The Journal of Open Source Software

Adaptive Resonance Lib: A Python package for Adaptive Resonance Theory (ART) models

Niklas M. Melton [1¶], **Dustin Tanksley** [1], and **Donald C. Wunsch II** [1]

1 Missouri University of Science and Technology, Rolla, Missouri, United States of America ¶ Corresponding author

DOI: 10.21105/joss.07764

Software

- Review
- Repository
- Archive

Editor: Marcel Stimberg

Reviewers:

- @TahiriNadia
- @chenxinye

Submitted: 15 November 2024
Published: 20 October 2025

Summary

The Adaptive Resonance Theory Library (**artlib**) is a Python library that implements a wide range of Adaptive Resonance Theory (ART) algorithms. **artlib** currently supports eight elementary ART models and 12 compound ART models, including Fuzzy ART (Gail A. Carpenter et al., 1991b), Hypersphere ART (Anagnostopoulos & Georgiopulos, 2000), Ellipsoid ART (Anagnostopoulos & Georgiopulos, 2001a, 2001b), Gaussian ART (Williamson, 1996), Bayesian ART (Vigdor & Lerner, 2007), Quadratic Neuron ART (M.-C. Su & Liu, 2001; M. Su & Liu, 2005), ART1 (Gail A. Carpenter & Grossberg, 1987a), ART2 (Gail A. Carpenter et al., 1991a; Gail A. Carpenter & Grossberg, 1987b), ARTMAP (G. A. Carpenter et al., 1991), Simplified ARTMAP (Gotarredona et al., 1998), SMART (Bartfai, 1994), DeepARTMAP (Niklas M. Melton et al., 2025b), TopoART (Tscherepanow, 2010), Dual Vigilance ART (Silva et al., 2019b), CVIART (Silva et al., 2022), BARTMAP (Xu et al., 2012; Xu & II, 2011), Fusion ART (Tan et al., 2007), FALCON (Tan, 2004), and TD-FALCON (Tan et al., 2008). These models can be applied to tasks such as unsupervised clustering, supervised classification, regression, and reinforcement learning (Silva et al., 2019a). This library provides an extensible and modular framework where users can integrate custom models or extend current implementations, allowing for experimentation with existing and novel machine learning techniques.

In addition to the diverse set of ART models, **artlib** offers implementations of visualization methods for various cluster geometries, along with pre-processing techniques such as Visual Assessment of Tendency (VAT; Bezdek & Hathaway, 2002), data normalization, and complement coding.

Adaptive Resonance Theory (ART)

ART is a class of neural networks known for solving the stability-plasticity dilemma, making it particularly effective for classification, clustering, and incremental learning tasks (Grossberg, 1976, 1976, 1980, 2013; Silva et al., 2019a). ART models are designed to dynamically learn and adapt to new patterns without catastrophic forgetting, making them ideal for real-time systems requiring continuous learning.

Over the years, dozens of ART variations have been published (Silva et al., 2019a), extending the applicability of ART to nearly all learning regimes, including reinforcement learning (Tan, 2004; Tan et al., 2008), hierarchical clustering (Bartfai, 1994), topological clustering (Tscherepanow, 2010), and biclustering (Xu et al., 2012; Xu & II, 2011). These numerous models provide an ART-based solution for most machine learning use cases. However, the rapid pace of bespoke model development, coupled with the challenges students face in learning ART's foundational principles, has contributed to a scarcity of open-source, user-friendly implementations for most

ART variants.

The ability of ART to preserve previously learned patterns while learning new data in real-time has made it a powerful tool in domains such as robotics, medical diagnosis, and adaptive control systems. **artlib** aims to extend the application of these models in modern machine learning pipelines, offering a unique and approachable toolkit for leveraging ART's strengths.

Statement of Need

The Adaptive Resonance Library (**artlib**) is essential for researchers, developers, and educators interested in adaptive neural networks, specifically ART algorithms. While deep learning dominates machine learning, ART models offer unique advantages in incremental and real-time learning environments due to their ability to learn new data without forgetting previously learned information.

Currently, no comprehensive Python library implements a variety of ART models in an open-source, modular, and extensible manner. **artlib** fills this gap by offering a range of ART implementations that integrate seamlessly with machine learning workflows, including scikit-learn's `Pipeline` and `GridSearchCV` (Pedregosa et al., 2011). The library is designed for ease of use and high performance, with all modules leveraging Python's scientific stack: i.e. NumPy (C. R. Harris et al., 2020), SciPy (Virtanen et al., 2020), and scikit-learn (Pedregosa et al., 2011), a subset of modules also provide C++ and Torch implementations for even faster numerical computation.

The modular design of **artlib** enables users to create novel compound ART models, such as Dual Vigilance Fusion ART (Silva et al., 2019b; Tan et al., 2007) or Quadratic Neuron SMART (Bartfai, 1994; M.-C. Su & Liu, 2001; M. Su & Liu, 2005). This flexibility offers powerful experimental and time-saving benefits, allowing researchers and practitioners to evaluate models on diverse datasets efficiently.

Additionally, the library serves as a valuable educational tool, providing well-documented code and familiar APIs to support hands-on experimentation with ART models. It is ideal for academic courses, personal projects, and research in artificial intelligence and machine learning, making **artlib** a versatile resource.

artlib is actively maintained and designed for future extension, allowing users to create new ART models, adjust parameters for specific applications, and explore ART's potential for novel research problems. Its integration with popular Python libraries ensures its adaptability to current machine learning challenges.

Comparison to Existing Implementations

While there are several open-source repositories that provide Python implementations of specific ART models (AI OpenLab, 2023; Birkjohann, 2023; Dilekman, 2022; Dixit, 2020; Inyoot, 2021; Ray11641, 2023; Stepanov, 2022; Valixandra, 2021; Wan, 2022), they lack modularity and are limited in scope, often implementing just one or two models. For instance, MATLAB-based ART toolboxes (Boitet, 2022; M. Harris, 2000; Schwenker, 2004b, 2004a) provide implementations of Fuzzy ART, TopoART, ART1, and ARTMAP models, but they lack the flexibility and modularity required for broader experimentation. The most significant existing ART implementation exists in Julia and provides just five models (Petrenko & Wunsch, 2022) but, like the previously listed MATLAB-based toolboxes, it is not easily accessible to Python-based work flows and lacks a modular design. Further, no public ART implementations provide configurable match-tracking (Niklas M. Melton et al., 2025a).

These existing implementations of ART models may provide standalone versions of individual models, but they are often not designed to integrate seamlessly with modern Python libraries such as scikit-learn, NumPy, and SciPy. As a result, researchers and developers working in Python-based environments face challenges when trying to incorporate ART models into their machine learning pipelines.

In contrast, **artlib** offers a comprehensive and modular collection of ART models, including both elementary and compound ART architectures. It is designed for interoperability with popular Python tools, enabling users to easily integrate ART models into machine learning workflows, optimize models using scikit-learn's `GridSearchCV`, and preprocess data using standard libraries. Further, **artlib** provides users the flexibility to construct their own compound ART modules (those ART modules deriving properties from other, elementary modules) which may or may not exist in published literature. **artlib** also provides a template in the source code to encourage users to develop and experiment with their own custom ART algorithms. This flexibility and integration make **artlib** a powerful resource for both research and practical applications.

Acknowledgements

This research was supported by the National Science Foundation (NSF) under Award Number 2420248. The project titled EAGER: Solving Representation Learning and Catastrophic Forgetting with Adaptive Resonance Theory provided essential funding for the completion of this work.

We would also like to thank Gail Carpenter and Stephen Grossberg for their feedback regarding this project and their immense contribution to machine learning by pioneering Adaptive Resonance Theory.

References

AI OpenLab. (2023). ART: Adaptive Resonance Theory implementation. In *GitHub repository* (Version v1.0.0). GitHub. https://github.com/AIOpenLab/art

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001a). Ellipsoid ART and ARTMAP for incremental clustering and classification. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, *2*, 1221–1226. https://doi.org/10.1109/IJCNN.2001.939535

Anagnostopoulos, G. C., & Georgiopoulos, M. (2001b). Ellipsoid ART and ARTMAP for incremental unsupervised and supervised learning. *Applications and Science of Computational Intelligence IV*, *4390*, 293–304. https://doi.org/10.1117/12.421180

Anagnostopoulos, G. C., & Georgiopulos, M. (2000). Hypersphere ART and ARTMAP for unsupervised and supervised, incremental learning. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, *6*, 59–64. https://doi.org/10.1109/IJCNN.2000.859373

Bartfai, G. (1994). Hierarchical clustering with ART neural networks. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, *2*, 940–944. https://doi.org/10.1109/ICNN.1994.374307

Bezdek, J. C., & Hathaway, R. J. (2002). VAT: A tool for visual assessment of (cluster) tendency. *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, *3*, 2225–2230 vol.3. https://doi.org/10.1109/IJCNN.2002.1007487

Birkjohann, C. (2023). Art-python: Adaptive Resonance Theory in Python. In *GitHub repository* (Version v0.1.0). GitHub. https://github.com/cbirkj/art-python

Boitet, M. (2022). TopoART neural networks. In *MATLAB Central File Exchange*. Mathworks. https://www.mathworks.com/matlabcentral/fileexchange/

118455-topoart-neural-networks

Carpenter, Gail A., & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, *37*(1), 54–115. https://doi.org/10.1016/S0734-189X(87)80014-2

Carpenter, Gail A., & Grossberg, S. (1987b). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, *26*(23), 4919–4930. https://doi.org/10.1364/AO.26.004919

Carpenter, G. A., Grossberg, S., & Reynolds, J. H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *[1991 Proceedings] IEEE Conference on Neural Networks for Ocean Engineering*, 341–342. https://doi.org/10.1109/ICNN.1991.163370

Carpenter, Gail A., Grossberg, S., & Rosen, D. B. (1991a). ART 2-a: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, *4*(4), 493–504. https://doi.org/10.1016/0893-6080(91)90045-7

Carpenter, Gail A., Grossberg, S., & Rosen, D. B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, *4*(6), 759–771. https://doi.org/10.1016/0893-6080(91)90056-B

Dilekman, K. (2022). Artificial neural network - Adaptive Resonance Theory. In *GitHub repository* (Version v0.1.0). GitHub. https://github.com/KeremDlkmn/artificial-neural-network-adaptive-resonance-theory

Dixit, M. (2020). Adaptive Resonance Theory. In *GitHub repository* (Version v1.0.0). GitHub. https://github.com/MeetiDixit/AdaptiveResonanceTheory

Gotarredona, T. S., Barranco, B. L., & Andreou, A. (1998). *Adaptive Resonance Theory microchips circuit design techniques*. Norwell, MA: Kluwer. https://doi.org/10.1007/978-1-4419-8710-5

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*(3), 121–134. https://doi.org/10.1007/978-94-009-7758-7_12

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, *87 1*, 1–51. https://doi.org/10.1007/978-94-009-7758-7_1

Grossberg, S. (2013). Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*, *37*, 1–47. https://doi.org/10.1016/j.neunet.2012.09.017

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Harris, M. (2000). ART1S: Adaptive Resonance Theory 1 (ART1). In *MATLAB Central File Exchange*. Mathworks. https://www.mathworks.com/matlabcentral/fileexchange/93-art1s-zip

Inyoot. (2021). ART: Adaptive Resonance Theory implementation in Python. In *GitHub repository* (Version v0.1.0). GitHub. https://github.com/inyoot/art

Melton, Niklas M., Brito da Silva, L. E., & Wunsch, D. C. (2025a). An extensive analysis of match-tracking methods for ARTMAP. *2025 IEEE Symposium on Computational Intelligence in Health and Medicine (CIHM)*. https://doi.org/10.1109/CIHM64979.2025.10969482

Melton, Niklas M., Silva, L. E. B. da, Petrenko, S., Wunsch II, D., & others. (2025b). Deep ARTMAP: Generalized hierarchical learning with adaptive resonance theory. *arXiv Preprint arXiv:2503.07641*.

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

Petrenko, S., & Wunsch, D., II. (2022). AdaptiveResonance.jl: A Julia implementation of Adaptive Resonance Theory (ART) algorithms. *Journal of Open Source Software*, *7*. https://doi.org/10.21105/joss.03671

Ray11641. (2023). Artpy: Adaptive Resonance Theory in Python. In *GitHub repository* (Version v0.1.0). GitHub. https://github.com/Ray11641/artpy

Schwenker, F. (2004a). ART1, fuzzy ART, ARTMAP, fuzzy ARTMAP neural networks. In *MATLAB Central File Exchange*. Mathworks. https://www.mathworks.com/matlabcentral/fileexchange/3070-art1-fuzzyart-artmap-fuzzyartmap

Schwenker, F. (2004b). Fuzzy ART and fuzzy ARTMAP neural networks. In *MATLAB Central File Exchange*. Mathworks. https://www.mathworks.com/matlabcentral/fileexchange/4306-fuzzy-art-and-fuzzy-artmap-neural-networks

Silva, L. E. B. da, Elnabarawy, I., & Wunsch II, D. C. (2019a). A survey of Adaptive Resonance Theory neural network models for engineering applications. *Neural Networks*, *120*, 167–203. https://doi.org/10.1016/j.neunet.2019.09.012

Silva, L. E. B. da, Elnabarawy, I., & Wunsch II, D. C. (2019b). Dual vigilance fuzzy Adaptive Resonance Theory. *Neural Networks*, *109*, 1–5. https://doi.org/10.1016/j.neunet.2018.09.015

Silva, L. E. B. da, Rayapati, N., & Wunsch, D. C. (2022). iCVI-ARTMAP: Using incremental cluster validity indices and Adaptive Resonance Theory reset mechanism to accelerate validation and achieve multiprototype unsupervised representations. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(12), 9757–9770. https://doi.org/10.1109/TNNLS.2022.3160381

Stepanov, S. (2022). ARTpy: A Python implementation of Adaptive Resonance Theory. In *GitHub repository* (Version v1.0.0). GitHub. https://github.com/DeadAt0m/ARTpy

Su, M.-C., & Liu, T.-K. (2001). Application of neural networks using quadratic junctions in cluster analysis. *Neurocomputing*, *37*(1-4), 165–175. https://doi.org/10.1016/S0925-2312(00)00343-X

Su, M., & Liu, Y. (2005). A new approach to clustering data with arbitrary shapes. *Pattern Recognition*, *38*(11), 1887–1901. https://doi.org/10.1016/j.patcog.2005.04.010

Tan, A.-H. (2004). FALCON: A fusion architecture for learning, cognition, and navigation. *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, *4*, 3297–3302. https://doi.org/10.1109/IJCNN.2004.1381208

Tan, A.-H., Carpenter, G. A., & Grossberg, S. (2007). Intelligence through interaction: Towards a unified theory for learning. *International Symposium on Neural Networks*, 1094–1103. https://doi.org/10.1007/978-3-540-72383-7_128

Tan, A.-H., Lu, N., & Xiao, D. (2008). Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. *IEEE Transactions on Neural Networks*, *19*(2), 230–244. https://doi.org/10.1109/TNN.2007.905839

Tscherepanow, M. (2010). TopoART: A topology learning hierarchical ART network. *Inter-*

national Conference on Artificial Neural Networks, 157–167. https://doi.org/10.1007/978-3-642-15825-4_21

Valixandra, F. (2021). Adaptive Resonance Theory. In *GitHub repository* (Version v1.0.0). GitHub. https://github.com/flppvalxndra/Adaptive-Resonance-Theory

Vigdor, B., & Lerner, B. (2007). The Bayesian ARTMAP. *IEEE Transactions on Neural Networks*, *18*(6), 1628–1644. https://doi.org/10.1109/TNN.2007.900234

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wan, Y. (2022). ART2: Adaptive Resonance Theory 2 implementation. In *GitHub repository* (Version v1.0.0). GitHub. https://github.com/YuChangWan/ART2

Williamson, J. R. (1996). Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, *9*(5), 881–897. https://doi.org/10.1016/0893-6080(95)00115-8

Xu, R., & II, D. C. W. (2011). BARTMAP: A viable structure for biclustering. *Neural Networks*, *24*, 709–716. https://doi.org/10.1016/j.neunet.2011.03.020

Xu, R., II, D. C. W., & Kim, S. Y. (2012). *Methods and systems for biclustering algorithm* (Patent No. US9,043,326B2). Filed January 28, 2012, issued May 26, 2015. https://patents.google.com/patent/US9043326B2/en