Virgo: Scalable Unsupervised Classification of Cosmological Shock Waves

Max Lamparth[1,5*] **Ludwig M. Böss**[2,5] **Ulrich Steinwandel**[3] **Klaus Dolag**[2,4]

[1] Physics Department, Technical University of Munich
[2] Universitäts-Sternwarte München, Ludwig Maximilian University of Munich
[3] Center for Computational Astrophysics, Flatiron Institute
[4] Max Planck Institute for Astrophysics, Garching, Germany
[5] Excellence Cluster ORIGINS, Garching, Germany

Abstract

Cosmological shock waves are essential to understanding the formation of cosmological structures. To study them, scientists run computationally expensive high-resolution 3D hydrodynamic simulations. Interpreting the simulation results is challenging because the resulting data sets are enormous, and the shock wave surfaces are hard to separate and classify due to their complex morphologies and multiple shock fronts intersecting. We introduce a novel pipeline, VIRGO, combining physical motivation, scalability, and probabilistic robustness to tackle this unsolved unsupervised classification problem. To this end, we employ kernel principal component analysis with low-rank matrix approximations to denoise data sets of shocked particles and create labeled subsets. We perform supervised classification to recover full data resolution with stochastic variational deep kernel learning. We evaluate on three state-of-the-art data sets with varying complexity and achieve good results. The proposed pipeline runs automatically, has few hyperparameters, and performs well on all tested data sets. Our results are promising for large-scale applications, and we highlight now enabled future scientific work.

1 Introduction

Cosmological structures form by gravitationally accreting mass from their surroundings [e.g. 3, 44, 29]. As galaxies fall into clusters, they dissipate their energy in the form of shock waves in the diffuse gas between them, labeled as the intra-cluster medium (ICM) [e.g. 6, 35, 36, 52, 53, 38, 47]. In these systems, the evolution of shock waves is the primary driver setting the global physical properties [e.g. 41, 48, 49, 20, 54, 46, 10]. These shock waves are defined as discontinuities in density and temperature, propagating through the ICM. They are powerful accelerators of relativistic particles, which we can observe as synchrotron emission sources from merging galaxy clusters [51]. Modeling these cosmological systems with state-of-the-art simulations requires modern supercomputers, as there is large degeneracy in the possible geometry. The produced data sets contain up to $\mathcal{O}(10^{10})$ particles, which we need to interpret to make conclusions about formation scenarios. However, shock wave structures in galaxy clusters form highly complex shapes and surfaces (see Fig. 1), and collisions between them lead to a superposition of different shock waves with overlapping geometries. From first principles, we can not make a simple, prior connection between in-falling substructures and shock wave surfaces. This setup poses a complex unsupervised classification problem for an unknown number of target classes in which we must find, separate, and label coherent shock wave structures in simulated data. To this end, we propose a novel, physically motivated, and fully scalable

*Corresponding author: max.lamparth@tum.de

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

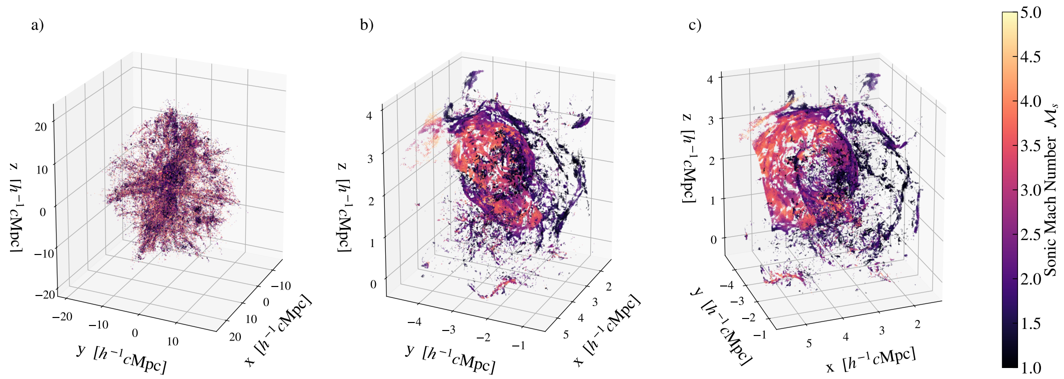

Figure 1: Simulated data set (CLUSTHD$_2$) containing all particles with a detected shock in the velocity range $\mathcal{M}_s \in [1, 5]$. *a)* Full simulation domain. *b)* Manual selection showing the complex shock structure of multiple ongoing merger shocks. We want to determine all shock surface particles as separate labeled groups and remove non-shock wave particles. *c)* Same as *b)*, but rotated by 45°.

machine learning pipeline to solve the outlined problem. We separate the unsupervised classification task by creating labels for a random subset of each data set and then training a classifier on that subset. For pre-processing, we exploit the non-stationarity of the problem with kernel principal component analysis (kernel PCA) [18, 40] and use Gaussian mixture models (GMM) [1] to pre-clean the data from unwanted non-shock wave particles. For the subset classification, we further use physically motivated kernel functions with kernel PCA, Nyström approximation [9, 37] and employ an agglomerative clustering (friends-of-friends (FoF) [5]) algorithm with an automatically set linking length. Finally, we use the labeled subset to train a stochastic variational deep kernel learning (SV-DKL, DKL) [14, 15, 56, 55] classifier to use our algorithm on the full data sets. For the first time, we can tackle this previously unsolved problem with our described pipeline and guarantee scalability for state-of-the-art and future data sets.

2 Background and Related Work

We utilize FoF and GMM for their flexibility and good scalability to large data sets [24]. The FoF algorithm is commonly used in astrophysics, e.g., for structure identification [5, 7]. We use the for physics applications attractive [22] kernel functions [39], as we can create physically motivated kernel functions with interpretable parameters and consider information like, e.g., symmetry or local density changes. For more flexible kernel functions, the authors of [56] introduced scalable deep kernel learning (DKL) to utilize the adaptive basis functions of a neural network (NN). A deep kernel NN is used as input for a base kernel of a Gaussian process [37] and their parameters are jointly trained. The DKL approach was expanded by [55], introducing stochastic variational deep kernel learning (SV-DKL). Gaussian processes have found applications in astrophysics [42, 50, 26, 23, 19, 25]. The authors of [22] also suggest SV-DKL for large-scale physics simulation interpolation. There have been analytical [43] and deep learning [27, 28] approaches to shock wave and supernovae identification. Clustering algorithms are commonly used in astrophysics [5, 59, 11, 34, 33]. In comparison, SV-DKL is the only approach offering Gaussian process non-parametric and statistical benefits, sufficient scalability and flexibility of applied metrics to solve our problem. To the best of our knowledge, this is the first application of DKL and SV-DKL to an astrophysical task. Previous work only hinted at DKL applications [30]. Furthermore, we are not aware of any preceding work having solved this unsupervised classification problem, in particular not with state-of-the-art resolution data sets.

3 Data Sets

We employ three data sets for the shock surface classification to gauge the flexibility of our pipeline for increasing physical complexity and different scales. All data sets were generated with OPEN-GADGET3[45, 2] (GNU). We name the data sets CLUSTHD (Fig. 1), CLUSTMHD and BOXMHD. The first two are ultra-high-resolution (magneto) hydrodynamics simulations (MHD) of a single massive galaxy cluster ($\sim 10^9$ particles), and BOXMHD is a high-resolution simulation of a large cosmological volume with many clusters ($\sim 10^{10}$ particles), but at lower resolution [2]. The magnetic fields lead to additional motions perpendicular to the shock propagation and more patchy shock wave surfaces. For BOXMHD, our problem scales from unsupervised classification of an unknown number of target classes from a single cluster to the same for an unknown number of clusters. We reduce the full data set of the simulation to only the parameters relevant for the actual shock surfaces. These are the spatial positions $\mathbf{x}$, the sonic Mach number $\mathcal{M}_\mathrm{s}$ and the shock normal vector $\hat{\mathbf{n}}_\mathrm{s}$.

4 VIRGO Model Pipeline

We propose a new pipeline to solve the unsupervised classification of an unknown number of cosmological shock waves in four separate steps:
1) For data pre-processing, we remove data points above a conservative Mach threshold ($\mathcal{M}_s \leq 15$) and rescale the data set to a zero mean and unit variance. We only use the particle position $\mathbf{x}$ and shock normal vector $\hat{\mathbf{n}}_s$. Each particle therefore is a 6-dimensional vector $\mathbf{q} = (x_x, x_y, x_z, \hat{n}_{sx}, \hat{n}_{sy}, \hat{n}_{sz})^\top$.
2) The raw simulation output is noisy with non-shock wave particles and not centered, as is illustrated in Fig. 1a. We use an RBF kernel with the Nyström approximation on the particle positions $\mathbf{x}$ for kernel PCA. We use GMM in the feature space with expectation maximization to separate the actual cluster of shock waves from non-shock wave particles by density estimation.
3) We construct a physically motivated composite kernel k_V by adding two separate composite kernels made up of Matérn-$\frac{5}{2}$ kernels k_M and linear kernels k_L

$$\begin{aligned} k_1(\mathbf{q}, \mathbf{q}') &= k_\mathrm{M}(\mathbf{x}, \mathbf{x}') \cdot k_\mathrm{L}(\mathbf{x}, \mathbf{x}') \\ k_2(\mathbf{q}, \mathbf{q}') &= k_\mathrm{M}(\mathbf{x}, \mathbf{x}') \cdot k_\mathrm{L}(\hat{\mathbf{n}}_s, \hat{\mathbf{n}}_s') \\ k_\mathrm{V}(\mathbf{q}, \mathbf{q}') &= k_1(\mathbf{q}, \mathbf{q}') + k_2(\mathbf{q}, \mathbf{q}'). \end{aligned}$$

k_1 creates a non-stationary kernel for spatial information, whereas k_2 combines local spatial information with shock normal directions of the particles. We combine k_V with the Nyström approximation and PCA, accepting a reduction of the data set to a random subset for computational limitations. The resulting feature space enables separation with a fixed linking length β FoF algorithm. We estimate β with the average n-next-neighbor distance in the resulting feature space.
4) We use this labeled subset to train an SV-DKL classifier. With the deep kernel, we gain a locally adaptable similarity metric required for robust classification. The SV-DKL framework allows us to achieve fast inference and good scalability, as we are not limited by the size of the data set.
Our approach is distinctly scalable, as we can downsize the data set at each step only to recover full resolution with the SV-DKL at the end. We collect our analysis and tools in a Python software package to be available for future work, called VIRGO[3]. The package utilizes already implemented features of PyTorch [31] (BSD), GPyTorch [12] (MIT), scikit-learn [32] (BSD) and pyfof [13] (MIT).

5 Experiments

We evaluate our pipeline on the data sets from Sec. 3[4]. Different time steps of one simulation, indicated by an index, are quasi-independent data sets to be solved due to morphing structures and changing number of target classes. As there exist no labeled data sets, we must verify the results visually by the coherence of the shock wave surface classification and the removal of non-shock particles. In our studies, we observe that any other approach visibly over- or under-segments the shock waves. We show the denoising and centering process representative for CLUSTHD$_2$ in Fig. 2. Our approach accurately separates the dense cluster region from the general simulation output. Should more structures be present, we increase the number of GMM components and obtain reliable results for all tested data sets. We classify the denoised result as described in step 3) while reducing the data set in size to a random subset. However, we recover full resolution with the SV-DKL classifier trained on the labeled subset. This final classification does not depend on the choice of the random subset in

[2]You can find more information, as well as a movie of some of the simulations at `http://www.magneticum.org/complements.html#Compass`

[3]**V**ariational **I**nference package for unsupe**R**vised classification of (inter-)**G**alactic sh**O**ck waves. The source code is publicly available at https://github.com/maxlampe/virgo (MIT)

[4]Used hardware, training and model parameters are stated in the appendix.

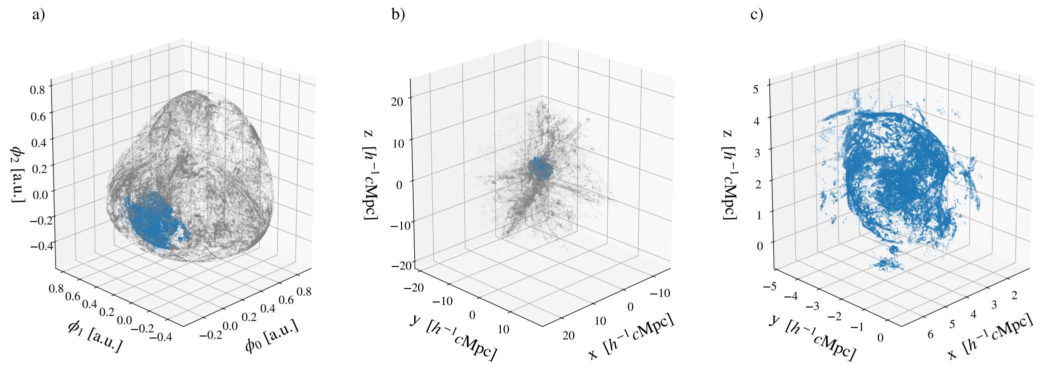

Figure 2: Denoising process of Sec. 4 step 2) for CLUSTHD$_2$ data set from Fig. 1. *a)* GMM fitted in kernel-PCA space with low-density noise (gray) and high-density cluster component (blue). *b)* Labeled data from *a)*, but in physical space. *c)* Resulting denoised data set for further analysis.

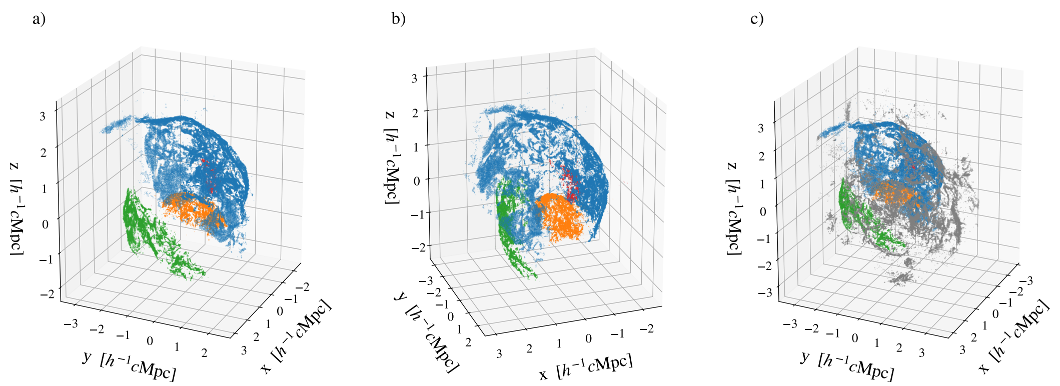

Figure 3: Full resolution reconstruction of CLUSTHD$_2$ with the SV-DKL classification. This result is the final VIRGO output for the raw input in Fig. 1. *a)* Labeled data set with SV-DKL classifier, step 4) in Sec. 4. *b)* same as a), but rotated by 45°. *c)* Same as a), but with non-shock wave particles.

the previous step. Fig. 3 shows the reconstructed and labeled data set of CLUSTHD$_2$. The complex morphology of the shock waves and its substructures are restored and correctly labelled. We compare the SV-DKL against a k-nearest-neighbor (k-NN, k = 10) classifier and a fully connected NN (like deep kernel NN) in Tab. 1. The SV-DKL outperforms the other methods in accuracy. However, k-NN achieves decent accuracy, and we recommend it as a cost-effective replacement for online applications. VIRGO also successfully separates and labels shock waves on the more complex BOXMHD data set. We repeat step 2) for this data set twice to deal with the multiple cluster objects. This additional step is required to single out dense objects and do single cluster analysis. In addition, VIRGO shows signs of generalization, as we used the trained classifier from the labeled subset of CLUSTHD$_2$ on the full data set of CLUSTHD$_3$ and obtained good results as well. However, this requires the same amount of target shock wave classes. Overall, VIRGO solves the outlined classification problem of cosmological shock waves and delivers robust results on all tested data sets.

Table 1: Comparing average test accuracies on the labeled subsets of the data after step 3) in Sec. 4 for different methods on different data sets for ten independent runs.

Method	CLUSTHD$_1$	CLUSTHD$_2$	CLUSTHD$_3$	CLUSTMHD$_1$	BOXMHD$_1$
k-NN	97.10 ± 0.34	96.61 ± 0.32	97.19 ± 0.30	96.57 ± 0.39	96.69 ± 0.48
FC-NN	95.33 ± 1.32	95.51 ± 0.84	96.63 ± 0.41	96.19 ± 0.50	95.05 ± 0.69
SV-DKL	$\mathbf{97.57 \pm 0.54}$	$\mathbf{97.00 \pm 0.49}$	$\mathbf{98.36 \pm 0.18}$	$\mathbf{98.08 \pm 0.16}$	$\mathbf{98.02 \pm 0.37}$

6 Discussion and Limitations

We demonstrated the capability of VIRGO to capture the irregular shapes of shock wave surfaces. For future work, we propose using VIRGO to improve large-scale galaxy-cluster simulations by increasing the efficiency of particle injections at shock structures [e.g. 8, 58, 57, 4] or to study supernovae remnants [17]. We determine the linking length estimator from step 3) to be most prone to error and limitation. However, our data sets are insufficient to construct an estimator for this hyper-parameter without overfitting. Also, labeling errors in step 3) will be propagated by the SV-DKL. The Gaussian process might correct minor errors, but this will not fix larger misclassifications. For applications to more complex data sets, VIRGO should be combined with a better structure finder and a criterion for actual shock wave detection. Future work should verify the robustness of our chosen hyper-parameters in a broader set of simulated data, as this might pose a challenge for users. We also propose training the same DKL over different data sets with SV-DKL to yield a more generalizable solution. The DKL could be combined with PCA and k-NN to achieve better computational scalability, classification for an unknown number of shock waves in a cluster, and robustness regarding the linking length hyper-parameter, as the pre-trained DKL could even replace or at least improve the subset labeling of step 2).

We introduced a novel, physically motivated, and scalable pipeline. The unsupervised classification problem of cosmological shock waves was successfully solved for the first time. We hope our work inspires other astrophysics and physical sciences applications with (SV)DKL.

7 General Impact Statement

We believe our proposed pipeline will increase progress in cosmology simulation studies and lead to new applications in astrophysics. While our approach is unique and distinct, it is limited to specific synthetic data sets which are generated and therefore don't necessitate privacy or fairness considerations. The data sets we are using do not contain any personally identifiable information or offensive content. Hence, we think a broader impact discussion is not applicable. Furthermore, given our specific data structure (spatial points and shock normal vectors), we see no possible applications outside natural sciences and for other purposes, even with malicious intent. We cannot outline any societal impact beyond scientific interpretations of cosmological structure formation.

Acknowledgments and Disclosure of Funding

ML, LMB and KD are acknowledging support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germanys Excellence Strategy – EXC-2094 – 390783311. KD acknowledges support for the COMPLEX project from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement ERC-2019-AdG 860744. The hydro-dynamical simulations were carried out at the Leibniz Supercomputer Center (LRZ) in Garching under the project pr86re and the CCA clusters "rusty" located in New York City as well as the cluster "popeye" located at the San Diego Supercomputing center (SDSC). UPS is supported by the Simons Foundation through a Flatiron Research Fellowship at the Center for Computational Astrophysics of the Flatiron Institute. The Flatiron Institute is supported by the Simons Foundation.

References

[1] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[2] A. M. Beck, G. Murante, A. Arth, R. S. Remus, A. F. Teklu, J. M. F. Donnert, S. Planelles, M. C. Beck, P. Förster, M. Imgrund, K. Dolag, and S. Borgani. An improved SPH scheme for cosmological simulations. *MNRAS*, 455(2):2110–2130, January 2016. doi: 10.1093/mnras/stv2443.

[3] Edmund Bertschinger. Simulations of structure formation in the universe. *Annual Review of Astronomy and Astrophysics*, 36(1):599–654, 1998. doi: 10.1146/annurev.astro.36.1.599. URL `https://doi.org/10.1146/annurev.astro.36.1.599`.

[4] Ludwig M. Böss, Ulrich P. Steinwandel, Klaus Dolag, and Harald Lesch. CRESCENDO: An on-the-fly Fokker-Planck Solver for Spectral Cosmic Rays in Cosmological Simulations. *arXiv e-prints*, art. arXiv:2207.05087, July 2022.

[5] Marc Davis, George Efstathiou, Carlos S Frenk, and Simon DM White. The evolution of large-scale structure in a universe dominated by cold dark matter. *The Astrophysical Journal*, 292:371–394, 1985.

[6] K. Dolag, F. Vazza, G. Brunetti, and G. Tormen. Turbulent gas motions in galaxy cluster simulations: the role of smoothed particle hydrodynamics viscosity. *MNRAS*, 364(3):753–772, December 2005. doi: 10.1111/j.1365-2966.2005.09630.x.

[7] K. Dolag, S. Borgani, G. Murante, and V. Springel. Substructures in hydrodynamical cluster simulations. *MNRAS*, 399(2):497–514, October 2009. doi: 10.1111/j.1365-2966.2009.15034.x.

[8] J. Donnert and G. Brunetti. An efficient Fokker-Planck solver and its application to stochastic particle acceleration in galaxy clusters. *MNRAS*, 443(4):3564–3577, October 2014. doi: 10.1093/mnras/stu1417.

[9] Petros Drineas, Michael W Mahoney, and Nello Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(12), 2005.

[10] Drummond Fielding, Eliot Quataert, Davide Martizzi, and Claude-André Faucher-Giguère. How supernovae launch galactic winds? *MNRAS*, 470(1):L39–L43, September 2017. doi: 10.1093/mnrasl/slx072.

[11] Bin Fu, Kai Ren, Julio López, Eugene Fink, and Garth Gibson. Discfinder: A data-intensive scalable cluster finder for astrophysics. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 348–351, 2010.

[12] Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. *CoRR*, abs/1809.11165, 2018. URL `http://arxiv.org/abs/1809.11165`.

[13] Simon Gibbons. pyfof - friends of friends cluster finding in python, 2019. URL `https://github.com/simongibbons/pyfof`.

[14] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[15] James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[17] Hans-Thomas Janka, Florian Hanke, Lorenz Hüdepohl, Andreas Marek, Bernhard Müller, and Martin Obergaulinger. Core-collapse supernovae: Reflections and directions. *Progress of Theoretical and Experimental Physics*, 2012(1):01A309, 2012.

[18] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374 (2065):20150202, 2016.

[19] Konstantin Karchev, Adam Coogan, and Christoph Weniger. Strong-lensing source reconstruction with variationally optimised gaussian processes. *Monthly Notices of the Royal Astronomical Society*, 2021.

[20] Chang-Goo Kim and Eve C. Ostriker. Momentum Injection by Supernovae in the Interstellar Medium. *ApJ*, 802(2):99, April 2015. doi: 10.1088/0004-637X/802/2/99.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Max Lamparth, Mattis Bestehorn, and Bastian Märkisch. Gaussian processes and bayesian optimization for high precision experiments, 2022. URL `https://arxiv.org/abs/2205.07625`.

[23] Florent Leclercq. Bayesian optimization for likelihood-free cosmological inference. *Physical Review D*, 98(6):063511, 2018.

[24] Mahmoud A Mahdi, Khalid M Hosny, and Ibrahim Elhenawy. Scalable clustering algorithms for big data: a review. *IEEE Access*, 2021.

[25] Andrew C Miller, Lauren Anderson, Boris Leistedt, John P Cunningham, David W Hogg, and David M Blei. Mapping interstellar dust with gaussian processes. *arXiv preprint arXiv:2202.06797*, 2022.

[26] Christopher J Moore, Christopher PL Berry, Alvin JK Chua, and Jonathan R Gair. Improving gravitational-wave parameter estimation using gaussian process regression. *Physical Review D*, 93(6):064001, 2016.

[27] Nathaniel R Morgan, Svetlana Tokareva, Xiaodong Liu, and Andrew Morgan. A machine learning approach for detecting shocks with high-order hydrodynamic methods. In *AIAA Scitech 2020 Forum*, page 2024, 2020.

[28] Robert Morgan, Brian Nord, Keith Bechtol, SJ González, E Buckley-Geer, A Möller, JW Park, AG Kim, S Birrer, M Aguena, et al. Deepzipper: A novel deep-learning architecture for lensed supernovae identification. *The Astrophysical Journal*, 927(1):109, 2022.

[29] Thorsten Naab and Jeremiah P. Ostriker. Theoretical Challenges in Galaxy Formation. , 55(1):59–109, August 2017. doi: 10.1146/annurev-astro-081913-040019.

[30] Ji Won Park, Ashley Villar, Yin Li, Yan-Fei Jiang, Shirley Ho, Joshua Yao-Yu Lin, Philip J Marshall, and Aaron Roodman. Inferring black hole properties from astronomical multivariate time series with bayesian attentive neural processes. *arXiv preprint arXiv:2106.01450*, 2021.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] María Sol Pera, Gabriel Ignacio Perren, A Moitinho, Hugo Daniel Navone, and Ruben Angel Vazquez. pyupmask: an improved unsupervised clustering algorithm. *arXiv preprint arXiv:2101.01660*, 2021.

[34] Gabriel Ignacio Perren, Ruben Angel Vazquez, and Andres Eduardo Piatti. Asteca: Automated stellar cluster analysis. *Astronomy & Astrophysics*, 576:A6, 2015.

[35] C. Pfrommer, T. A. Enßlin, V. Springel, M. Jubelgas, and K. Dolag. Simulating cosmic rays in clusters of galaxies - I. Effects on the Sunyaev-Zel'dovich effect and the X-ray emission. *MNRAS*, 378(2):385–408, June 2007. doi: 10.1111/j.1365-2966.2007.11732.x.

[36] C. Pfrommer, R. Pakmor, K. Schaal, C. M. Simpson, and V. Springel. Simulating cosmic ray physics on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 465(4):4500–4529, nov 2016. doi: 10.1093/mnras/stw2941.

[37] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, second edition, 2006.

[38] Kevin Schaal and Volker Springel. Shock finding on a moving mesh - I. Shock statistics in non-radiative cosmological simulations. *MNRAS*, 446(4):3992–4007, February 2015. doi: 10.1093/mnras/stu2386.

[39] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2018.

[40] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[41] Leonid Ivanovich Sedov. Propagation of strong shock waves. *Journal of Applied Mathematics and Mechanics*, 10:241–250, January 1946.

[42] Arman Shafieloo, Alex G Kim, and Eric V Linder. Gaussian process cosmography. *Physical Review D*, 85 (12):123530, 2012.

[43] Ben Snow, Andrew Hillier, Giulia Murtas, Gert JJ Botha, and Stefano Camera. Shock identification and classification in 2d magnetohydrodynamiccompressible turbulence—orszag–tang vortex. *Experimental Results*, 2, 2021.

[44] Rachel S. Somerville and Romeel Davé. Physical Models of Galaxy Formation in a Cosmological Framework. , 53:51–113, August 2015. doi: 10.1146/annurev-astro-082812-140951.

[45] Volker Springel. The cosmological simulation code GADGET-2. *MNRAS*, 364(4):1105–1134, December 2005. doi: 10.1111/j.1365-2966.2005.09655.x.

[46] Ulrich P. Steinwandel, Benjamin P. Moster, Thorsten Naab, Chia-Yu Hu, and Stefanie Walch. Hot phase generation by supernovae in ISM simulations: resolution, chemistry, and thermal conduction. *MNRAS*, 495(1):1035–1060, June 2020. doi: 10.1093/mnras/staa821.

[47] Ulrich P. Steinwandel, Ludwig M. Boess, Klaus Dolag, and Harald Lesch. On the small scale turbulent dynamo in the intracluster medium: A comparison to dynamo theory. *arXiv e-prints*, art. arXiv:2108.07822, August 2021.

[48] Geoffrey Taylor. The Formation of a Blast Wave by a Very Intense Explosion. I. Theoretical Discussion. *Proceedings of the Royal Society of London Series A*, 201(1065):159–174, March 1950. doi: 10.1098/rspa.1950.0049.

[49] Geoffrey Taylor. The Formation of a Blast Wave by a Very Intense Explosion. II. The Atomic Explosion of 1945. *Proceedings of the Royal Society of London Series A*, 201(1065):175–186, March 1950. doi: 10.1098/rspa.1950.0050.

[50] Rutger van Haasteren and Michele Vallisneri. New advances in the gaussian-process approach to pulsar-timing data analysis. *Physical Review D*, 90(10):104012, 2014.

[51] R. J. van Weeren, F. de Gasperin, H. Akamatsu, M. Brüggen, L. Feretti, H. Kang, A. Stroe, and F. Zandanel. Diffuse radio emission from galaxy clusters. *Space Science Reviews*, 215(1), feb 2019. doi: 10.1007/s11214-019-0584-z.

[52] F. Vazza, K. Dolag, D. Ryu, G. Brunetti, C. Gheller, H. Kang, and C. Pfrommer. A comparison of cosmological codes: properties of thermal gas and shock waves in large-scale structures. *MNRAS*, 418(2):960–985, December 2011. doi: 10.1111/j.1365-2966.2011.19546.x.

[53] F. Vazza, M. Brüggen, D. Wittor, C. Gheller, D. Eckert, and M. Stubbe. Constraining the efficiency of cosmic ray acceleration by cluster shocks. *MNRAS*, 459(1):70–83, June 2016. doi: 10.1093/mnras/stw584.

[54] Stefanie Walch and Thorsten Naab. The energy and momentum input of supernova explosions in structured and ionized molecular clouds. *MNRAS*, 451(3):2757–2771, August 2015. doi: 10.1093/mnras/stv1155.

[55] Andrew G Wilson, Zhiting Hu, Russ R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[56] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[57] Georg Winner, Christoph Pfrommer, Philipp Girichidis, and Rüdiger Pakmor. Evolution of cosmic ray electron spectra in magnetohydrodynamical simulations. *MNRAS*, 488(2):2235–2252, September 2019. doi: 10.1093/mnras/stz1792.

[58] D. Wittor, F. Vazza, and M. Brüggen. Testing cosmic ray acceleration with radio relics: a high-resolution study using MHD and tracers. *MNRAS*, 464(4):4448–4462, February 2017. doi: 10.1093/mnras/stw2631.

[59] Yanxia Zhang and Yongheng Zhao. Automated clustering algorithms for classification of astronomical objects. *Astronomy & Astrophysics*, 422(3):1113–1121, 2004.

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]
   (b) Did you describe the limitations of your work? [Yes] See Sec. 6 "Discussion and Limitations"
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 7 "General Impact Statement"
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [No] We are not including theoretical results. Where appropriate, we state taken assumptions.
   (b) Did you include complete proofs of all theoretical results? [No] We are not including theoretical results.
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 publicly available, see Sec. 4 (MIT license) and all data sets will be made available at request or can be generated with the stated software, see Sec. 3. We also have an anonymized, supplementary zip file with code and data, which is available at request, as we can only submit one pdf file for this workshop on the submission page.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 5 and Sec. A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Sec. 5
   (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 Sec. A
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]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See anser to checklist question 3.a)
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We are only using self-generated data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Sec. 7
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? [No] We did not used 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? [No] We did not used 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? [No] We did not used crowdsourcing or conducted research with human subjects.

A Appendix

The deep kernel NN is adjusted to our problem regarding the accuracy and a minimal number of parameters and is a fully connected NN with ReLU activation functions after each layer. We use an RBF kernel for the additive base kernels. The inducing points and training parameters are set up as in [55] and [16], unless stated otherwise below. We use a standard softmax likelihood. We optimized all parameters in steps 1) to 3) from Sec. 4 to run all experiments on a Linux machine with two 2.4 GHz CPU cores and 8 GB RAM to highlight its efficiency. However, step 4) required a GPU with 16 GB RAM. We list any distinct parameters for our pipeline and its training: For the denoising step 2), we use an RBF kernel with the Nyström approximation $m = 100$, PCA with $k = 5$ components, and GMM with 2 or 5 components, depending on the data set. The physically motivated kernel k_V of step 3) is used with Nyström approximation $m = 500$, PCA with $k = 6$ components, and we estimate the FoF linking length β as the average 20-next-neighbor distance. The deep kernel NN is set up with six input features, $n_h = 1$ hidden layer of size 20, $n_f = 10$ output features, and without pre-training. We trained the SV-DKL of step 4) over 20 epochs, with a batch size of 1024, a grid of 64 inducing points, an overall learning rate of $\alpha = 0.1$, a learning rate scheduler decreasing it by 0.1 after 50 and 75% of the training time, decreased learning rate for the Gaussian process parameters $\alpha_{GP} = \alpha/100$, L2 regularization with weight decay of 10^{-4}, and an ADAM optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [21]. We employed an 80/10/10 spit between training, validation and test sets.