A&A, 711, A164 (2026)
https://doi.org/10.1051/0004-6361/202557384

Astronomy & Astrophysics

Extracting latent representations from X-ray spectra

Classification, regression, and accretion signatures of Chandra sources

N. O. Pinciroli Vago[1,2,3,⋆], R. Martínez-Galarza[3,⋆], and R. Amato[2]

[1] Department of Electronics, Information and Bioengineering, Politecnico di Milano, via G. Ponzio, 34, 20133 Milan, Italy
[2] INAF – Osservatorio Astronomico di Roma, Via Frascati 33, 00078, Monte Porzio Catone, Italy
[3] Center for Astrophysics | Harvard & Smithsonian, 60 Garden Street, Cambridge, MA 02138, USA

Received 23 September 2025 / Accepted 15 May 2026

ABSTRACT

Context. Spectral signatures are crucial in the era of large X-ray surveys, as effective methods for source identification and classification are needed due to the scale of the data they yield. Automatic machine learning methods have proven useful for such tasks, but so far they have not been applied to large spectral datasets, such as the Chandra Source Catalog.
Aims. Our aim was to develop a compact and physically meaningful representation of Chandra X-ray spectra using deep learning. To verify that the learned representation captures relevant information, we evaluated it through classification, regression, and interpretability analyses, and we measured the mutual information between spectral and time-domain properties of these sources to aid in future identification of transient events.
Methods. We used a transformer-based autoencoder to compress X-ray spectra into representations in an eight-dimensional latent space. Astrophysical source types and physical summary statistics were compiled from external catalogs. We evaluated the learned representation in terms of spectral reconstruction accuracy, clustering performance regarding eight known astrophysical source classes, and correlation with physical quantities such as hardness ratios and hydrogen column densities (N_H).
Results. Upon reconstruction, clustering in the latent space yielded a balanced classification accuracy of ∼40% across the eight source classes, increasing to ∼69% when restricted to Active Galactic Nuclei and stellar-mass compact objects exclusively. Moreover, latent features correlate with spectral and temporal properties, suggesting that the compressed representation captures physically relevant information.
Conclusions. Features learned directly from X-ray spectra capture relevant physical information as effectively as human-extracted features that require additional computations. They can be used for both classification and regression in large surveys, and they also share mutual information with time-domain properties. The methodology presented in this paper can be adapted to existing and upcoming X-ray catalogs.

Key words. accretion, accretion disks – methods: data analysis – methods: statistical – X-rays: binaries – X-rays: general

1. Introduction

Large catalogs containing hundreds of thousands of astrophysical high-energy sources have been compiled using data obtained by X-ray telescopes, including the Chandra X-ray Observatory (Weisskopf et al. 2000), XMM-Newton (Jansen et al. 2001), the Neil Gehrels Swift Observatory (Gehrels et al. 2004), and eROSITA (Predehl et al. 2021). These catalogs contain important diagnostics for the physical characterization of the sources they contain (Evans et al. 2024; Webb et al. 2020; Evans et al. 2020; Merloni et al. 2024), such as hardness ratios, fluxes, and variability estimates, as well as reprocessed data products, namely light curves and spectra, crucial to identifying the nature of the large fraction of serendipitous unlabeled X-ray sources. There is also growing interest in systematic searches of X-ray sources with specific spectral properties. For example, in the context of active galactic nuclei (AGNs), it is generally assumed that the typically hard spectrum of active galaxies results from nonthermal processes in the corona (Galeev et al. 1979), but a population of soft thermal AGNs has been identified (Arnaud et al. 1985; Iwasawa et al. 2024), a subset of which hosts emerging classes of transient events, such as tidal disruption events (Guolo et al. 2024) and quasi-periodic eruptions (QPEs; Miniutti et al. 2019; Arcodia et al. 2021; Chakraborty et al. 2021). Additionally, spectrally hard, fast X-ray transients (FXTs; Lin et al. 2022; Quirola-Vásquez et al. 2022) have recently been associated with neutron star mergers and other accreting compact object phenomena. Many of these sources, especially the extragalactic ones, do not have multiwavelength counterparts for classification, and therefore it becomes imperative to design effective methods of classification when only X-ray data are available. Complicating the picture, the spectra of astrophysical sources evolve over time, introducing additional variability that is informative of the physical processes but hard to evaluate in the Poisson regime of X-ray photon counting (Luo et al. 2016). Source classification, therefore, remains a cornerstone of X-ray astronomy, particularly in view of current and forthcoming missions with survey capabilities, such as eROSITA, AXIS (Mushotzky 2018), and NewAthena (Cruise et al. 2025). In particular, understanding to what extent spectral features relate to time-domain anomalies is crucial for the study of AGNs, QPEs, and FXTs. Classification can be attempted with traditional data analysis techniques, typically through spectral modeling aimed

⋆ Corresponding authors: nicolooreste.pinciroli@polimi.it; jmartine@cfa.harvard.edu

at deriving physically meaningful quantities. This approach is employed, for instance, in the Chandra Source Catalog (CSC) (Evans et al. 2024), which collects hundreds of thousands of spectra at different significance levels and provides a set of relevant summary statistics for each of them (e.g., counts, fluxes, best-fit parameters, hardness ratios, variability diagnostics). In addition to classification, regression techniques can be used to estimate continuous physical parameters directly from summary statistics. These methods complement traditional spectral modeling by enabling the rapid characterization of source properties across large datasets without the computational cost of individual fitting. Alternatively, self-supervised approaches can learn representations of astrophysical data across multiple modalities. These techniques have proven effective in diverse fields such as astronomy (Orwat-Kapola et al. 2022; Howie et al. 2025; Rizhko & Bloom 2025; Cheng et al. 2021, 2020), medicine (Huang et al. 2023; Yuan et al. 2024), genomics (Richter et al. 2025), industrial applications (Zangrando et al. 2022; Xu 2025), and remote sensing (Alosaimi et al. 2023; Rajaei et al. 2024). These machine learning (ML) methods enable more effective downstream tasks compared to those that require the manual extraction of features, such as classification, regression, and cross-modal prediction of astrophysical quantities (see, e.g., Parker et al. 2024). Because of their data-driven nature, ML algorithms introduce less bias than model-dependent approaches and can expose novel patterns and clusters in the data (Duffy et al. 2025), which in turn can provide new insights into physical processes and even help identify previously unknown astronomical sources.

Autoencoders are a type of artificial neural network (ANN) (Yang & Li 2015; Kingma & Welling 2022) that learn to compress the input data into a low-dimensional representation (or latent space) in a self-supervised fashion, and they reconstruct the input from the compressed latent representation. Autoencoders are particularly useful for anomaly detection (Chen et al. 2018), feature extraction (Meng et al. 2017), clustering (Tavakoli et al. 2020), and data denoising (Majumdar 2019). In astrophysics, they can help identify anomalies in large amounts of data, as done for galaxies (Storey-Fisher et al. 2021; Gagliano & Villar 2023; Liang et al. 2023; Arcelin et al. 2021) as well as in real-time light curves of different types of objects (Muthukrishna et al. 2022). Autoencoders can be used to speed up the computation time for parameter estimates (Tregidga et al. 2024) and for population studies (Saxena 2025), and they can also be used to detect outliers based on spectral and variability properties (e.g., the discovery of the extragalactic FXT XRT 200515, Dillmann et al. 2025). Transformers are another popular architectural choice for sequential data. They process sequences, such as spectra, using a mechanism called attention (Vaswani et al. 2017) that quantifies the level of relatedness between any pair of sequence elements. In this work, we present a compressed representation of Chandra X-ray spectra, projecting the data in a low-dimensional space using an autoencoder based on a transformer architecture. We used the resulting learned representations to reconstruct the input spectra and classify X-ray sources according to their physical properties. Specifically, we trained our transformer autoencoder to reconstruct the input spectra, cluster the resulting latent space, and evaluate the predictive power of the resulting embeddings on three downstream tasks, X-ray source type classification, regression on physically informative summary statistics, and symbolic regression (SR), in order to derive mathematical relations between the derived low-dimensional latent space and physical quantities such as X-ray fluxes. We evaluated the performance of the learned representations through the classification of eight types of astronomical objects, finding that despite the expected degeneracies between astrophysical types, the learned representations have a similar performance to manually extracted features but with a lower computational cost. On the regression front, we demonstrate that the learned features capture physically meaningful spectral summaries such as the hardness ratio and several parameters of physical spectral models, including time variability, thus stressing their usefulness as compressed summaries of the X-ray spectra and highlighting mutual information between spectral and time-domain properties. The method can be adapted for the study of X-ray spectra from other facilities, such as XMM-Newton, eROSITA, and other upcoming X-ray missions. The rest of the paper is organized as follows: Section 2 presents the dataset and data preprocessing. Section 3 presents the metrics and the algorithms used in this work. Section 4 presents quantitative results, and Sect. 5 presents the conclusions and a discussion on potential future directions for research based on this work.

2. Data

2.1. Dataset

We used a set of background-subtracted X-ray spectra of point-like sources from the CSC (v. 2.1, Evans et al. 2024). The CSC provides the pulse height amplitude (PHA) files associated with each source detection as well as the corresponding redistribution matrix files (RMFs)[1] and auxiliary response files (ARFs)[2], which contain information on the channel-energy conversion and the effective area of the mission, respectively, and are necessary to convert the detected counts into physical units, thus allowing for spectral modeling. We grouped the source spectra using the tools of the Chandra Interactive Analysis of Observations (CIAO) software (Fruscione et al. 2006), with a fixed count binning (i.e., each spectrum is binned differently in energy in order to achieve a fixed value of ten counts per bin) across all spectra. For context, the spectra of all sources in the CSC were fit using four canonical spectral models: power-law, blackbody, bremsstrahlung, and a collisional plasma model (APEC, Smith et al. 2001). These models were selected by the CSC pipeline because they robustly represent the primary X-ray emission mechanisms found in astrophysical sources. The CSC spectra and associated response files used here were downloaded using the scripts from the Multimodal Universe repository (Angeloudi et al. 2024), a large collection of ML-ready astronomical datasets, including Chandra. The spectra are filtered based on the number of X-ray counts in the source detection ($N_c \geq 100$), the signal to noise ratio (S/N) (S/N ≥ 5), and the off-axis angle ($\theta \leq 10$), given that the point spread function (PSF) degrades significantly at large angles due to the telescope optics. In addition to the processed spectra, we also compile a set of labels for each X-ray detection, specifying the astrophysical class of the corresponding source, collected from the labeled dataset prepared by Yang et al. (2022a): AGN, cataclysmic variable (CV), high-mass star (HM-STAR), high-mass X-ray binary (HMXB), low-mass star (LM-STAR), low-mass X-ray binary (LMXB), pulsar or isolated neutron star (NS), or young stellar object (YSO). This results in ≈25 000 spectra, ≈3200 of which are associated with a known class. We use the subset of labeled X-ray detections as the test set, while the unlabeled data are used for training (≈80% of the unlabeled data) and validation (≈20% of the unlabeled data). Since the class distribution for

[1] https://cxc.cfa.harvard.edu/ciao/dictionary/rmf.html
[2] https://cxc.cfa.harvard.edu/ciao/dictionary/arf.html

the training and validation sets is unknown, it is not guaranteed to be the same as the test set. However, the main results presented in this work (e.g., clustering) use only the test set. The test set labels were grouped in two alternative ways using either the eight distinct classes defined above or only two classes, AGNs and stellar-mass compact objects, with the latter class comprising HMXBs, LMXBs, CVs, and NSs. The two-class split was done to test whether the automatically learned features can be used to distinguish between supermassive black holes (SMBHs) and stellar-mass compact objects when only the X-ray spectra are available, as they may display similar spectral properties.

In addition to the astrophysical classes, we collected summary statistics (i.e., the physical quantities reported in Table A.1) from the CSC and fluxes from Yang et al. (2022a). The summary statistics are relevant for the physical characterization of each detection. We used them to show the correlation with the latent space extracted by the autoencoder (see Sect. 3.1.4). The fluxes were combined using SR (see Sect. 3.1.5). The correlation between the fluxes and latent space variables instead is not shown, as the spectra were rescaled and made independent of the absolute flux values.

2.2. Preprocessing

Each spectrum consists of a set of photon count rates (in counts per second per keV) for each energy bin in a range covering from 0.5 to 8 keV. Because the spectra have variable bin sizes, whereas the transformer autoencoder needs a fixed-length input, we resample the spectra onto a regular grid of energies via interpolation. Due to the energy-dependent noise level in the proportionality between charge and photon energy in the CCD detector, the energy resolution of Chandra spectra is higher at lower energies. Therefore, we model the spectra using a denser sampling at lower energies and a sparser sampling at higher energies by defining a logarithmic grid in base 10 with $N = 400$ points, spanning from a minimum energy of 0.5 keV to a maximum energy of 8 keV. The flux values are linearly interpolated using `interp1d`[3]. We determined the optimal number of grid points N by testing values starting at 100 in increments of 50. At each step, the mean absolute error (MAE) between the original and resampled spectra on the test set was computed. The improvement in MAE was evaluated using bootstrapping to estimate 5σ confidence intervals for each step. To balance reconstruction accuracy and the length of the resampled spectra, the procedure was terminated when the 5σ intervals of consecutive MAE values first overlapped. Linear interpolation was chosen because it acts as a low-pass filter (i.e., it reduces high-frequency components, which also prevents the introduction of additional peaks), as shown in Jokanović et al. (2015).

We normalize all the spectra in the count rate dimension to make it fit in the interval $[0, 1]$ using min-max normalization, implemented through `MinMaxScaler`[4], as shown in Fig. 1. For a resampled spectrum $\tilde{f}$ with flux values $\tilde{f}_i$, the normalized flux values $\hat{f}_i$ are given by

$$\hat{f}_i = \frac{\tilde{f}_i - \min(\tilde{f})}{\max(\tilde{f}) - \min(\tilde{f})} \quad (1)$$

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

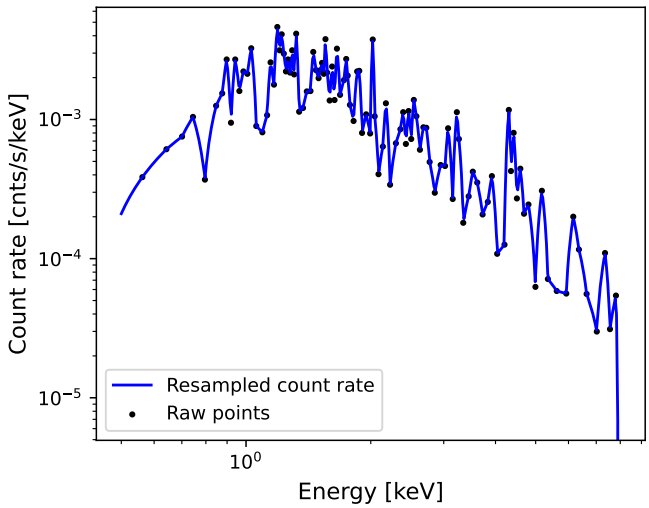


Fig. 1. Example of resampling with $N = 400$ points.

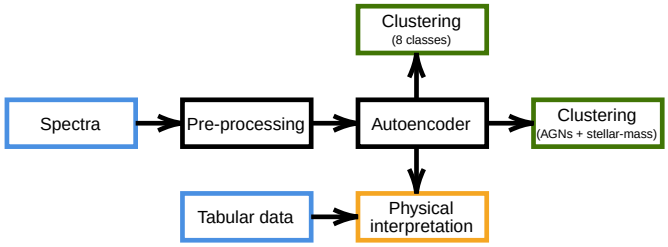


Fig. 2. Our pipeline. The blue rectangles indicate the inputs, the green rectangles indicate the clustering outputs, and the orange rectangle indicates the physical interpretation of the results.

The goal of normalization is to capture the shape of spectra rather than focusing on their absolute values. Additionally, normalization of the input has been shown to be beneficial in ML (Géron 2025).

3. Methods

This section illustrates the ML architectures, our pipeline, and the quantitative evaluation metrics.

3.1. Pipeline

Figure 2 presents our pipeline for learning the latent representations, clustering them, using them for source type identification, and finally explaining the results through mathematical relations between the latent representations and physical quantities. The pipeline is divided into six steps:

1. The spectra are preprocessed (Section 2.2).
2. A transformer autoencoder is trained to reconstruct the spectra and to compute a latent representation of the data (Section 3.1.1).
3. The test set latent space vectors are clustered (Section 3.1.2) and evaluated in terms of astrophysical class separation.
4. The clusters are visualized using t-distributed stochastic neighbor embedding (tSNE) in two dimensions (Section 3.1.3).
5. The physical quantities presented in Table A.1 (excluding fluxes, since we normalized all the spectra) are obtained from the latent space through regression (Section 3.1.4).

Table 1. Overview of the autoencoder hyperparameters, their values, and their meaning.

Hyperparameter	Values	Explanation
Learning rate (LR)	[1e-4, 5e-4, 1e-3]	Controls the step size in gradient descent, determining how quickly the model learns.
Embedding dimensions	[8, 16]	The size of the encoded representation in the latent space.
Number of layers	[1, 2, 4]	The number of layers in the encoder and decoder subnetworks.
Number of attention heads	[8, 16]	Number of different attention mechanisms used in each layer.

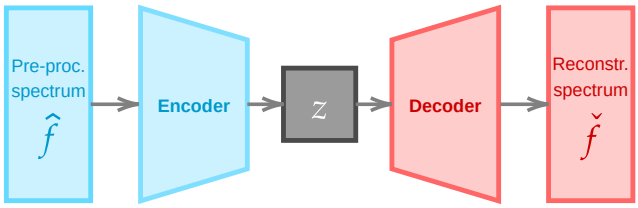


Fig. 3. Autoencoder architecture. The autoencoder takes as input the resampled normalized spectrum ($\hat{f}$), encodes it into a compressed representation (z), and reconstructs it as $\check{f}$.

6. Mathematical relations between fluxes in different energy bands and the latent space dimensions are presented for sources of types AGN and YSO in the test set (Section 3.1.5).

3.1.1. Autoencoder

We designed a deep autoencoder architecture that takes as input the original resampled normalized spectra $\hat{f}$ and outputs the reconstructed spectra $\check{f}$, learning a compressed representation (or latent space) of the spectra (z), as outlined in Figure 3. An overview of the autoencoder architecture is given in Appendix B.1. Both the encoder and the decoder are implemented using the Transformer architecture (Shamsolmoali et al. 2023; Cheng et al. 2024; Wu et al. 2024; Vaswani et al. 2017). Each input sequence of resampled flux values is passed through a linear layer to obtain embedding vectors of dimension `embedding_dim`. Learnable positional encodings $\mathbf{P}$ of shape (L, `embedding_dim`) are added to capture information about the ordering of points in the spectra. The resulting sequence is processed by a stack of `num_layers` Transformer encoder layers, each employing multihead self-attention with `nhead` attention heads (i.e., parallel attention mechanisms that allow the model to focus on different parts of the spectrum at the same time), enabling the model to learn global dependencies across the sequence. After encoding, the sequence is pooled over the length dimension using adaptive average pooling to obtain a single latent vector $\mathbf{z} \in \mathbb{R}^{\texttt{embedding\_dim}}$. For reconstruction, the decoder receives $\mathbf{z}$, applies a linear embedding, adds positional encodings, and passes the result through an analogous stack of Transformer decoder layers. The output layer projects the resulting sequence to produce the reconstructed spectra. The network is optimized using the Adam optimizer (Kingma & Ba 2014). The network is trained for a maximum of 200 epochs, with early stopping on the validation MAE (patience $p = 20$) to prevent overfitting. Table 1 provides an overview of the variable hyperparameters, their respective values, and their meanings. The transformer-based autoencoder is chosen over other autoencoder architectures because transformers are effective at modeling long-range correlations. In X-ray spectra, such correlations extend beyond local dependencies that are typical of time series data. The self-attention mechanism within transformers allows the model to efficiently identify and represent these nonlocal relationships across the entire input. In astrophysical terms, this allows the autoencoder to capture physical relationships between spectral features separated by large energy gaps.

3.1.2. Clustering

Clustering is an unsupervised learning technique used to group similar data points, so that the points in each cluster are more similar to each other than to points in other clusters (Jain et al. 1999; Saxena et al. 2017; Fotopoulou 2024). In this work, we use clustering to find associations between X-ray spectra in the learned latent space representations for the validation set and evaluate the usefulness of the representation for the two classification tasks introduced in Sect. 2. We compare eight clustering algorithms belonging to two families: hard clustering and soft clustering (see also Table B.1, where each algorithm is labeled according to its clustering type). In hard clustering, each data point is assigned to exactly one cluster, meaning there is no overlap in membership between clusters (Jain et al. 1999). In contrast, soft clustering (or fuzzy clustering) assigns each data point to multiple clusters, with membership probabilities that reflect the degree of association with each cluster (Jain et al. 1999). Appendix B.2 presents the clustering algorithms in more detail.

3.1.3. t-distributed stochastic neighbor embedding

The goal of tSNE[5] (Van der Maaten & Hinton 2008) is to find a low-dimensional representation (in this work, bi-dimensional) of high-dimensional data (here, the latent space created by the autoencoder) that preserves local pairwise similarities between the original points. Unlike principal component analysis (PCA) (Greenacre et al. 2022), tSNE can model complex and nonlinear relationships, making it suitable for exploring structures in high-dimensional data. Appendix B.3 presents the algorithm in detail.

3.1.4. Regression

Regression is a technique used to quantify the relationship between input variables and one or more continuous target variables. In this work, we use Huber regression (Owen 2007) due to its robustness to outliers. This regression technique uses a piecewise loss function that is quadratic for small residuals and linear for large ones, being robust to outliers as a result. To ensure robustness and generalization, we used five-fold cross-validation. First, the labeled test set is partitioned into five subsets. Then, for each fold, the model is trained on four subsets

[5] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

and validated on the remaining one, with performance metrics (here, MAE, see Sect. 3.2) averaged across all folds. The results are computed first on the entire test set and then by grouping observations according to their associated statistics (see Table A.1). In this second case, for each observation, the best-fitting model (either black body, power law, bremsstrahlung, or collisional plasma) is selected, and only the quantities associated with it are estimated.

3.1.5. Symbolic regression

Symbolic regression (Makke & Chawla 2024) searches for underlying mathematical expressions that describe the relationships between variables. Unlike traditional regression methods, which require assumptions about the form of the underlying model (Maulud & Abdulazeez 2020), SR is able to discover adequate expressions that combine variables autonomously. The expressions are represented as trees. The nodes in the trees represent either operators and functions (e.g., +, −, ×, ÷, sin, exp) or variables and constants. In this work, SR is used to find relations between the latent space components and the fluxes presented in Table A.1 using 4 operations (+, −, ×, ÷) and a function (log). The use of log is motivated by the fact that flux values span several orders of magnitude, making log-transformation a natural choice. From the extracted formulas, we manually derive simplified versions proportional to the original ones to identify simple correlations with the latent space dimensions. Appendix B.4 presents the SR algorithm in more detail.

3.2. Metrics

The results were evaluated in three steps:

- Autoencoder reconstruction: assessment of the reconstruction ability of the autoencoder (see Sect. 3.2.1).
- Clustering performance: evaluation of the clustering results on the test set for both the eight-class and two-class cases (see Sect. 3.2.2).
- Physical interpretation of the latent space: evaluation of the relationship between the latent space and physical quantities for all the sources. Evaluation of the correlation between physical interpretable formulas and the latent space dimensions for AGNs (see Sect. 3.2.3).

Each step is explained in more detail in the following subsections.

3.2.1. Autoencoder

The reconstruction ability of the autoencoder is quantified using the MAE between the resampled normalized spectra and their corresponding reconstructions. For a given resampled and normalized spectrum $\hat{f}$ with count rates $\hat{f}_i$ and estimated count rates $\check{f}_i$, the MAE is defined as

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\check{f}_i - \hat{f}_i|, \tag{2}$$

where n represents the number of points in the spectrum. The MAE ranges between 0 and 1, with lower values indicating a better reconstruction ability (i.e., a better autoencoder performance). Unlike mean square error (MSE), which squares errors and effectively amplifies outliers, MAE applies a linear penalty. This makes MAE significantly more robust, as extreme values do not dominate the metrics. It is also easier to interpret, as it measures average relative error on the normalized spectra. Moreover, it avoids the bias MSE would have in high-flux regions, since squaring errors in MSE emphasizes deviations at large count values. On the other hand, MAE is less smooth than MSE, which can lead to slower or less stable optimization.

3.2.2. Clustering

The clustering performance is evaluated on the labeled test set using a contingency matrix (Bonaccorso 2019). A contingency matrix is a tool used to compare the correspondence between ground truth (GT) labels and the predicted labels (i.e., clusters) produced by the clustering algorithm. In a generic clustering problem, let m denote the number of GT classes $\{C_1, C_2, ..., C_m\}$ and k denote the number of predicted clusters $\{K_1, K_2, ..., K_k\}$. For this study, the number of clusters is fixed such that $k = m = 8$ (for the algorithms that allow for this, see Appendix B.2). In the contingency matrix M, each row corresponds to a GT class, and each column corresponds to a cluster. A cell M_{ij} indicates the number (or proportion) of samples from class C_i assigned to cluster K_j. In this work, the contingency matrix is adjusted for each class, yielding the normalized matrix M', defined as

$$M'_{ij} = \frac{M_{ij}}{\sum_{j=1}^{k} M_{ij}} \quad \forall i, j. \tag{3}$$

As a result, the sum of the elements in each row of the normalized matrix is equal to one:

$$\sum_{j=1}^{k} M'_{ij} = 1 \quad \forall i. \tag{4}$$

The evaluation procedure consists of two steps: the computation of the contingency matrix, followed by the assignment of each cluster to a corresponding GT class. The assignment of clusters to GT classes is determined to maximize the trace of the normalized contingency matrix, $Tr(M')$, which is the sum of the elements on its main diagonal. Maximizing $Tr(M')$ directly corresponds to maximizing the balanced accuracy (Grandini et al. 2020), defined as

$$ACC_B = \frac{1}{k}\sum_{i=1}^{k} M'_{ii} = \frac{Tr(M')}{k}. \tag{5}$$

The optimal assignment of clusters to classes is computed using the Kuhn-Munkres algorithm, as implemented in `scipy.optimize.linear_sum_assignment`[6] (Crouse 2016). For the best balanced accuracy, the uncertainty is computed using bootstrapping on the clusters, with a 99% confidence interval and $n = 10000$ bootstrapping iterations. Uncertainties obtained using bootstrapping account for class imbalance and the limited number of test samples, allowing a fair comparison of the results. For simplicity and compactness, the other results are reported without uncertainties.

3.2.3. Regression and symbolic regression

The regression results are evaluated using the MAE, consistent with the autoencoder case. Performance is compared against a baseline, defined as the mean value of the target variable, and the improvement is reported as a percentage. The evaluation of

[6] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

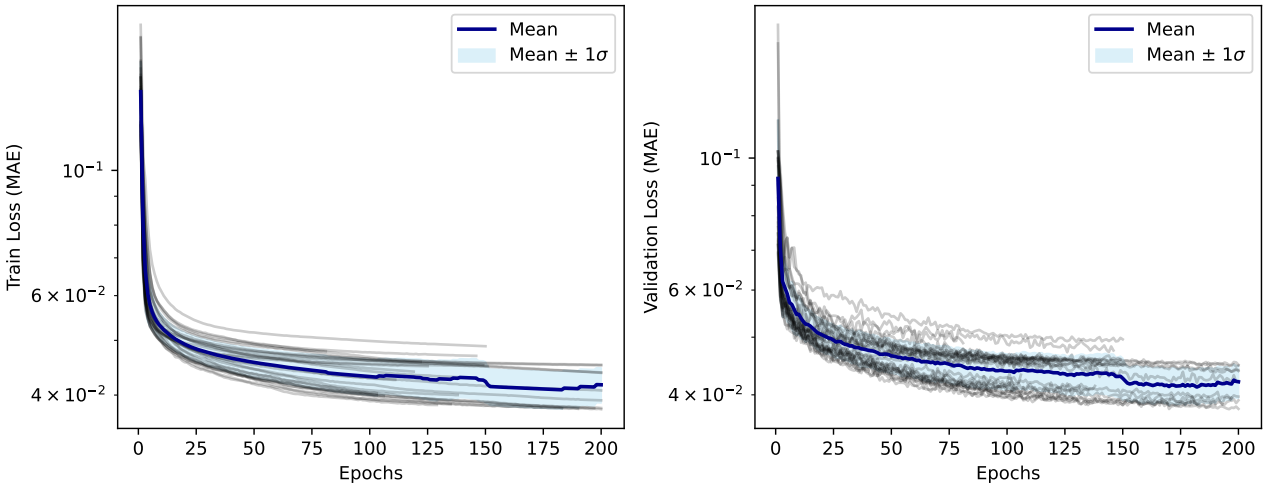


Fig. 4. Training (left panel) and validation (right panel) losses of the autoencoder, measured using MAE. The dark blue line is the mean loss across different hyperparameter configurations in the grid search (see Section 3.1.1), the light blue region indicates the mean $\pm 1\sigma$, and the gray lines indicate the loss for different hyperparameters configurations. The number of epochs varies across the experiments because of early stopping.

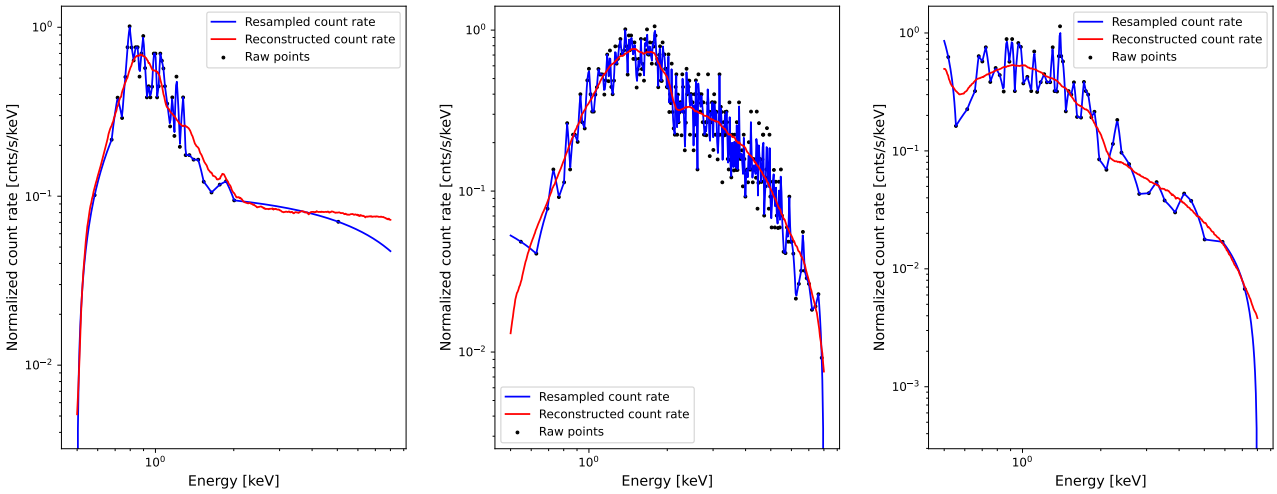


Fig. 5. Examples of autoencoder reconstructions (red) compared to the input resampled normalized spectra (blue). Left: source better reconstructed at lower energies (source: 2CXO J034639.3+240611, Obs. ID: 17250, type: LM-STAR). Center: source better reconstructed at higher energies (source: 2CXO J111438.8+324133, Obs. ID: 3137, type: AGN). Right: source reconstructed comparably at all energies (source: 2CXO J100433.8+411234, Obs. ID: 14498, type: AGN).

SR formulas is performed using two correlation coefficients: the Pearson correlation coefficient ρ[7], which measures linear correlations, and the Spearman correlation coefficient r_s[8], which assesses rank-based correlations (Martinez-Gil & Chaves-Gonzalez 2020). The Pearson correlation coefficient is defined as

$$\rho = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{s_X s_Y}. \quad (6)$$

The Spearman correlation coefficient is defined as

$$r_s = \frac{1}{n} \sum_{i=1}^{n} \frac{X_i Y_i - \bar{X}\bar{Y}}{s_X s_Y}. \quad (7)$$

In both cases, n represents the number of samples, X_i and Y_i represent the values associated with the i-th sample (i.e., the latent space variable and the physical formula value), $\bar{X}$ and $\bar{Y}$ are their respective means, and s_X and s_Y are their standard deviations. Both ρ and r_s range from -1 to 1, with larger absolute values indicating stronger correlations.

4. Results and discussion

We now report the results of applying the transformer autoencoder framework to our dataset of Chandra spectra. We first evaluate the reconstruction capabilities of the autoencoder. We then report on the classification and regression tasks and investigate interpretability through symbolic regression.

4.1. Spectral reconstruction

Figure 4 shows the training and validation losses for the autoencoder. A nonzero loss suggests that the autoencoder reconstructs a smoothed version of the input spectra, as illustrated in

[7] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html
[8] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html

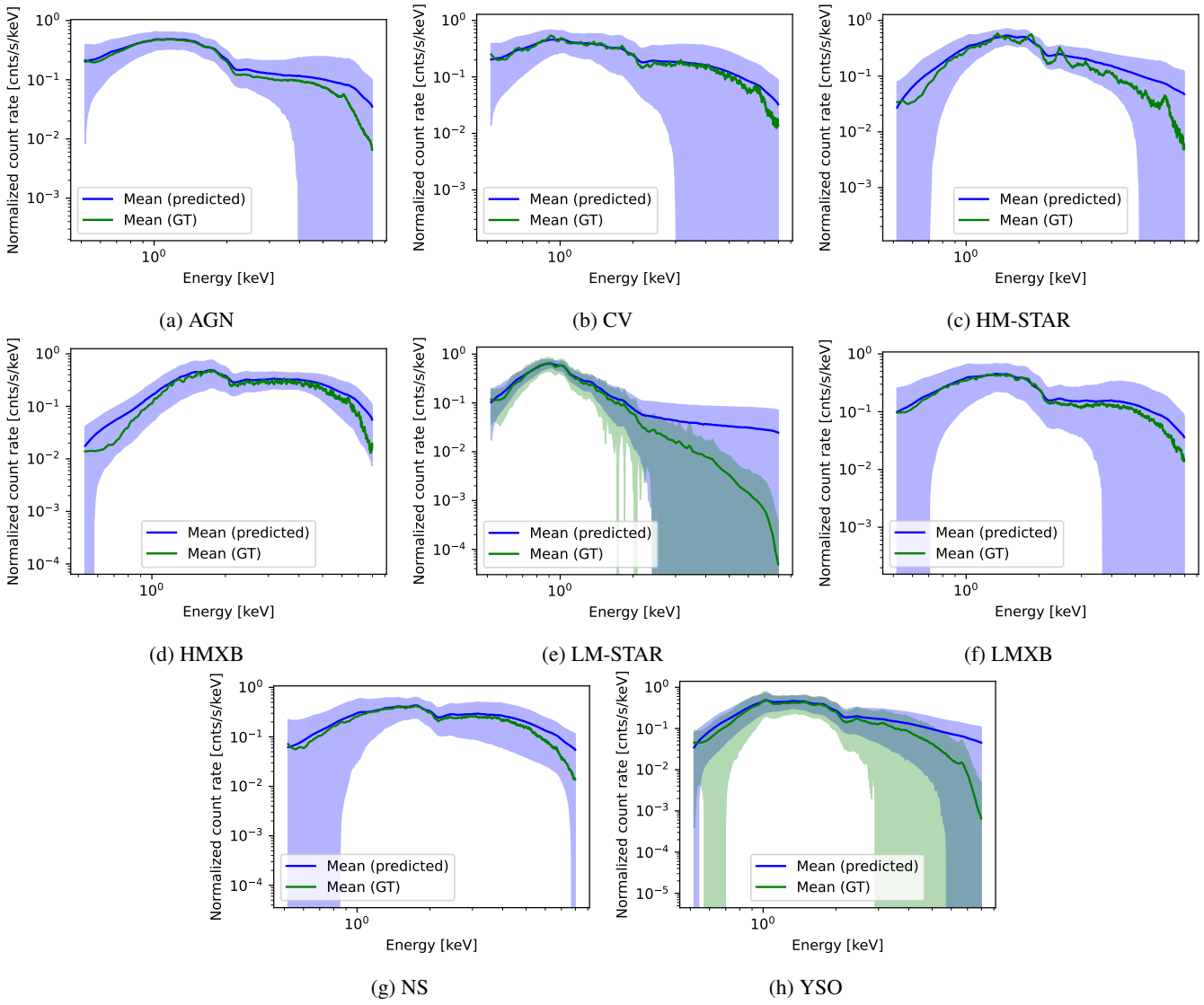


Fig. 6. Comparison of the mean spectra (green) and the corresponding reconstructed spectra (blue) for each class. The shaded regions represent the 1σ standard deviation around the mean.

Fig. 5. The use of a logarithmic scale on the y-axis and the different scales in the three spectra count ranges visually accentuate differences between the reconstructed and the original spectra, which appear larger at lower count rates. The reconstruction in the left panel has MAE $\approx$ 0.0337, the one in the central panel has MAE $\approx$ 0.0523 and the one in the right panel has MAE $\approx$ 0.0612. In the presented examples, the largest errors correspond to regions with more fluctuations in the count rate values. In all hyperparameter configurations, the losses on both the train and validation sets decrease, indicating the absence of overfitting. All the configurations result in a validation set MAE < 0.06. Overall, the autoencoder's performance is consistent across different hyperparameter settings, with a relatively small spread in performance, suggesting that the model is robust to hyperparameter choices. In all experiments, both the training and validation losses decrease rapidly within the first $\approx$25 training epochs. Fig. 6 shows the comparison, on the labeled test set and for each class, between the original spectra and the reconstructed spectra. The solid green line is the mean value for the original spectra, and the solid blue line is the mean value for the spectra reconstructed by the autoencoder. The shaded regions indicate a $\pm 1\sigma$ standard deviation around the mean. Overall, the reconstruction means are better for lower energies ($E \lesssim 3$ keV), and the original spectra mean values are within the 1σ intervals around the reconstructed spectra mean values. The most significant differences are observed for LM-STARs and YSOs. Both classes are characterized by lower normalized count rates at higher energies. In addition, LM-STARs are rare (96 out of 3241, or $\approx$3%). Still, the average reconstruction at high energies lies within 1σ of the average ground truth spectra. Reconstruction quality does not depend directly on the class since the autoencoder is unsupervised. However, since the autoencoder is predominantly trained on hard spectra (emission dominates in the 2–8 keV range), it implicitly learns to expect higher fluxes at high energies, which in turn leads to poor generalization on this class, whose soft spectra deviate from the norm.

4.2. Classification

Figure 7 shows the contingency matrix for the best hyperparameters' configuration (LR of 10^{-4}, eight embedding dimensions, two layers, eight attention heads, and a Gaussain mixture model

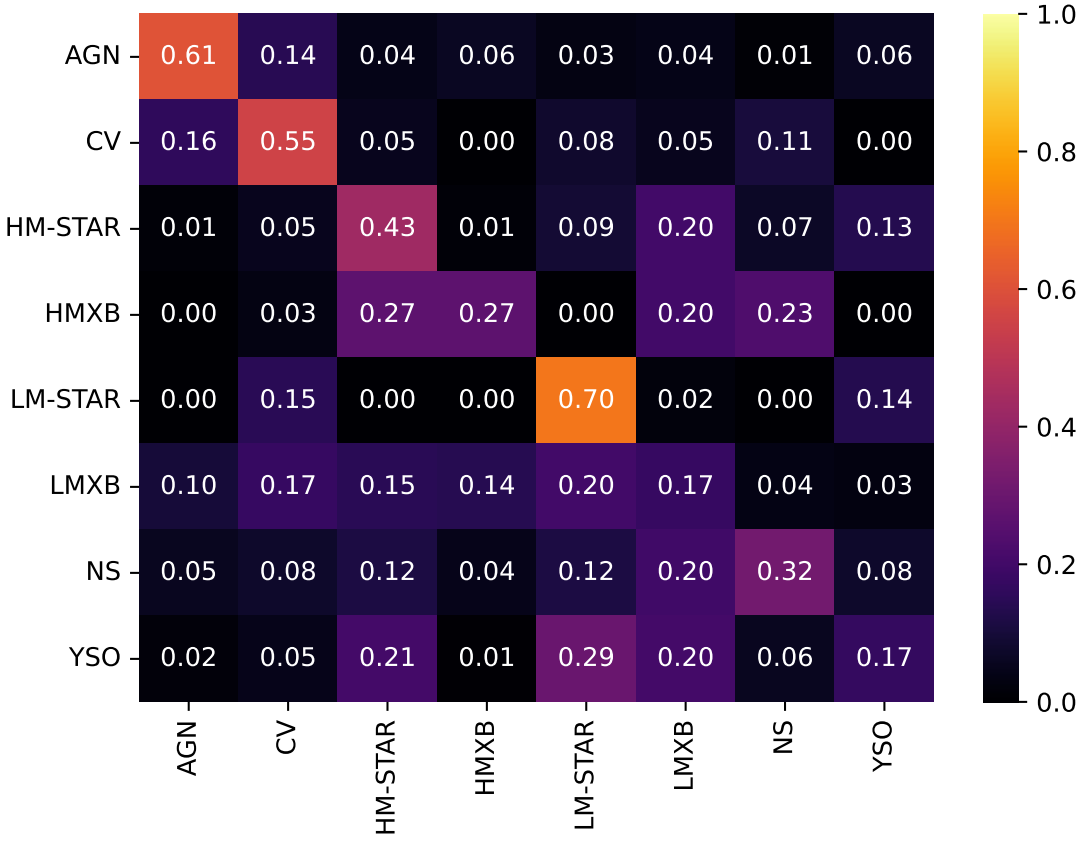


Fig. 7. Best contingency matrix, showing a balanced accuracy of ≈40%. The GT classes are shown on the *y*-axis, and the predicted classes are shown on the *x*-axis.

(GMM) with a "full" covariance type). The balanced accuracy is ≈40.17%, significantly higher than the 12.5% expected from an uninformed random classifier. Gaussain mixture model outperforms alternative algorithms because, unlike distance-based methods, it assigns probabilistic memberships that reflect that there is no one-to-one correspondence between spectral characteristics and source classes. The "full" covariance type allows each mixture component to capture distributions better aligned with continuous physical properties (e.g., hardness ratios), which the autoencoder captures along latent dimensions. Appendix C.1 presents a thorough comparison of different clustering algorithms and autoencoder hyperparameters.

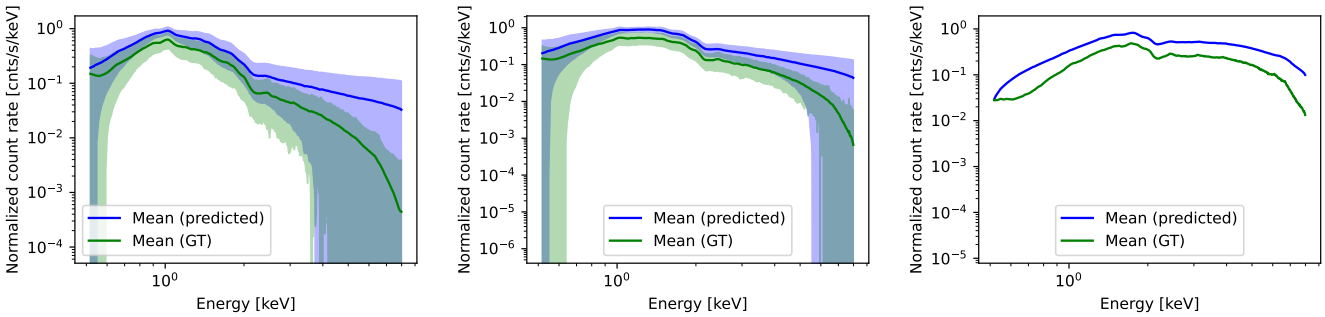
4.2.1. Spectral degeneracies

Our pipeline performs particularly well in classifying AGNs and LM-STARs. Interestingly, although LM-STAR spectra are not well reconstructed, their reconstruction patterns are still clearly distinguishable from those of other spectra. The lowest performance, instead, is observed for HMXBs, LMXBs, and YSOs. Note that the physical grounding of the latent space is further supported by the symbolic regression analysis (Section 4.3) and the regression results (Section 4.4), which we refer to throughout this discussion. To assess the significance of the differences between classes, for each resampled energy bin in the spectra, we computed

$$m = \frac{|\mu_A - \mu_B|}{\sqrt{\sigma_A^2 + \sigma_B^2}}, \tag{8}$$

where A and B represent the two classes under comparison. Higher values of m indicate significant differences between the spectral shapes of the two classes, while lower values suggest that the spectra are similar. If $m < 1$ for the majority of energy bins, the spectral differences between the classes are not considered statistically significant. Based on the contingency matrix, we identified the following main sources of confusion:

- YSOs and LM-STARs: in YSOs, X-ray emission arises from a combination of processes: particles accelerated by strong rotation-powered magnetic fields, heating of the corona, heating of the disk due to accretion, and shocks in outflows and jets. In contrast, LM-STARs (typically old main-sequence stars) exhibit X-ray emission primarily due to coronal heating, also linked to magnetic activity from rotation, but are not determined by accretion or jets. We noticed statistically significant differences between spectra ($>1\sigma$) only for $E \lesssim 2$ keV. Overall, $m < 1$ for ≈78% of the points in the predictions and ≈86% in the GT.
- YSOs and HM-STARs: the spectra of the two classes also have $m < 1$ for 100% of the energy bins. Massive YSOs drive powerful stellar winds whose X-ray signatures overlap with those of HM-STARs, which also produce hard emission from stellar-wind shocks and magnetic activity, making the two classes spectrally similar at the coarse energy resolution considered here.
- HMXBs and HM-STARs: an HM-STAR is the massive stellar companion in an HMXB system (Tan 2021; Ge et al. 2024). In HMXBs, the X-ray emission is due to accretion, including wind interactions and magnetic activity. In isolated HM-STARs, X-rays are generated mainly by stellar winds and magnetic processes. Since the autoencoder latent



(a) -1 ≤ hard_hs < -0.33 (b) -0.33 ≤ hard_hs < 0.33 (c) 0.33 ≤ hard_hs < 1

Fig. 8. Comparison of mean reconstructed (blue) and ground-truth (green) spectra for three hard_hs ranges: (a) soft spectra ($-1 \leq$ hard_hs < -0.33), (b) intermediate spectra ($-0.33 \leq$ hard_hs < 0.33), and (c) hard spectra ($0.33 \leq$ hard_hs ≤ 1). Solid lines show the mean spectrum across all sources in each bin, and shaded regions indicate the 1σ standard deviation around the mean.

variables are correlated with physical variables rather than directly with classes (cf Section 3.1.4), the clustering algorithm is not able to fully separate the spectra associated with different classes. Moreover, the two classes have $m < 1$ for 84% of the energy bins in predicted data and for 90% of the energy bins in GT data. The only significant difference between spectra is observed for $E \approx 4$ keV.

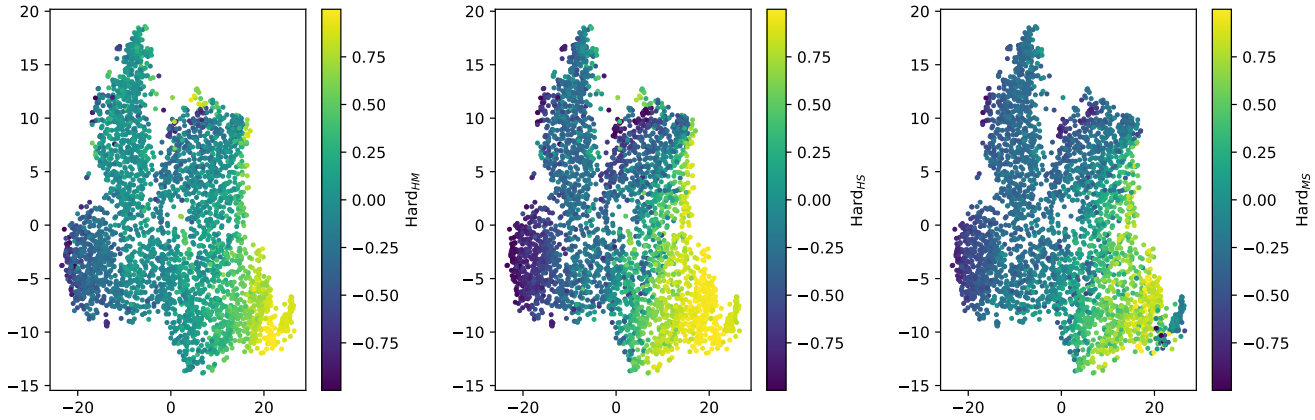
- AGNs and CVs: approximately 14% of AGNs are classified as CVs, and about 16% of CVs are classified as AGNs, suggesting spectral similarities notwithstanding the different physical nature of the two classes. In both cases, accretion is the primary mechanism of X-ray emission, producing both thermal radiation from hot disks and coronae and nonthermal emission from accelerated particles. The spectra of the two classes also have $m < 1$ for 100% of the energy bins.
- LM-STARs and CVs: approximately 15% of LM-STARs are mispredicted as CVs. This is physically motivated by the fact that the X-ray emission in CVs and low-mass stars is a thermal plasma, with similar temperatures, though the nature of the systems is fundamentally different. In CVs, the thermal plasma comes from the accretion flow between the companion star and the white dwarf, whereas in low-mass stars, thermal emission comes from the corona. The spectra of the two classes also have $m < 1$ for 100% of the points in the predictions and 98% in the GT, which implies that X-ray spectral signatures are insufficient to discern between the two.
- HM-STARs and LMXBs: both are characterized by relatively hard spectra compared to other classes (e.g., LM-STARs). Moreover, the spectra of the two classes also have $m < 1$ for 100% of the energy bins.
- YSOs and LMXBs: approximately 20% of YSOs are mispredicted as LMXBs. Both classes can produce hard X-ray emission, resulting in overlapping spectral shapes. The two classes have $m < 1$ for 100% of the energy bins, indicating similar spectra.
- HMXBs and NSs: some confusion is observed between these two classes. In general, HMXBs are expected to have a distinct hard X-ray emission related to Comptonization associated with accretion, whereas neutron stars, if truly isolated, are characterized by thermal blackbody emission. However, if the HMXB is in a low-accretion, quiescent state, thermal emission from the disk can dominate. Similarly, HMXB can also be heavily absorbed and embedded in strong winds, which can distort its rich spectrum and make it appear featureless. Both of these effects can be amplified at low signal-to-noise ratios.

4.2.2. Clustering

Figure 8 compares mean reconstructed and ground-truth spectra (similarly to Figure 6) for three ranges of hard_hs (see Table A.1): $[-1, -0.33]$, $[-0.33, 0.33]$, and $[0.33, 1]$ across the full test set, corresponding to soft, intermediate, and hard X-ray spectra, respectively. The hardness ratio hard_hs $= (F_{\rm hard} - F_{\rm soft})/(F_{\rm hard} + F_{\rm soft})$ quantifies the relative balance between the soft band (∼0.5–1.2 keV) and the hard band (∼2–8 keV) in Chandra data. The reconstruction quality varies across the energy range. For soft spectra (panel a), the model reconstructs the low-energy part of the spectrum well but shows larger discrepancies at higher energies. This behavior is consistent with regression to the mean, a well-known tendency of neural networks to predict values close to the dataset average for inputs at the extremes of the training distribution. Sources with very soft spectra provide the network with very few hard-band photons, so predictions revert toward mean values in that energy regime. Importantly, however, the 1σ deviations of the predictions still encompass the range of observed hard photons. For hard spectra (panel c), the reconstruction is more uniform across energy ranges. A limitation in spectral reconstruction at extreme energies does not imply that the latent representation fails to encode the corresponding spectral information. As Figure 9 and Table 2 show, the latent embeddings correlate strongly with the hardness ratios: for hard_hs, the mean absolute error is 0.18, representing a 64% improvement over the mean baseline, with an R^2 of 0.83. This indicates that the latent space encodes the global balance between soft and hard X-ray emission, even when the pixel-level reconstruction of the hard band is imperfect for the softest sources.

As expected from the contingency matrix, spectra from different classes and clusters exhibit some shared features. For instance, HM-STARs, LM-STARs, and YSOs spectra tend to be softer. This does not prevent the latent space from encoding physical properties, which can be used for downstream regression, as we show in Section 4.4. In general, the latent space is more informative of the summary statistics, such as the hardness ratios, than it is of the specific classes of objects.

To conclude, this clustering analysis demonstrates that the autoencoder can capture continuous physical properties better than discrete astrophysical class labels (see also Section 4.3).



(a) `hard_hm` (b) `hard_hs` (c) `hard_ms`

Fig. 9. Visualization based on tSNE and color coded using three hardness ratios. Colors range from blue (lower hardness ratios) to yellow (higher hardness ratios). Similar hardness ratio values tend to be close to each other.

Table 2. Regression results.

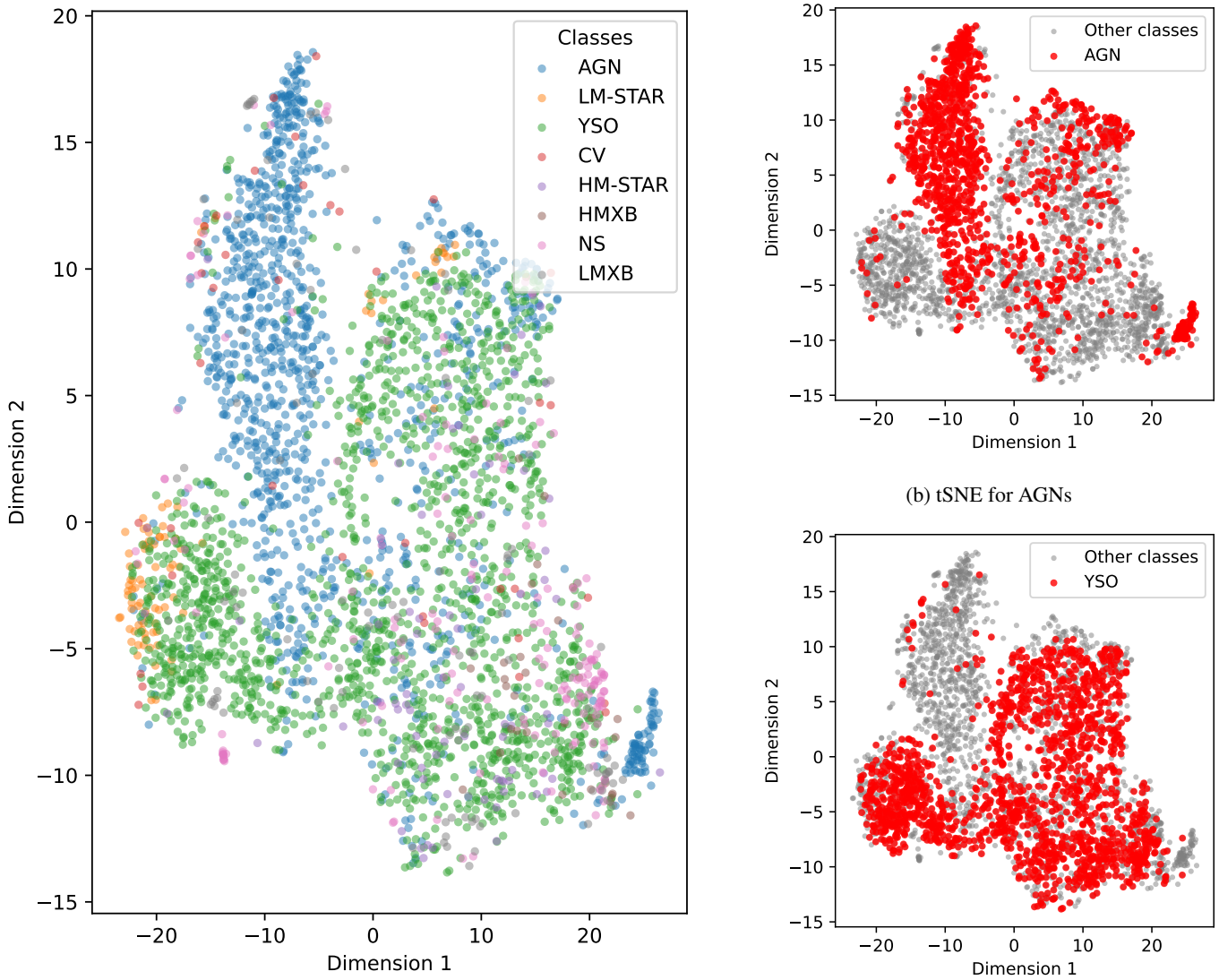
	All test set			Best model set			
Variable	Model	Base	Impr.%	Model	Base	Impr.%	Samples
hard_hs	0.18	0.49	64	–	–	–	–
hard_ms	0.14	0.35	60	–	–	–	–
hard_hm	0.11	0.27	60	–	–	–	–
var_prob_b	0.35	0.40	13	–	–	–	–
var_index_b	3.12	3.69	15	–	–	–	–
powlaw_gamma	0.62	0.90	32	0.56	0.86	35	1609
powlaw_nh	49.96	95.46	48	47.85	85.89	44	1609
bb_kt	1.29	2.38	46	1.01	1.87	46	188
bb_nh	34.70	70.09	50	144.37	289.97	50	188
brems_kt	9.26	15.65	41	3.31	4.70	30	541
brems_nh	74.46	155.04	52	33.74	77.28	56	541
apec_kt	2.59	3.72	30	1.03	1.37	25	904
apec_abund	0.41	0.49	17	0.33	0.41	20	904
apec_z	0.13	0.15	8	0.06	0.07	10	904
apec_nh	50.22	100.42	50	23.02	53.95	57	904

Notes. "Model" refers to the MAE of our method, while "Base" refers to the baseline. For global variables (top), model-specific sets are not applicable.

The overlap of source types within the latent space is not a failure of feature extraction, but a consequence of inter-class similarity. This analysis confirms that the learned representations are physically grounded, even if the spectra of sources with higher hardness ratios are better modeled because of their abundance.

4.2.3. Dimensionality reduction

Figure 10 shows a two-dimensional representation of the autoencoder latent space after training, as obtained using the tSNE dimensionality reduction method. Although the autoencoder was not trained for classification, the latent space can be used to differentiate between some classes, while acknowledging that some confusion remains in underrepresented classes. This highlights the importance of the self-supervised pretraining of the autoencoder and its potential as a foundation model for X-ray datasets. Section 4.4, however, shows that the autoencoder is effective at regression tasks, as it captures spectral properties. The most represented classes (see also Figure 10) are AGN (blue) and YSO (green), with the latter partially overlapping in tSNE space with less-represented classes, such as LM-STARs (orange) and NSs (pink). Considering the case of LM-STARs, their mean spectra appear visually different from those of YSOs, but their 1σ standard deviation bands overlap significantly across most energy bins, as reflected by $m < 1$ for ≈78% of bins in the predictions and ≈86% in the GT (see also Section 4.2.1). The two predominant classes, on the other hand, occupy distinct regions in the learned representation: YSOs are concentrated mostly in the bottom area, and most AGNs are clustered in the upper-left part of the plot. The AGNs span a wider range due to their physical heterogeneity, and their spectral shapes partially overlap with those of some YSOs (see also Figure 6). The overlaps with other classes depend on the intra-class variability and inter-class similarity of spectra, which for some couples of classes are not significantly different, as shown in Section 4.2.1.



(a) tSNE for all the classes

(b) tSNE for AGNs

(c) tSNE for YSOs

Fig. 10. Two-dimensional tSNE representations of the autoencoder latent space for (a) all the test set objects with assigned classes, (b) AGNs examples, and (c) YSOs examples.

Despite evident overlap between the YSOs and other types, some classes occupy adjacent, relatively distinct regions that indicate similarities in the X-ray spectral shape, which can be attributed to specific emission mechanisms. For example, the X-ray luminosities and overall spectral features of M-dwarf stars and YSOs can be similar, as they are both powered by magnetic activity near their surfaces. While both groups exhibit similar spectral features driven by magnetic activity, their separation suggests distinct activity regimes (e.g., accretion-dominated versus magnetically dominated) rather than fundamentally different spectral states. Interestingly, tSNE reveals a distinct cluster of AGNs in the bottom right corner of the bidimensional representation of the latent space, near YSOs and NSs, as they have similar physical properties (e.g., `hard_hm` and `hard_hs`, see also Figure 9). Upon closer inspection, all observations within this region correspond to multiple distinct detections of a single source: 2CXO J195928.3+404401. This source stands out due to its significantly higher hardness ratios compared to the rest of the AGN population in the dataset. 2CXO J195928.3+404401 is the nucleus of Cygnus A (Carilli & Barthel 1996), a powerful radio galaxy with strong jets coming out of the SMBH, and shocking the surrounding medium, which results in a very hard spectrum, compared with both other radio galaxies and average AGNs. Moreover, the spectrum of this source is among the hardest in the dataset, and it is well distinct from the bulk of the AGN class (see Figure 9).

The tSNE maps suggest that the representations that the autoencoder has learned are useful for classification. We now investigate to what extent the learned latent embeddings can discriminate between AGNs and stellar-mass compact objects. This is a relevant task because, in the absence of a redshift estimation, these two types of accretion-powered systems can be spectrally similar, despite the difference in the mass of the central accretor. In particular, the spectral photon index (Γ), which parametrizes the slope of a power law spectrum, for X-ray binaries in the low/hard state has a similar range compared to AGNs ($\Gamma \sim 1.5$) and similarities have been reported in the relationship between the X-ray spectral hardness and the Eddington

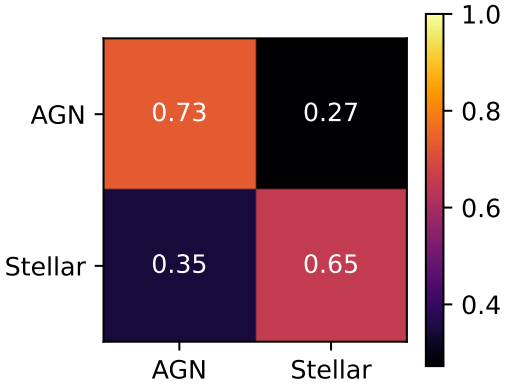


Fig. 11. Contingency matrix for AGNs and stellar-mass compact objects, with a balanced accuracy of ≈69%.

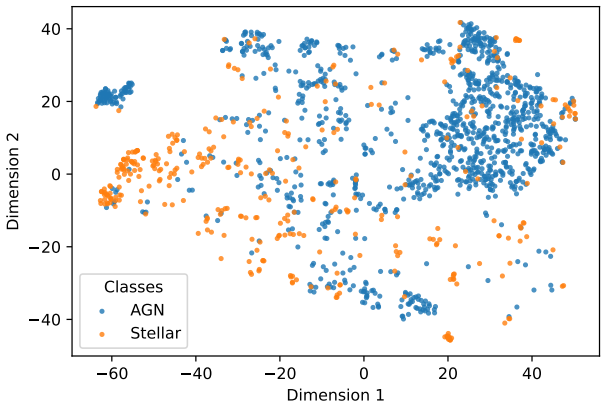


Fig. 12. Output of tSNE for clustering AGNs (blue) and stellar-mass compact objects (orange).

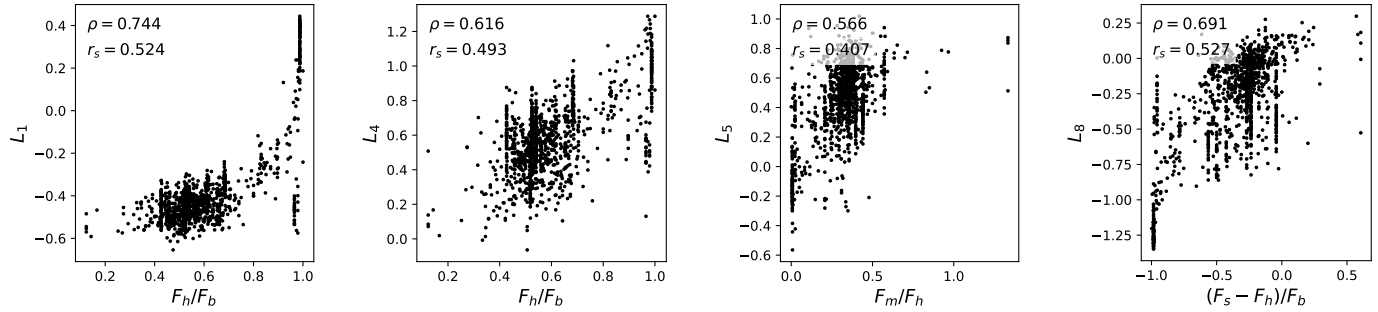
ratio in both AGNs and stellar-mass compact objects (Trichas et al. 2013). We group stellar-mass compact objects to include LMXBs, HMXBs, NSs, and CVs, and compare their latent representations to objects classified as AGNs. We fit GMM with the best hyperparameters found for the eight-class clustering on the reduced test set containing only objects from the two selected classes. Figure 11 shows the contingency matrix, corresponding to a balanced accuracy of ≈69%. Figure 12 presents the output of tSNE on the latent space for the reduced test set. We find that spectral information alone is sufficient to partially separate AGNs for compact objects of stellar mass, with the AGNs occupying the upper right corner of the representation space. As we show in Section 4.4, this relates to the average hardness ratio of the sources, with most AGNs having intermediate `hard_hs` values, and most stellar-mass compact objects displaying a relatively high hardness ratio in the same energy band, with the exception of the outlying Cygnus A detections, which are closer in the representation space to LMXBs and close to each other.

4.2.4. Comparative performance

Overall, the representations learned using the autoencoder do not underperform with respect to other automatic classification methods that only use the X-ray data. For example, Pérez-Díaz et al. (2024) achieve a 61% accuracy in a four-type classification task (relying on GMM) using tabulated properties from the CSC (v 2.0), which do not have the same representation/reconstruction power as the autoencoder features learned in our method. However, methods that use the full information from the event files in a representation learning framework, rather than only the spectral information, might have an advantage over spectral-only methods like the one presented here, as they can use variability information as well. Song et al. (2025) achieve a 60% accuracy in a similar eight-label task. However, their approach differs fundamentally, as they use a neural field estimator to model the photon arrival times (light curves), capturing variability information that is not included in spectra. In contrast, our method relies solely on spectral shapes. Moreover, the analysis of Song et al. (2025) is more computationally expensive at inference time because an optimization loop (gradient descent) is used to find the latent vector that best fits the data.

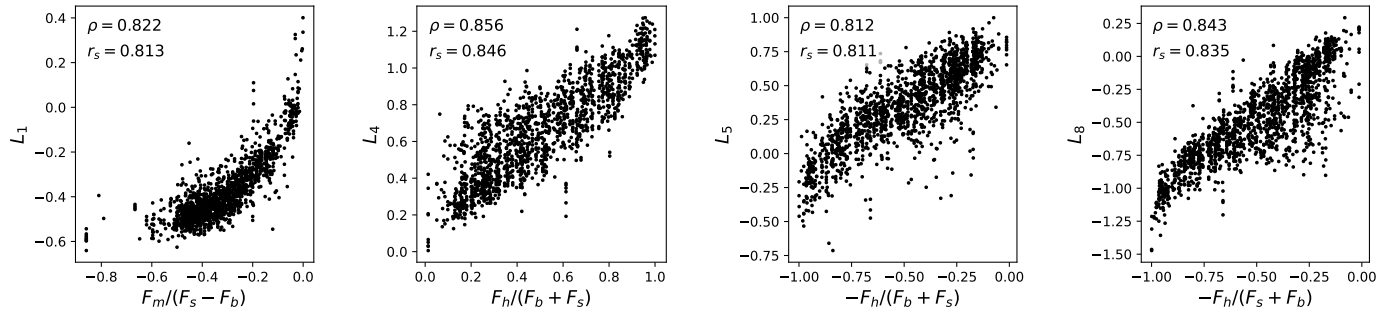
4.3. Physical correlations

Features learned using self-supervised approaches, such as the one used here, either compare fairly or overperform with respect to methods that use manually extracted features in downstream tasks of classification. For example, as we have pointed out, our spectral classifier has a similar accuracy to the one presented in Pérez-Díaz et al. (2024), which uses very carefully crafted, human-interpretable features manually extracted as part of the CSC processing, which include both spectral and time variability properties. Also, Song et al. (2025) demonstrate that automatically learned latent features for light curve reconstruction lead to an F1 score of 0.69 in the YSO versus AGN classification task, comparable to algorithms that use a much richer set of multiwavelength features (e.g., Yang et al. 2022a). However, it is of little use to have specialized automatically extracted features if they are not interpretable. If they overperform with respect to manually extracted features at classification, it is reasonable to expect that they relate to physically interpretable features, or to some combination of physical features that provide them with the observed predicted power. In order to investigate this, we performed symbolic regression to mathematically relate the latent features L_i to physically interpretable quantities, such as X-ray fluxes and hardness ratios. In the following paragraphs, we focus on sources of classes AGN and YSO, as the majority of astrophysical sources at high energies fall in these two groups. Figure 13 shows some of the most significant correlations for AGNs and reports both the Pearson correlation coefficient ρ and the Spearman correlation coefficient r_s for each case. Latent features L_1 and L_4 are respectively strongly and moderately correlated with the fractional flux emitted in the hard X-ray band (2–8 keV), F_h/F_b, a crucial diagnostic of the physical nature of the emission, as nonthermal processes tend to dominate emission in the hard band. AGNs with high L_1 and L_4 values are therefore likely to emit strongly through nonthermal processes such as inverse Comptonization, or they can also be heavily obscured AGNs, for which the soft X-ray emission is absorbed by dust. Latent feature L_5 is moderately correlated ($\rho \approx 0.566$, $r_s \approx 0.407$) with F_m/F_h, the ratio between the medium band (1.2–2 keV) flux and the hard band flux, which provides additional information about the emission mechanisms. Sources with enhanced medium band emission include thermal plasma sources with $kT \leq 1$–2 keV, such as supernova remnants and coronal emission from active stars, as well as sources with strong atomic emission lines in the 1–2 keV range, such as X-ray binaries with photoionized winds. Finally, the eighth latent dimension (L_8) is moderately correlated ($\rho \approx 0.691$, $r_s \approx 0.527$) with $(F_s - F_h)/F_b$, the relative fractional emission in the soft band (0.5–1.2 keV) with respect to the hard band. Sources with a high L_8 are thus likely to include



(a) L_1 (b) L_4 (c) L_5 (d) L_8

Fig. 13. Relevant correlations between latent space dimensions and physical formulas obtained from tabulated data for AGNs.



(a) L_1 (b) L_4 (c) L_5 (d) L_8

Fig. 14. Relevant correlations between latent space dimensions and physical formulas obtained from tabulated data for YSOs.

super soft sources AGNs, which are dominated by the thermal emission from the accretion disk, showing no hard coronal emission. Tidal Disruption Events also populate this soft regime. Figure 14 shows the derived correlations for YSOs. Latent features L_4 and L_5 are strongly correlated/anticorrelated with $F_h/(F_b + F_s)$, which relates to the fractional flux emitted in the hard X-ray band. Hard X-ray emission in YSOs is associated with coronal particle acceleration by magnetic reconnection events that produce X-ray flares, which are more common in young, rapidly rotating stars. Therefore, latent features L_4 and L_5 are good indicators of magnetic activity in stars. Latent feature L_1 is strongly anticorrelated with $F_m/(F_b - F_s)$, which is related to the fraction of the X-ray flux emitted in the 1–2 keV range. So, a low value of L_1 might indicate sources with thermal coronal plasma at intermediate temperatures, $kT \sim 0.8{-}1.5$ keV ($\sim 10{-}20 \times 10^6$ K), usually peaking in this energy range. These automatically learned features thus represent spectral summaries that have a direct link to specific physical mechanisms of X-ray emission in both AGNs and YSOs. They are an alternative to but also related to commonly used diagnostics, such as the hardness ratios. However, they carry more information than commonly used diagnostics about the specific spectral shape and time domain properties of X-ray sources and significantly reduce computational costs associated with computing these human-extracted properties.

4.4. Regression

We now investigate if the representations relate to human-interpretable properties that are associated with the physics of the X-ray sources. Specifically, we train a Huber regressor to predict the physically interpretable properties described in Table A.1 using the learned representations as input. Table 2 reports the results for the entire test set (column "All test set") as well as for each variable computed only on the subset of samples corresponding to the best-fitting spectral model. The "best model" refers to the spectral model that achieved the best fit statistic for each source. For example, for `powlaw_gamma`, in the "best model" set, we only consider sources that were best fit with the power-law model. All results are compared against a baseline (in which the mean value of each target variable is used as the predicted value for the entire test set). The column "Impr. (%)" indicates the percentage improvement with respect to the mean baseline. To illustrate the usefulness of the representations for regression on a crucial summary statistic, namely the hardness ratio, in Figure 9, we show the correlation between the learned latent representations and this quantity using a color coded tSNE projection. We note a clear correlation between the latent space and the three hardness ratios, which describe the overall spectral shape of the X-ray sources. Apart from the hardness ratios, which are model-independent quantities well captured by the learned neural summaries, Table 2 also lists several model-dependent physical parameters, such as the power law spectral slope (Γ) which indicates the acceleration of electrons in the hot plasma via synchrotron emission or inverse Comptonization, the blackbody temperature (`bb_kt`) when a thermal component is present, the hydrogen column density in the line of sight to the source (`n_H`), and even nonspectral diagnostics, such as the time variability of the X-ray flux (`var_prob_b`) of the source. When appropriate spectral models are considered ("Best model set" in the table), information gain with respect to the baseline model can be well above 50% for the hydrogen column densities, and close to 50% for the black body temperatures. There is also information gain for parameters such as

Table 3. Hardness ratio estimation comparison.

Variable	MSE		R^2	
	This work	Song et al. (2025)	This work	Song et al. (2025)
hard_hs	0.06	0.01	0.83	0.94
hard_ms	0.04	0.02	0.77	0.87
hard_hm	0.02	0.01	0.82	0.88

Notes. Comparison with the results reported in Song et al. (2025).

temporal flux variability, which is not intuitively related to the spectral shape of the source. This indicates that the features learned through spectral reconstruction capture multiple information about X-ray sources, and that there is mutual information between their spectral and temporal properties. We also compare the performance of our approach to other self-supervised learning methods, such as the Poisson Process Autodecoder (PPAD) presented in Song et al. (2025). The latter is a ResNet architecture trained to predict the mean Poisson arrival rate of individual X-ray photons in a particular energy range, using the individual photon event recordings (as opposed to the spectrum) as the input. Table 3 shows the results. The PPAD performs better in regression tasks with respect to our model, which indicates the advantage of using the fine-grained individual photon information with respect to coarser spectral bins. We note, however, that event lists can be extremely long (tens of thousands of photons), and that the PPAD is inherently inefficient at inference time, which makes it an unlikely tool for the analysis of large repositories of X-ray sources.

5. Summary and conclusions

In this work, we have addressed the challenges of characterizing large catalogs of high-energy X-ray sources. To this end, we have presented a self-supervised deep learning framework based on a transformer autoencoder that generates compressed low-dimensional representations of Chandra X-ray spectra. Our primary objective was to evaluate whether these learned latent representations can capture fundamental physical properties and serve as a robust alternative to manually extracted features for downstream tasks, including source classification, parameter regression, and symbolic regression, with a lower computational cost. The autoencoder compresses the spectra into an eight-dimensional latent space and preserves enough physical information about each source to allow for spectral reconstruction. The latent space has clustering properties that generally preserve astrophysical classes despite not having been trained on clustering. Known degeneracies in spectral shape between classes are also preserved, although the model is able to successfully separate between large accretors, mostly AGNs, and stellar-size accretors (e.g., X-ray binaries) when redshift information is not available. When clustering observations into eight classes (AGN, CV, HM-STAR, HMXB, LM-STAR, LMXB, NS, and YSO) the balanced accuracy is ∼40%, but it significantly increases for binary classification between AGNs and stellar-mass compact objects, reaching ∼69%, which is comparable to the performance of methods that use human-extracted features. The learned representation also preserves summaries that are relevant to the identification of astrophysically compelling sources that are being actively sought in large X-ray catalogs, such as TDEs, QPEs, and FXTs, as well as summaries related to temporal variability. Our framework is therefore interpretable and immediately applicable to dedicated searches for these objects. As next-generation X-ray surveys continue to produce vast amounts of spectral data, unsupervised and self-supervised methods will be able to capture relevant information, which in turn can be used for classification, discovery, and the training of foundation models in astrophysics.

Our approach has some limitations. The test set consists of ∼3200 labeled spectra, and class imbalance impacts the clustering performance. Some classes, such as LMXBs, are underrepresented, and some are spectrally similar to other types, making them more challenging to distinguish. Moreover, this work does not consider uncertainties on fluxes or faint sources. These constraints highlight the need for more comprehensive labeled datasets and techniques to incorporate uncertainty in reconstruction and self-supervised learning. Improvements to our proposed framework can incorporate uncertainties (Farrell et al. 2023) in the flux values to provide uncertainty estimations for the predictions. The regression techniques presented can also be used to estimate physical quantities from the latent space. The method can also be adapted to incorporate observations across multiple wavelengths and data in multiple modalities (Cuoco et al. 2022; Pinciroli Vago & Fraternali 2023; Buck & Schwarz 2024; Rizhko & Bloom 2024; Casado & García 2024). A deeper analysis of the similarities between the variability in spectra and light curves can also be an area for further research. This work contributes to the broader field of representation learning in astrophysics by demonstrating its applicability to high-energy spectral data. Future extensions will include multimodal learning by combining spectra with temporal or spatial information, anomaly detection in the latent space to discover rare sources, and transfer learning across different X-ray instruments.

Acknowledgements. N.O.P.V. acknowledges support from INAF and CINECA for granting 125 000 core hours on the Leonardo supercomputer (project INA24_C6B05). RA acknowledges financial support from INAF through the grant "INAF-Astronomy Fellowships in Italy 2022 - (GOG)" and the Bando Ricerca Fondamentale INAF 2024 Large Grant "Timing the Ultra Luminous x-ray Pulsars (TULiP)". The authors gratefully acknowledge the AstroAI multimodal group at CfA for their support, valuable suggestions, and insightful discussions. This research has made use of data obtained from the Chandra Source Catalog, provided by the Chandra X-ray Center (CXC). This research has also made use of software provided by the Chandra X-ray Center (CXC) in the application packages CIAO[9] and Sherpa[10].

References

Ackermann, M. R., Blömer, J., Kuntze, D., & Sohler, C. 2014, Algorithmica, 69, 184

Alosaimi, N., Alhichri, H., Bazi, Y., Ben Youssef, B., & Alajlan, N. 2023, Sci. Rep., 13, 433

Angeloudi, E., Audenaert, J., Bowles, M., et al. 2024, in The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track

[9] https://cxc.cfa.harvard.edu/ciao/

[10] https://cxc.cfa.harvard.edu/sherpa/

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. 1999, ACM SIGMOD Record, 28, 49
Arcelin, B., Doux, C., Aubourg, E., Roucelle, C., & (The LSST Dark Energy Science Collaboration). 2021, MNRAS, 500, 531
Arcodia, R., Merloni, A., Nandra, K., et al. 2021, Nature, 592, 704
Arnaud, K. A., Branduardi-Raymont, G., Culhane, J. L., et al. 1985, MNRAS, 217, 105
Banzhaf, W., Koza, J., Ryan, C., Spector, L., & Jacob, C. 2000, IEEE Intell. Syst. Appl., 15, 74
Bengio, Y., Courville, A., & Vincent, P. 2013, IEEE Trans. Pattern Anal. Mach. Intell., 35, 1798
Bishop, C. 2006, Pattern Recognition and Machine Learning (Berlin: Springer)
Bonaccorso, G. 2019, Hands-On Unsupervised Learning with Python (UK: Packt Publishing)
Buck, T., & Schwarz, C. 2024, Deep Multimodal Representation Learning for Stellar Spectra (USA: NeurIPS)
Carilli, C., & Barthel, P. 1996, A&A Rev., 7, 1
Casado, J., & García, B. 2024, arXiv e-prints [arXiv:2404.17720]
Chakraborty, J., Kara, E., Masterson, M., et al. 2021, ApJ, 921, L40
Chen, Z., Yeo, C. K., Lee, B. S., & Lau, C. T. 2018, in 2018 Wireless Telecommunications Symposium (WTS), 1
Cheng, T.-Y., Li, N., Conselice, C. J., et al. 2020, MNRAS, 494, 3750
Cheng, T.-Y., Huertas-Company, M., Conselice, C. J., et al. 2021, MNRAS, 503, 4446
Cheng, C., Liu, W., Fan, Z., Feng, L., & Jia, Z. 2024, Neural Netw., 172, 106111
Crouse, D. F. 2016, IEEE Trans. Aerospace Electronic Syst., 52, 1679
Cruise, M., Guainazzi, M., Aird, J., et al. 2025, Nat. Astron., 9, 36
Cuoco, E., Patricelli, B., Iess, A., & Morawski, F. 2022, Nat. Comput. Sci., 2, 479
Dillmann, S., Martínez-Galarza, J. R., Soria, R., Stefano, R. D., & Kashyap, V. L. 2025, MNRAS, 537, 931
Duffy, C., Hassanshahi, M., Jastrzebski, M., & Malik, S. 2025, Quantum Mach. Intell., 7,
Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. 1996, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96 (Portland, Oregon: AAAI Press), 226
Evans, P. A., Page, K. L., Osborne, J. P., et al. 2020, ApJS, 247, 54
Evans, I. N., Evans, J. D., Martínez-Galarza, J. R., et al. 2024, ApJS, 274, 22
Farrell, D., Baldi, P., Ott, J., et al. 2023, J. Cosmology Astropart. Phys., 2023, 016
Fotopoulou, S. 2024, Astron. Comput., 48, 100851
Fruscione, A., McDowell, J. C., Allen, G. E., et al. 2006, SPIE, 6270, 586
Gagliano, A., & Villar, V. A. 2023, arXiv e-prints [arXiv:2312.16687]
Galeev, A. A., Rosner, R., & Vaiana, G. S. 1979, ApJ, 229, 318
Ge, H., Tout, C. A., Chen, X., et al. 2024, ApJ, 975, 254
Gehrels, N., Chincarini, G., Giommi, P., et al. 2004, ApJ, 611, 1005
Grandini, M., Bagli, E., & Visani, G. 2020, arXiv e-prints [arXiv:2008.05756]
Greenacre, M., Groenen, P. J. F., Hastie, T., et al. 2022, Nat. Rev. Methods Primers, 2, 1
Guolo, M., Gezari, S., Yao, Y., et al. 2024, ApJ, 966, 160
Géron, A. 2025, Hands-On Machine Learning with Scikit-Learn and PyTorch (USA: O'Reilly Media)
Hinton, G. E., & Salakhutdinov, R. R. 2006, Science, 313, 504
Howie, S., Cheng, T.-Y., & Baugh, C. M. 2025, MNRAS, 544, 3124
Huang, S.-C., Pareek, A., Jensen, M., et al. 2023, npj Digital Medicine, 6, 74
Iwasawa, K., Liu, T., Boller, T., et al. 2024, A&A, 684, A153
Jain, A. K., Murty, M. N., & Flynn, P. J. 1999, ACM Comput. Surv., 31, 264
Jansen, F., Lumb, D., Altieri, B., et al. 2001, A&A, 365, L1
Jokanović, B., Amin, M., & Dogaru, T. 2015, IEEE Geosci. Remote Sensing Lett., 12, 204
Kingma, D. P., & Ba, J. 2014, arXiv e-prints [arXiv:1412.6980]
Kingma, D. P., & Welling, M. 2022, Auto-Encoding Variational Bayes (The Netherlands: University of Amsterdam)
Li, H., Li, J., Guan, X., et al. 2019, in 2019 15th International Conference on Computational Intelligence and Security (CIS), 78
Liang, Y., Melchior, P., Lu, S., Goulding, A., & Ward, C. 2023, AJ, 166, 75
Lin, D., Irwin, J. A., Berger, E., & Nguyen, R. 2022, ApJ, 927, 211
Luo, B., Brandt, W. N., Xue, Y. Q., et al. 2016, ApJS, 228, 2
MacQueen, J. 1967, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Statistics (USA: University of California Press), 1, 281
Majumdar, A. 2019, IEEE Trans. Neural Netw. Learn. Syst., 30, 312
Makke, N., & Chawla, S. 2024, Artif. Intell. Rev., 57, 2
Martinez-Gil, J., & Chaves-Gonzalez, J. M. 2020, Expert Syst. Appl., 160, 113663
Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. 2011, in Artificial Neural Networks and Machine Learning – ICANN 2011, ed. T. Honkela, W. Duch, M. Girolami, & S. Kaski (Berlin, Heidelberg: Springer), 52
Maulud, D., & Abdulazeez, A. M. 2020, J. Appl. Sci. Technol. Trends, 1, 140
Mei, Y., Chen, Q., Lensen, A., Xue, B., & Zhang, M. 2023, IEEE Trans. Evol. Comput., 27, 621
Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J. 2017, in 2017 International Joint Conference on Neural Networks (IJCNN), 364
Merloni, A., Lamer, G., Liu, T., et al. 2024, VizieR Online Data Catalog: 368, J/A+A/682/A34
Miniutti, G., Saxton, R. D., Giustini, M., et al. 2019, Nature, 573, 381
Mushotzky, R. 2018, SPIE, 10699 1069929
Muthukrishna, D., Mandel, K. S., Lochner, M., Webb, S., & Narayan, G. 2022, MNRAS, 517, 393
Müllner, D. 2011, arXiv e-prints [arXiv:1109.2378]
Ng, A., Jordan, M., & Weiss, Y. 2001, in Advances in Neural Information Processing Systems, eds. T. Dietterich, S. Becker, & Z. Ghahramani (Cambridge: MIT Press), 14
Orwat-Kapola, J. K., Bird, A. J., Hill, A. B., Altamirano, D., & Huppenkothen, D. 2022, MNRAS, 509, 1269
Owen, A. 2007, Contem. Math., 443, 59
Parker, L., Lanusse, F., Golkar, S., et al. 2024, MNRAS, 531, 4990
Peng, C. 2024, Appl. Comput. Eng., 102, 183
Pinciroli Vago, N. O., & Fraternali, P. 2023, Neural Comput. Appl., 35, 19253
Predehl, P., Andritschke, R., Arefiev, V., et al. 2021, A&A, 647, A1
Pérez-Díaz, V. S., Martínez-Galarza, J. R., Caicedo, A., & D'Abrusco, R. 2024, MNRAS, 528, 4852
Quirola-Vásquez, J., Bauer, F. E., Jonker, P. G., et al. 2022, A&A, 663, A168
Rajaei, A., Abiri, E., & Helfroush, M. S. 2024, Sci. Rep., 14, 29820
Rasmussen, C. 1999, in Advances in Neural Information Processing Systems, eds. S. Solla, T. Leen, & K. Müller (Cambridge: MIT Press), 14
Reynolds, D. 2009, in Encyclopedia of Biometrics, eds. S. Z. Li, & A. Jain (Boston, MA: Springer US), 659
Richter, T., Bahrami, M., Xia, Y., Fischer, D. S., & Theis, F. J. 2025, Nat. Mach. Intell., 7, 68
Rizhko, M., & Bloom, J. S. 2024, in Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges
Rizhko, M., & Bloom, J. S. 2025, AJ, 170, 28
Saxena, A. 2025, arXiv e-prints [arXiv:2511.05439]
Saxena, A., Prasad, M., Gupta, A., et al. 2017, Neurocomputing, 267, 664
Shamsolmoali, P., Zareapoor, M., Zhou, H., Tao, D., & Li, X. 2023, IEEE Trans. Image Proc., 32, 4486
Smith, R. K., Brickhouse, N. S., Liedahl, D. A., & Raymond, J. C. 2001, ApJ, 556, L91
Song, Y., Villar, V. A., Martínez-Galarza, R., & Dillmann, S. 2025, ApJ, 988, 143
Steinhaus, H., & others. 1956, Bull. Acad. Polonaise Sci., 1, 801
Storey-Fisher, K., Huertas-Company, M., Ramachandra, N., et al. 2021, MNRAS, 508, 2946
Sutskever, I., Vinyals, O., & Le, Q. V. 2014, in NIPS'14, Proceedings of the 28th International Conference on Neural Information Processing Systems (Cambridge, MA, USA: MIT Press), 2, 3104
Tan, C. 2021, J. Phys. Conf. Ser., 2012, 012119
Tavakoli, N., Siami-Namini, S., Adl Khanghah, M., Mirza Soltani, F., & Siami Namin, A. 2020, SN Appl. Sci., 2, 937
Tregidga, E., Steiner, J. F., Garraffo, C., Rhea, C., & Aubin, M. 2024, MNRAS, 529, 1654
Trichas, M., Green, P. J., Constantin, A., et al. 2013, ApJ, 778, 188
Van der Maaten, L., & Hinton, G. 2008, J. Mach. Learn. Res., 9, 2579
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, in Advances in Neural Information Processing Systems (USA: Curran Associates, Inc.), 30
Wang, Y., Yao, H., & Zhao, S. 2016, Neurocomputing, 184, 232
Webb, N. A., Coriat, M., Traulsen, I., et al. 2020, A&A, 641, A136
Weisskopf, M. C., Tananbaum, H. D., Speybroeck, L. P. V., & O'Dell, S. L. 2000, SPIE, 4012, 2
Wu, W., Wang, W., Jia, X., & Feng, X. 2024, Eng. Appl. Artif. Intell., 133, 108612
Xu, W. 2025, J. Mater. Inform., 5, N/A
Yang, T., & Li, X. 2015, MNRAS, 452, 158
Yang, H., Hare, J., Kargaltsev, O., et al. 2022a, ApJ, 941, 104
Yang, Z., Xu, B., Luo, W., & Chen, F. 2022b, Measurement, 189, 110460
Yuan, H., Chan, S., Creagh, A. P., et al. 2024, npj Digital Medicine, 7, 91
Zangrando, N., Fraternali, P., Petri, M., Pinciroli Vago, N. O., & Herrera González, S. L. 2022, Energy Inform., 5, 62
Zhang, T., Ramakrishnan, R., & Livny, M. 1996, ACM SIGMOD Record, 25, 103

Appendix A: Physical variables

Table A.1: Physical variables collected for showing their correlation with the latent space extracted by the autoencoder.

Variable	Description
F_s	Average soft band flux (0.5 - 1.2 keV)
F_b	Average broad band flux (0.5 - 7 keV)
F_m	Average medium band flux (1.2 - 2 keV)
F_h	Average hard band flux (2 - 7 keV)
hard_hs	hard (2.0-7.0 keV) - soft (0.5-1.2 keV) energy band hardness ratio
hard_ms	medium (1.2-2.0 keV) - soft (0.5-1.2 keV) energy band hardness ratio
hard_hm	hard (2.0-7.0 keV) - medium (1.2-2.0 keV) energy band hardness ratio
var_prob_b	Gregory-Loredo variability probability values
var_index_b	Intra-observation Gregory-Loredo variability index
powlaw_gamma	photon index, defined as $F_E \propto E^{-\gamma}$, of the best fitting absorbed power-law model spectrum to the source region aperture pulse invariant (PI) spectrum
powlaw_nh	N_H column density of the best fitting absorbed power-law model spectrum to the source region aperture PI spectrum
bb_kt	temperature (kT) of the best fitting absorbed black body model spectrum to the source region aperture PI spectrum
bb_nh	N_H column density of the best fitting absorbed black body model spectrum to the source region aperture PI spectrum
brems_kt	temperature (kT) of the best fitting absorbed bremsstrahlung model spectrum to the source region aperture PI spectrum
brems_nh	N_H column density of the best fitting absorbed bremsstrahlung model spectrum to the source region aperture PI spectrum
apec_kt	temperature (kT) of the best fitting absorbed astrophysical plasma emission code (APEC) model spectrum to the source region aperture PI spectrum
apec_abund	abundance of the best fitting absorbed APEC model spectrum to the source region aperture PI spectrum
apec_z	redshift of the best fitting absorbed APEC model spectrum to the source region aperture PI spectrum
apec_nh	N_H column density of the best fitting absorbed APEC model spectrum to the source region aperture PI spectrum

Appendix B: Methods

Appendix B.1. Autoencoder

Autoencoders consist of two main components (Yang et al. 2022b): the encoder and the decoder. The encoder maps the input data (the resampled normalized spectra $\hat{f}$) into a lower-dimensional space known as the latent space. Deep encoders are typically implemented using multiple layers of ANNs (Hinton & Salakhutdinov 2006; Wang et al. 2016), such as fully connected networks (Bengio et al. 2013), convolutional neural networks (CNNs) (Masci et al. 2011), or recurrent architectures (Sutskever et al. 2014), which progressively reduce the dimensionality of the input. For a single training example, the output of the encoder is a low-dimensional latent vector, denoted as z. The set of latent vectors for all examples captures the salient characteristics of the input data while discarding irrelevant information. The decoder, on the other hand, takes as input the latent vectors and attempts to reconstruct the original input data $\hat{f}$ as $\check{f}$. Like deep encoders, deep decoders consist of multiple layers of a neural network. The quality of the reconstruction, as measured by an appropriate loss function, such as the MAE, indicates how effectively the latent vectors capture the critical information in the input data. Accurate reconstruction demonstrates that the encoder has successfully preserved the most significant features in the latent space. Autoencoders are trained to minimize the reconstruction error, which quantifies the difference between the original input $\hat{f}$ and the reconstructed output $\check{f}$. Accordingly, the loss function $\mathcal{L}$ used during training depends on both $\hat{f}$ and $\check{f}$. By optimizing this loss function, the autoencoder learns to represent the input data efficiently in the latent space while retaining the ability to reconstruct it accurately.

Appendix B.2. Clustering

In this work, we compared different clustering algorithms, which rely on different principles:

- KMeans[11] (Steinhaus & others 1956; MacQueen 1967) partitions the data into k clusters, updating the cluster centroids iteratively to minimize the within-cluster variance. Each data point is assigned to the cluster with the closest centroid.
- GMM[12] (Reynolds 2009) models the data as a mixture of k Gaussian distributions (or components), each with its mean, covariance, and weight.
- Bayesian GMM[13] (Rasmussen 1999; Bishop 2006) extends GMM by incorporating Bayesian priors on the distributions parameters.
- Agglomerative clustering[14] (Müllner 2011; Ackermann et al. 2014) is a hierarchical clustering algorithm that starts by treating each data point as a single cluster and iteratively merges close clusters.
- Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)[15] (Zhang et al. 1996) creates a hierarchical structure with data summaries and is typically used in large datasets.
- Spectral clustering[16] (Ng et al. 2001) is a technique that uses the eigenvalues of a similarity matrix to reduce the dimensionality of the data before using an additional clustering algorithm (here, KMeans).
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[17] (Ester et al. 1996) does not require specifying the number of clusters. It identifies clusters based on the density of data points, grouping together points that are closely packed and marking points in low-density regions as outliers.
- Ordering Points To Identify Cluster Structure (OPTICS)[18] (Ankerst et al. 1999) is an extension of DBSCAN designed to handle clusters with different densities.

The clustering algorithms' hyperparameters are illustrated in detail in Table B.1.

Appendix B.3. t-distributed stochastic neighbor embedding

The first step of tSNE is measuring the similarity between each pair of points in a dataset in the original high-dimensional space, combining the Euclidean distance and a Gaussian distribution. The similarity between two points i and j is defined as

$$p_{ij} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)}, \tag{B.1}$$

where σ_i is the variance associated with the Gaussian distribution for each x_i. tSNE representations also depend on two hyperparameters: perplexity and early exaggeration. Perplexity is related to the number of nearest neighbors considered for computing the pairwise distances to preserve. Larger perplexity values capture global structures, while smaller values emphasize local relationships. Early exaggeration is a parameter that amplifies pairwise similarities during the initial optimization stages. In the low-dimensional space, tSNE also defines a pairwise similarity between couples of points based on the Student's t-distribution. The algorithm aims to minimize the difference between high- and low-dimensional similarities. The optimization is performed through gradient descent. In this work, tSNE is applied with a perplexity of 200 and an early exaggeration of 100. All plots show points belonging to different GT classes using different colors to facilitate comparisons.

Appendix B.4. Symbolic regression

Finding nonlinear mathematical expressions that approximate data requires an efficient search strategy. For this reason, we use `PySRRegressor`[19] to perform a heuristic search based on genetic programming (GP) (Banzhaf et al. 2000; Mei et al. 2023). Genetic programming relies on three fundamental operations on the trees:

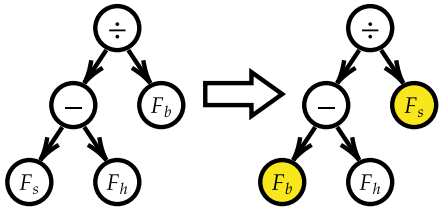
- Mutation: A tree is modified randomly (Fig. B.1a).
- Crossover (or recombination): Two trees are combined to create two new trees (Fig. B.1b).
- Selection: The best trees are selected.

[11] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[12] https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html
[13] https://scikit-learn.org/stable/modules/generated/sklearn.mixture.BayesianGaussianMixture.html
[14] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
[15] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html
[16] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html
[17] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
[18] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html
[19] https://astroautomata.com/PySR/api/

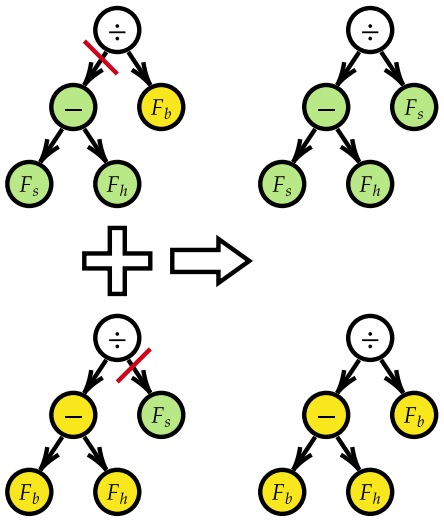
Table B.1: Overview of the selected clustering algorithms, their hyperparameters, and their clustering types.

Algorithm	Hyperparameters	Type
KMeans	`init`: ['k-means++', 'random'], `max_iter`: [30000], `tol`: [1e-5]	Hard
GMM	`covariance_type`: ['full', 'diag', 'tied', 'spherical'], `tol`: [1e-5], `max_iter`: [30000]	Soft
Bayesian GMM	`covariance_type`: ['full', 'diag', 'tied', 'spherical'], `tol`: [1e-5], `max_iter`: [30000]	Soft
Agglomerative	`linkage`: ['ward', 'complete', 'average', 'single']	Hard
BIRCH	`threshold`: `np.linspace(0.1, 0.5, 41)`	Hard
Spectral Clustering	`affinity`: ['nearest_neighbors', 'rbf']	Hard
DBSCAN	`eps`: `np.linspace(0.1, 2, 20)`, `min_samples`: [5, 10, 15]	Hard
OPTICS	`min_samples`: [5, 10, 15], `xi`: [0.05, 0.1, 0.2], `min_cluster_size`: [0.05, 0.1, 0.2]	Hard

Notes. The notation `np.linspace(start, stop, num)` indicates an array of `num` evenly spaced numbers starting with `start` and ending with `stop`.



(a) Mutation from the expression $(F_s - F_h)/F_b$ to the expression $(F_b - F_h)/F_s$. The nodes highlighted in yellow are different than those in the first tree.



(b) Crossover from the expressions $(F_s - F_h)/F_b$ and $(F_b - F_h)/F_s$. Two possible resulting expressions are $(F_s - F_h)/F_s$ and $(F_b - F_h)/F_b$.

Fig. B.1: Examples of mutation and crossover when applying GP to SR

During the search for optimal trees, their quality is assessed using MSE. For the values l_i in the latent dimension L_i and the corresponding values e_i generated by a tree's expression, MSE is calculated as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(e_i - l_i)^2. \tag{B.2}$$

A GP algorithm consists of four steps:

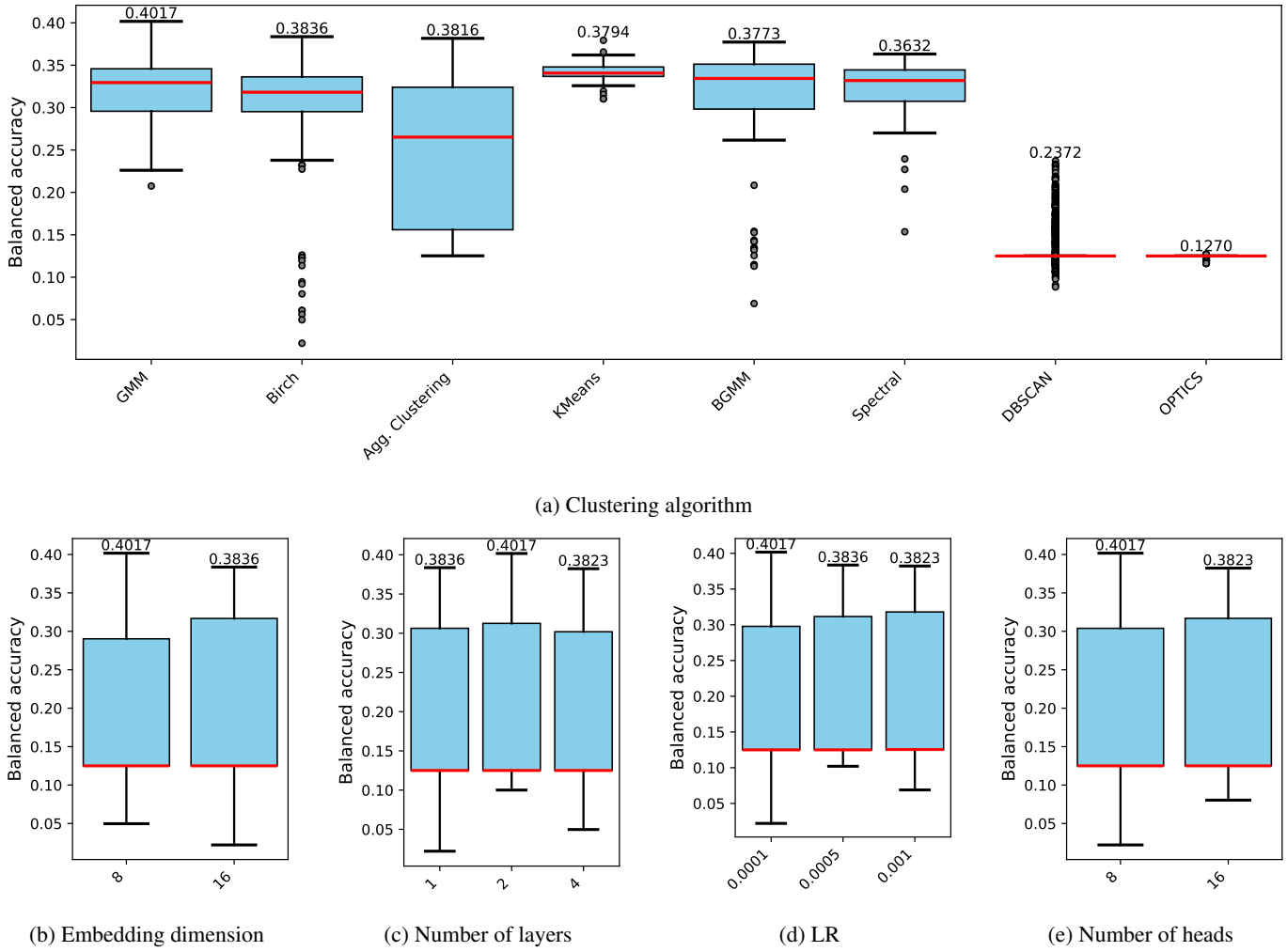
1. Initialization: generate an initial population of N_t random trees, where each tree represents a random physical expression.
2. Selection: evaluate the candidate trees using the MSE between the expression represented by the tree and the latent dimension data. Retain only the top N_b trees with the lowest MSE values.
3. Evolution: apply genetic operations (mutation, crossover, and selection) to the population of trees over multiple iterations to refine their quality.
4. Termination: stop the evolutionary process once a termination condition is satisfied (here, a predefined number of iterations)

As a result, the trees progressively improve with every iteration, producing easily interpretable physical formulas that more accurately approximate the latent space dimensions. In this work, the maximum depth of the trees is $d = 5$ (to limit the number of variables to 3 and the number of operators to 2), the maximum complexity of the expressions is $c_{max} = 7$, and the number of iterations before termination is $N_{it} = 400$. The best model is the one with the lowest MSE. All other hyperparameters are kept at their default values.

Appendix C: Results

Appendix C.1. Clustering

Figure C.1 summarizes the effect of the hyperparameters listed in Table 1 and the clustering algorithms summarized in Table B.1 on balanced accuracy. All the figures use box plots to visualize the results. A box plot presents distributions using five main components: the median (the red line inside the box representing the 50th percentile), the lower quartile (Q1) and upper quartile (Q3) (i.e., the bottom and top lines of the box, corresponding to the 25th and 75th percentiles), the interquartile range (IQR) (i.e., the difference between Q3 and Q1), the whiskers (lines extending from the box to the smallest and largest values within 1.5 × IQR from Q1 and Q3), and the outliers (individual points beyond the whiskers). In each figure, the box plots show the distribution of performance metrics when only one hyperparameter is fixed at a specific value, while all other hyperparameters are allowed to vary freely. The clustering algorithm choice introduces the most significant variations in balanced accuracy. The remaining



(a) Clustering algorithm

(b) Embedding dimension (c) Number of layers (d) LR (e) Number of heads

Fig. C.1: Impact of different autoencoder hyperparameters and clustering algorithms on the balanced accuracy. The red horizontal bars indicate the median values, and the black horizontal bars that delimit the light blue region indicate the first and the third quartiles. The clustering algorithm box plots are sorted by decreasing maximum balanced accuracy.

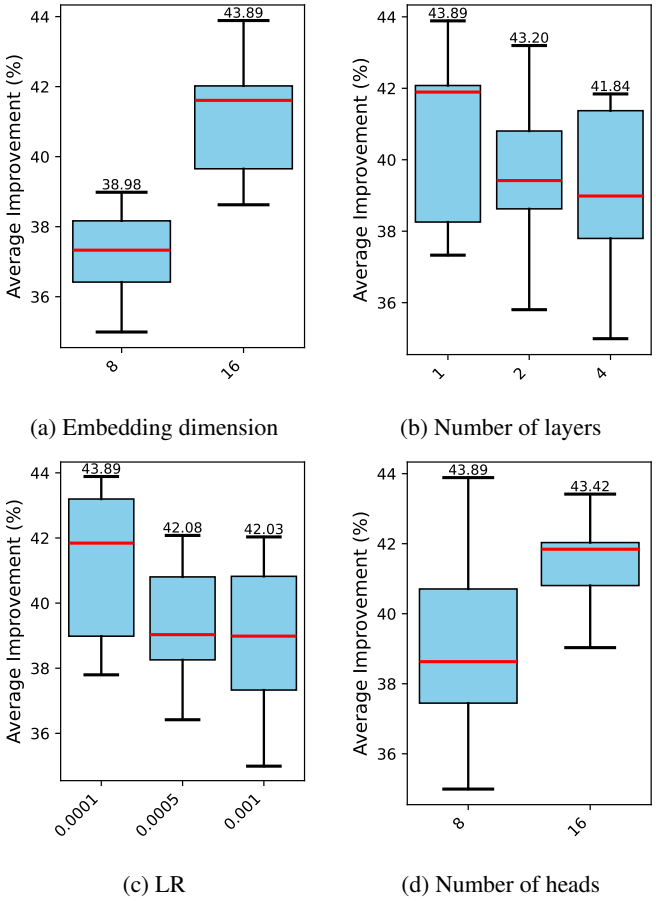
hyperparameters pertain to the autoencoder and have a smaller impact on balanced accuracy:

- Embedding dimensions: using eight embedding dimensions results in a small latent space, which provides the best performance. This outcome indicates that the most relevant spectral features are captured in a smooth spectrum, while finer-grained structures are less significant.
- Number of layers: The optimal architecture consists of two layers, balancing the simplicity of a single-layer model and the potential overfitting risks of a four-layer model.
- LR: A LR of 10^{-4} yields the best balanced accuracy. Large rates lead to worse latent representations due to overfitting (Li et al. 2019) and ineffectiveness in optimization (Peng 2024).
- Number of heads: Using eight heads is preferable as the autoencoder captures essential features in the latent space while avoiding focusing on irrelevant or noisy features.

Some clustering algorithms allow the number of clusters to be explicitly specified, unlike DBSCAN and OPTICS, which determine clusters based on density. Algorithms that allow one to define the number of clusters have better performance compared to those that rely on density-based clustering methods. Among all, GMM achieves the highest balanced accuracy ($\approx 40\%$). KMeans also performs well, achieving a balanced accuracy of 37.94% (approximately 2% lower than GMM) while being the least sensitive to variations in other hyperparameters. Overall, the choice of the clustering algorithms' hyperparameters introduces a significant variation in balanced accuracy, compared with the uncertainty computed on the highest balanced accuracy ($40.5^{+4.4}_{-4.2}\%$ for a 99% confidence interval around the median). Simpler configurations perform better because they avoid overfitting and effectively capture the most relevant spectral information.

Appendix C.2. Regression

Figure C.2 summarizes the influence of the hyperparameters listed in Table 1 on the average MAE improvement across all predicted physical quantities, with respect to a mean baseline (i.e., predicting the mean value of the training set). The embedding dimension shows the most significant impact, with an improvement of $\approx 5\%$ when increasing the dimension from eight to 16. This result is in contrast with that of the clustering scenario, where smaller embedding dimensions lead to better results. In regression, a higher-dimensional latent space retains more details, which are relevant to the prediction of physical quantities.



(a) Embedding dimension (b) Number of layers

(c) LR (d) Number of heads

Fig. C.2: Impact of different autoencoder hyperparameters and regression algorithm on the average MAE improvement over the baseline. The red horizontal bars indicate the median values, and the black horizontal bars that delimit the light blue region indicate the first and the third quartiles.

This representation supports fine-grained reconstructions that benefit the downstream regression task. Clustering, instead, benefits from smaller embeddings, which emphasize coarse-grained features and neglect irrelevant detail.