Cross-Temporal Spectrogram Autoencoder (CTSAE): Unsupervised Dimensionality Reduction for Clustering Gravitational Wave Glitches

Yi Li[1] Yunan Wu[2] Aggelos K. Katsaggelos[1,2]
[1] The Department of Computer Science, Northwestern University, Evanston, IL, 60208
YiLi2023.1@u.northwestern.edu
[2] The Department of Electrical Computer Engineering, Northwestern University, Evanston, IL, 60208
yunanwu2020@u.northwestern.edu, aggk@eecs.northwestern.edu

Abstract

The advancement of The Laser Interferometer Gravitational-Wave Observatory (LIGO) has significantly enhanced the feasibility and reliability of gravitational wave detection. However, LIGO's high sensitivity makes it susceptible to transient noises known as glitches, which necessitate effective differentiation from real gravitational wave signals. Traditional approaches predominantly employ fully supervised or semi-supervised algorithms for the task of glitch classification and clustering. In the future task of identifying and classifying glitches across main and auxiliary channels, it is impractical to build a dataset with manually labeled ground-truth. In addition, the patterns of glitches can vary with time, generating new glitches without manual labels. In response to this challenge, we introduce the Cross-Temporal Spectrogram Autoencoder (CTSAE), a pioneering unsupervised method for the dimensionality reduction and clustering of gravitational wave glitches. CTSAE integrates a novel four-branch autoencoder with a hybrid of Convolutional Neural Networks (CNN) and Vision Transformers (ViT). To further extract features across multi-branches, we introduce a novel multi-branch fusion method using the CLS (Class) token. Our model, trained and evaluated on the GravitySpy O3 dataset on the main channel, demonstrates superior performance in clustering tasks when compared to state-of-the-art semi-supervised learning methods. To the best of our knowledge, CTSAE represents the first unsupervised approach tailored specifically for clustering LIGO data, marking a significant step forward in the field of gravitational wave research. The code of this paper is available at https://github.com/Zod-L/CTSAE

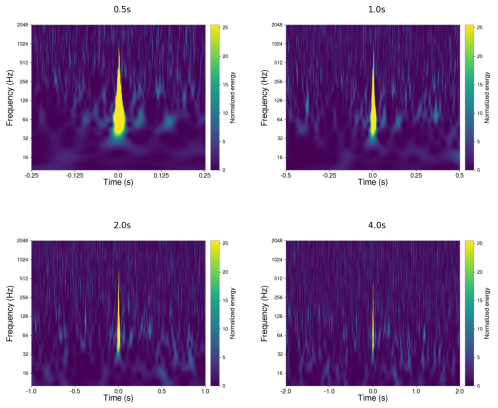

Figure 1. An example of a blip glitch with four spectrograms corresponding to time windows of 0.5 s, 1.0 s, 2.0 s, and 4.0 s. The horizontal axis, the vertical axis and the color intensity in each time-frequency bin represent time, frequency and the energy level, respectively.

1. Introduction

The detection of gravitational waves has revolutionized our cosmic exploration by opening a new window into the universe. The Laser Interferometer Gravitational-Wave Observatory (LIGO) marked a milestone by confirming these spacetime ripples for the first time in September 2015 [1, 3]. These groundbreaking observations are contingent upon highly sensitive detection systems capable of distinguishing the faintest spacetime fluctuations amidst a myriad of environmental and instrumental noises [4]. Among these, non-Gaussian noise bursts, or glitches, present significant challenges to the clarity and reliability of gravitational wave detections [2]. The Gravity Spy dataset comprises time-frequency spectrograms capturing the characteristics of glitches. Each glitch instance consists of four frequency spectrograms with distinct time windows: 0.5 s, 1.0 s, 2.0 s, and 4.0 s. An example of glitch spectrograms is shown in

Fig. 1.

To address this challenge, the Gravity Spy project combines volunteer efforts and machine learning to categorize the glitches in LIGO's time-series data [30]. Despite the project's success in glitch classification, a notable gap remains in identifying and classifying glitches across the main channel and auxiliary channels monitored by LIGO detectors [2], which is essential for understanding the causal relations between these channels. Given the infeasibility of labeling data from all channels in the future GravitySpy 2.0 dataset, we propose an unsupervised learning model, called CTSAE, to uncover the underlying correlations between different glitches. This method aims to enhance the glitch identification process and the accuracy of gravitational wave detections, thereby advancing the frontier of space technology and exploration. Since the GravitySpy O4 dataset on the main and auxiliary channels is still under the early stage of construction, we apply CTSAE to the GravitySpy O3 dataset on the main channel where only main channel glitches are involved to evaluate the cluster performance. The contribution of our paper is as follows:

- We develop a novel four-branch autoencoder that integrates CNN and ViT to process glitches across four different time window durations, facilitating spatial feature extraction of glitch characteristics.
- We introduce a novel CLS fusion module designed for effective inter-branch communication, enabling the extraction of temporal glitch features by capturing dynamic changes over time.
- CTSAE is the first method to cluster gravitational wave glitches in an unsupervised learning manner, achieving superior performance over existing semi-supervised methods deployed by Gravity Spy that rely on partial training labels.

2. Related Work

Deep Learning-Based Approaches for Glitch Classification and Clustering Deep learning methods [5–7, 11, 19, 28] have been widely used for classifying and clustering glitches. For instance, Coughlin *et al*. [11] demonstrated the effectiveness of utilizing the VGG16 architecture [24] for glitch classification tasks. Later on, Wu *et al*. [28] introduced a multi-view fusion model, combined with an attention mechanism to improve the glitch classification performance. They both extended their works to the clustering task by leveraging their models as feature extractors, facilitating the clustering of unidentified classes. In DIRECT [6], a contrastive learning framework was established to train a deep feature extraction model, utilizing true class labels to improve its learning efficacy. Similarly, Bahaadini *et al*. [7] explored a semi-supervised learning approach, where a virtual adversarial model was developed and trained with a mixture of labeled and unlabeled data. Regarding the task of glitch classification and clustering, existing works primarily focus on models trained under conditions of full supervision or semi-supervision, necessitating the use of pre-labeled glitch data. To the best of our knowledge, our work is the first unsupervised learning method for clustering LIGO glitches.

Autoencoder-based Self-supervised Dimensionality Reduction Typically, to reduce computational cost and remove irrelevant information, high-resolution images are reduced to a low-dimensional space before undergoing clustering. Traditional approaches to image dimensionality reduction largely rely on shallow machine learning algorithms, such as Principal Component Analysis (PCA). However, with the rapid development of deep learning over recent years, deep neural networks have been introduced for dimensionality reduction in a self-supervision manner. Autoencoder (AE) [23] stands out as a prominent self-supervised algorithm widely used for both dimensionality reduction and representation learning. Despite its simplicity in implementation and training, AE is prone to overfitting, especially with limited training data. To avoid the issue of merely copying the input without capturing high-level features, several variants of AE [15, 20, 26] are proposed. A common strategy to mitigate overfitting is data augmentation [8, 9, 15, 17, 21, 26, 27, 31]. For instance, denoising autoencoder [26, 27] augments training data by introducing random noise to the input image, with the AE then trained to remove this added noise. Zhang *et al*. [31] introduced input corruption by removing color channels while Pathak *et al*. [21] proposed a context encoder that was trained by the inpainting of randomly masked images. He *et al*. [17] achieved state-of-the-art performance in feature extraction by masking random image patches and reconstructing on the unmasked patches. In tasks related to clustering and representation learning, it is important to model the similarity and dissimilarity between images. Contrastive learning [9, 15] aims to learn representations that bring similar images closer together in the feature space while keeping dissimilar ones apart. It leverages data augmentation to generate positive samples in the absence of ground truth class labels. While these methods are effective in extracting high-level features from images, they predominantly rely on data augmentation, which is not suitable for our Gravity Spy dataset. Augmenting glitch spectrograms can significantly distort their physical semantics. Therefore, we choose to use a standard AE framework in this work.

Combining CNN and ViT for Enhanced Feature Extraction Recent advancements have been seen in the combinations of CNN and ViT, yielding great success across different tasks. Such combinations excel at capturing both global and local features, which is critical for effective unsupervised clustering. Previous studies have shown the benefits of both sequential [13, 14, 25] and parallel structures [10,

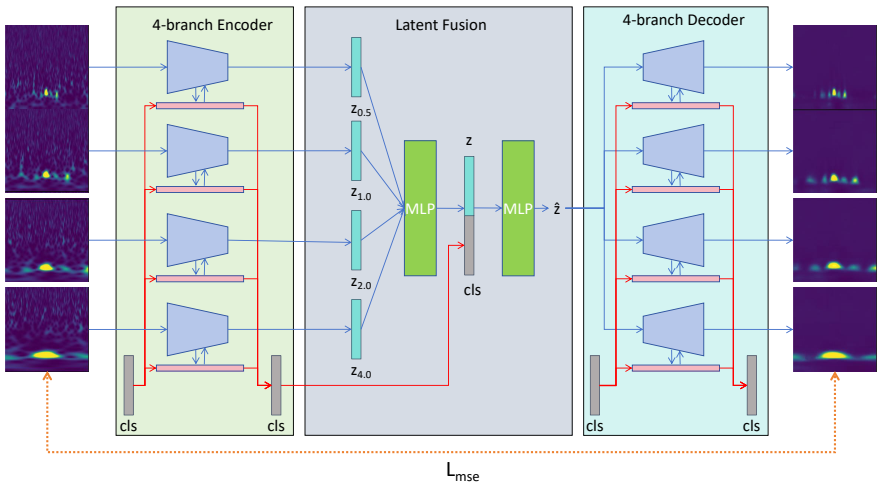

Figure 2. The architecture of CTSAE. The input comprises a glitch with four spectrograms of different time-window durations (0.5 s, 1.0 s, 2.0 s and 4.0 s). Four CNN-ViT encoders encode each spectrogram to extract high-level features, interconnected via s shared CLS token. These features, along with the shared CLS token, are fused by an MLP into a low-dimensional latent vector. This latent code is then shared among four decoders to generate spectrograms of different durations. Decoders communicate through a shared CLS token, similar to the encoder setup.

22] in combining the convolutional and transformer-based methods. Srinivas *et al*. [25] improved the performance of instance segmentation and object detection by replacing the last three CNN bottleneck blocks in ResNet [16] with self-attention mechanisms. d'Ascoli *et al*. [13] proposed a gated positional self-attention (GPSA) layer with a "soft" convolutional inductive bias where each self-attention layer decides whether to behave as a convolutional layer based on the context. In the LeViT model [14], the initial patchification process is replaced with a compact CNN encoder. While these sequential combinations have achieved substantial improvements, they do not possess an advantage for information exchange within the CNN part of the multi-branch structures due to feature unalignment problems. In terms of parallel structures, Peng *et al*. [22] introduced Conformer for image classification, featuring separate convolution and transformer branches linked via Feature Coupling Units (FCU). Moreover, Chen *et al*. [10] designed a lightweight Mobile-Former, which establishes a bidirectional connection between the MobileNet [18] and transformer branches. Aiming to capture high-level features across glitches from all four time-window durations, our work aligns with the parallel structure [22], which we expand into a four-branch autoencoder. This improved model effectively addresses the global-local feature fusion from convolutions and transformers problem.

3. Method

To construct a latent space that is simultaneously smooth, discriminative, and low-dimensional, we introduce a multi-branch AE to extract high-level features from glitch spectrograms. Our CTSAE model leverages the CNN-ViT blocks [22], combining convolution and vision transformer [12] to effectively capture both global and local glitch features. The distinct branches of CTSAE are interconnected through an innovative fusion strategy we have developed.

3.1. Overview

In the context of unsupervised learning algorithms, distinguishing glitch spectrograms from different classes presents a significant challenge, especially when they exhibit similar patterns across specific timescales. To effectively distinguish those closely similar classes, features from different timescales should be analyzed. Given that the Gravity Spy project generates glitches in four different time window durations, we propose CTSAE, a four-branch AE for feature extractions, as illustrated in Fig. 2. Our model processes an input glitch composed of four spectrograms, each representing a different duration, denoted as $I_{0.5}, I_{1.0}, I_{2.0}, I_{4.0}$. These four spectrograms are parallelly fed into four CNN-ViT encoders E_i, which compresses them into a lower-dimensional vector z_i in the latent space following:

$$z_i = E_i(I_i) \quad \forall i \in \{0.5, 1.0, 2.0, 4.0\}. \tag{1}$$

Interconnectivity between four encoders is achieved via a shared CLS token, a learnable parameter, facilitated by CLS Fusion modules. These vectors are then fed into a fully connected layer, producing an output z, which is subsequently concatenated with the shared CLS token x_{cls} to

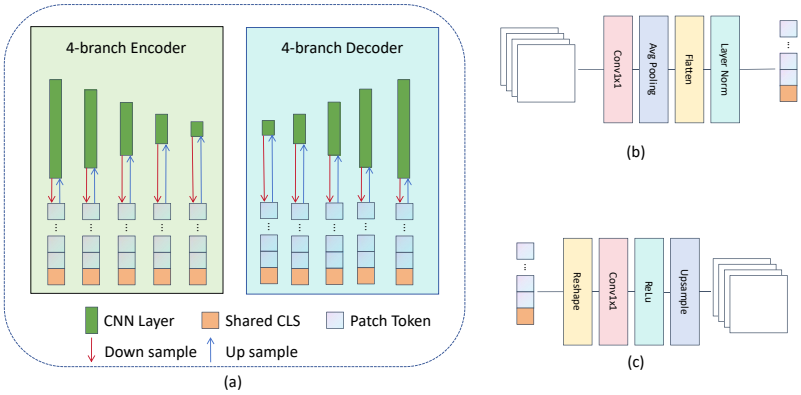

Figure 3. (a) The detailed architecture of each encoder/decoder branch. Information exchange between CNN layers and attention layers is achieved by downsampling and upsampling modules. (b) The architecture of the downsampling module. (c) The architecture of the upsampling module.

form the latent code $\hat{z}$. The process can be formulated as follows:

$$z = [z_{0.5} \mid z_{1.0} \mid z_{2.0} \mid z_{4.0}]W_1^T + b_1, \tag{2}$$

$$\hat{z} = [z \mid x_{cls}]W_2^T + b_2, \tag{3}$$

where W_1, b_1, W_2 and b_2, are parameters of the fully connected layers. Finally, the latent code $\hat{z}$ is decoded by the four-branch decoder D_i to reconstruct the spectrograms $\hat{I}_i$ in four time durations.

$$\hat{I}_i = D_i(\hat{z}) \;\; \forall i \in \{0.5, 1.0, 2.0, 4.0\}. \tag{4}$$

The reconstructed spectrograms $\hat{I}_i$, along with the original inputs I_i are used to compute the reconstruction loss, which is defined as follows:

$$L = \sum_{i=0.5,1.0,2.0,4.0} L_{mse}(I_i, \hat{I}_i) \tag{5}$$

where L_{mse} represents the mean square error loss. The decoders are discarded during inference and glitches are clustered using the low-dimensional latent code $\hat{z}$.

3.2. CNN-ViT Encoder/Decoder

For unsupervised clustering of glitches, integrating both global and local features is crucial for effectively distinguishing between different types of glitches. To this end, we employ an encoder and decoder architecture based on CNN and ViT to construct our multi-branch AE. Specifically, our encoders and decoders utilize CNN-ViT blocks, each comprised of two ResNet bottleneck layers, a self-attention module, along with downsampling and upsampling modules, as depicted in Fig. 3. The CNN component adheres to a standard encoder-decoder configuration. Throughout the encoding phase, the resolution of feature maps is progressively halved, while their channel dimension is expanded at each subsequent layer. Conversely, the decoding phase mirrors the encoding process, employing transposed convolution to upsample the feature maps, thus restoring their original image size. The encoded local information from the CNN module is fed to the attention module via a downsampling process, as illustrated in Fig. 3 (b). A 1×1 convolution is applied first to align the CNN features with the attention tokens. This step transforms the feature map dimensions from shape $(N \times C \times H \times W)$ to $(N \times K \times H \times W)$, where N, H, W, C, K are the batch size, image height, image width, number of channels in the CNN features, and the embedding size of the self-attention module, respectively. The aligned feature map is then downsampled by an average pooling, followed by a vector flatten operation and a normalization layer. Patch tokens should maintain the same receptive field as the network forward, hence, the stride of downsampling is decreased by half as the resolution of feature maps halves. The downsampled output, denoted as x_d, is then combined with the patch token x_t from the self-attention module, following:

$$\hat{x_t} = x_d + x_t, \tag{6}$$

$$y_t = \text{attn}(\hat{x_t}). \tag{7}$$

where attn represents the self-attention layer. The output y_t is upsampled (Fig. 3 (c)) and then fed back to the CNN module, which can provide global contextual information, enriching the local feature map x_c through addition:

$$\hat{x_c} = y_t + x_c, \tag{8}$$

$$y_c = \text{fuse}(\hat{x_c}). \tag{9}$$

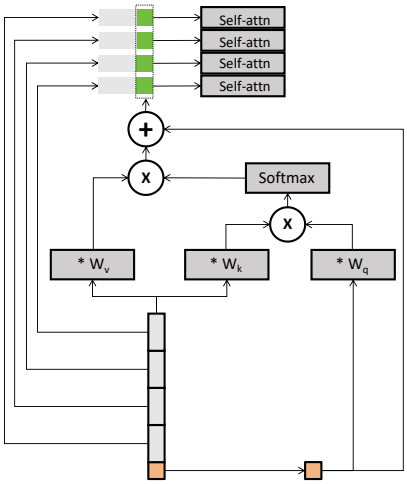

Figure 4. Our CLS fusion module. The shared CLS token queries all patch tokens to gather global information from all four branches. It is then concatenated with each branch to provide abstract information.

where fuse is a ResNet bottleneck layer fusing the local feature map x_c and the globally upsampled output y_t.

3.3. CLS Fusion Module

To effectively capture features across multiple time-window durations, it's crucial to facilitate information exchange between the branches of our model. A straightforward way is to concatenate the patch tokens from all four branches and apply self-attention across the aggregated tokens. However, this approach is impractical for our purpose because glitch spectrograms from different durations do not exhibit spatial alignment, rendering most cross-branch patch pairs irrelevant. Incorporating these pairs not only significantly increases computational demands but also introduces noise into the model. To address these potential issues, we introduce an innovative fusion technique centered around the use of a shared CLS token, which acts as a mediator for inter-branch communication, as depicted in Fig. 4. Specifically, this process begins with the CLS token x_{cls} functioning as the query in interactions with patch tokens x_i from each branch:

$$Q = x_{cls} W_q^T + b_q, \tag{10}$$

$$K = [x_{cls} \mid x_{0.5} \mid x_{1.0} \mid x_{2.0} \mid x_{4.0}] W_k^T + b_k, \tag{11}$$

$$V = [x_{cls} \mid x_{0.5} \mid x_{1.0} \mid x_{2.0} \mid x_{4.0}] W_v^T + b_v, \tag{12}$$

$$\hat{x_t} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{13}$$

$$x_t = x_t + \hat{x_t}. \tag{14}$$

where d_k denotes the embedding dimension for the query token Q. The updated CLS token is concatenated with the patch tokens from each branch for the subsequent standard self-attention processing. The use of shared attention weights $W_q, b_q, w_k, b_k, w_v, b_v$ is applied to all branches as well as the CLS fusion process to ensure consistent parameters in mediating interactions. Notably, employing a shared CLS token enables branches to extract more discriminative features of glitches by considering global context without being overwhelmed by irrelevant local patch details. Furthermore, the use of common attention weights fosters a form of soft interconnection among branches, enhancing the model's ability to integrate and differentiate multi-duration features effectively.

4. Experiments

4.1. Setup

Dataset Our study focuses on the GravitySpy O3 dataset on the main channel [29]. We sample 41745 glitches categorized into 23 different classes from the full dataset, collecting data from both Hanford and Livingston detectors. The detailed distribution of each class is shown in Tab. 1. As you can see, this dataset has a complicated distribution with higher intra-class and inter-class variance, which poses a greater challenge for the clustering task. For the purpose of training and evaluation, we pick 70% of the data for training and 10% for validation and the remaining 20% of data for testing. Notably, the class labels here are not used for model training but only for evaluating the clustering performance.
Evaluation Metrics To assess the performance of our unsupervised clustering algorithm, we adopt two standard metrics: normalized mutual information (NMI) and adjustable rand index (ARI), as described in [6]. NMI is defined as:

$$NMI(Z; \hat{Z}) = \frac{I(Z; \hat{Z})}{\sqrt{H(Z) \times \hat{Z} \times}} \tag{15}$$

where Z and $\hat{Z}$ denote the ground-truth cluster and the predicted cluster, and I and H represent mutual information and entropy, respectively. NMI measures the mutual dependence between the predicted and the ground truth distributions, with its values normalized within the range $[0, 1]$. A higher NMI value indicates a stronger correlation between these distributions. Additionally, we use ARI to quantify the similarity between the ground truth and predicted clusters. ARI adjusts the Rand index RI, a metric that calculates the proportion of sample pairs correctly grouped

Class	train	val	test
1080Lines	845	121	242
1400Ripples	1235	177	353
Air_Compressor	1258	180	360
Blip	1674	239	479
Blip_Low_Frequency	1939	277	555
Chirp	51	7	15
Extremely_Loud	2074	297	593
Fast_Scattering	1990	285	569
Helix	94	14	27
Koi_Fish	2258	323	646
Light_Modulation	333	48	96
Low_Frequency_Burst	1774	254	507
Low_Frequency_Lines	1750	250	500
No_Glitch	1996	285	571
Paired_Doves	1148	164	329
Power_Line	497	71	143
Repeating_Blips	562	81	161
Scattered_Light	2729	390	780
Scratchy	400	57	115
Tomte	2015	288	576
Violin_Mode	442	63	127
Wandering_Line	66	10	19
Whistle	2016	288	577
Total	29146	4169	8340

Table 1. Data sample distribution across different classes in the training, validation, and testing sets of our dataset

together or apart in the same or different clusters, respectively:

$$RI = \frac{C_1 + C_2}{\binom{n}{2}} \tag{16}$$

where n is the total number of samples, C_1 is the count of sample pairs correctly placed in the same cluster according to both the ground truth and prediction, and C_2 is the count of sample pairs correctly separated into different clusters by both. ARI is then adjusted and defined as:

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)} \tag{17}$$

where $E(RI), max(RI)$ are the expectation and maximum of RI, respectively, and $-1 \leq ARI \leq 1$. A higher ARI score indicates greater concordance between the clustering assignments.

4.2. Implementation Detail

Before being fed into the model, all spectrograms undergo a standard preprocessing where they are normalized to the range of [-1, 1] and resized from dimensions of 480x575 to 224x224 pixels. Our CTSAE architecture incorporates 13 CNN-ViT blocks to construct both encoders and decoders components. The encoder is organized into four parts. First, a single CNN-ViT block converts the input spectrogram into feature maps and patch tokens. The rest three parts each contain three blocks, where the second and third parts reduce the feature maps from 224x224 to 112x112, and then further down to 56x56, respectively. The last part downsamples the feature maps twice, resulting in an output in 14x14. These feature maps are finally pooled to 4x4, which are then flattened to form the latent codes z_i, as mentioned in Sec. 3. The decoder mirrors the encoder's structure but inverts the downsampling process with upsampling transposed convolutions, ensuring a symmetrical architecture. In the encoding phase, an embedding size of 384 is used for the self-attention components. Following [17], the decoder is designed to be more compact, with a reduced embedding size of 192 to optimize computational efficiency. Our model's training was executed on 8 Tesla V100 GPUs, requiring approximately 4 days to complete 200 epochs. All experiments are conducted on our validation and testing dataset. The selection of the optimal model was based on its performance on the validation set. K-Means algorithm is utilized to cluster the extracted latent codes and the random seed is fixed for a fair comparison.

Method	ARI	NMI
DIRECT [6]	0.2137	0.4281
VAT-1% [7]	0.4054	0.5963
VAT-5% [7]	**0.5938**	**0.7011**
CTSAE	**0.4091**	**0.6362**

Table 2. Results comparison between our CTSAE approach and DIRECT, VAT with 1% of labeled data, and VAT with 5% of labeled data.

4.3. Comparison

As we are the first in applying unsupervised learning to the task of glitches clustering, we benchmark our proposed method against existing semi-supervised clustering methods [6, 7]. Results are shown in Tab. 2. We first compare our method with DIRECT [6] which is currently deployed to the official GravitySpy pipeline to find new glitch classes. We utilize its most recent version of checkpoints for a direct comparison on our same test set. Our approach outperforms DIRECT by a large margin even in the absence of supervision, suggesting that our unsupervised approach has the potential to substantially enhance the LIGO glitch identification process upon implementation. Furthermore, we compare our method with the state-of-the-art clustering algorithm, virtual adversarial training (VAT), on the GravitySpy data [7]. We reproduce the VAT model [7] and experiment

Model	Branches #	Recon-MSE	ARI	NMI
CNN	1	0.0782	0.2792	0.5464
ViT	1	0.0175	0.1243	0.3139
CNN-ViT	1	**0.0167**	**0.3118**	**0.5647**
No Fusion	4	0.0325	0.3186	0.5691
All-attention	4	0.0141	0.3518	0.6170
CLS Fusion(CTSAE)	4	**0.0137**	**0.4091**	**0.6362**

Table 3. Results of the ablation study conducted on both single-branch and multi-branch models. We compare CNN-only, ViT-only, and CNN-ViT AEs within a single branch. For multi-branch AEs, we compare different fusing strategies, including no fusion, All-attention and CLS Fusion.

with VAT under various ratios of labeled to unlabeled data. We observe that while our model does not exceed the performance of VAT trained with 5% labeled and 95% unlabeled data, it presents considerable improvements over VAT configured with 1% labeled data. Notably, VAT still needs labeled dataset while our method is fully unsupervised. Given the upcoming tasks on the GravitySpy O4 dataset on the main and auxiliary channels, i.e., to find correlations between main channel glitches and auxiliary channel glitches, which will contain significantly less than 1% labeled data, we anticipate our model to set new benchmarks in clustering glitches sourced from the GravitySpy O4 dataset on the main and auxiliary channels.

4.4. Ablation Study

The Multi-branch Architecture We begin with examining the necessity of incorporating glitch spectrograms acorss all four durations in our analysis. This is assessed by comparing the performance of our multi-branch architecture, which utilizes spectrograms from all four durations, against a single-branch architecture that processes only 4.0 s duration spectrograms. To make a fair comparison, the architecture of the single-branch model mirrors that of each individual branch within our proposed multi-branch framework. As is shown in Tab. 3, the multi-branch model demonstrates superior performance relative to its single-branch counterpart. This discrepancy in performance can be attributed to the fact that glitches, even when belonging to distinct classes, may exhibit similar patterns within a singular time duration. Such similarities obscure clear class distinctions, which can be effectively resolved only by aggregating and analyzing data across all four time durations. Therefore, extracting features from all four time-window durations achieves a more comprehensive understanding and accurate clustering of glitches.

CNN-ViT We further assess the effectiveness of the CNN-ViT hybrid block in comparison to the CNN-only block and the ViT-only block. To avoid the impact of different multi-branch fusion strategies, this evaluation is conducted using single-branch autoencoder models. The CNN-only and ViT-only configurations are derived by omitting the complementary component from the CNN-ViT block, resulting in three distinct single-branch autoencoders, each built upon the same architectural framework but differing in their foundational blocks. All AEs are trained and tested on the data with only 4.0 s spectrograms. For the models based on CNN and CNN-ViT, the encoder outputs are encoded into low-dimensional latent vectors. Conversely, in the ViT-based model, the encoder's embedding tokens, inclusive of the CLS token, proceed through several fully connected layers before being fed into the decoder, with the CLS token being specifically leveraged for clustering in the test phase as described in [17].

As shown in Tab. 3, the ViT-only model exhibits the lowest performance, a phenomenon potentially linked to inadequate constraints applied to the CLS token. In the context of the Masked AE [17], the CLS token will further undergo additional fine-tuning for clustering tasks, forcing the CLS token to encode critical global information. However, in unsupervised settings, there lacks a direct mechanism to ensure the CLS token aggregately represents global features, rendering a solely ViT-based autoencoder less effective for unsupervised learning tasks due to its inherent characteristics. Moreover, the CNN-ViT hybrid model outperforms the CNN-based model in both NMI and ARI metrics, indicating that the inclusion of a ViT branch facilitates superior global information capture and, consequently, enhances overall performance. An additional benefit of integrating attention mechanisms within this setup is the facilitation of efficient information fusion across branches, achieved through our proposed CLS Fusion module.

CLS Fusion Module Finally, we investigate various multi-branch fusion schemes, including methods with No Fusion, the All-attention as outline in Sec.3.3, and our proposed CLS Fusion module. As shown in Tab. 3, both fusion strategies achieve superior ARI and NMI scores compared to the model without any fusion. This again affirms the critical role of cross-branch interaction for enhanced clustering outcomes. Compared with the All-attention approach, our CLS Fusion module obtains a better performance while reduc-

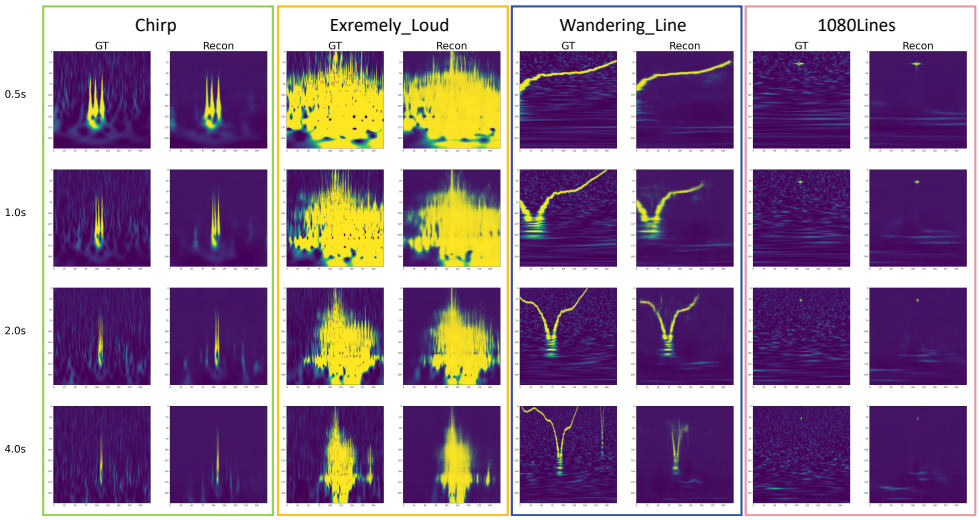

Figure 5. Reconstruction results on test data. Each column represents the spectrograms of the same glitch across four time windows: 0.5 s, 1.0 s, 2.0 s, and 4.0 s, from top to bottom. From left to right, the columns represent input glitches and their corresponding reconstructed glitches. Four samples are selected from the classes Chirp, Extremely Loud, Wandering Line, and 1080Line, respectively.

ing the computation complexity. This superiority can be attributed to two primary factors. Firstly, our CLS Fusion module forces the CLS token to communicate with tokens from all branches before per-branch self-attention. As discussed in the **CNN-ViT** context, imposing such a communicative constraint is advantageous for clustering activities. Besides, while the All-attention approach considers the correlation between each pair of tokens across all branches, it tends to generate an excess of redundant correlations. For instance, the majority of patches within 4.0 s spectrograms lie beyond the coverage of those in 0.5 s spectrograms, potentially introducing irrelevant noise to the refined features discerned from shorter duration spectrograms. Our CLS Fusion strategy mitigates this issue by channeling all cross-branch communications through the CLS token, thereby restricting interactions to within individual branches and effectively eliminating redundancy.

4.5. Reconstruction Results

We also present the reconstruction results of our CTSAE model. As shown in Tab. 3, our CTSAE also achieves the lowest reconstruction error among the compared models, suggesting its superior capability in capturing and accurately encoding the spectral information of glitches. Fig. 5 illustrates that the reconstructed spectrograms effectively retain the integral structure of the glitches, significantly reducing background noise in the process. This fidelity in reconstruction not only demonstrates the effectiveness of CTSAE in encoding relevant information but also opens avenues for future research. We aim to explore the potential of CTSAE to generate glitch spectrograms that preserve their physical semantics without compromise, effectively solving the issue of imbalanced classes shown in Tab. 1.

5. Conclusion

In this paper, we propose CTSAE, the first unsupervised model designed for dimensionality reduction and clustering of gravitational wave glitches. By integrating CNNs with ViTs, we develop a multi-branch autoencoder, enhanced with a novel inter-branch CLS fusion module. This model has been trained on the Gravity Spy 1.0 dataset. During inference, multi-duration glitch spectrograms are encoded into a low-dimensional latent space via encoders, facilitating the clustering of glitches within the test set. Experiments show that our CTSAE model outperforms the existing state-of-the-art semi-supervised methods even without reliance on any labeled data. In the future, we aim to extend our research to the Gravity Spy 2.0 dataset, which promises richer cosmic signal data from both main and auxiliary channels. Given the ongoing development and current lack of manual labeling in Gravity Spy 2.0, we plan to deploy our CTSAE model to this newer dataset and investigate enhancements, particularly incorporating conditions related to characteristics of LIGO instruments and sensors.

Acknowledgement

The authors wish to express their sincere gratitude to the community-science volunteers of the Gravity Spy project. We also extend our thanks to ManLeong Chan for his insightful comments that significantly enhanced the quality of this manuscript. Furthermore, this work was supported by the NSF's LIGO Laboratory, a major facility fully funded by the National Science Foundation.

References

[1] J Aasi, B P Abbott, R Abbott, T Abbott, M R Abernathy, K Ackley, C Adams, and et al. Adams. Advanced ligo. *Classical and Quantum Gravity*, 32(7):074001, 2015. 1

[2] Benjamin P Abbott, R Abbott, TD Abbott, MR Abernathy, F Acernese, K Ackley, M Adamo, C Adams, T Adams, P Addesso, et al. Characterization of transient noise in advanced ligo relevant to gravitational wave signal gw150914. *Classical and Quantum Gravity*, 33(13):134001, 2016. 1, 2

[3] Benjamin P Abbott, Richard Abbott, TDe Abbott, MR Abernathy, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Observation of gravitational waves from a binary black hole merger. *Physical review letters*, 116(6):061102, 2016. 1

[4] Benjamin P Abbott, Rich Abbott, Thomas D Abbott, Sheelu Abraham, Fausto Acernese, Kendall Ackley, Carl Adams, Vaishali B Adya, Christoph Affeldt, Michalis Agathos, et al. A guide to ligo–virgo detector noise and extraction of transient gravitational-wave signals. *Classical and Quantum Gravity*, 37(5):055002, 2020. 1

[5] Sofia Alvarez-Lopez, Annudesh Liyanage, Julian Ding, Raymond Ng, and Jess McIver. Gspynettree: A signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, 2023. 2

[6] Sara Bahaadini, Neda Rohani, Aggelos K Katsaggelos, Vahid Noroozi, Scott Coughlin, and Michael Zevin. Direct: Deep discriminative embedding for clustering of ligo data. In *2018 25th ieee international conference on image processing (icip)*, pages 748–752. IEEE, 2018. 2, 5, 6

[7] Sara Bahaadini, Yunan Wu, Scott Coughlin, Michael Zevin, and Aggelos K Katsaggelos. Discriminative dimensionality reduction using deep neural networks for clustering of ligo data. *arXiv preprint arXiv:2205.13672*, 2022. 2, 6

[8] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 2

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[10] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5270–5279, 2022. 2, 3

[11] S Coughlin, S Bahaadini, N Rohani, M Zevin, O Patane, M Harandi, C Jackson, V Noroozi, S Allen, J Areeda, et al. Classifying the unknown: discovering novel gravitational-wave detector glitches using similarity learning. *Physical Review D*, 99(8):082002, 2019. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[13] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021. 2, 3

[14] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: A vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12259–12269, 2021. 2, 3

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pages 1735–1742. IEEE, 2006. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 6, 7

[18] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[19] Seraphim Jarov, Sarah Thiele, Siddharth Soni, Julian Ding, Jess McIver, Raymond Ng, Rikako Hatoya, and Derek Davis. A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*, 2023. 2

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[21] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[22] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 367–376, 2021. 3

[23] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985. 2

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[25] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16519–16529, 2021. 2, 3

[26] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[27] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 2

[28] Yunan Wu, Michael Zevin, Christopher PL Berry, Kevin Crowston, Carsten Østerlund, Zoheyr Doctor, Sharan Banagiri, Corey B Jackson, Vicky Kalogera, and Aggelos K Katsaggelos. Advancing glitch classification in gravity spy: Multi-view fusion with attention-based machine learning for advanced ligo's fourth observing run. *arXiv preprint arXiv:2401.12913*, 2024. 2

[29] Michael Zevin, Scott Coughlin, Eve Chase, Sara Allen, Sara Bahaadini, Christopher Berry, Kevin Crowston, Mabi Harandi, Corey Jackson, Vicky Kalogera, et al. Gravity spy volunteer classifications of LIGO glitches from observing runs O1, O2, O3a, and O3b. 2022. 5

[30] Michael Zevin, Corey B Jackson, Zoheyr Doctor, Yunan Wu, Carsten Østerlund, L Clifton Johnson, Christopher PL Berry, Kevin Crowston, Scott B Coughlin, Vicky Kalogera, et al. Gravity spy: lessons learned and a path forward. *The European Physical Journal Plus*, 139(1):100, 2024. 2

[31] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016. 2