# From Particles to Fluids: Dimensionality Reduction for Non-Maxwellian Plasma Velocity Distributions Validated in the Fluid Context

**Daniel da Silva**
Heliophysics Sciences Division
NASA Goddard Spaceflight Center
Greenbelt, MD 20770
daniel.e.dasilva@nasa.gov

**Christopher Bard**
Heliophysics Sciences Division
NASA Goddard Spaceflight Center
Greenbelt, MD 20770
christopher.bard@nasa.gov

## Abstract

Gases and plasmas can be modeled in both a statistical sense (as a collection of discrete particles) and a continuum sense (as a continuous distribution). A collection of discrete particles is often modeled using a Maxwellian velocity distribution, which is useful in many scenarios but limited by the assumption of thermal equilibrium. In this work, we develop an architecture to learn a low-dimensional, general parameterization of the velocity distribution from scientific instrument plasma data. Such parameterizations have direct applications in data compression and simplified downstream learning algorithms. We verify that this dimensionally-reduced distribution preserves the key underlying physics of the data after reconstruction, specifically looking at the fluid parameters as derived from the instrument plasma moments (e.g., density, velocity, temperature). Finally, we present evidence for an information bottleneck arising from the relationship between the number of reduced parameters and the quality of reconstructed fluid parameters. Applying this learned architecture to data compression, we achieved a 30X compression ratio with what were deemed as acceptable losses.

## 1 Introduction

The field of physics has a long history of utilizing dimensionality reduction methods to distill data, including but not limited to spherical harmonics, the Fourier Transform, and the wavelet transform. Here, we present a technique for performing dimensionality reduction on ion counts distributions from the Fast Plasma Instrument of the Magnetosphere Multiscale mission using a data-adaptive method powered by neural networks. This has applications to both feeding low-dimensional parameterizations of the counts distributions into other machine learning algorithms, and the problem of data compression to reduce transmission volume for space missions. The algorithm presented here is lossy, and in this work, we present the technique of validating the reconstruction performance with calculated plasma moments under the argument that preserving the moments also preserves fluid-level physics, and in turn a degree of scientific validity. Code can be found online under the MIT license at `https://github.com/ddasilva/plasma-compression-neurips-2022`.

## 2 Data

We use data from the Fast Plasma Investigation Dual Ion Spectrometers (FPI/DIS) instrument on board the Magnetospheric Multiscale (MMS) satellite mission [1] [6]. MMS studies the Earth's space environment, specifically studying how magnetic fields and plasmas interact within the context of Earth's interaction with the solar wind. We utilize the subset of MMS data from its dayside orbit,

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

when the spacecraft were up to 70,080 km (11 Earth radii) away from Earth towards the direction of the Sun. This region of space is covered by low-density collisionless plasma, generally with fewer than 50 particles per $\mathrm{cm}^3$.

FPI/DIS measures a three-dimensional picture of ion plasma every 150 milliseconds, sensing what is in effect a histogram of particles over velocity space. More specifically, DIS generates an ion detection rate $C(\phi, \theta, E)$ at the moving spacecraft location, which is the detection rate of ions within a wedge of velocity space (in units of counts). $C(\phi, \theta, E)$ is proportional to the phase-space density $f(\phi, \theta, E)$ through scaling by a calibration factor. In these expressions, $\phi$ and $\theta$ are the azimuth and elevation look directions in the Geocentric Solar Ecliptic (GSE) coordinate system, and $E$ is the energy of the particle. The instrument resolution is $(N_\phi, N_\theta, N_E) \rightarrow (32, 16, 32)$. The azimuth and elevation angles are linearly spaced with full $4\pi$-steradian coverage, and the energies are log-spaced between 10 eV and 30 keV. The training and test data are split 90/10 from all data during Phase 4B of the mission (November 29, 2018 - April 13, 2019). MMS data is available for free under a Creative Commons license at `https://lasp.colorado.edu/mms/sdc/public/`.

## 3 Methods

We use a patch-based, multi-network autoencoder architecture (**Figure 1**) to perform dimensionality reduction on the instrument counts distributions $C(\phi, \theta, E)$. Based on previous work in [2] to remove compression artifacts from this same data, we use a single-layer hidden network operating on patches of the 3D structure, using rectified linear unit (ReLU) activations on the hidden and final layers. Using ReLU in the final layer guarantees the positivity of the output counts, which is a key property of the data. Advantages of the single-layer architecture over convolutional networks for this type of data is discussed in [2].

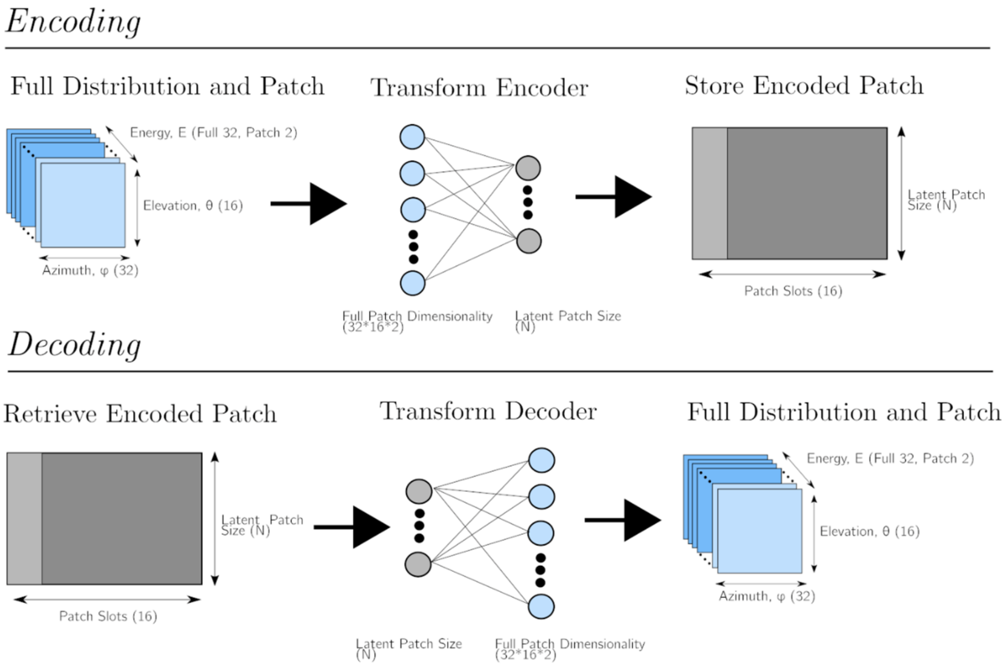

Figure 1: The encoding/decoding steps of the dimensionality reduction method. A patch "slot" corresponds to a column in the matrix storing all latent representations. Each patch slot (column) is a latent representation for a single patch. Please note that the algorithm includes a normalization step to guarantee the preservation of the mean number of counts per energy shell (not illustrated in this diagram).

We use data patches of $C(\phi, \theta, E)$ consisting of all look directions and two neighboring energy levels. The network architecture is duplicated and trained independently between each patch location in energy space. In other words, a separate but architecturally identical network is trained for each patch location. This allows each network to tailor itself to specific portions of velocity space. We define the loss function as the pixel-wise mean squared error between the original and reconstructed counts, $C(\phi, \theta, E)$ vs $\tilde{C}(\phi, \theta, E)$ within the patch. We experimented with using the reconstructed fluid variables (which can be derived from the counts distributions) in the loss function but these networks were extremely difficult to train. The network was trained with ADAM optimizer with a learning rate of 0.001[5]. Finally, we force the mean of the each energy shell $E$ equal the original energy shell's mean to ensure the preservation of raw counts:

$$C^E = \text{Energy Shell } E \text{ from Original Patch } C \tag{1}$$

$$\tilde{C^E} = \text{Energy Shell } E \text{ from Reconstructed Patch } Decoder(Encoder(C)) \tag{2}$$

$$\tilde{C}^E_{adjusted} = \tilde{C}^E * \frac{\mathrm{Mean(C^E)}}{\mathrm{Mean(\tilde{C^E})}} \tag{3}$$

**Figure 1** illustrates the dimensionality reduction method applied on a per-patch basis. Starting with a patch selected from the training set, the transform encoder encodes the patch into a latent representation with dimensional size N (where N is a user-selected parameter). The fully encoded counts distribution from the autoencoder is thus 16×N, where 16 is the number of patch slots for the DIS data. The process for decoding is the reverse of encoding: the latent representation is run through the middle and final layers of the autoencoder network. The decoded representation of each patch is stored in an array holding the full reconstructed count distribution.

A limitation of this method (and all auto-encoders) is the confined ability to generalize the encoding process to types of data not found or underrepresented in the training set. This may pose an issue for "once-in-a-lifetime" observations where-in the corresponds to a very rare (but scientifically interesting) event in nature. The extent of this limitation is subject to on-going investigation. These networks were trained on Amazon Web Services (AWS) using a single NVIDIA Tesla T4 GPU over about 12 hours.

## 4 Validation and Fluid Variable Information Bottleneck

Since the observed ion count distributions are created from underlying physical processes, we must ensure that the reconstructed data is also physically consistent. We check how well the dimensionality reduction is able to preserve fluid variables; this is used as a proxy for how well the final reconstruction preserves fundamental continuum conservation laws. The fluid variables are themselves moments of the reconstructed ion velocity distribution, which is in turn a per-pixel scaled version of the counts distribution modeled by the autoencoder. A dimensionality reduction which performs well to preserve, e.g., $\rho$ and $\vec{v}$ on data will also do well to preserve $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0$. Therefore, the performance in reconstructed fluid parameters serves as strong indicator of the ability to preserve continuum-level physics.

There is mathematical support that strong agreement in fluid variables (moments) leads to agreement in the distribution function. The result of the Hausdorff moment problem states that if two distributions $f_1(\phi, \theta, E)$ and $f_2(\phi, \theta, E)$ have the same moments $M_1^{(k)} = M_2^{(k)}$ for all $k = 0, 1, 2, \ldots, \infty$ then it is necessarily true that $f_1 = f_2$ [3][4] [7]. This is only guaranteed when $f_1$ and $f_2$ are defined on a bounded space and is not necessarily true for distributions defined on unbounded space. Here, our velocity spaces are bounded by $|\vec{v}| < c$. We note a major limitation of this theorem is that it is stated in terms of absolute equality for an infinite sequence, and similarly concludes that $f_1 = f_2$ exactly. In practice absolute equality is not achievable and therefore the literal interpretation is limited. However, we believe the intuitive principle of the theorem is still useful and provides a foundation of mathematical support for the methodology.

The size of the latent representation ($N$) for each patch is taken as an independent parameter. For each $N$, the performance of the fluid variable reconstruction is evaluated in terms of the fluid-variable-wise correlation coefficient $r^2$ between original and reconstructed data. **Figure 2** shows the $r^2$ vs the dimensionality reduction fraction (DRF) for the first four fluid variables. We see an information bottleneck around a DRF of 0.05-0.10 (10-20X) wherein further dimensionality quickly reduces the quality of the fluid variables.

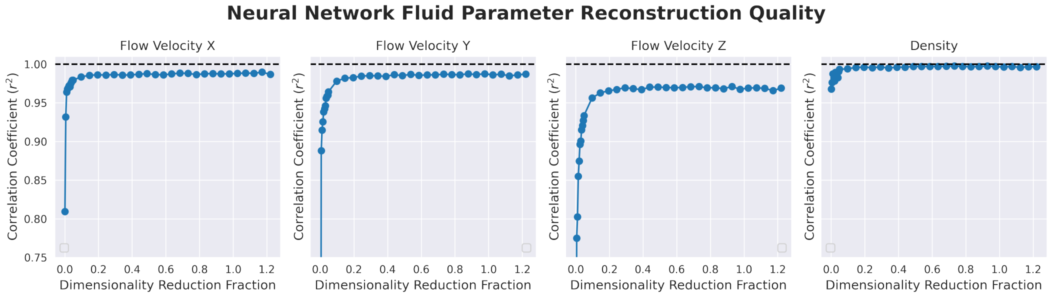

Figure 2: Correlation coefficient $r^2$ between the original and reconstructed plasma data (density and velocity components) as a function of the dimensionality reduction fraction (DRF). The DRF is defined as the size of the latent representation for each patch relative to the original dimension, with 1.0 indicating no dimensional reduction. We observe that an information bottleneck, which prevents accurate reconstructions, occurs for dimensional reductions between 10-20x (DRF $\approx 0.05 - 0.10$).

## 5 Applications to Data Compression

Potential use cases for this method include reducing the required satellite transmission budget for observed data. **Figure 3** shows that the compression demonstrates strong ability to capture the energy spread of the distribution as well as transitions between cold and hot plasma in the spectrogram perspective. Because we preserve the mean counts for each patch, the relative error in the spectrogram perspective is extremely low.

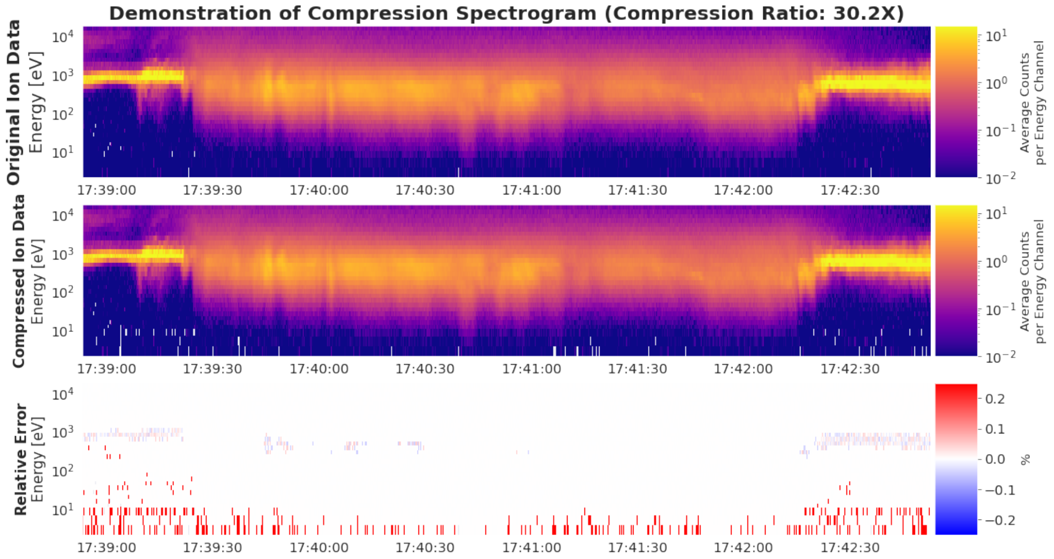

Figure 3: Demonstration of the compression algorithm in its end-to-end form displaying the original ion data and the compressed (reconstructed) ion data. This uses a version of the model with a latent patch size of N=100 corresponding to DRF=0.098. Quantization is used to trim 6 bits off the fractional part of IEEE 16-bit floating point latent representation, and the GZIP software implementing DEFLATE is used for lossless entropy coding. Relative error is computed as $(\text{original} - \text{reconstructed})/\text{original}$.

Features such as bimodal populations (first 10 seconds) and populations skewed with tails extending to lower energies (about 10-25 seconds in) are preserved. Between 17:41:00 and 17:41:30 the dip in the peak of the main population is also well preserved. The background flux of the spectrogram (dark purple) is sufficiently similar between the original and reconstructed data. Error in the high flux areas are caused only by quantization in the stored calculated mean.

## 6 Potential Broader Impact

The method has potential to support other machine learning algorithms, particularly in space physics, by reducing plasma data to a form with fewer parameters. This makes the data easier to process and train on. The compression algorithm also has the potential to reduce spaceflight costs for future space missions, where-in the cost to utilize a radio transmission telemetry network is a major cost particularly for high data-rate spacecraft. In addition to cost savings, data compression relieves load on over-congested telemetry networks that may not have bandwidth to spare. The authors believe that such compression algorithms will benefit both scientific spaceflight mission development as well as society through greater scientific return. The inherent impact on the broader world is neutral otherwise, as the algorithm is agnostic to how the data may be used. However, since our application is specifically for plasma data, it is difficult to imagine how this would negatively impact society and human relations.

## References

[1] JL Burch et al. “Magnetospheric multiscale overview and science objectives”. In: *Space Science Reviews* 199.1 (2016), pp. 5–21.

[2] D. da Silva et al. “Neural Network Repair of Lossy Compression Artifacts in the September 2015 to March 2016 Duration of the MMS/FPI Data Set”. In: *Journal of Geophysical Research (Space Physics)* 125.4, e27181 (Apr. 2020), e27181. DOI: 10.1029/2019JA027181.

[3] Felix Hausdorff. “Summationsmethoden und momentfolgen. I”. In: *Mathematische Zeitschrift* 9.1 (1921), pp. 74–109.

[4] Felix Hausdorff. “Summationsmethoden und momentfolgen. II”. In: *Mathematische Zeitschrift* 9.3 (1921), pp. 280–299.

[5] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[6] C Pollock et al. “Fast plasma investigation for magnetospheric multiscale”. In: *Space Science Reviews* 199.1 (2016), pp. 331–406.

[7] James Alexander Shohat and Jacob David Tamarkin. *The problem of moments*. Vol. 1. American Mathematical Society (RI), 1950.

## 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] We believe the abstract accurately reflects the paper’s contributions and scope.
   (b) Did you describe the limitations of your work? [Yes] We have a paragraph dedicated to limitations of the method at the end of the "Methods" section, as well as a paragraph dedicated to discussing imperfections in the compressed spectrogram ("Applications to Data Compression" section) and how they may affect scientific conclusions.
   (c) Did you discuss any potential negative societal impacts of your work? [No] The authors believe that any proposed negative societal impacts of this work would be a stretch of the imagination.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] The authors have read them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Not a theoretical result
   (b) Did you include complete proofs of all theoretical results? [N/A] Not a theoretical result

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] Code is available at a Github location found in the introduction.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Yes, the data split is discussed int he data section and the hyperparameters are discussed in the methods section
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Figure 2 represents a correlation coefficient that indicates a metric of error.
   (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] Yes, comment at the end of the methods section on how we used 12 hours of training on a single GPU (NVIDIA Tesla T4) node on AWS.
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] Yes, we cited the datasets (with URLS to download the data), as well as the papers.
   (b) Did you mention the license of the assets? [Yes] Yes. The data is under Creative Commons
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] There is no supplemental material, but a link to the data is included.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] No consent was requested because the data was produced and release with the intent to be used by the community.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] There is no data relating to humans where PII or offensive content applies.
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] No crowdsourcing or conducted research with human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No crowdsourcing or conducted research with human subjects
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No crowdsourcing or conducted research with human subjects