Detecting Spatiotemporal Lightning Patterns: An Unsupervised Graph-Based Approach

Emma Benjaminson
Carnegie Mellon University
ebenjami@andrew.cmu.edu

Satyarth Praveen
Lawrence Berkeley National Labs
satyarth@lbl.gov

Giulia Luise
University College London
g.luise.16@ucl.ac.uk

J. Emmanuel Johnson
Institut des Géosciences de l'Environnement, UGA
johnsonj@univ-grenoble-alpes.fr

Richard Strange
Trillium Technologies, Inc.
richard@trillium.tech

Milad Memarzadeh
USRA, NASA Ames Research Center
milad.memarzadeh@nasa.gov

Nadia Ahmed
University of California, Irvine
ahmedn@uci.edu

Abstract

Accurate measures of lightning activity can be used to predict extreme weather events in advance, saving lives and property. However, the current hand-crafted filtering algorithm for identifying true lightning events from data captured by the GLM onboard NOAA's GOES-R satellites is only 70% accurate, with a 5% false alarm rate. This work applies unsupervised learning techniques to the large volume and high temporal resolution GLM dataset in an effort to detect lightning within raw data signals. We present a novel data processing pipeline for the GLM Level 0 products and case study comparison of two approaches to dimensionality reduction and clustering to sort the data by similar patterns. These clusters could then be labeled by a domain expert to accurately distinguish between noise and true lightning events. We demonstrate that autoencoders with graph convolution layers can learn a translationally invariant representation of the dataset which allows for k-means clustering to group samples that have similar spatiotemporal patterns together. This is a first step towards building a machine learning pipeline for improving false event filtering to identify lightning and enhance predictive abilities in the face of increasingly frequent extreme weather events.

1 Introduction

Lightning flashes can be excellent predictors of severe weather, such as tornadoes, large hail and high winds. Lightning data is used by meteorologists to warn communities of oncoming severe weather many minutes in advance, protecting built infrastructure and human lives. The National Oceanic and Atmospheric Administration (NOAA) currently has two Geostationary Operational Environmental Satellite (GOES) weather satellites in geostationary orbit that have instruments to sense optical emissions from lightning-illuminated cloud tops in the Western Hemisphere. This instrument, known as the Geostationary Lightning Mapper (GLM), captures and records the intensity

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

of light that is incident on any camera pixel with a brightness above a certain threshold. Each of these records is considered a single "event" and is down-linked to receivers on the Earth's surface to create the raw GLM Level 0 dataset. Events are recorded every two milliseconds at a fixed wavelength of 777.4 nm [1, 2]. Each pixel corresponds to a region on the Earth's surface between 8 and 14 km in length, depending on the distance of that pixel from the nadir [3]. Each event can represent either true lightning or spurious noise caused by light glinting off of reflective surfaces like lakes, direct sunlight incident to the camera [4], radiation striking the camera sensors, instrument noise, and other artifacts. The current Level 1b Ground Processing Algorithms (GPAs) in use by NASA and NOAA, which use a series of filters based on domain knowledge, can distinguish true lightning events from these spurious events with an accuracy of approximately 70% with 5% false alarms [2, 5].

This work endeavors to replace the Level 1b GPAs with an unsupervised, data-driven, machine learning-based approach that can identify true lightning events based on their spatiotemporal patterns. It has been demonstrated that lightning events tend to occur as a sequence of pulses in a short time window, and that they tend to be clustered in space around a given storm system [3]. We hypothesize that unsupervised machine learning algorithms can identify these behaviors in the large dataset obtained by the GLM sensor, and use these learned trends to identify lightning more accurately than current methods. We present a two-step unsupervised learning approach that learns a low-dimensional latent representation of raw GLM Level 0 data, and then clusters the samples in the resultant latent space. We test two different dimensionality reduction algorithms (autoencoders with standard convolution layers and graph convolution layers [6]). During our experiments, we found that the learned representation must be translationally invariant[1]. All of the code used in this submission is available on our Gitlab repository[2].

2 Methods

2.1 Data

The size of the dataset from the active GLM onboard the GOES-R satellites is on the order of tens of terabytes, having been collected since 2017. A sample of raw data is presented in Figure 1. Given the immensity of the data, we prepared a preliminary dataset from May 18, 2021, a month characterized by high storm activity over the contiguous United States (CONUS). The events that are registered by the GLM sensor are originally stored as a large image capturing the entire disk of Earth. This image is 1300×1372 pixels in dimension. The image was tiled across space and time, with dimensions chosen based on computation time, the algorithms in use, and spatiotemporal signature of a lightning event, as described below. Domain experts define a lightning flash as lasting approximately 330ms and extending over tens of kilometers, so we chose a tile size on this order of magnitude. We filtered each tile, removing isolated events in either time or space, since they are unlikely to be lightning activity.

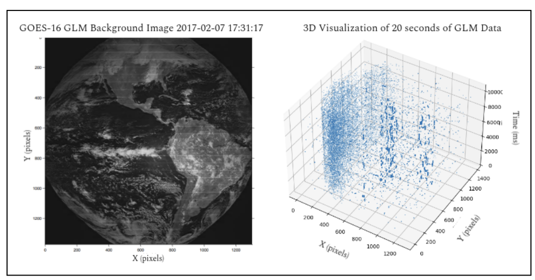

Figure 1: Raw data from GOES satellites. Left: Field of view for GOES-16. Right: 3D visualization of 20 seconds of unfiltered data.

As noted above, during experimentation we found that the learned representation of this dataset must be translationally invariant. The data patching process described here generates patches that contain groups of possible lightning events, but the location of those groups of events within the image does not have any real-world significance. We hypothesize that it is the shape of the event groupings that contains important information about whether they represent true lightning events, not the groupings' positions within the images. As pointed out by Kayhan and van Gemert and Sun et al., the semantic information contained within image-based datasets should be invariant to its position within the

[1] We also tested using the Gromov-Wasserstein distance [7] with k-means clustering and found that it was translationally invariant as well; however it did not scale well for our large dataset.

[2] https://gitlab.com/frontierdevelopmentlab/fdl-2021-lightning-upgrade-2-public-facing/lightning-ml4ps-submission

images [8, 9]. Thus as we review the results of our work, we will focus on how well each method was able to learn and sort images in a translationally-invariant fashion.

2.2 2D Convolution

The task of lightning identification (regardless of sensor) suffers from limitations in spatial coverage, detection efficiency, false alarms and other challenges which makes generating a ground truth dataset elusive. To compensate, we built a baseline approach that we could use to compare against more advanced techniques to measure model performance. Our baseline consisted of an autoencoder with four convolution layers that condense the data to a lower-dimensional latent space (we refer to this model as Conv-AE). For this model, we tiled the data into images that were 32 pixels $\times$ 32 pixels $\times$ 250 frames, which was equivalent to a spatiotemporal window of 320 km $\times$ 320 km $\times$ 0.5 seconds. We collapsed all the tiled data across the temporal domain by averaging the event frequency and event intensity and computing the event intensity standard deviation, and encoding these values for each pixel as a 3-channel input. We performed a hyperparameter sweep to determine the best parameters for the convolution layers and the size of the latent space and we present the optimal values in the next section. We applied a k-means clustering algorithm with Euclidean distance to the latent space.

2.3 Graph Convolution

We hypothesized that a graph-based representation might also be robust to translation. We implemented an autoencoder with two graph convolution network (GCN) layers [6] that together performed the following graph convolution operation on the input data to obtain the latent representation, Z:

$$Z = \sigma(\hat{A}\ \mathrm{ReLU}\big(\hat{A}XW^{(0)}\big)W^{(1)}) \tag{1}$$

where $\sigma(\cdot)$ is the softmax function, $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, $\tilde{D}$ is the renormalized degree matrix and $W^{(i)}$ are the weights of each layer [6]. For this model, we used patches of size 8 pixels $\times$ 8 pixels $\times$ 250 frames. These patches were converted to a graph-based representation of the data, $\mathcal{G} = (A, X)$ where A was a 64 $\times$ 64 adjacency matrix to encode the spatial arrangement of the events, and X was a feature matrix containing the same 3 channels of data for each event as used in the image representation. We performed a hyperparameter sweep to find the optimal values for this model (referred to as GCN-AE).

3 Experiments

We selected 10,000 sample data patches for our experiments (7,000/3,000 train/test split) [10]. We tested log normalization but found that non-normalized data provided the best results. We used tools provided by *Weights and Biases* [11] to perform hyperparameter sweeps for both of our model implementations. All work was done on cloud computing resources provided by Microsoft. The data processing was done on a virtual machine with 32 cores and 128GB of RAM; all model training was done on a virtual machine with 12 cores, 2 Tesla K80 GPUs and 112GB of RAM. For the Conv-AE, we found the best parameters were a latent dimension of 64, with a learning rate of 0.001 and a batch size of 512. Using a similar approach, we found that the optimal parameters for the GCN-AE [6, 12] were a hidden layer of dimension 32 and a latent dimension of 16, with a learning rate of 0.005 and dropout of 0.5. The batch size was 128.

4 Results and Discussion

Table 1: Cluster Metrics

Metric	Conv-AE	GCN-AE
Silhouette score	**0.697**	0.659
Davies-Bouldin score	0.781	**0.586**
Calinski-Harabasz score	5066	**8240**

The latent space learned by Conv-AE is visualized in Figure 2 [13]. The color-coding indicates the cluster label applied to each sample point by a standard k-means clustering algorithm using Euclidean distance. We found that the optimal number of clusters was 3 when testing over the range $\{2, 3, ...15\}$. In both the middle and right plots, the samples pulled from clusters 1

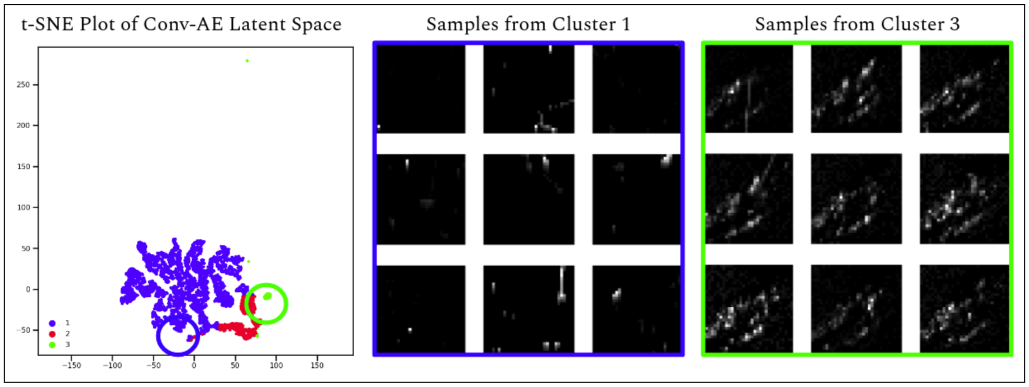

Figure 2: Latent space learned by Conv-AE. Left: t-SNE plot of latent space with cluster labels. Circles indicate sampled regions containing images shown in middle and right subfigures. Middle: 9 samples of cluster 1 (blue) showing very little translation invariance between images. Right: 9 samples of cluster 3 (green) showing all images have events in the same part of the image.

and 3 largely show activity appearing in the same region of the image, although for images of size 32×32 there are not necessarily 9 highly similar images in the dataset, which can make direct comparisons difficult. While examining the results, we found that samples pulled from specific regions in the t-SNE plot often had similar patterns, and some amount of variation in position in the image. The separation between clusters is quantified by the evaluation metrics presented in Table 1[3].

For comparison, the latent space for our GCN-AE is visualized in Figure 3. Again, the color-coding indicates the label assigned via a standard K-means clustering algorithm with Euclidean distance. We found that the optimal number of clusters was 3 when testing over the range $\{2, 3, ...15\}$. This time, when we inspected the contents of each cluster, we found that the same cluster contained examples of similar event patterns in different parts of the image, although it should be noted that it is easier to find highly similar examples for images of size 8×8 pixels. In examining our cluster metrics as shown in Table 1, we found that the silhouette coefficients for both models were about the same, but the GCN-AE had a lower Davies-Bouldin score (indicating better cluster partitioning) and a higher Calinski-Harabasz score (indicating more dense and distinct clusters).

Overall, our preliminary findings indicate that both the Conv-AE and GCN-AE models were able to find a latent representation that could be clustered and displayed some degree of translation

[3]Note that these metrics are most meaningful when applied to convex clusters and in this context we cannot guarantee that our clusters meet this requirement, so these metrics may be of limited utility.

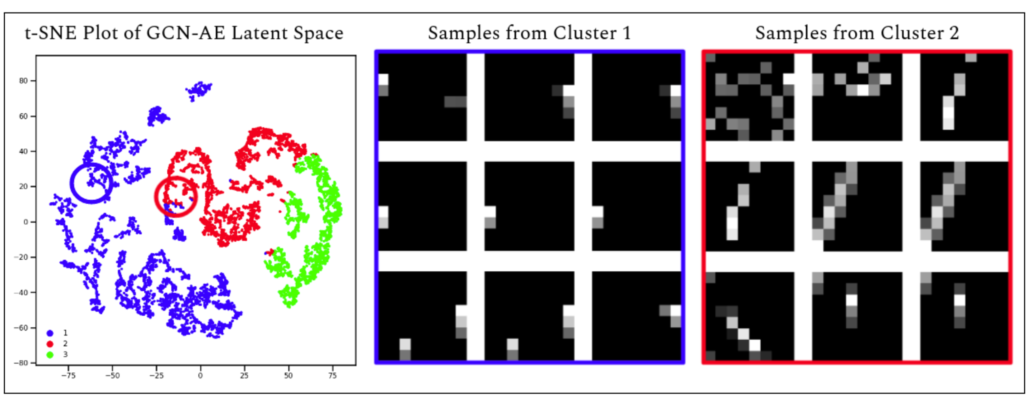

Figure 3: Representation of latent space learned by GCN-AE. Left: t-SNE plot of latent space with cluster labels. Circles indicate sampled regions containing images shown in middle and right subfigures. Middle: 9 samples of cluster 1 (blue) with translation invariance between images. Right: 9 samples of cluster 2 (red) also demonstrating translation invariance.

invariance. Our results suggest that the GCN-AE may be better at finding translation-invariant patterns in the dataset, and it may be better at learning a latent representation which leads to distinct clusters. However, the GCN-AE is limited to processing smaller images than the Conv-AE because the adjacency matrix is not a space-efficient representation of the data.

5 Next Steps

In this preliminary work, we identified two approaches to finding spatiotemporal patterns in the data that are translationally invariant. We now know that this is an important property for any algorithm designed to distinguish between noise and true lightning events based on identifying common spatiotemporal patterns in the data. While this approach does not scale well for larger tiles, our chosen tile size is an appropriate size for capturing a single typical lightning event. One long-term goal for this work is to provide automated capabilities for accurately predicting lightning events which could involve applying time-series forecasting approaches such as [14] to this dataset. We found that even a single day generates an enormous amount of data, so we propose that one way to scale this approach up would be to use online learning to continuously train a model as data is received from the GOES satellites.

6 Ethical Considerations

If we consider this work with respect to the dataset presented here, the dataset's precision is too coarse (one pixel is 8-14km in length) to represent a threat to the privacy of individuals within the instrument's field of view. The dataset is also maintained by NASA and NOAA who are responsible for maintaining the security of the raw data. This data is unlikely to enable activities that are harmful to the environment and our primary goal is to predict extreme weather for the benefit of the general public. It is possible that incorrect predictions based on our data could lead to false alarms or delayed warnings, but our work is an early prototype and has not yet been implemented in a public system. If we consider how this work might be applied to other datasets, it is possible that the models we use here to learn a latent representation could learn discriminating patterns if presented with, for example, images of faces or other human-related data.

Acknowledgments and Disclosure of Funding

This work has been enabled by the Frontier Development Lab (FDL.ai). FDL is a co-operative agreement between NASA, the SETI Institute (seti.org) and Trillium Technologies Inc, in partnership with Lockheed Martin, Microsoft and Weights and Biases. This work was in part supported with advice from Samantha Edgington and Clem Tillier. The NOAA/NASA GOES-R GLM program supported Lockheed Martin's generation of the Level 0 data set used in this work.

EB acknowledges funding by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1745016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Steven J. Goodman, Richard J. Blakeslee, William J. Koshak, Douglas Mach, Jeffrey Bailey, Dennis Buechler, Larry Carey, Chris Schultz, Monte Bateman, Eugene McCaul, and Geoffrey Stano. The GOES-R Geostationary Lightning Mapper (GLM). *Atmospheric Research*, 125-126:34–49, may 2013.

[2] Jeffrey C Smith, Robert L Morris, Clemens Rumpf, Randolph Longenbaugh, Nina McCurdy, Christopher Henze, and Jessie Dotson. An automated bolide detection pipeline for goes glm. *Icarus*, page 114576, 2021.

[3] Scott D Rudlosky, Steven J Goodman, Katrina S Virts, and Eric C Bruning. Initial geostationary lightning mapper observations. *Geophysical Research Letters*, 46(2):1097–1104, 2019.

[4] Michael Peterson. Removing solar artifacts from geostationary lightning mapper data to document lightning extremes. *Journal of applied remote sensing*, 14(3):032402, 2020.

[5] Eric C Bruning, Clemens E Tillier, Samantha F Edgington, Scott D Rudlosky, Joe Zajic, Chad Gravelle, Matt Foster, Kristin M Calhoun, P Adrian Campbell, Geoffrey T Stano, et al. Meteorological imagery for the geostationary lightning mapper. *Journal of Geophysical Research: Atmospheres*, 124(24):14285–14309, 2019.

[6] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[7] Facundo Mémoli. Gromov–Wasserstein Distances and the Metric Approach to Object Matching. *Foundations of Computational Mathematics 2011 11:4*, 11(4):417–487, apr 2011.

[8] Osman Semih Kayhan and Jan C Van Gemert. On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location. *arXiv preprint arXiv:2003.07064*, 2020.

[9] Bing Sun, Jufu Feng, and Guoping Wang. On the translation-invariance of image distance metric. *Applied Informatics 2015 2:1*, 2(1):1–12, Nov 2015.

[10] NOAA National Centers for Environmental Information. Goes-r series program, (2019): Noaa goes-r series geostationary lightning mapper (glm) level 0 data.

[11] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[12] Daehan Kim. Variational graph auto-encoder in pytorch, https://github.com/DaehanKim/vgae_pytorch, mit license, August 2020.

[13] Tarun Narayanan and Drew Bollinger. Interactive viewer, https://github.com/spaceml-org/Interactive-TSNE, June 2021.

[14] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Conguri Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *arXiv preprint arXiv:2103.07719*, 2021.

[15] Douglas M Mach. Geostationary lightning mapper clustering algorithm stability. *Journal of Geophysical Research: Atmospheres*, 125(5):e2019JD031900, 2020.

[16] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017.

[17] Chunfeng Song, Feng Liu, Yongzhen Huang, Liang Wang, and Tieniu Tan. Auto-encoder based data clustering. In *Iberoamerican congress on pattern recognition*, pages 117–124. Springer, 2013.

[18] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[19] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020.

[20] Kaspar Märtens and Christopher Yau. Basisvae: Translation-invariant feature-level clustering with variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 2928–2937. PMLR, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract and Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Sections 4 and 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Authors have read guidelines and ensured compliance.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code is available on a Gitlab repository. The URL is provided in Section 1. Data is available from NASA/NOAA.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] This results presented here are qualitative in nature and are intended to serve as a proof of concept for future work.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3 and References.
   (b) Did you mention the license of the assets? [Yes] See References. Data used in this work is publicly available and managed by NASA/NOAA.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code is available on a Gitlab repository. The URL is provided in Section 1. Raw data is available from NASA and NOAA and included in References.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The data is publicly available and consent was obtained from NOAA/NASA.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]