`PyAWD`: A Library for Generating Large Synthetic Datasets of Acoustic Wave Propagation with Devito

Tribel Pascal[0009−0006−3701−1223]
Gianluca Bontempi[0000−0001−8621−316X]

Machine Learning Group, Université Libre de Bruxelles, Belgique
http://mlg.ulb.ac.be/

Abstract. Seismic data is often sparse and unevenly distributed due to the high costs and logistical challenges associated with deploying physical seismometers, limiting the application of Machine Learning (ML) in earthquake analysis. To address this gap, we introduce `PyAWD`, a Python library designed to generate high-resolution synthetic datasets simulating spatio-temporal acoustic wave propagation in both two-dimensional and three-dimensional heterogeneous media. By allowing fine control over parameters such as wave speed, external forces, spatial and temporal discretization, and media composition, `PyAWD` enables the creation of ML-scale datasets that capture the complexity of seismic wave behavior. We illustrate the library's potential with an epicenter retrieval task, showcasing its suitability for designing complex, accurate seismic problems that support advanced ML approaches in the absence or lack of dense real-world data.

Keywords: Wave Simulation · Pytorch Datasets · Machine Learning · Python Library · Seismic Data · Spatio-Temporal Analysis

1 Introduction

Earthquakes, sudden and intense geological events, affect lives and infrastructure. Predicting them and accurately interpreting measurements is difficult due to the complex behavior of seismic waves in varied fields. Seismic data challenges supervised ML models as they require high-resolution data that links wave propagation to specific features, such as epicenter or wave timings. Seismometers, though precise, are costly and cover limited areas, resulting in sparse datasets [Woollam et al. 2022]. Consequently, ML models often lack the details needed to effectively model seismic behavior. To overcome these data constraints, we introduce `PyAWD`, a Python library for creating synthetic seismic datasets with high spatio-temporal detail. It simulates acoustic wave propagations [Demanet 2015]through 2D and 3D fields, yielding PyTorch-compatible data that can improve ML pipelines by providing intricate wave and spatial information. Recent ML advancements hold promise for earthquake prediction and epicenter detection [Ridzwan and Md Yusoff 2023], but existing datasets limit further progress. Studies emphasize the need for high-resolution data to unlock ML's full potential in seismology [Lehmann et al. 2024]. By producing synthetic datasets that mimic geological conditions, `PyAWD` enables effective ML model training. Past work highlights the lack of synthetic datasets for seismic ML studies [S. Mostafa Mousavi and Beroza 2023; Lehmann et al. 2024]. Data sources like satellite images and local seismometer readings [Cambrin and Garza 2024; S Mostafa Mousavi et al. 2019]offer limited spatial coverage. Existing tools like *Synthoseis* [Merrifield et al. 2022]simulate 2D data, but comprehensive customizable 3D solutions are nonexistent. `PyAWD` addresses this gap by enriching spatial data for ML applications. `PyAWD` provides essential features for researchers:

– Customization of material properties, simulation duration, spatial/temporal resolution, and source details.
– Comprehensive data access: Seismograms or full spatial overviews enable studies not feasible with current real-life datasets, which consist only of *seismograms*, temporal measurement on (usually a small amount of) chosen spatial points.
– Integration with ML frameworks: PyTorch-compatible datasets aid ML model development for seismic and other geophysical applications.

The paper is organized as follows: Section 2 covers `PyAWD`'s methods, algorithms, and specifications. Section 3 presents a toy example study, focused on the question of *epicenter retrieval*, to show the usefulness of `PyAWD`. Section 4 situates our findings within related research, noting limitations and proposing future directions.

2 Methods/Materials and Methods

`PyAWD` is a tool for simulating wave propagation phenomena in complex media, allowing customization of parameters like wave speed, attenuation, and external forces. A standout feature is its handling of spatio-temporally varying propagation fields, enabling simulations in heterogeneous environments with different material properties. `PyAWD` integrates with `Pytorch`, producing datasets for Deep Learning applications. Visualization tools are also implemented, supporting both 2D and 3D representations of wave propagation, enhancing interpretation of simulations. It also offers tools for visualizing specific points in the wave field, aiding detailed analyses of the wave behavior. `PyAWD` uses the anisotropic nondispersive Acoustic Wave Equation [Demanet 2015], given by:

$$\frac{d^2u}{dt^2} = c\nabla^2 u - \alpha\frac{du}{dt} + f \tag{1}$$

where u is the displacement field, t is time, c the wave speed, α an attenuation factor, and f an external force. The ∇ operator is the *Laplacian* operator, yielding the vector of spatial derivatives of u. The parameter c introduces spatio-temporal heterogeneity, varying over time and space, modeled as $c(x, y, t)$. Users can define custom fields, simulating complex geological structures with different material properties. Figure 1 shows a wave propagation simulation with a variable c field, modeling temperature variation. Another example is the Marmousi field (Figure 2) [Versteeg 1994], provided as a preset in `PyAWD`, simulating complex geological formations. The external force is essential for realistic simulations. `PyAWD` includes an explosion-shaped force field, shown in Figure 3. The simulations

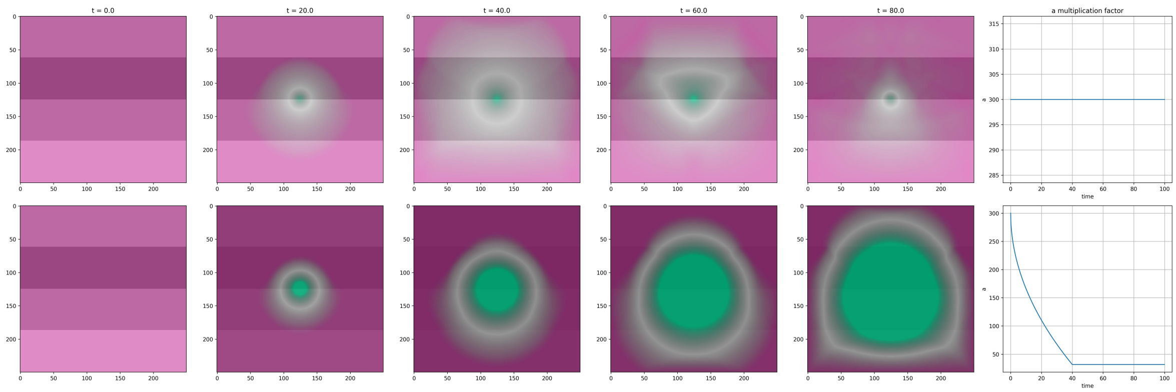

Fig. 1. Example of wave propagation with spatial and spatio-temporal varying propagation speed fields.

run on Devito [Luporini et al. 2020; Louboutin et al. 2019], a Python-based finite difference solver for partial differential equations (PDEs). Devito uses SymPy [Meurer et al. 2017]for symbolic problem definitions, generating optimized C++ code, which is JIT-compiled for performance. This enables focus on algorithmic design over low-level coding. Devito's optimizations enhance code performance, but `PyAWD` abstracts the numerical complexity for ease of use, particularly for seismologists and ML engineers. Seismic analysis benefits from neural network applications in fields like earthquake detection [S. Mousavi et al. 2019]and phase picking [Ross, Meier, and Hauksson 2018]. `PyAWD` is compatible with PyTorch [Paszke et al. 2017], allowing data to be processed using tools like `DataLoaders` and `torchvision transformations` [maintainers and contributors 2016]. PyTorch provides scalability and easy integration with standard Python libraries, allowing the connection seamlessly to tools like SciPy and sklearn. `PyAWD` is designed for interdisciplinary users, offering Jupyter Notebooks that introduce key concepts and demonstrate the library functions, covering:

– Acoustic Wave Equation solutions in Devito,
– Interrogators usage (an abstraction of *seismometers*, whose place is customizable),
– Dataset generation,
– Heterogeneous field simulations,
– Spatio-temporal field variations

A complete documentation is also available[1].

3 Validation and results

In this section, we demonstrate a toy example of `PyAWD` for *epicenter retrieval* [Perol, Gharbi, and Denolle 2018], aimed at determining wave epicenter coordinates (x_0, y_0) based on ground motion recorded at specific locations, stored in *seismograms*. Traditional methods use three or more seismometers [Braile 2002]to estimate *epicentral distance* using triangulation, yet fewer than three devices yield

[1] `https://pascaltribel.github.io/pyawd/`

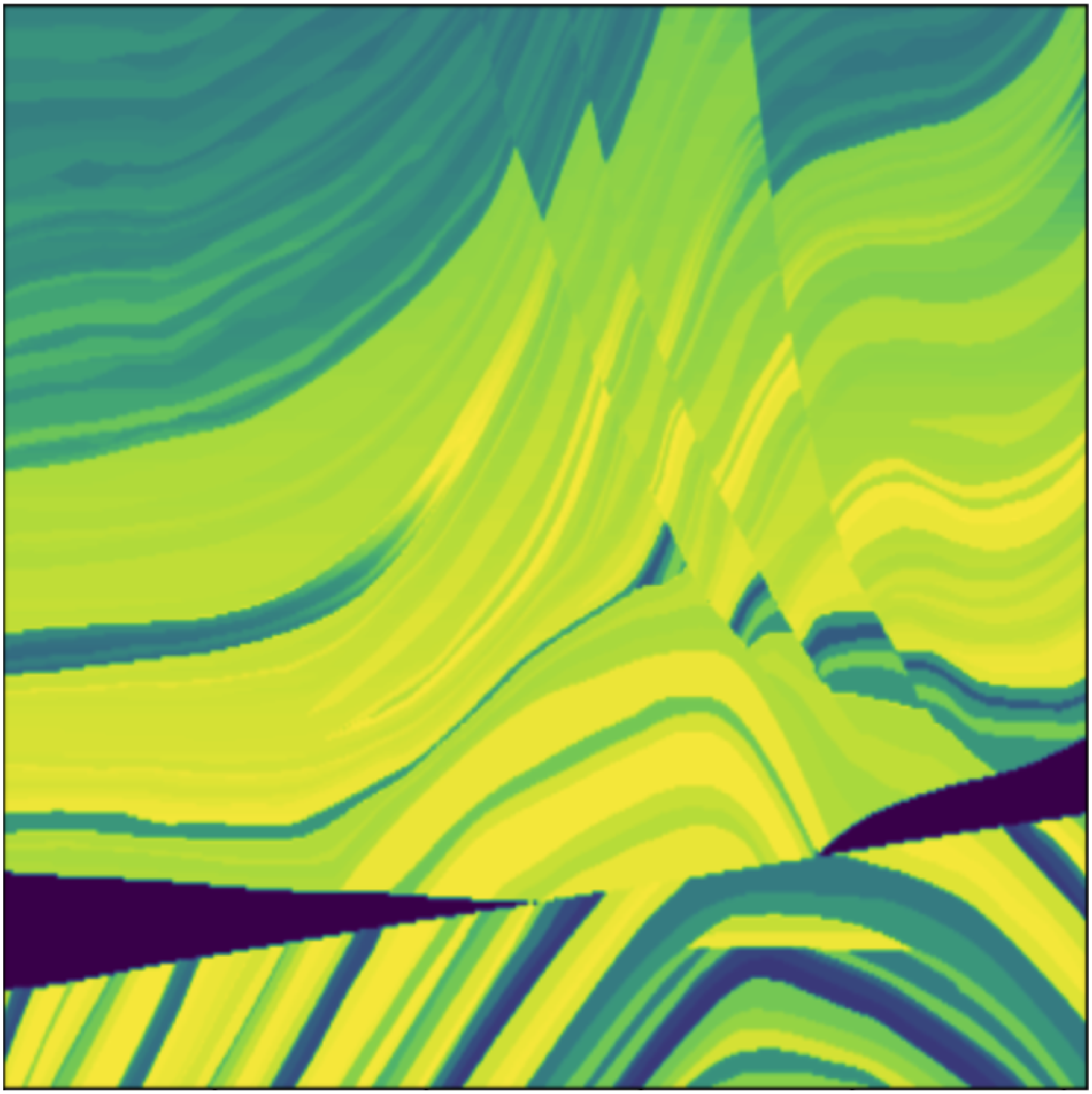

Fig. 2. Marmousi field preset in `PyAWD`. Darker colors indicate slower speeds.

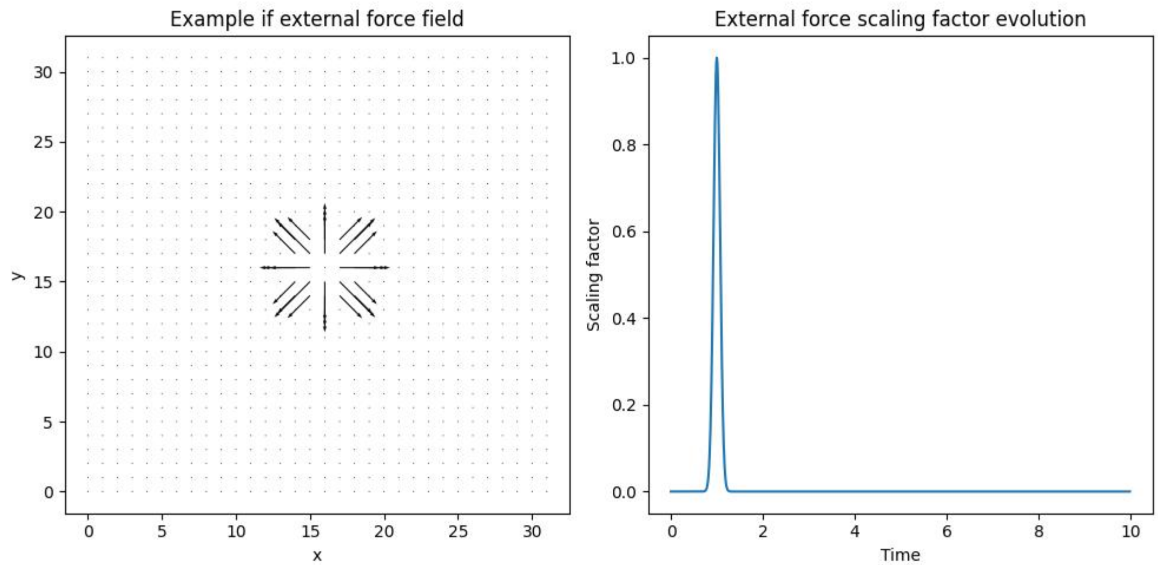

Fig. 3. External force example: left shows vector scaling; right shows scaling factor over time.

uncertainty. Recent Machine Learning advances show that seismogram data holds valuable information for this task [Perol, Gharbi, and Denolle 2018; Noda et al. 2012; Saad et al. 2023; Türkmen et al. 2024]. We hypothesize that reflection interferences within heterogeneous propagation fields provide valuable information. This could reduce epicentral uncertainty. This behavior is inspired by how humans localize sound using limited reference points [Carlini, Bordeau, and Ambard 2024]. To validate PyAWD's realism and utility, we generate 2D acoustic wave simulations in the heterogeneous Marmousi field (Figure 2), producing two datasets: one for training (4096 simulations) and one for testing (512 simulations) over a 256×256 grid. Two interrogators at coordinates $(-64, 0)$ and $(64, 0)$ record ground motion for 10 seconds at 100Hz. Epicenter positions and parameters vary within field boundaries and intervals, ensuring a diverse dataset. An example is shown in Figures 4 and 5. We approach

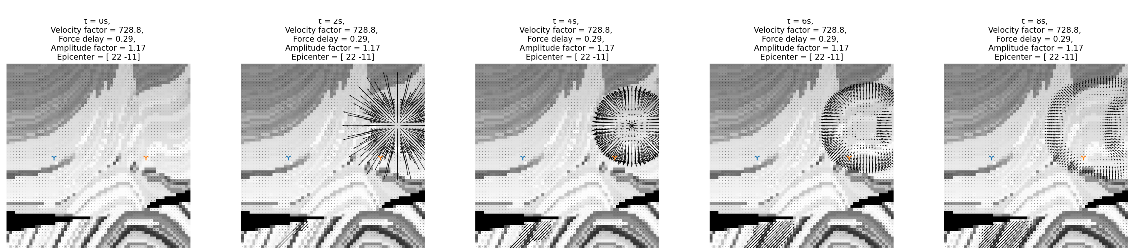

Fig. 4. Example of propagating wave in the Marmousi field, with two interrogators.

the regression problem with 14 ML architectures using raw seismograms, including:

Baseline : Constant average of training dataset

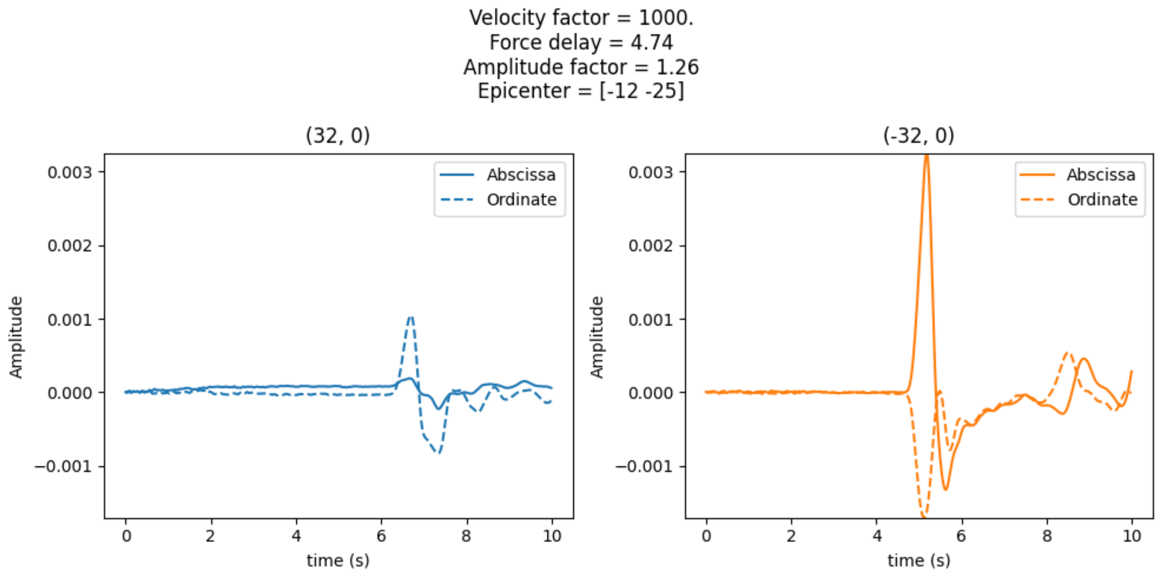

Fig. 5. Example of interrogators response following the simulation in Figure 4

Linear Models : Linear Model, Ridge Regression, Stochastic Gradient Descent
Neighbors Model : KNN Neighbors
Support Vector Machines : Support Vector Regression [Chang and Lin 2011]
Tree Model : Decision Tree [Breiman et al. 1984]
Ensemble Models : Extra Trees [Geurts, Ernst, and Wehenkel 2006], XGBoost [Chen and Guestrin 2016], Random Forest [Breiman 2001]
Deep Learning : Multi-layer Perceptron [Hinton 1989], Temporal CNN (`TCNN`) [Lea et al. 2016]

We also applied these models to features extracted from seismograms using `tsfresh` [Christ et al. 2018], which provides:

- **Basic Statistics:** Absolute energy, maximum values, and mean.
- **Trend and Autocorrelation:** Time trends and autocorrelation patterns.
- **Complexity Measures:** Entropy and compression-based complexity.
- **Peak Detection:** Peaks and their duration.
- **Crossing and Ratio Analysis:** Series threshold crossings and value ratios.

Most models are implemented in `sklearn` [Pedregosa et al. 2011], except for the `TCNN` which is implemented in PyTorch [Paszke et al. 2017]. Model performance on the testing dataset is assessed using the *mean squared error* (`MSE`). Figures 6 and 8 present the `MSE` for each model on test data. Figure 7 shows raw data model predictions, for the first and second coordinates, and the spatial predicted/expected discrepancies. Green points mark actual epicenters, and red points, linked by blue lines to corresponding green points, are predictions. Line length indicates prediction error: shorter lines mean higher accuracy. Figure 9 shows the feature-based predictions. The fact that simple mod-

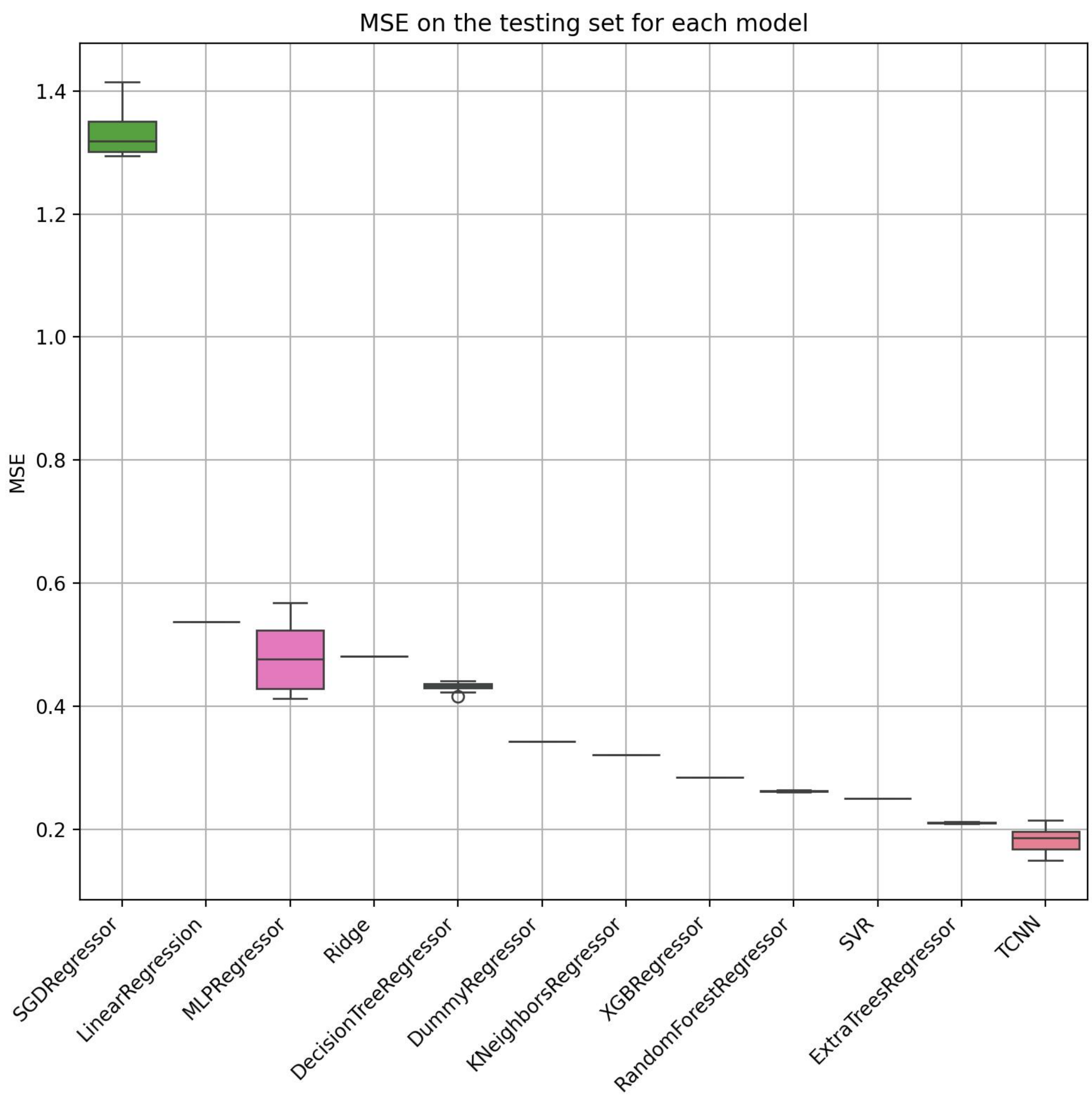

Fig. 6. Testing set errors with architectures trained on raw seismograms.

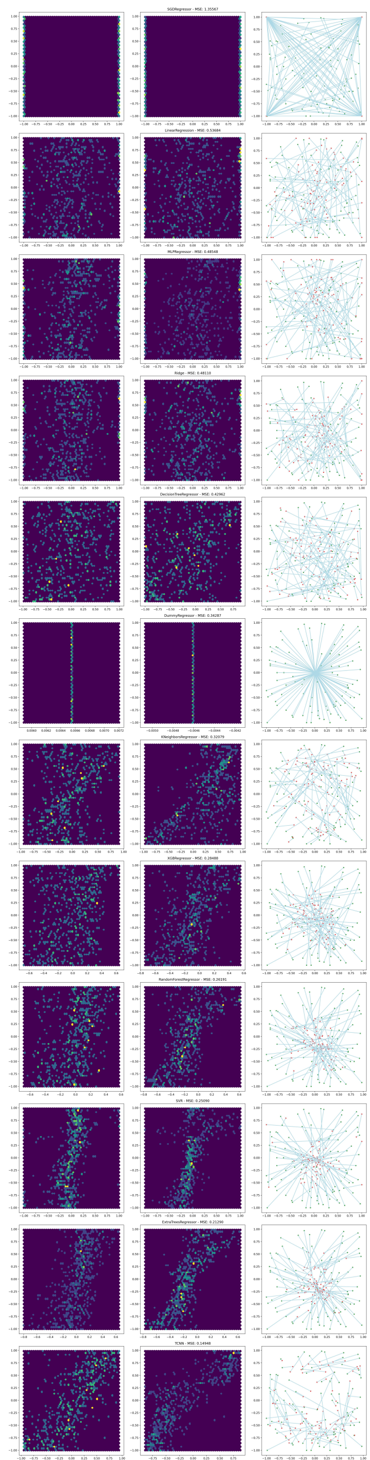

Fig. 7. Spatial comparison of predicted and expected epicenter locations for raw seismogram-trained architectures.

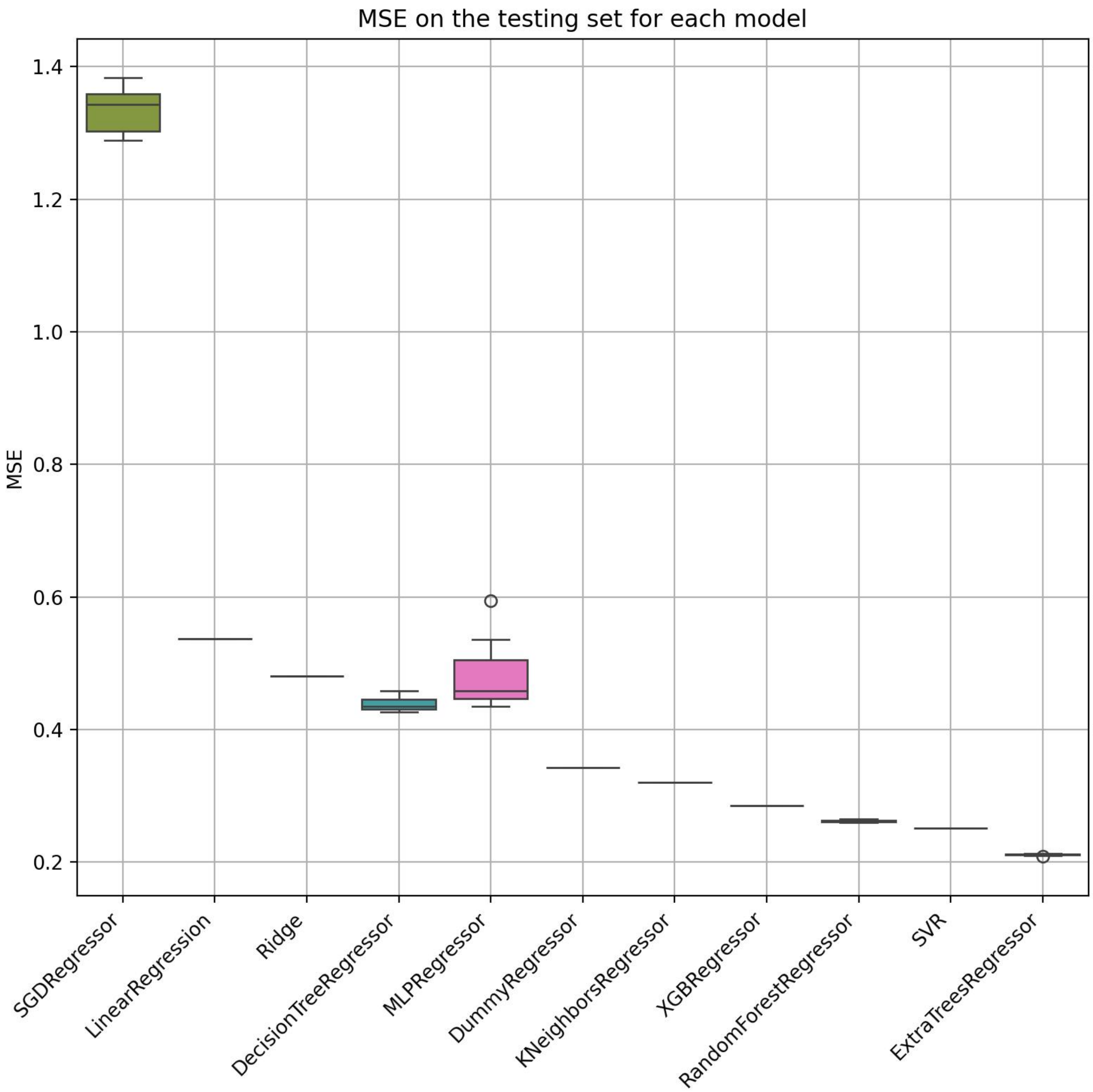

Fig. 8. Testing set errors with feature-engineered architectures.

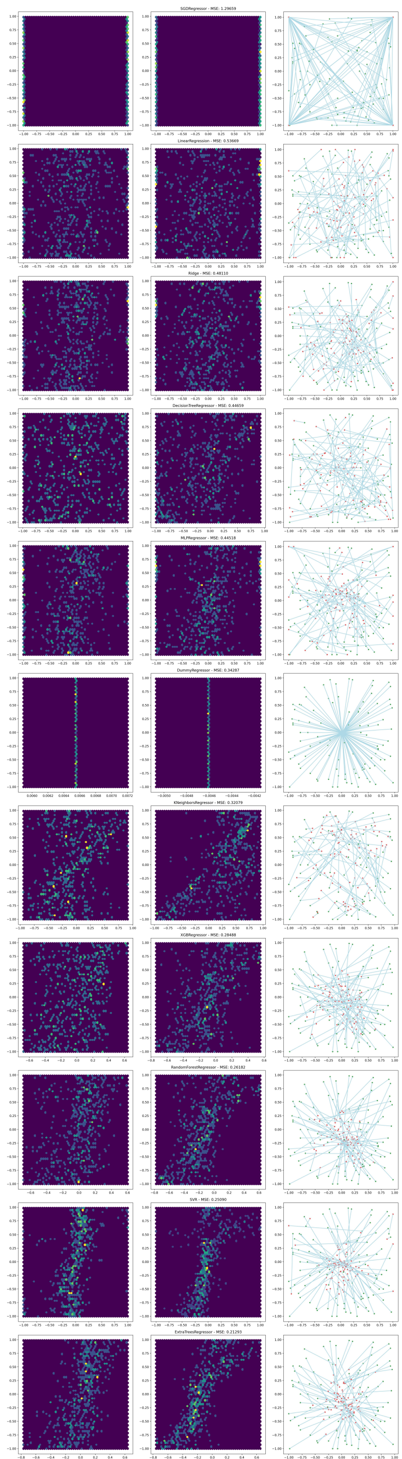

Fig. 9. Spatial comparison of predicted and expected epicenter locations for feature-engineered architectures.

els struggle with generalization indicates a challenging task. Indeed, linear model, ridge regression, decision tree and SVM fail to capture the observation-epicenter relationship. Non-linear models, particularly Extra Trees and TCNN, excel, with noticeable discrepancies across coordinates likely due to field asymmetry. Raw data models also surpass feature-engineered ones, suggesting further complex modeling approaches as proposed in the literature [Saad et al. 2023; Türkmen et al. 2024].

4 Discussion

Our results show PyAWD as a valuable simulation tool that aids Machine Learning in seismic wave analysis. It has been shown on the question of epicenter retrieval with minimal seismometer data. Experiments indicate that complex architectures like Temporal Convolutional Neural Networks (TCNNs) are effective for handling this spatio-temporal data, supporting the importance of advanced models in seismic applications. More importantly, PyAWD offers a sufficient and cost-efficient alternative to deploying numerous seismometers, by generating high-resolution synthetic seismic datasets. Especially, the problem of determining the best placement for a pair of measurement stations to retrieve an earthquake epicenter can be tackled using PyAWD in further work.While PyAWD alleviates some data acquisition limitations, challenges remain. Synthetic data may not fully replicate real-world seismic complexities, such as environmental noise or unexpected wave interactions. Future efforts should focus on implementing different governing propagation equations. Moreover, the generation of realistic heterogeneous propagation fields is not a trivial part that has to be studied as well. The impact of PyAWD extends beyond earthquake epicenter studies. It can simulate various seismic events, allowing research in areas such as resource exploration, hazard assessment, and infrastructure monitoring. By offering control over material properties and wave sources, PyAWD broadens the scope for geophysical studies and potential new applications. In conclusion, PyAWD provides a customizable tool that supports Machine Learning research in seismic analysis by generating synthetic datasets efficiently. While challenges in integrating synthetic and real-world information remain, PyAWD shows strong potential for enhancing seismic research and data accessibility.

5 Conclusion

This paper has introduced PyAWD, a Python tool that enhances seismic analysis through customizable wave simulations integrated with Machine Learning. By facilitating synthetic data creation, PyAWD helps ML-based earthquake analysis research. Future enhancements to PyAWD include refining its wave solver with options like the Elastic Wave Equation, finite-element or finite-volume methods. However, its current modularity allows the support for different equations and conditions, enabling varied simulations.

Code availability section The author has no competing interest to claim. Pascal Tribel and Gianluca Bontempi are affiliated to *TRusted AI Labs* (TRAIL). Gianluca Bontempi is supported by the Service Public de Wallonie Recherche under grant nr. 2010235-ARIAC by *DigitalWallonia4.ai*. Computational resources have been provided by the *Consortium des Equipements de Calcul Intensif* (CECI), funded by the *Fonds de la Recherche Scientifique de Belgique* (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region. The authors would like to thank Corentin Caudron (ULB) for his insightful remarks.

- Name of the Library: PyAWD
- Contact: pascal.tribel@ulb.be
- Hardware requirements: None
- Program language: Python 3.x

The source codes are available for downloading at the link: https://github.com/pascaltribel/pyawd

References

Braile, Larry (Nov. 2002). *The AS-1 Seismograph – Operation, Filtering, S-P Distance Calculation, and Ideas for Classroom Use*. Tech. rep. Explorations in Earth Science, supported by IRIS and the National Science Foundation. Purdue University.

Breiman, L. (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32.

Breiman, L. et al. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth.

Cambrin, Daniele Rege and Paolo Garza (2024). *QuakeSet: A Dataset and Low-Resource Models to Monitor Earthquakes through Sentinel-1*. arXiv: 2403.18116 [cs.CV]. URL: https://arxiv.org/abs/2403.18116.

Carlini, Alessandro, Camille Bordeau, and Maxime Ambard (2024). “Auditory localization: a comprehensive practical review”. In: *Frontiers in Psychology* 15. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2024.1408073.

Chang, Chih-Chung and Chih-Jen Lin (2011). “LIBSVM: A library for support vector machines”. In: *ACM transactions on intelligent systems and technology (TIST)* 2.3, pp. 1–27.

Chen, Tianqi and Carlos Guestrin (2016). “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 785–794. DOI: 10.1145/2939672.2939785.

Christ, Maximilian et al. (2018). “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)”. In: *Neurocomputing* 307, pp. 72–77. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.03.067. URL: https://www.sciencedirect.com/science/article/pii/S0925231218304843.

Demanet, Laurent (2015). *Wave Equations*. Lecture notes. Massachusetts Institute of Technology. Chap. 1.

Geurts, P., D. Ernst, and L. Wehenkel (2006). “Extremely Randomized Trees”. In: *Machine Learning* 63.1, pp. 3–42.

Hinton, Geoffrey E. (1989). "Connectionist Learning Procedures". In: *Artificial Intelligence* 40.1, pp. 185–234.

Lea, Colin et al. (2016). *Temporal Convolutional Networks for Action Segmentation and Detection*.

Lehmann, Fanny et al. (Jan. 2024). "Synthetic ground motions in heterogeneous geologies: the HEMEW-3D dataset for scientific machine learning". In: *Earth System Science Data*. DOI: 10.5194/essd-2023-470.

Louboutin, M. et al. (2019). "Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration". In: *Geoscientific Model Development* 12. DOI: 10.5194/gmd-12-1165-2019.

Luporini, Fabio et al. (Apr. 2020). "Architecture and Performance of Devito, a System for Automated Stencil Computation". In: *ACM Trans. Math. Softw.* 46. DOI: 10.1145/3374916.

maintainers, TorchVision and contributors (2016). *TorchVision: PyTorch's Computer Vision library*. https://github.com/pytorch/vision.

Merrifield, Tom P. et al. (2022). "Synthetic seismic data for training deep learning networks". In: *Interpretation* 10.3, SE31–SE39. DOI: 10.1190/INT-2021-0193.1.

Meurer, Aaron et al. (Jan. 2017). "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3. DOI: 10.7717/peerj-cs.103.

Mousavi, S Mostafa et al. (2019). "STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI". In: *IEEE Access*.

Mousavi, S. Mostafa and Gregory C. Beroza (2023). "Machine Learning in Earthquake Seismology". In: *Annual Review of Earth and Planetary Sciences* 51.Volume 51, 2023, pp. 105–129. ISSN: 1545-4495. DOI: https://doi.org/10.1146/annurev-earth-071822-100323. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-earth-071822-100323.

Mousavi, S.Mostafa et al. (July 2019). "CRED: A Deep Residual Network of Convolutional and Recurrent Units for Earthquake Signal Detection". In: *Scientific Reports* 9. DOI: 10.1038/s41598-019-45748-1.

Noda, Shunta et al. (Mar. 2012). "Improvement of back-azimuth estimation in real-time by using a single station record". In: *Earth Planets and Space* 64, pp. 305–308. DOI: 10.5047/eps.2011.10.005.

Paszke, Adam et al. (2017). "Automatic differentiation in PyTorch". In: *NIPS-W*.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Perol, Thibaut, Michaël Gharbi, and Marine Denolle (2018). "Convolutional neural network for earthquake detection and location". In: *Science Advances* 4. DOI: 10.1126/sciadv.1700578.

Ridzwan, Nurafiqah and Siti Harwani Md Yusoff (Mar. 2023). "Machine learning for earthquake prediction: a review (2017–2021)". In: *Earth Science Informatics* 16. DOI: 10.1007/s12145-023-00991-z.

Ross, Zachary, Men-Andrin Meier, and Egill Hauksson (Apr. 2018). "P Wave Arrival Picking and First-Motion Polarity Determination With Deep Learning". In: *Journal of Geophysical Research: Solid Earth* 123. DOI: 10.1029/2017JB015251.

Saad, Hagar et al. (Sept. 2023). "EQConvMixer: A Deep Learning Approach for Earthquake Location From Single-Station Waveforms". In: *IEEE Geoscience and Remote Sensing Letters* 1, pp. 1–1. DOI: 10.1109/LGRS.2023.3312324.

Türkmen, Melek et al. (May 2024). "Deep Learning-based Epicenter Localization using Single-Station Strong Motion Records". In: *Geophysical Research Letters*.

Versteeg, Roelof (1994). "The Marmousi experience: Velocity model determination on a synthetic complex data set". In: *The Leading Edge* 13.9, pp. 927–936. DOI: 10.1190/1.1437051.

Woollam, Jack et al. (Mar. 2022). "SeisBench—A Toolbox for Machine Learning in Seismology". In: *Seismological Research Letters* 93.3, pp. 1695–1709. ISSN: 0895-0695. DOI: 10.1785/0220210324. eprint: https://pubs.geoscienceworld.org/ssa/srl/article-pdf/93/3/1695/5596371/srl-2021324.1.pdf. URL: https://doi.org/10.1785/0220210324.