entropy. For reconstructions of the data during decoding, the same relative probabilities of symbols are used as during encoding. The partitions which are generated by these probabilities are used to localize the real-number whose leading binary digits gives the uncompressed data [32, 51, 61].

Our model for $\tilde{\rho}(\delta)$ for the probabilities is obtained by fitting a shifted Gaussian-like form $\tilde{\rho}(\delta) = q(\delta)/Z$, where $Z = \sum_\delta q(\delta)$. The $q(\delta)$ are weights that combine a normally distributed density and uniform distribution of the form $P(x; \mu, \sigma^2, c_0) = (2\pi\sigma^2)^{-1/2}\exp(-(x-\mu)^2/2\sigma^2) + c_0$ with $\mu = 0, \sigma^2 = (\ell/3)^2$, and $c_0 = 10^{-2}$. To compress the weights $\{w_k\}$ we use lossy quantization of the latent weights $\tilde{w} = \mathcal{Q}(w)$, such as using 16-bit floating-points, [24, 43, 49]. This provides for $z$ the compressed representation $(c, \tilde{w})$. We use these methods to provide further compression of our data in addition to the sparsity.

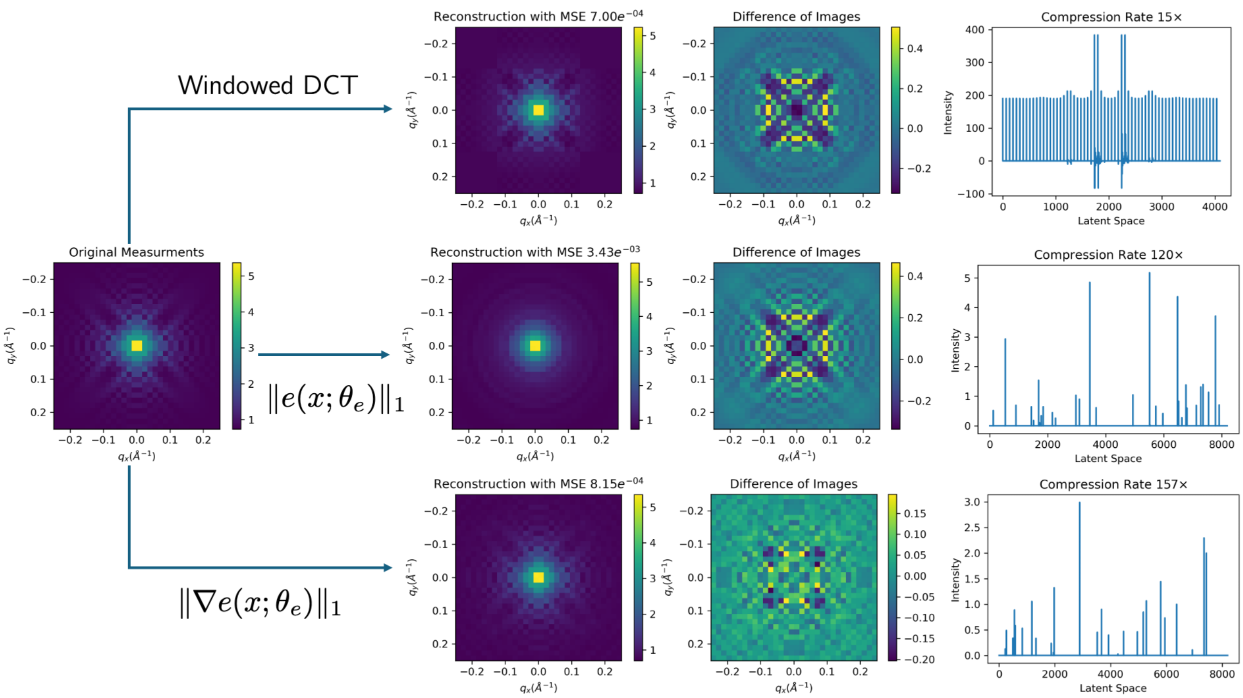

**Figure 4.** We show a representative testing input $x_j$ for the SAS application in the first column, while its reconstructions using a traditional windowed DCT on the top, sparse autoencoder networks $(1, 2, 1/2, z)$ in the middle and $(1, 2, 5, \nabla z)$ on the bottom are presented in the second column. The average pixel reconstruction error $\|d(e(x_j;\theta_e);\theta_d)-x_j\|_2/n$ are $7.04 \times 10^{-4}$, $3.43 \times 10^{-3}$ and $8.15 \times 10^{-4}$, respectively, where $n$ is the number of pixels. We show the latent space variable $z_j = e(x_j; \theta_e)$ in the third column. The latent variable $z_j$ each contains 264, 34 and 26 non-zero elements. With an original image size of $64 \times 64$ the resulting compression ratio $\|x_j\|_0$ to $\|e(x_j;\theta_e)\|_0$ and $\|x_j\|_0$ to $\|\nabla e(x_j;\theta_e)\|_0$ are $15 : 1$, $120 : 1$ and $157 : 1$.

## 3. Numerical Investigations

We illustrate the significant advantages our novel approach carries on simulated small angle scattering (SAS) data, a technique that is ubiquitous across the world's X-ray light and neutron facilities. The central objective of small-angle scattering (SAS) experiments is to non-invasively probe the nanoscale structure of materials by capturing information about their size, shape, internal arrangement, and overall morphology. This makes SAS a powerful tool for characterizing material properties across a wide range of scientific disciplines. We utilize the tool SASView [50], a community-based tool used at experimental facilities to analyze and simulate SAS experiments. By using SASView we are able to accurately approximate the broad range of experimental data that can be produced by different instruments and different facilities, for different conditions and material samples. This also give us the opportunity to validate our methods for newly generated data. For all SAS experiments simulated, we set the number of sensors to be uniformly spaced with $n = 64 \times 64$. Measurement sensors for SAS experiments are some form of charge-coupled device (CCD), so uniform spacing is common. Given limited range of material properties (molecular arrangement, size, shape, and structure), a low-dimensional manifold may effectively represent this data, facilitating the possibility of high data compression rates.

All networks in this section use the same autoencoder architecture depicted in Figure 2 and this architecture follows that depicted in Figure 1. For a given input of size $n$ and loss defined in Equation (10), we adopt the notation $\left(\frac{m}{n}, \frac{\ell}{n}, \lambda, f(z)\right)$ to uniquely define all networks used in this paper (e.g., $n$, input and target dimension, $m$ intermediate network layer dimension, $\ell$ latent dimension, $\lambda$ sparsity regularization parameter, and the geometric structure selector $f$). We further report that all networks were trained using RAdam optimization, introduced in 2019 [37] and implemented by Pytorch [18], with learning rate of $1 \times 10^{-4}$ a batch size of 512 and 50 epochs used for the statistics of Table 1. We note that training and performance of all networks in this study displayed sensitivity to optimization method, batch size, learning rate, and number of epochs – a common issue for nonlinear stochastic optimization problems such as those that occur in machine learning. We keep all these network architecture choices fixed throughout, with the slight modification of additional epochs for the main example, in order to show the general characteristics of sparse optimization in the latent space of autoencoder. All computations were performed on Oak Ridge National Laboratory's (ORNL) Compute and Data Environment for Science (CADES) cluster [16]. In the initial phases of our investigation, we used convolution-type architectures which generally use convolution operators at the beginning and end of the network. We observed that the filtering aspect of the convolutional neural network dominated and hindered all information from reaching the latent space layer that connects the encoder and decoder.

We demonstrate the advantages of our proposed method for realistic configurations of SASView [25]. We are able to highly compress all simulations from this package. We begin by randomly generating 50,000 images using the aforementioned sensor configuration of $n = 64 \times 64$, which is characteristic of the range of SAS experimental data collected at scattering facilities. The representative result of this investigation is presented in Figure 4, which demonstrates high compression rates with high accuracy for the networks $(1, 2, {}^{1}/_{2}, z)$ and $(1, 2, 5, \nabla z)$. For comparison, we also present the results of using a windowed Discrete Cosine Transform (DCT) [6], a common technique used in many image and video compression methods. We use a standard window size of eight pixels.

We demonstrate that we have captured all realistic configurations of SASView [25], using our train networks in Figure 5. Here we sample again a much denser set of measurements using 150,000 images for testing. The takeaway from this analysis is that we can maintain the same level of compression and accuracy for both testing and training data. Additionally, it is demonstrated for the same number of network parameters, the sparsity promoting function

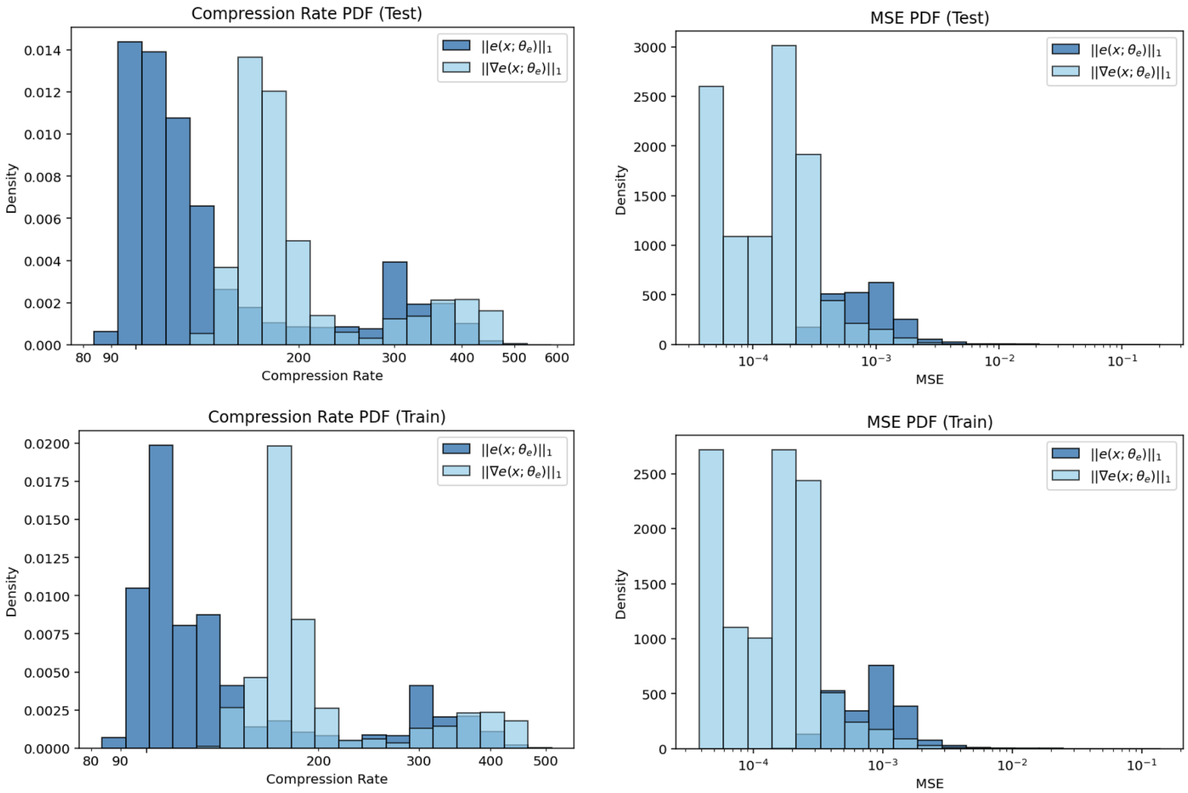

**Figure 5.** We show the error and compression rate probability densities for the sparse autoencoder networks $(1, 2, {}^{1}/_{2}, z)$ and $(1, 2, 5, \nabla z)$. The top left plot displays the distributions of the testing error, which consists of 150,000 independently random SAS images generated by SASView post-training. The bottom left plot displays the distributions in the errror in the 50,000 random SAS images generated by SASView for training. The mean training errors of both approaches are $2.48 \times 10^{-3}$ and $8.00 \times 10^{-4}$, respectively. Correspondingly, the mean training compression rates are 153× and 216×. Note that these values only alter insignificantly for the testing set.

$f(z) = \nabla z$ significantly improved compression rates and accuracy. Again this is visually represented in Figure 4 where this increased accuracy is able to maintain a more complex scattering pattern that can occur in SAS experiments.

Figure 6 presents an ablation study that quantifies the sensitivity of the latent space in sparse autoencoder networks, specifically $(1, 2, {}^{1}/_{2}, z)$ and $(1, 2, 5, \nabla z)$. The figure illustrates the difference in the ratio when a latent space term is zeroed compared to its normal state. Most latent space variables exhibit low sensitivity; however, there is a small subset of variables that show high variability, which aligns with the sparsity patterns observed in both the training and testing datasets.

Our methods can also be combined with further lossy and lossless methods to obtain

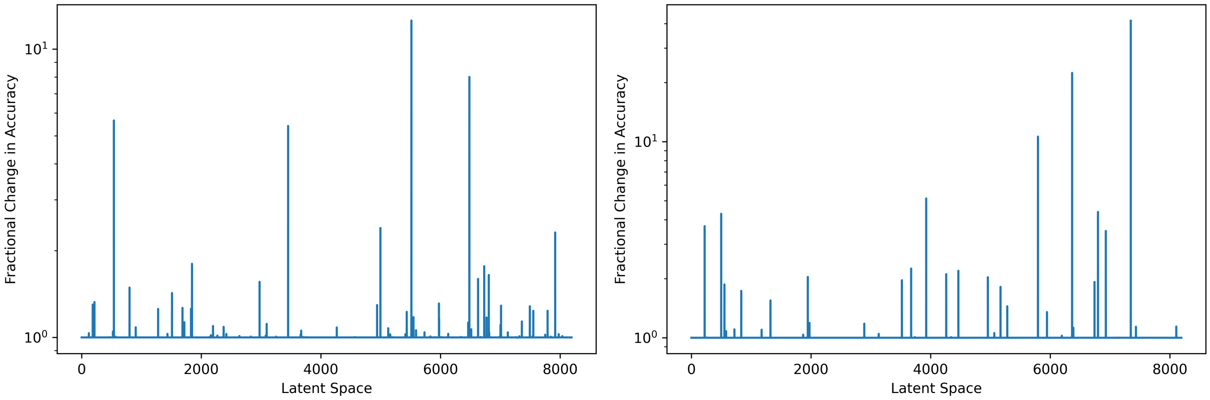

**Figure 6.** Ablation study in the latent space for sparse autoencoder networks $(1, 2, {}^1\!/_2, z)$ and $(1, 2, 5, \nabla z)$, on the left and right respectively. Here we report the ratio ${\|d(e(x;\theta_e)_{i\mapsto 0};\theta_d)-x\|_2}/{\|d(e(x;\theta_e);\theta_d)-x\|_2}$ where $e(x;\theta_e)_{i\mapsto 0}$ represents replacing the $i^{th}$ element in the latent space with zero.

further compression. As discussed above, we represent the information in $z$ as a list of differences in the $M$ indices of non-zero entries to obtain the sequence $\delta_1, \ldots, \delta_{M-1}, \iota$ and the weights $w_k$ at these indices to obtain $z \to (\delta, w)$. Here, we use arithmetic coding $\mathcal{A}$ to develop for $\delta$ lossless compression methods $c = \mathcal{A}(\delta; \rho(\cdot))$ [32, 51, 61]. We also further could quantize $\mathcal{Q}$ for lossy compression of $w$ as $\tilde{w} = \mathcal{Q}(w)$, such as using lower-precision floating-points [24, 43]. We currently use the full precision floating-point representations of the data. For $\delta$, we leverage that the probability distribution $\rho(\delta)$ will tend to skew to the left, for the SAS data see Figure 7. As an initial model for this distribution, we use a Gaussian-like form $\rho(\delta) = q(\delta)/Z$, where $Z = \sum_\delta q(\delta)$ where $q(\delta)$ is normally distributed with density $P(\delta; 0, \sigma^2)$ where $P(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp(-(x-\mu)^2/2\sigma^2)$. To help ensure efficient encoding on future samples, we used parameters having good coverage with $\sigma^2 = 10^3$ and $c_0 = 10^{-3}$. We found the lossless compression methods provide on average a compressed representation 79% of the uncompressed $\delta$. Combining this with lossy quantization of the weights from 64-bit floating-points to 16-bit floating-points [24, 43, 49], yields an overall compressed representation $(c, \tilde{w})$ that is 52% of the uncompressed case. These methods provide an additional factor of around 2× to the already favorable compression ratios achieved by the sparsity.

In order to compare and demonstrate the potential of high dimensional sparse autoencoding, we present the results of Table 1 which compares a fully connected encoder-decoder with the typical hourglass framework without the sparsity promoting $L^1$-norm in (10), to encoder-decoder with sparsity promoting loss functions. This table also compares the effect of increasing the dimension of the latent space, where the statistics are obtained by running twenty random initial conditions for each tested network architecture in the table using the same testing and training data described at the beginning of this section. The first entry in

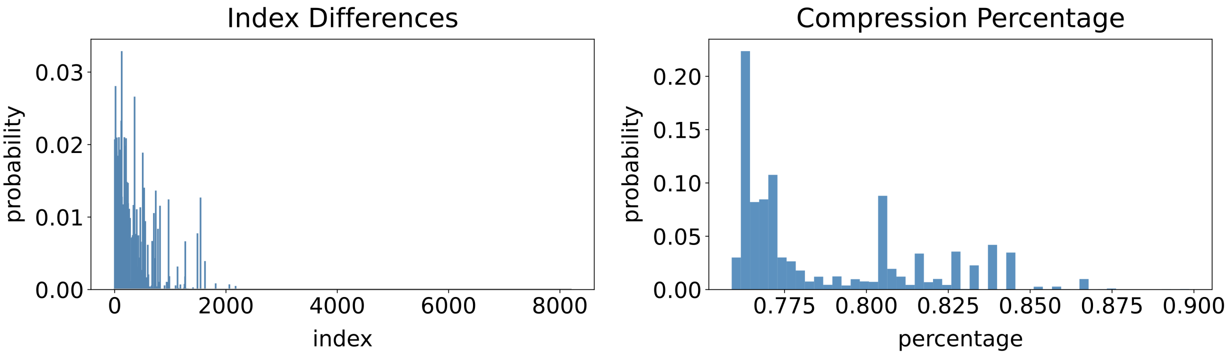

**Figure 7.** For further compression of $z$ with our arithmetic entropy encoding, we show the distribution of index differences $\rho(\delta_k)$ for our representation $z \to (\delta, w)$ (left). For the SAS scattering data, we show the further compression reductions in percentage obtained for the index differences $\delta$ (right).

the table displays the mean and standard deviation for the relative $L^2$-norm and compression rate for typical hourglass framework with architecture $(1/4, 1/8, 0, \cdot)$. It can be seen that for this type of network, the mean testing relative error is 3.6% with compression rate 10×. Note, we report average pixel error in Figure 4 and report here that the corresponding relative errors for the first example is 6.81% and 3.39% respectively. The errors are comparable to the relative errors reported in Table 1. Every other row in this table represents a different sparse autoencoder architecture, and the $\lambda$ parameter has been chosen to best approximate the error of first entry.

We can make some general observations from Table 1. First, as seen from the first three rows, that even when the network architecture and training is held constant, adding in a sparsity term into the loss function can provide additional compression without the loss of accuracy. Two, enlarging the latent space provides significant improvement to compression when sparse loss functions is used in the loss function. Finally, the type of sparse structure selector $f$ used in the loss function makes a difference in the accuracy and compression. Additionally, we report that for all autoencoders trained the weights were randomly chosen by adding normal distributed random noise to each weight before training, and the distribution of the relative error after one epoch was $8.2 \pm 1.5 \times 10^{-1}$ for all models in Table 1. The network operates consistently under different random initial conditions.

We investigate the impact of resolution on compression and introduce a new structural loss function for latent space, the Laplacian operator. The results are presented in Figures 8 and 9. We utilize two sparse autoencoder networks: $(1, 2, 1/2, z)$ and $(1, 2, 6, \nabla^2 z)$. Comparing these figures, we observe that employing the Laplacian operator to structure the latent space enhances compression at higher resolutions and effectively clusters non-zero elements in the latent space. In contrast, at lower resolutions, the clustering constraint in the latent space is

| **Architecture** | **Realitive $L^2$** | | **Compression Rate** | |
|---|---|---|---|---|
| $\left(\frac{m}{n}, \frac{\ell}{n}, \lambda, f(z)\right)$ | **Test** | **Train** | **Test** | **Train** |
| $(1/4, 1/8, 0, \cdot)$ | $3.6(4)\times10^{-2}$ | $3.5(4)\times10^{-2}$ | $1.0(1)\times10^{1}$ | $1.0(1)\times10^{1}$ |
| $(1/4, 1/8, 1\times10^{-3}, z)$ | $3.5(4)\times10^{-2}$ | $3.4(4)\times10^{-2}$ | $1.2(1)\times10^{1}$ | $1.2(1)\times10^{1}$ |
| $(1/4, 1/8, 2\times10^{-3}, \nabla z)$ | $3.6(4)\times10^{-2}$ | $3.6(4)\times10^{-2}$ | $1.2(2)\times10^{1}$ | $1.2(2)\times10^{1}$ |
| $(1/2, 1/4, 7\times10^{-3}, z)$ | $2.9(6)\times10^{-2}$ | $2.8(6)\times10^{-2}$ | $2.2(4)\times10^{1}$ | $2.2(4)\times10^{1}$ |
| $(1/2, 1/4, 3\times10^{-2}, \nabla z)$ | $2.8(7)\times10^{-2}$ | $2.7(6)\times10^{-2}$ | $2.2(5)\times10^{1}$ | $2.1(5)\times10^{1}$ |
| $(3/4, 1/2, 3\times10^{-2}, z)$ | $3.0(6)\times10^{-2}$ | $2.9(6)\times10^{-2}$ | $4.4(9)\times10^{1}$ | $4.5(8)\times10^{1}$ |
| $(3/4, 1/2, 1.5\times10^{-1}, \nabla z)$ | $3.1(6)\times10^{-2}$ | $3.1(6)\times10^{-2}$ | $5.2(9)\times10^{1}$ | $5.1(9)\times10^{1}$ |
| $(1, 1, 1\times10^{-1}, z)$ | $3.2(7)\times10^{-2}$ | $3.3(7)\times10^{-2}$ | $7.8(9)\times10^{1}$ | $7.8(9)\times10^{1}$ |
| $(1, 1, 9\times10^{-1}, \nabla z)$ | $3.6(7)\times10^{-2}$ | $3.6(7)\times10^{-2}$ | $1.0(1)\times10^{2}$ | $1.0(1)\times10^{2}$ |

**Table 1.** This table displays the mean and standard deviation for the relative $L^2$ norm, $\|d(e(x_j;\theta_e);\theta_d)-x_j\|_2/\|x_j\|_2$, and compression rate, $n/\|f(e(x_j;\theta_e))\|_0$, for different autoencoder architectures. The statistical notation, for example $1.57(1)\times10^{2}$, implies $157 \pm 1$. The uncertainty was determined by training twenty autoencoders on the same testing and training dataset discussed at the beginning of Section 3.

more restrictive, resulting in lower compression performance compared to the architecture shown in Figure 8.

Figure 10 illustrates the relationship between MSE and compression rate in relation to the size of the training dataset. Notably, the compression rates for training and testing datasets align closely. A transition point is observed in the compression rate: once the training dataset reaches a certain size, the compression significantly increases. This suggests that the autoencoder requires sufficient information to effectively determine highly sparse representations in the latent space.

The impact on MSE is also noteworthy. The MSE for the training and testing datasets only converge when there is an adequate amount of training data. As expected, the testing set's MSE increases when insufficient data is used for training, indicating that a substantial amount of data is necessary for the autoencoder to learn to compress all SAS-generated data effectively. Interestingly, the MSE for the training set also rises with smaller training datasets, which is surprising since one might expect memorization to occur with limited data.

## 4. Conclusion and Future Work

Our proposed sparse autoencoder methods present a novel approach to lossy compression tailored to scientific data. By extending the principles of compressed sensing, we introduce a framework that effectively captures and may preserve essential features of scientific datasets. Incorporating sparsity-promoting regularizations within our autoencoder architecture significantly enhances the encoding process, enabling more efficient representation

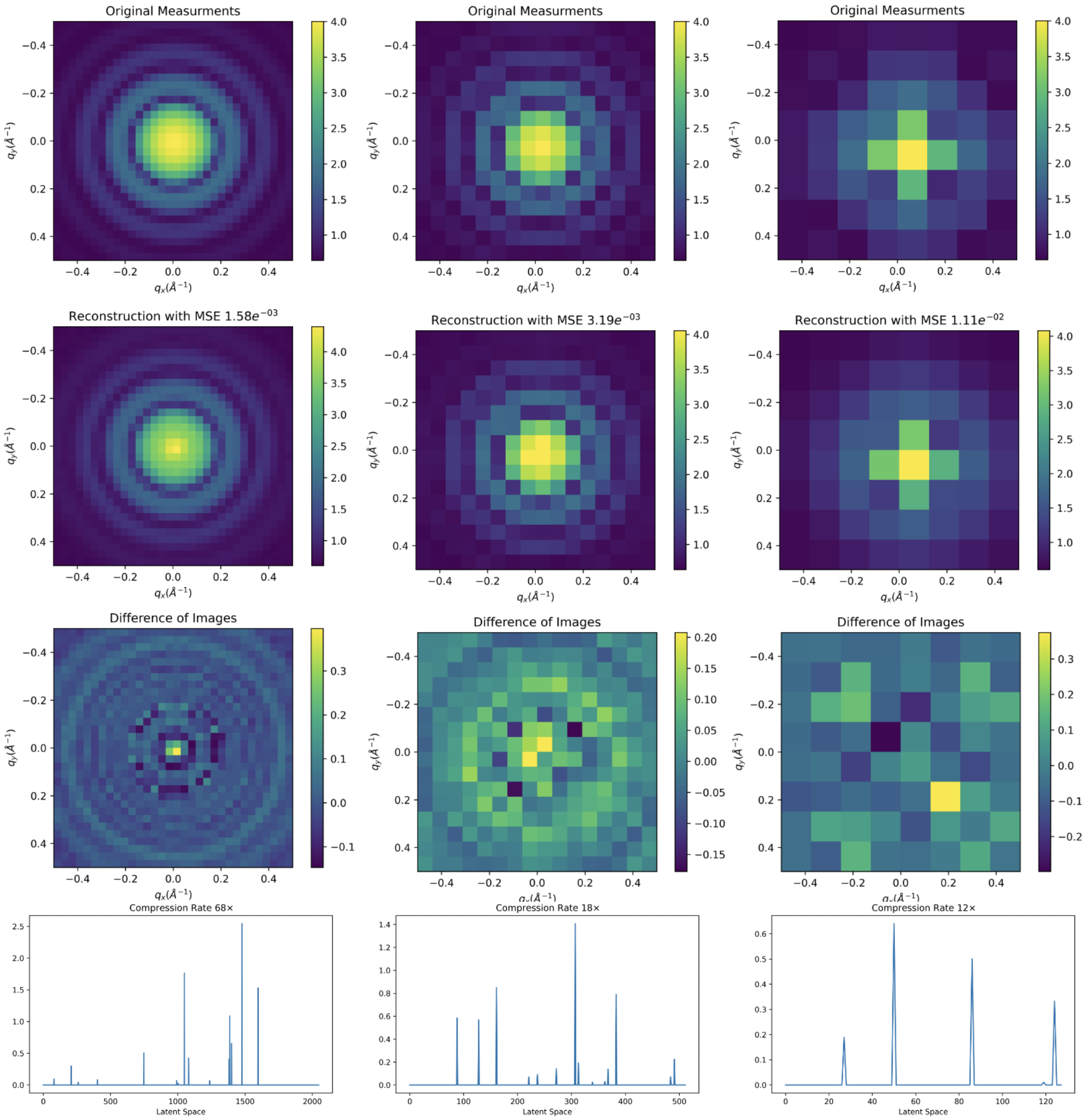

**Figure 8.** Top row displays the same SAS image with resolution of $32 \times 32$, $16 \times 16$, and $8 \times 8$, from left to right. The second row shows the reconstruction of compression of this image using a sparse autoencoder network of $(1, 2, {}^{1}\!/_{2}, z)$ for each resolution. We show the MSE for each reconstruction of this SAS image in the corresponding title, and report that the mean testing errors on the 150,000 independently random SAS images was $3.56 \times 10^{-3}$, $5.37 \times 10^{-3}$, and $5.84 \times 10^{-3}$, respectively. Correspondingly, the mean training compression rates are 106×, 28×, and 12×. Here the third row shows the difference of the reconstructed and original image and the last row shows the latent space with compression rate for this SAS image.

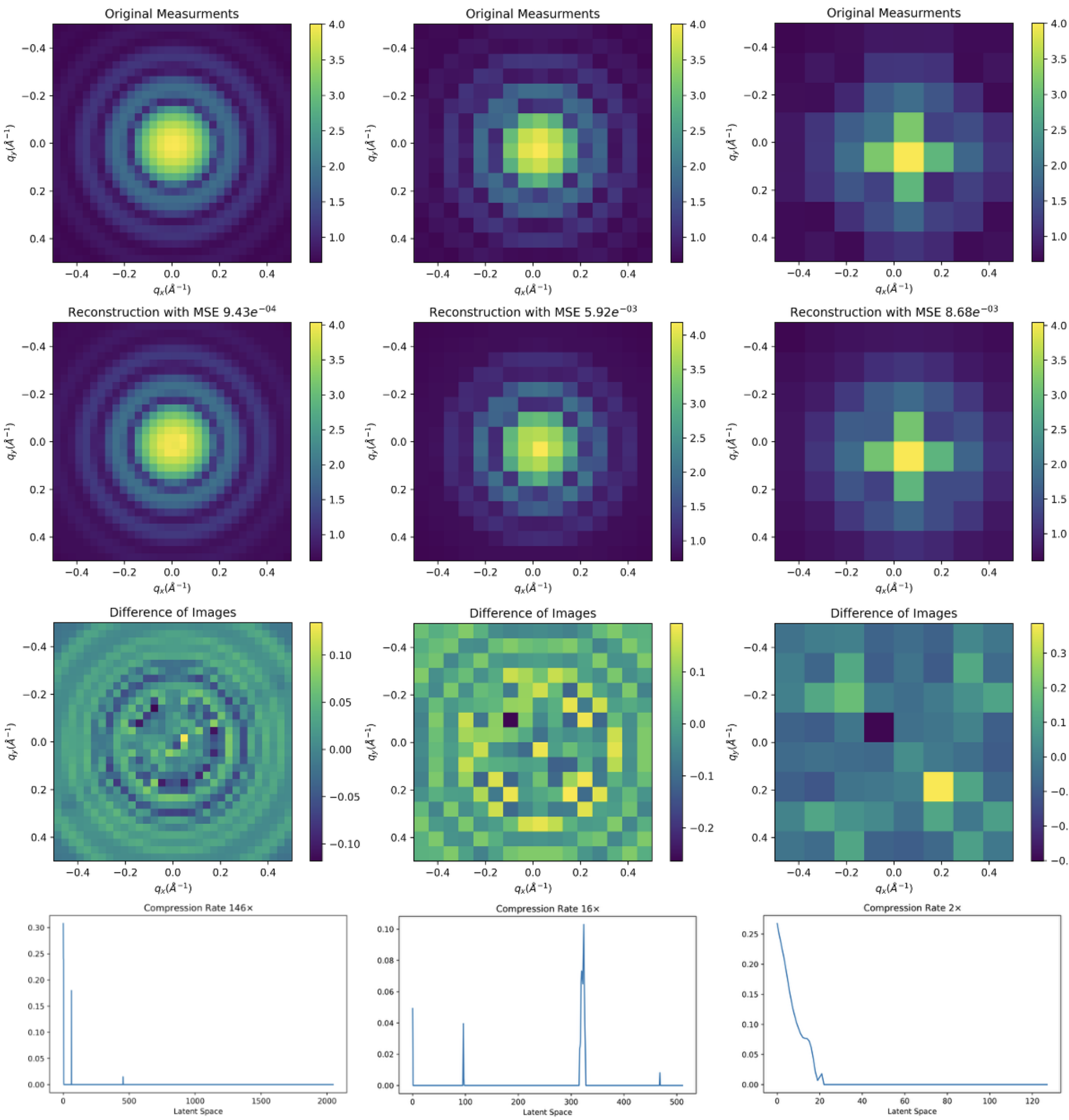

**Figure 9.** Top row displays the same SAS image with resolution of 32 × 32, 16 × 16, and 8 × 8, from left to right. The second row shows the reconstruction of compression of this image using a sparse autoencoder network of $(1, 2, 6, \nabla^2 z)$ for each resolution. We show the MSE for each reconstruction of this SAS image in the corresponding title, and report that the mean testing errors on the 150,000 independently random SAS images was $4.12 \times 10^{-3}$, $5.05 \times 10^{-3}$, and $3.83 \times 10^{-3}$, respectively. Correspondingly, the mean training compression rates are 131×, 22×, and 3×. Here the third row shows the difference of the reconstructed and original image and the last row shows the latent space with compression rate for this SAS image.

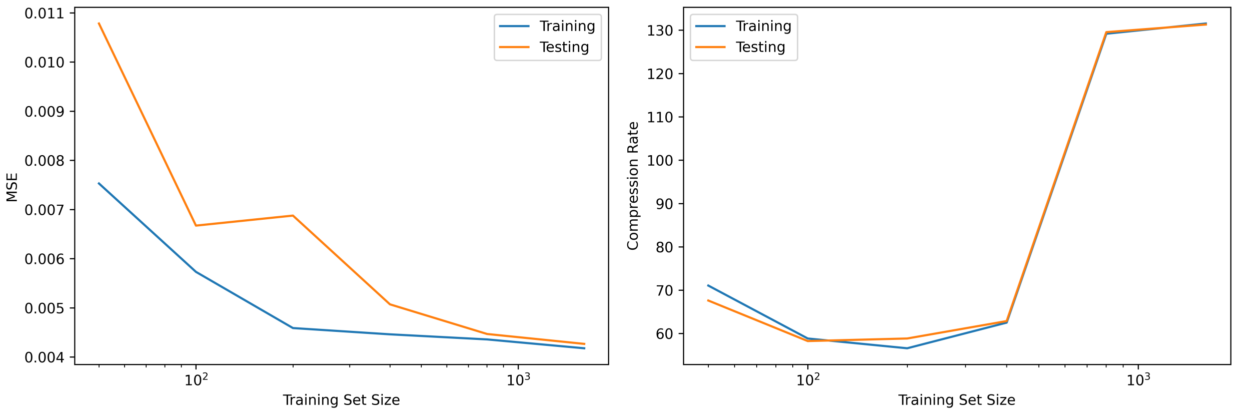

**Figure 10.** Comparison of the MSE for training and testing set data as a function of the training set size (left). Comparison of the compression rate for training and testing set data as a function of the training set size (left).

of complex scientific data.

In our numerical experiments, we have demonstrated the superiority of our approach in preserving critical signal features during compression. Our findings highlight the importance of utilizing high-dimensional latent spaces, which provide ample capacity to capture features and structures inherent in scientific data. By introducing a sparse feature selector of the encoded latent variable, our methods offer a flexible and efficient means of obtaining sparse representations in the latent space. Our proposed method introduces a robust learning strategy that enables substantial compression ratios without compromising data quality. Consequently, our novel approach paves the way for efficient storage, transmission, and analysis of large-scale scientific datasets, addressing the growing challenges associated with managing and processing scientific data.

Our future research efforts on sparse autoencoder methods for scientific data compression will for instance explore network architectures, generative models, and attention mechanisms. Data compression and related inverse problems–such as those arising in medical imaging applications like MRI and CT–present compelling directions for future research We will further explore hybrid compression techniques combining lossy and lossless approaches. We will investigate domain-specific applications and scalability and efficiency are other important areas for future research. By addressing these challenges and opportunities, we can further advance the state-of-the-art in scientific data compression and meet the growing demands of modern scientific research.

## Acknowledgments

This work was partially supported by the National Science Foundation (NSF) under grant DMS-2152661 (M. Chung) and grant DMS-2306101 (P. Atzberger). This work was partially supported by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE), Office of Advanced Scientific Computing Research (R. Archibald). The US government retains and the publisher, by accepting the work for publication, acknowledges that the US government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the submitted manuscript version of this work, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (`http://energy.gov/downloads/doe-public-access-plan`).

## Bibliography

[1] B. M. Afkham, J. Chung, and M. Chung. “Goal-oriented Uncertainty Quantification for Inverse Problems via Variational Encoder-Decoder Networks”. In: *arXiv preprint arXiv:2304.08324* (2023) (cited on page 2).

[2] P. Baldi and K. Hornik. “Neural networks and principal component analysis: Learning from examples without local minima”. In: *Neural networks* 2.1 (1989), pages 53–58 (cited on page 2).

[3] C. Bonneville et al. “A Comprehensive Review of Latent Space Dynamics Identification Algorithms for Intrusive and Non-Intrusive Reduced-Order-Modeling”. In: *arXiv preprint arXiv:2403.10748* (2024) (cited on page 2).

[4] H. Bourlard and Y. Kamp. “Auto-association by multilayer perceptrons and singular value decomposition”. In: *Biological cybernetics* 59.4 (1988), pages 291–294 (cited on page 1).

[5] T. Bricken et al. “Towards monosemanticity: Decomposing language models with dictionary learning”. In: *Transformer Circuits Thread* (2023), page 2 (cited on page 3).

[6] V. Britanak, P. C. Yip, and K. R. Rao. *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Elsevier, 2010 (cited on page 12).

[7] E. J. Candes, J. K. Romberg, and T. Tao. “Stable signal recovery from incomplete and inaccurate measurements”. In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59.8 (2006), pages 1207–1223 (cited on page 4).

[8] E. J. Candes and T. Tao. “Decoding by linear programming”. In: *IEEE transactions on information theory* 51.12 (2005), pages 4203–4215 (cited on page 4).

[9] E. J. Candès, J. Romberg, and T. Tao. “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. In: *IEEE Transactions on information theory* 52.2 (2006), pages 489–509 (cited on page 8).

[10] X. Chen et al. "A sub-Nyquist rate sampling receiver exploiting compressive sensing". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.3 (2010), pages 507–520 (cited on page 8).

[11] N. Cheng et al. "Bi-fidelity variational auto-encoder for uncertainty quantification". In: *Computer Methods in Applied Mechanics and Engineering* 421 (2024), page 116793 (cited on page 2).

[12] Z. Cheng et al. "Deep convolutional autoencoder-based lossy image compression". In: *2018 Picture Coding Symposium (PCS)*. IEEE. 2018, pages 253–257 (cited on page 2).

[13] J. Chung and M. Chung. "Optimal regularized inverse matrices for inverse problems". In: *SIAM Journal on Matrix Analysis and Applications* 38.2 (2017), pages 458–477 (cited on page 6).

[14] M. Chung and R. A. Renaut. "A variable projection method for large-scale inverse problems with $\ell^1$ regularization". In: *Applied Numerical Mathematics* 192 (2023), pages 297–318 (cited on page 4).

[15] M. Chung et al. "Paired Autoencoders for Inverse Problems". In: *arXiv preprint arXiv:2405.13220* (2024) (cited on page 2).

[16] T. Compute and D. E. for Science (CADES). *www.cades.ornl.gov.* 2024 (cited on page 12).

[17] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006. ISBN: 0471241954. URL: `http://dx.doi.org/10.1002/047174882X` (cited on pages 4, 9, 10).

[18] R. P. documentation. *pytorch.org/docs/stable/generated/torch.optim.RAdam.html.* 2024 (cited on page 12).

[19] D. L. Donoho. "For most large underdetermined systems of linear equations the minimal $\ell^1$-norm solution is also the sparsest solution". In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59.6 (2006), pages 797–829 (cited on page 4).

[20] B. Dumitrescu and P. Irofti. *Dictionary learning algorithms and applications*. Springer, 2018 (cited on page 8).

[21] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Volume 2. Springer, 2010 (cited on page 4).

[22] S. Friedland and A. Torokhti. "Generalized rank-constrained matrix approximations". In: *SIAM Journal on Matrix Analysis and Applications* 29.2 (2007), pages 656–659 (cited on page 6).

[23] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016 (cited on page 1).

[24] R. M. Gray and D. L. Neuhoff. “Quantization”. In: *IEEE transactions on information theory* 44.6 (1998), pages 2325–2383 (cited on pages 9, 10, 14).

[25] W. T. Heller, M. Doucet, and R. K. Archibald. “Sas-temper: Software for fitting small-angle scattering data that provides automated reproducibility characterization”. In: *SoftwareX* 16 (2021), page 100849. ISSN: 2352-7110. DOI: `https://doi.org/10.1016/j.softx.2021.100849`. URL: `https://www.sciencedirect.com/science/article/pii/S235271102100128X` (cited on page 12).

[26] X. Jiang et al. “A novel sparse auto-encoder for deep unsupervised learning”. In: *2013 Sixth international conference on advanced computational intelligence (ICACI)*. IEEE. 2013, pages 256–261 (cited on page 3).

[27] S. H. Kabil and H. Bourlard. “From Undercomplete to Sparse Overcomplete Autoencoders to Improve LF-MMI Speech Recognition”. In: *Interspeech 2022* (2022), pages 1061–1065 (cited on pages 2, 3).

[28] D. P. Kingma and M. Welling. “Auto-encoding variational Bayes”. In: *arXiv preprint arXiv:1312.6114* (2013) (cited on page 2).

[29] M. A. Kramer. “Autoassociative neural networks”. In: *Computers & chemical engineering* 16.4 (1992), pages 313–328 (cited on page 1).

[30] K. Kreutz-Delgado et al. “Dictionary learning algorithms for sparse representation”. In: *Neural computation* 15.2 (2003), pages 349–396 (cited on page 8).

[31] R. Kumar et al. “High-fidelity audio compression with improved RVQGAN”. In: *Advances in Neural Information Processing Systems* 36 (2024) (cited on page 2).

[32] G. G. Langdon. “An introduction to arithmetic coding”. In: *IBM Journal of Research and Development* 28.2 (1984), pages 135–149 (cited on pages 10, 14).

[33] D. Lappas, V. Argyriou, and D. Makris. “Fourier transformation autoencoders for anomaly detection”. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pages 1475–1479 (cited on page 2).

[34] J. Lee, A. Rangarajan, and S. Ranka. “Nonlinear-by-Linear: Guaranteeing Error Bounds in Compressive Autoencoders”. In: *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing*. 2023, pages 552–561 (cited on page 2).

[35] Q. Li et al. “Deep sparse autoencoder and recursive neural network for EEG emotion recognition”. In: *Entropy* 24.9 (2022), page 1187 (cited on page 3).

[36] J. Liu et al. “Exploring autoencoder-based error-bounded compression for scientific data”. In: *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE. 2021, pages 294–306 (cited on page 2).

[37] L. Liu et al. “On the variance of the adaptive learning rate and beyond. arXiv 2019”. In: *arXiv preprint arXiv:1908.03265* (2019) (cited on page 12).

[38] C. Louizos, M. Welling, and D. P. Kingma. "Learning sparse neural networks through $L_0$ regularization". In: *arXiv preprint arXiv:1712.01312* (2017) (cited on page 3).

[39] A. Majumdar. "An autoencoder based formulation for compressed sensing reconstruction". In: *Magnetic resonance imaging* 52 (2018), pages 62–68 (cited on page 3).

[40] A. Makhzani and B. Frey. "K-sparse autoencoders". In: *arXiv preprint arXiv:1312.5663* (2013) (cited on page 3).

[41] G. Martino, D. Moroni, and M. Martinelli. "Are We Using Autoencoders in a Wrong Way?" In: *arXiv preprint arXiv:2309.01532* (2023) (cited on pages 3, 5).

[42] Q. Meng et al. "Relational autoencoder for feature extraction". In: *2017 International joint conference on neural networks (IJCNN)*. IEEE. 2017, pages 364–371 (cited on page 2).

[43] J.-M. Muller et al. *Handbook of floating-point arithmetic*. Springer, 2018 (cited on pages 10, 14).

[44] E. Newman, J. M. Solomon, and M. Chung. "Image reconstructions using sparse dictionary representations and implicit, non-negative mappings". In: *arXiv preprint arXiv:2312.03180* (2023) (cited on page 8).

[45] A. Ng et al. "Sparse autoencoder". In: *CS294A Lecture notes* 72.2011 (2011), pages 1–19 (cited on page 3).

[46] H.-A. T. Nguyen, T. H. Le, and T. D. Bui. "A deep wavelet sparse autoencoder method for online and automatic electrooculographical artifact removal". In: *Neural Computing and Applications* 32.24 (2020), pages 18255–18270 (cited on page 3).

[47] E. Norlander and A. Sopasakis. "Latent space conditioning for improved classification and anomaly detection". In: *arXiv preprint arXiv:1911.10599* (2019) (cited on page 5).

[48] E. Plaut. "From principal subspaces to principal components with linear autoencoders". In: *arXiv preprint arXiv:1804.10253* (2018) (cited on page 2).

[49] A. Polino, R. Pascanu, and D. Alistarh. "Model compression via distillation and quantization". In: *arXiv preprint arXiv:1802.05668* (2018) (cited on pages 10, 14).

[50] S. project. *sasview.org*. 2024 (cited on page 11).

[51] J. Rissanen and G. G. Langdon. "Arithmetic coding". In: *IBM Journal of research and development* 23.2 (1979), pages 149–162 (cited on pages 10, 14).

[52] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pages 259–268 (cited on page 5).

[53] M. Sakurada and T. Yairi. "Anomaly detection using autoencoders with nonlinear dimensionality reduction". In: *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*. 2014, pages 4–11 (cited on page 2).

[54] S. Scardapane et al. “Group sparse regularization for deep neural networks”. In: *Neurocomputing* 241 (2017), pages 81–89 (cited on page 3).

[55] S. Tariyal et al. “Deep dictionary learning”. In: *IEEE Access* 4 (2016), pages 10096–10109 (cited on page 2).

[56] L. Theis et al. “Lossy image compression with compressive autoencoders”. In: *International conference on learning representations*. 2022 (cited on page 2).

[57] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pages 267–288 (cited on page 4).

[58] I. Tošić and P. Frossard. “Dictionary learning”. In: *IEEE Signal Processing Magazine* 28.2 (2011), pages 27–38 (cited on page 8).

[59] P. Vincent et al. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pages 1096–1103 (cited on page 2).

[60] Y. Wang, H. Yao, and S. Zhao. “Auto-encoder based dimensionality reduction”. In: *Neurocomputing* 184 (2016), pages 232–242 (cited on page 2).

[61] I. H. Witten, R. M. Neal, and J. G. Cleary. “Arithmetic coding for data compression”. In: *Communications of the ACM* 30.6 (1987), pages 520–540 (cited on pages 10, 14).

[62] P. Wu et al. “Reduced order model using convolutional auto-encoder with self-attention”. In: *Physics of Fluids* 33.7 (2021) (cited on page 2).

[63] G. Zhang, Y. Liu, and X. Jin. “A survey of autoencoder-based recommender systems”. In: *Frontiers of Computer Science* 14 (2020), pages 430–450 (cited on page 2).

[64] Y. Zhu et al. “Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data”. In: *Journal of Computational Physics* 394 (2019), pages 56–81 (cited on page 2).