Multilinear Factorized Representations for LIGO Glitches in Label-scarce Settings

Rutuja Gurav
University of California Riverside
rgura001@ucr.edu

Barry C. Barish
University of California Riverside
barry.barish@ucr.edu

Evangelos E. Papalexakis
University of California Riverside
epapalex@cs.ucr.edu

ABSTRACT

The discovery of gravitational waves by the Advanced Laser Interferometer Gravitational-wave Observatory (LIGO) has ushered in a new era in astrophysics. Successful detection of gravitational wave signals requires ground-based interferometers like LIGO and Virgo to be exquisitely isolated from environmental and instrumental noise. Albeit the highly sophisticated design of the detectors which carefully mitigates effects of most types of noise, they are still susceptible to non-Gaussian noise transients called *glitches*. As they can mask or mimic real gravitational wave signals and given their high rate of occurrence, proper characterization and classification of *glitches* is necessary.

State-of-the-art machine learning approaches for glitch classification involves training deep neural networks which notoriously require a large number of human annotations, currently provided by the Gravity Spy project [19]. As the operational sensitivity of LIGO increases across several detectors, the number of glitch occurrences are bound to increase. This increase can render human annotation of all potential glitches infeasible, resulting in inevitable label scarcity.

In this work, we set out to explore the problem of characterizing and classifying glitches in a label-scarce setting. First, we propose a tensor-based unsupervised representation, leveraging techniques from multilinear algebra, which discovers meaningful structure that correlates well with human annotations, while uncovering subtle intraclass variation. This result serves as a *proof of concept* for conducting glitch exploration in the future, where a vast number of glitches are expected to be unlabeled. Subsequently, we use our tensor-based representation in a thought experiment to measure its effectiveness in enhancing the performance of state-of-the-art deep transfer learning models, when the number of labels is severely decreased.

KEYWORDS

LIGO glitches, label scarcity, deep transfer learning, tensor decomposition

ACM Reference Format:
Rutuja Gurav, Barry C. Barish, and Evangelos E. Papalexakis. 2019. Multilinear Factorized Representations for LIGO Glitches in Label-scarce Settings. In *Proceedings of* 25^{th} *ACM SIGKDD Workshop: FEED 2019 (FEED Workshop, KDD).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.475/123_4

FEED Workshop, KDD, 2019, Anchorage, Alaska USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION

Einstein's general theory of relativity in 1916 predicted the existence of gravitational waves as perturbations in the fabric of spacetime caused by interactions between massive accelerating objects in the universe. The first gravitational wave signal, `GW150914`, detected in 2015 by LIGO stretched spatial lengths by a distance smaller than the diameter of a proton. Physically measuring such an incredibly small change in length using ground-based detectors requires huge and highly sensitive interferometers like LIGO.

Although the highly sophisticated design of LIGO detectors isolate the sensitive instruments from most types of non-astrophysical noise, the detectors are susceptible to non-Gaussian noise transients called *glitches*. The sources of glitches are environmental and instrumental in nature. Glitches have a high rate of occurrence, complex morphologies and wide variation in their duration and frequency range. Classifying glitches into morphologically distinct classes helps to characterize them and understand their origin such that some sources of noise can be fixed in upgrades to LIGO and others can be further understood [11]. Given their diversity, machine learning approaches have been used to automatically classify glitches.

Gravity Spy[19] is a citizen science project which aims to classify glitches into known classes using crowd-sourced human labeling of glitches which are represented on their web interface as time-frequency spectrogram images. They now maintain a comprehensive labeled dataset of numerous glitches that occurred in the first and second observing runs of LIGO. In previous works, S.Bahaadini et al [1] [2] discuss several popular machine learning classifiers trained using the Gravity Spy dataset and a deep transfer learning method is discussed by D. George et al. [6] which achieves state-of-the-art results.

Machine learning models, especially deep learning models, notoriously require a large amount of labeled training data because the number of parameters that need to be tuned by the learning algorithm can range up to several million. Given the small size of the Gravity Spy dataset ($\sim$ 8500 labeled glitches) in order to avoid overfitting, S. Bhaadini et al. [1][2] used relatively shallow neural networks which are trained from scratch and D. George et al. [6] use state-of-the-art deep neural networks by leveraging transfer learning to repurpose a pretrained deep neural network by fine-tuning it for the glitch classification task. As the LIGO experiment expands across the globe with the addition of new detectors and especially the increase in the operational sensitivity of the detectors, we expect to see a rise in occurrences of glitches. This necessitates us to perform glitch classification at a larger scale. To that end, in order to employ the state-of-the-art supervised learning models, the human labeling process is required to keep pace with the increasing number of glitches.

In this work, we set out to investigate the problem of glitch characterization and classification in a *label-scarce setting*. Our investigation first examines the extreme case where no human annotations are available, and subsequently analyzes the consequences of severely limiting the number of labels on the *state-of-the-art* models. We then explore how our proposed method can be used in conjunction with the state-of-the-art in order to compensate for the lack of adequate amounts of labels. In particular, our contributions are two-fold:

- **Unsupervised tensor-based glitch exploration**: We propose an unsupervised tensor-based representation of glitch data, leveraging methods from multilinear algebra and tensor analysis, that identifies distinct categories of glitches that correlate well with human annotations of the Gravity Spy project. Furthermore, our representation is able to identify subtle variation within classes defined by Gravity Spy. This observation serves as a proof of concept for the utility of our proposed representation in vastly unlabeled LIGO data.
- **Enhancing the state-of-the-art in label-scarce settings**: In order to measure the effectiveness of our unsupervised tensor decomposition in capturing useful structure in LIGO glitches, we propose a thought experiment in the form of an ensemble model, where we integrate our proposed tensor-based representation and an existing state-of-the-art deep transfer learning model, and evaluate the performance in label scarce settings.

In the remainder of the paper, we provide some background, enumerate the challenges and problems faced, and describe our solutions to those problems, which yield promising preliminary results.

2 BACKGROUND & PROBLEM STATEMENT

2.1 Impact of glitches on LIGO detections

Glitches adversely affect the searches for transient gravitational waves in LIGO signals. Glitches can mask or mimic a transient gravitational wave. For instance, `GW170817`, a gravitational wave generated by a collision event of a neutron star binary occurred during a glitch in the Livingston detector [14]. Luckily, the glitch signal was short and the gravitational wave signal was long which led to easy removal of the glitch from the signal. However, with upgrades to the detectors, the occurrence of glitches is bound to increase due to the increased sensitivity of the detector rendering the glitch problem an important one to address.

2.2 Gravity Spy Dataset

The Gravity Spy project[19] has made public a human-labelled dataset of ~ 8500 glitches from the first (O1) and second (O2) observing runs of LIGO. Some of the glitch classes were created by the experts in LIGO Scientific Collaboration (LSC), and more classes were identified by the human annotators based on visual inspection, and were subsequently added to the dataset upon review from LSC. The publicly available Gravity Spy dataset contains 22 classes as shown in Figure 1 viz. 20 morphologically distinct classes, 1 *No-Glitch* class and 1 *None-of-the-above* class which is a catch-all class for glitches which human classifiers could not classify into any of the 20 classes. Each glitch is visualized using 4 different time windows viz. 0.5, 1.0, 2.0 and 4.0 seconds. Details regarding the criteria for obtaining the glitches and creating this dataset can be found in [1] and [19] respectively.

In this work, we use a publicly available version of Gravity Spy dataset[1] which was kindly made available by S. Bahaadini et al consisting of human-labeled glitches which occurred during the O1 and O2 runs of LIGO.

As noted in [1], the 22 classes constituting this dataset are the best efforts of the curators to create a representative sample of different types of glitches that occurred during O1 and O2 runs. However, our work is a *proof of concept* glitch exploration and classification of any glitch dataset - labeled, unlabeled or partially labeled - and we use the Gravity Spy dataset as our benchmark.

2.3 Problem Statement

In this paper, we investigate the effects of label scarcity in analyzing LIGO glitches. The challenge is twofold:

(1) Different glitches occur with different frequencies, resulting in significant imbalances in the representation of every class of glitch in the data. Classes such as *Blip*, *Light Modulation* and *Low Frequency Burst* are well-represented with hundreds of examples available for each class whereas classes like *Paired Doves*, *Air Compressor* and *Chirp* have 27, 58 and 66 examples respectively. This class imbalance challenge is present even without label scarcity.

(2) If we assume label scarcity, the above challenge is further amplified, since the absolute number of available data points per glitch class, especially for some less frequent glitches, becomes extremely low (in the order of tens of examples). Such a low number of training examples is expected to negatively affect the performance of current state-of-the-art models which leverage deep learning, which is notorious for requiring large amounts of labeled data points per class.

In this paper, we explore solutions to the above challenges by solving the following two problems:

Problem 1: *Given a collection of unlabeled glitches, identify in an unsupervised manner inherent hidden structure in the data, which correlates with human annotations.*

Addressing Problem 1 can provide insights on how to tackle the following problem.

Problem 2: *Given a collection of glitches, where only a small percentage of them is labeled, train a classification model that works on par with state-of-the-art models which operates on a fully labeled dataset.*

Section 3 discusses our suggested approach to Problem 1, and subsequently, Section 4 discusses how lessons learned from addressing Problem 1 can be applied to solving Problem 2.

3 UNSUPERVISED TENSOR-BASED GLITCH EXPLORATION

An extreme instance of label scarcity is a complete lack of labels. In such instances, unsupervised methods are used to discover the

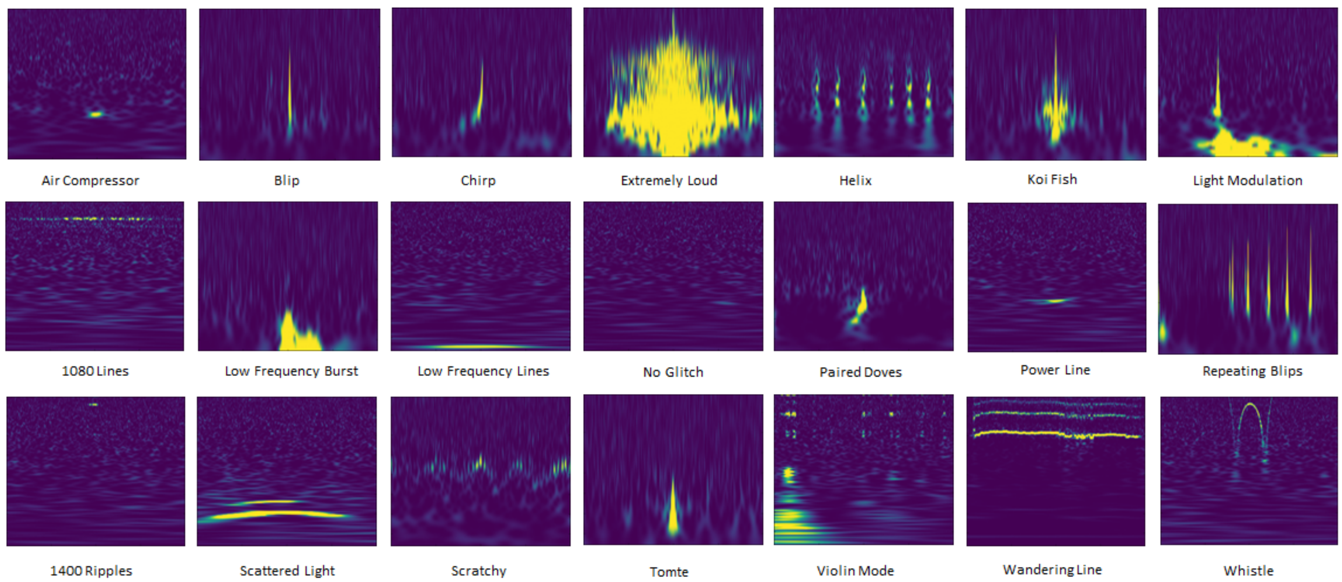

Figure 1: 2.0 second view of various types of glitches in the Gravity Spy dataset. None-of-the-Above glitch type excluded.

underlying structure in the data. We explore the glitch dataset by expressing it in from of a tensor created by stacking glitch images. Tensors or multi-way arrays are multi-dimensional extensions of matrices [15], and have been shown, time and time again to be very effective in modeling highly heterogeneous and multidimensional data. A tensor indexed by n variables is termed an n-mode tensor (e.g., a matrix is simply a 2-mode tensor). Tensor decompositions are data analysis tools that leverage multilinear algebra for extracting interpretable latent patterns from the data [9]. Thus, our glitch dataset is a good candidate for tensor analysis. Tensor-based clustering methods have been shown to be effective for high-dimensional data like images[18][4]. In this work, we use CP/PARAFAC tensor decomposition to find latent space embeddings for the glitches.

3.1 Tensor factorization for glitches

We obtain unsupervised representations of glitches by performing CP/PARAFAC decomposition on the glitch dataset represented as a 4-mode tensor $\mathcal{X}^{N,l_1,l_2,channels}$ where N is the number of glitches, l_1, l_2 denote the dimensions of the glitch spectrogram and *channels* denotes the number of channels used to encode the image. CP/PARAFAC decomposition of $\mathcal{X}$ is expressed as a sum of outer products of 4 factor matrices viz. A, B, C and D each with r components or columns.

$$\mathcal{X} \approx \sum_{r=1}^{R} A(:,r) \circ B(:,r) \circ C(:,r) \circ D(:,r)$$

For the computation of the CP/PARAFAC decomposition we used the implementation of the Alternating Least Squares method [15] in Tensorly [10]. Below we outline how we analyze the factorization results.

3.2 Analysis of factorization results

We obtained the factor matrix $A \in \mathbb{R}^{N \times R}$ for various values of the rank using 2 second view of the glitches. Each row of A is a R-length embedding of a glitch in latent space where R is the rank used for the tensor decomposition.

Table 1: Coverage analysis of factor matrices obtained using various decomposition ranks. For each rank, we show the percentage of ground truth classes present in the top-20 across all components.

rank	% classes covered in top-20
20	38.09
21	33.33
22	42.85
23	38.09

Computing rank of a tensor is an NP-hard problem [8]. Appropriate rank selection in case of tensors is an open problem and several heuristics have been proposed to determine the appropriate rank in [3] for dense tensors and an extension for sparse tensors is proposed in [13] and [12]. In this work, we have simply selected a range of ranks based on the number of ground-truth classes present in the Gravity Spy dataset.

To examine the underlying structure discovered by the tensor decomposition, we performed a homogeneity analysis on the factor matrix A. We take the top k values per column of the factor matrix A and examine the labels of the corresponding glitches. Figure 2 shows percentage of different classes appearing in top-20 in each column of factor matrix A of a rank 24 decomposition.

We also examine the extent to which the various ground-truth classes are covered across the various components of the decomposition using coverage analysis. We find the dominant label in the top-k values in each component of the factor matrix and determine how many of the ground-truth classes are present across all components combined. Table 1 shows the results of this analysis.

Factor Matrix Components

Glitch Classes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Air Compressor	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
Blip	0	0	50	10	0	5	0	0	0	5	0	0	5	30	0	10	10	5	30	25	30	0	0	0
Chirp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	10	0	0	0	0
Extremely Loud	25	30	5	0	5	5	5	5	95	10	30	80	15	0	0	45	5	15	10	0	25	95	15	10
Helix	0	10	0	0	5	0	0	0	0	0	0	0	5	0	0	0	0	0	0	15	0	0	0	0
Koi Fish	5	10	10	50	0	20	5	5	0	40	0	0	10	0	20	0	15	20	10	10	0	0	65	65
Light Modulation	20	0	0	5	40	15	0	10	5	0	0	0	5	0	10	0	10	10	15	0	0	5	5	5
1080 Lines	0	0	0	5	0	0	0	15	0	0	0	5	0	15	0	0	5	0	0	0	5	0	0	0
Low Frequency Burst	0	10	0	0	30	10	5	5	0	10	5	5	20	0	10	5	5	5	10	15	10	0	5	5
Low Frequency Lines	40	5	5	5	20	5	0	5	0	5	15	0	10	0	10	10	10	5	10	0	0	0	0	0
No Glitch	0	10	0	0	0	0	5	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
Paired Doves	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Power Line	0	0	5	0	0	5	25	35	0	20	0	0	0	25	10	0	15	20	0	0	0	0	0	0
Repeating Blips	5	5	0	0	0	5	5	0	0	0	5	0	0	5	0	0	0	5	0	15	5	0	5	5
1400 Ripple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Scattered Light	5	0	0	0	0	5	0	0	0	5	15	0	0	0	5	10	15	0	0	5	5	0	5	10
Scratchy	0	10	0	10	0	10	10	10	0	5	20	0	15	5	15	10	0	10	0	0	0	0	0	0
Tomte	0	5	5	10	0	10	5	0	0	0	0	0	0	0	0	0	0	0	5	0	5	0	0	0
Violin Mode	0	0	5	0	0	0	15	0	0	0	0	0	5	10	5	5	5	0	5	0	5	0	0	0
Wandering Line	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	5	0	0	0
Whistle	0	5	10	0	0	5	0	5	0	0	10	0	5	0	10	0	0	5	5	0	0	0	0	0

Figure 2: The figure shows the homogeneity analysis of CP/PARAFAC decomposition of glitch tensor using rank 24. Each column shows the percentage of each type of glitch appearing in the top-20 in each of the 24 components of the factor matrix. It can be observed that *Extremely Loud* glitch dominates the top-20 values of several components of the factor matrix. All types of glitches are suitably captured in at least one component.

Some gravity spy classes like *Extremely Loud* and *Low Frequency Burst* are catch-all classes and hence have a high intraclass variability. According to [1], *Extremely Loud* class contains high energy glitches and loud *Koi Fish* type of glitches are included in this class.

As shown in Figure 3, CP/PARAFAC decomposition of *Extremely Loud* class reveals clusters of similar glitches within the class and clearly isolates the loud, high energy *Koi Fish* type glitches.

This demonstrates the ability of tensor decomposition to glean good structure present in the glitch tensor with a selection of an appropriate rank. Thus, tensor-based methods would be effective for searching meaningful patterns in power excesses in LIGO signal and can potentially be used of exploring unmodeled sources of gravitational waves.

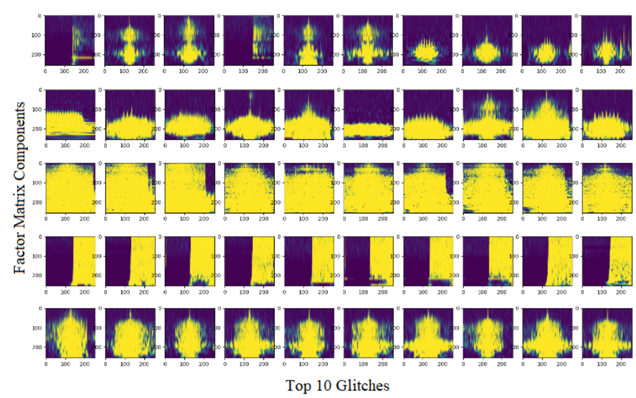

Figure 3: Each row in the figure represents the top-10 glitch instances from each component of the factor matrix obtained from CP/PARAFAC decomposition with rank 5. Observe that row 5 shows the cluster of loud *Koi Fish* type glitches isolated within the *Extremely Loud* class.

4 ENSEMBLE MODEL FOR LABEL-SCARCE GLITCH CLASSIFICATION

This section explains in detail the proposed method for an ensemble model for glitch classification using only a small fraction of the labels available in the dataset. We begin by understanding the method of deep transfer learning for classifying glitch spectrograms and the effect of label scarcity on this method. Thereafter, we propose an ensemble method in which we augment the deep transfer learning representations with tensor embeddings.

4.1 Deep Transfer Learning

Convolutional neural networks (CNNs) are currently the state-of-the-art technique in image classification. CNNs are excellent feature extractors which allow us to learn low dimensional representations from the raw image input. We can learn better and more sophisticated features by stacking layers in a CNN and making it deeper. This is in essence the principle behind deep learning.

Although deep CNNs are powerful tools, they have millions of parameters that need to be learned. Given the sheer number of parameters of the model, we require commensurate amount of labeled data to properly train the network and avoid the pitfall of *overfitting* where the network simply memorizes the features in the training examples instead of learning to generalize.

In many real use-cases, however, we often do not have resources to collect millions images and then label them in order to have ample training data for our deep network. In order to train deep CNNs using a small dataset like Gravity Spy, transfer learning is employed. It is common knowledge in the machine learning community that the initial layers of a deep CNN learn fairly generic features like edges and curves which are universal to all images. Thus, instead of training the network to learn these generic features from scratch, we can borrow a deep network trained on a massive dataset, albeit for a different task than ours, and fine-tune the deeper layers which learn abstract, data-specific features for our glitch classification task.

4.2 Effect of Label scarcity

D. George et al in [6] demonstrate that deep transfer learning achieves *state-of-the-art* results for glitch classification task using Gravity Spy dataset compared to other approaches which train models from scratch. They use 80% of the labeled glitches for fine-tuning various deep CNNs [16][17][7] pre-trained on the ImageNet[5] dataset and validate their trained model on the remaining 20% of the labeled glitches.

To simulate a label-scarce setting, we throttle the number of labeled instances available for training by randomly sampling 10% of the glitches for each class from the 80% split and use this sample for training. We used the remaining 20% glitches as hold-out set for validation across all experiments. It is to be noted that label scarcity further exacerbates the class imbalance problem.

The deep transfer learning method using limited labels is described in Algorithm 1.

Algorithm 1 Representations using Deep Learning

1: **Input:**
  Dataset D: $\{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^{l_1 \times l_2 \times channels}$ is a glitch spectrogram and y_i is the corresponding label.
  l_1, l_2 denote the dimensions of the glitch spectrogram and *channels* denotes the number of channels used to encode the image, typically $channels = 3$ for and RGB image. N is the total number of glitches in the dataset.
  Labeled training set L: Choose a small percentage of glitches and their labels at random from each class.
  Unlabeled set U: Treat rest of the glitches as unlabeled.
2: **Step 1:** Fine-tune pretrained deep network using L.
3: **Step 2:** Use trained network as feature extractor f and represent glitches in $L \cup U$ in latent space.
  $f : \mathbb{R}^{l_1 \times l_2 \times channels} \rightarrow \mathbb{R}^m$
  Obtain feature vectors as FV_L and FV_U for L and U respectively.
4: **Step 3:** Train a discriminator using FV_L and corresponding labels and evaluate on the hold-out set.

For the purposes of this paper we are going to exclude the *None of the Above* class from our analysis as this class does not have a consistent intraclass structure. The classification performance is measured using F1 score for each class which is the harmonic mean of the precision and recall.

We randomly split the Gravity Spy dataset into 80% training and 20% hold-out sets. To demonstrate the impact of using limited labels for training a deep CNN, we first trained the network using the whole 80% training data, and then on random 10% samples for 5 runs. The hold-out set was used of validation.

Table 2 shows the effect of label scarcity on a VGG19 [16] network pretrained on ImageNet [5] in terms of drop in average F1 score over 5 runs.

Table 2: Class-wise F1 scores of VGG19 fine-tuned on 10% and 80% of the labeled glitches respectively show that there is a drop in F1 score for every class as we reduce the amount of available labels during training. The under-represented classes (in bold) are worst affected.

glitch class	vgg19-10%	vgg19-80%	support
Air Compressor	0.790938685	0.827586207	12
Blip	0.946134684	0.981182796	374
Chirp	**0.672668127**	**1**	14
Extremely Loud	0.969669574	1	91
Helix	0.973149947	1	56
Koi Fish	0.955218569	0.990990991	166
Light Modulation	0.912167451	0.973913043	115
1080 Lines	0.908763929	0.96969697	66
Low Frequency Burst	0.906152506	0.957528958	132
Low Frequency Lines	0.920353467	0.941176471	91
No Glitch	0.65114379	0.821917808	37
Paired Doves	**0.426031746**	**0.909090909**	6
Power Line	0.963662563	0.966292135	91
Repeating Blips	0.916514915	0.964285714	57
1400 Ripple	0.847251596	0.929292929	47
Scattered Light	0.951348696	0.978494624	92
Scratchy	0.988632219	0.992907801	71
Tomte	0.733760082	0.867924528	24
Violin Mode	0.948745888	0.989473684	95
Wandering Line	**0.565641026**	**0.888888889**	9
Whistle	0.83488995	0.94017094	61

4.3 Ensemble model using Tensor Embeddings

As described in Algorithm 2, we use the embeddings obtained from the tensor decomposition to augment the feature vectors of each glitch obtained using the deep CNN. These combined CNN and tensor embeddings are used to train a discriminator to classify the glitches into 21 classes. Figure 4 depicts the block diagram of the ensemble setup.

The presented ensemble is a *thought experiment* that is meant to measure the effectiveness of the CP/PARAFAC factors. Note that there is no target/label leakage since the decomposition is unsupervised, albeit leveraging the structure of the entire dataset, as in typical semi-supervised settings. However, in a realistic deployed setting, we would have to project test instances to the already computed decomposition space and use that projection as the representation. To the best of our knowledge, such augmentation of a supervised model with an unsupervised one is an open problem which we reserve for future work.

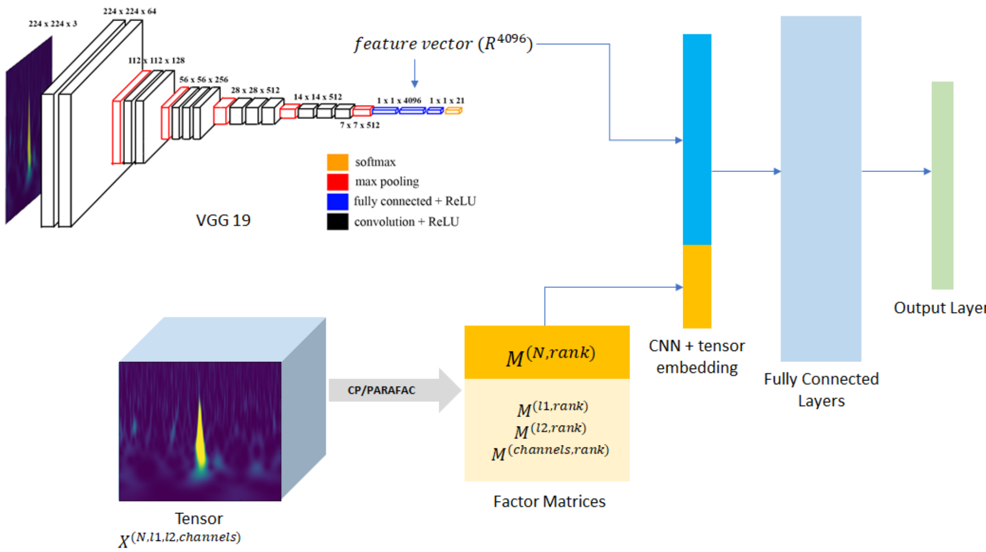

Figure 4: Block Diagram of Ensemble method which combines feature vectors from VGG19 fine-tuned on 10% of the labeled glitches and CP/PARAFAC decomposition of tensor representing all the glitches.

Algorithm 2 Ensemble Representation

1: **Input:**
R_1: Feature vectors for labeled glitches FV_L and feature vectors for unlabeled glitches FV_U as obtained in Algorithm 1.
R_2: Factor Matrix $A \in \mathbb{R}^{N \times R}$ where N is the total number of glitches and R is the decomposition rank used in CP/PARAFAC.

2: **Step 1:** Construct a latent representation of glitches by combining latent representation in R_1 with corresponding representation in R_2 for every glitch.
$Ensemble_L$: FV_L joined with corresponding row of R_2.
$Ensemble_U$: FV_U joined with corresponding row of R_2.

3: **Step 2:** Train a discriminator using $Ensemble_L$ and validated the trained model using $Ensemble_U$

We use a VGG19 trained on a random 10% sample from the training data as a base CNN and augment the feature vectors obtained from this base CNN with tensor embeddings from 5 different decomposition runs in our ensemble setup.

Figure 5 shows the average class-wise F1 scores across 5 runs as a measure of the classification performance of the ensemble model using various ranks for tensor decomposition. We see that augmenting CNN representation with tensor embedding improves the classification performance in following respects.

- Most classes perform better than the baseline VGG19 with appropriate selection of the CP/PARAFAC rank.
- More classes do better with lower rank tensor embeddings because low rank decomposition is more stable. As we go higher, we have low SNR. Improving tolerance of our decomposition can improve performance at higher ranks.
- Classes like *Paired Doves*, *Wandering Lines* and *Chirp* has 2, 3 and 5 examples respectively in the training dataset and are severely under-represented under label scarcity.

5 RELATED WORK

S.Bahaadini et al[2] have used the Gravity Spy dataset to perform glitch classification in multi-view mode. They use two ways to represent glitches in a multi-view mode viz. merged view where they concatenate the 4 images representing the 4 views of a glitch in a grid and use the resultant image for classification, and parallel view where they concatenate feature maps obtained by training 4 shallow CNNs on the 4 views independently and then use the concatenated feature maps for classification. They use the glitches from the first observing run of LIGO and employ a 75%,12.5% and, 12.5% training, validation and test split.

D. George et al[6] take the approach of deep transfer learning to classify glitch images in single view mode and fused view mode in which they encode 1.0 sec, 2.0 sec and 4.0 sec views of a glitch in the RGB channels of a new image. They employ a 80%, 20% training and validation split of their data to finetune various popular deep CNN architectures[7][16][17] pretrained on ImageNet[5].

To the best of out knowledge, this paper represents the first attempt as glitch-classification in a label scare setting.

6 CONCLUSIONS AND FUTURE DIRECTIONS

We present a *proof of concept* approach for tackling the problem of classifying LIGO glitches in a label scarce setting.

Our unsupervised tensor-based analysis shows that the simple CP/PARAFAC tensor decomposition is able to find coherent structure in the glitch dataset. We demonstrated the ability of CP/PARAFAC to discover the intraclass structure in the catch-all class *Extremely Loud* which agrees with the ground truth composition of that class as mentioned in [1]. In future work on tensor-based analysis of the glitches, we seek to investigate the crucial question of automatic rank selection for the tensor decomposition. We will also further study the different constraints, like sparsity and near-orthogonality, that we should impose on the factors in order to discover cleaner, more interpretable structure.

glitch-class	rank16	rank17	rank18	rank21	rank22	rank24	vgg19-10%
Air Compressor	0.8	0.792381	0.651429	0.721556	0.64	0.637594	0.7909387
Blip	0.977917	0.971861	0.975585	0.779007	0.96778	0.957481	0.9461347
Chirp	**0.763373**	**0.629524**	**0.786791**	**0.635293**	**0.708635**	**0.641734**	**0.6726681**
Extremely Loud	0.763368	0.769477	0.959159	0.776485	0.774048	0.729489	0.9696696
Helix	0.987516	0.99823	0.987386	0.99099	0.79823	0.992792	0.9731499
Koi Fish	0.960781	0.971778	0.975695	0.958999	0.964452	0.762866	0.9552186
Light Modulation	0.920353	0.901827	0.945295	0.920357	0.930897	0.848184	0.9121675
1080 Lines	0.965106	0.985075	0.979237	0.964182	0.97928	0.970693	0.9087639
Low Frequency Burst	0.923818	0.920617	0.943303	0.915748	0.946082	0.911988	0.9061525
Low Frequency Lines	0.967671	0.972074	0.96233	0.954238	0.923065	0.940686	0.9203535
No Glitch	0.889135	0.886005	0.842307	0.72678	0.882322	0.835769	0.6511438
Paired Doves	**0.556444**	**0.652821**	**0.462143**	**0.385348**	**0.510476**	**0.412015**	**0.4260317**
Power Line	0.969064	0.96995	0.965219	0.775479	0.960489	0.958114	0.9636626
Repeating Blips	0.934822	0.920429	0.915624	0.806786	0.900288	0.909373	0.9165149
1400 Ripple	0.88207	0.975129	0.971175	0.963139	0.971132	0.975171	0.8472516
Scattered Light	0.961426	0.966528	0.980255	0.954508	0.97252	0.930934	0.9513487
Scratchy	0.980719	0.991529	0.985974	0.975306	0.976711	0.985974	0.9886322
Tomte	0.859776	0.70295	0.827707	0.675095	0.845038	0.658655	0.7337601
Violin Mode	0.775937	0.985196	0.983101	0.94351	0.976882	0.976441	0.9487459
Wandering Line	**0.338661**	**0.645714**	**0.475458**	**0.239394**	**0.527233**	**0.32381**	**0.565641**
Whistle	0.893064	0.903229	0.879539	0.693833	0.879965	0.861037	0.8348899

Figure 5: Average Class-wise F1 score for ensemble model using various ranks over 5 different runs of tensor decomposition. The classes highlighted in bold are some of the most under-represented classes in the training data and highlighted in green are instances where ensemble outperforms baseline VGG19.

Finally, we demonstrated that the tensor-based augmentation of deep CNN features has the potential to reduce the impact of label scarcity to some extent on the *state-of-the-art* model used for classifying glitches. Further investigation into better representations, realistic deployment of our proposed ensemble, more suitable models and data augmentation for label scarce scenario is underway.

7 ACKNOWLEDGEMENTS

We are thankful to the people behind Gravity Spy [19] and especially to Sara Bahaadini from Northwestern University for providing us the latest datasets. The GPUs used for this research were donated by the NVIDIA Corporation. Research was supported by the National Science Foundation CDS&E Grant no. OAC-1808591.

REFERENCES

[1] Sara Bahaadini, Vahid Noroozi, Neda Rohani, Scott Coughlin, Michael Zevin, JR Smith, Vicky Kalogera, and A Katsaggelos. 2018. Machine learning for Gravity Spy: Glitch classification and dataset. *Information Sciences* 444 (2018), 172–186. https://zenodo.org/record/1476156#.XNr-3Y5KhPY

[2] Sara Bahaadini, Neda Rohani, Scott Coughlin, Michael Zevin, Vicky Kalogera, and Aggelos K Katsaggelos. 2017. Deep multi-view models for glitch classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2931–2935.

[3] Rasmus Bro and Henk AL Kiers. 2003. A new efficient method for determining the number of components in PARAFAC models. *Journal of Chemometrics: A Journal of the Chemometrics Society* 17, 5 (2003), 274–286.

[4] Xiaochun Cao, Xingxing Wei, Yahong Han, and Dongdai Lin. 2014. Robust face clustering via tensor decomposition. *IEEE transactions on cybernetics* 45, 11 (2014), 2546–2557.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[6] Daniel George, Hongyu Shen, and EA Huerta. 2017. Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO. *arXiv preprint arXiv:1706.07446* (2017).

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

[8] Christopher J Hillar and Lek-Heng Lim. 2013. Most tensor problems are NP-hard. *Journal of the ACM (JACM)* 60, 6 (2013), 45.

[9] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

[10] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. 2019. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research* 20, 1 (2019), 925–930.

[11] LK Nuttall. 2018. Characterizing transient noise in the LIGO detectors. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376, 2120 (2018), 20170286.

[12] Evangelos E Papalexakis. 2016. Automatic unsupervised tensor mining with quality assessment. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 711–719.

[13] Evangelos E Papalexakis and Christos Faloutsos. 2015. Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5441–5445.

[14] Jade Powell. 2018. Parameter estimation and model selection of gravitational wave signals contaminated by transient detector noise glitches. *Classical and Quantum Gravity* 35, 15 (2018), 155017.

[15] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. 2017. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing* 65, 13 (2017), 3551–3582.

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

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[18] Amara Tariq and Hassan Foroosh. 2015. T-clustering: Image clustering by tensor decomposition. In *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 4803–4807.

[19] Michael Zevin, Scott Coughlin, Sara Bahaadini, Emre Besler, Neda Rohani, Sarah Allen, Miriam Cabero, Kevin Crowston, Aggelos K Katsaggelos, Shane L Larson, and others. 2017. Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science. *Classical and Quantum Gravity* 34, 6 (2017), 064003.