chatter: a Python library for applying information theory and AI/ML models to animal communication

Mason Youngblood

Institute for Advanced Computational Science, Stony Brook University, USA
masonyoungblood@gmail.com

Software

- GitHub
- Documentation
- PyPI

Date: 12 December 2025

Summary

The study of animal communication often involves categorizing units into types (e.g. syllables in songbirds, or notes in humpback whales). While this approach is useful in many cases, it necessarily flattens the complexity and nuance present in real communication systems. `chatter` is a new Python library for analyzing animal communication in continuous latent space using information theory and modern machine learning techniques. It is taxonomically agnostic, and has been tested with the vocalizations of birds, bats, whales, and primates. By leveraging a variety of different architectures, including variational autoencoders and vision transformers, `chatter` represents vocal sequences as trajectories in high-dimensional latent space, bypassing the need for manual or automatic categorization of units. The library provides an end-to-end workflow—from preprocessing and segmentation to model training and feature extraction—that enables researchers to quantify the complexity, predictability, similarity, and novelty of vocal sequences.

Statement of Need

In recent years, animal behaviorists have started to use machine learning to project vocalizations into high-dimensional latent space (Goffinet et al., 2021; Merino Recalde, 2023; Sainburg et al., 2020). However, these methods are typically used to identify categories of vocalizations instead of enabling analysis of vocal sequences as continuous vectors, with some exceptions (Alam et al., 2024). `chatter` addresses this gap by providing an accessible, modular, and easy-to-use interface for applying these techniques. It complements existing tools like `pykanto` (Merino Recalde, 2023) and `AVGN` (Sainburg et al., 2020), which focus on discrete analysis, by offering a parallel workflow for continuous analysis. In doing so, it lowers the barrier to entry for researchers who are interested in computational methods but may not have the time or expertise to assemble a machine learning pipeline from scratch. The library is designed to be flexible enough for advanced users while providing sensible defaults for novices.

The core workflow of `chatter` has three steps, all managed by a configuration dictionary. First, users initialize the `Analyzer` class, which takes parameters like minimum and maximum frequency, amplitude thresholds, and denoising settings. The `Analyzer` handles the raw audio, preprocessing it with noise reduction, filtering, and compression, and segmenting it into individual acoustic units based on amplitude or other criteria. It also gives users working with birdsong data the ability to identify their focal species using BirdNET (Kahl et al., 2021). The resulting units are converted into spectrograms.

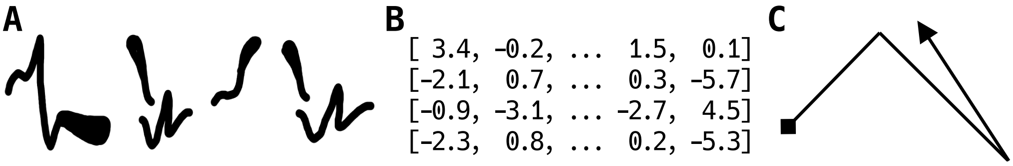

Figure 1: A basic diagram of the `chatter` workflow, showing the progression from spectrograms to latent features to visualizations in 2D space. Note that all of the information theoretic analysis occurs in original latent space, not in the reduced 2D space.

Next, the `Trainer` class is used to train a convolutional variational autoencoder to learn a high-dimensional latent representation of the vocalizations (Goffinet et al., 2021). This unsupervised approach is able to describe complex vocalizations in a relatively lossless way—allowing users to reconstruct the original sound from a set of learned acoustic features. These latent features are then extracted for all acoustic units. Users also have the option of collecting latent features by applying the DINOv3 vision transformer to spectrograms, which works especially well when treating sequences as units (i.e. working with spectrograms of whole sequences rather than units within sequences) (Siméoni et al., 2025). Finally, the library provides tools to visualize this latent space using dimensionality reduction techniques such as PaCMAP (Wang et al., 2021), an extension of UMAP that better captures global structure. Figure 2 illustrates this output for Cassin's vireo syllables (recordings from Hedley (2016)).

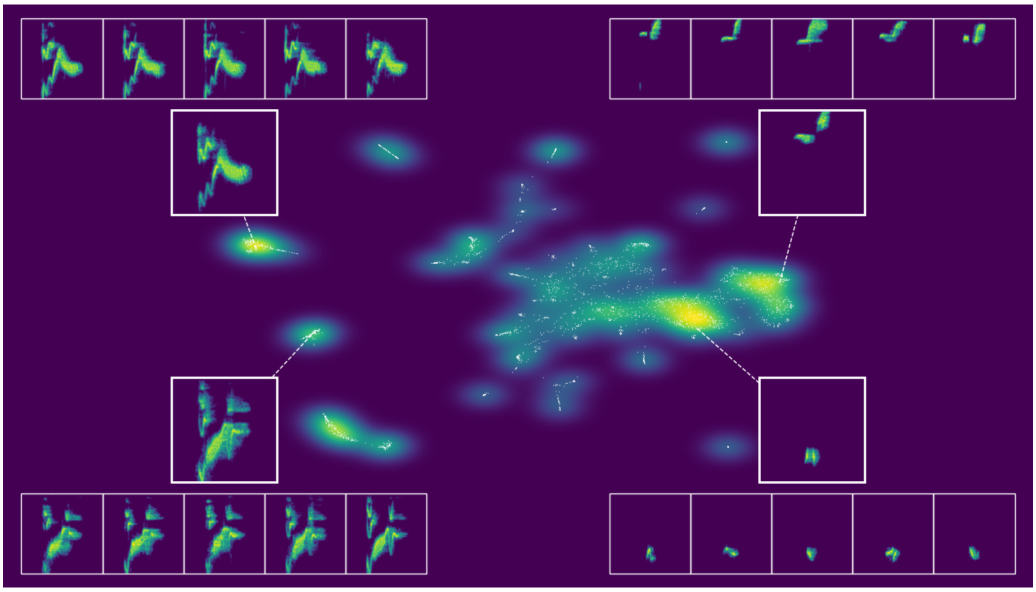

Figure 2: The latent space of Cassin's vireo syllables. The plot visualizes the syllables in a 2D latent space produced by applying PaCMAP to the latent features from a variational autoencoder, with representative spectrograms overlaid.

Finally, the `FeatureProcessor` class provides a variety of functions for downstream analysis. Complexity is quantified by calculating the path length (cosine distance) between consecutive acoustic units in latent space, normalized by duration, offering a continuous measure of acoustic diversity over time (Alam et al., 2024) similar to previous methods based on spectrogram cross-correlation (Sawant et al., 2022). To assess predictability, `chatter` trains a vector autoregression model on sequences of latent features (Sims, 1980). This model learns temporal dependencies and quantifies the probability of future states.

`chatter` also estimates the commonness and rarity of acoustic units using `denmarf`, a density estimation library that uses masked autoregressive flows to compute density in high-dimensional space (Papamakarios et al., 2017). This provides a measure of how common or rare a given vocalization is relative to the learned distribution. Finally, similarity between whole sequences of units is computed by applying dynamic time warping to the sequences of latent vectors (i.e. panel B in Figure 1). Collectively, these measures provide a fairly complex and nuanced description of animal vocal sequences.

References

Alam, D., Zia, F., & Roberts, T. F. (2024). The hidden fitness of the male zebra finch courtship song. *Nature*, *627*, 851–857. https://doi.org/10.1038/s41586-024-07207-4

Goffinet, J., Brudner, S., Mooney, R., & Pearson, J. (2021). Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires. *eLife*, *10*, e67855. https://doi.org/10.7554/eLife.67855

Hedley, R. W. (2016). Complexity, predictability, and time homogeneity of syntax in the songs of cassin's vireo. *PLOS ONE*, *11*(3), e0150822. https://doi.org/10.1371/journal.pone.0150822

Kahl, S., Wood, C. M., Eibl, M., & Klinck, H. (2021). BirdNET: A deep learning solution for avian diversity monitoring. *Ecological Informatics*, *61*, 101236. https://doi.org/10.1016/j.ecoinf.2021.101236

Merino Recalde, N. (2023). Pykanto: A python library to accelerate research on wild bird song. *Methods in Ecology and Evolution*, *14*(6), 1639–1648. https://doi.org/10.1111/2041-210X.14155

Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked autoregressive flow for density estimation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf

Sainburg, T., Thielk, M., & Gentner, T. Q. (2020). Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLOS Computational Biology*, *16*(10), e1008228. https://doi.org/10.1371/journal.pcbi.1008228

Sawant, S., Arvind, C., Joshi, V., & Robin, V. V. (2022). Spectrogram cross-correlation can be used to measure the complexity of bird vocalizations. *Methods in Ecology and Evolution*, *13*(2), 459–472. https://doi.org/10.1111/2041-210X.13765

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., … Bojanowski, P. (2025). *DINOv3*. https://arxiv.org/abs/2508.10104

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica*, *48*(1), 1–48. https://doi.org/10.2307/1912017

Wang, Y., Huang, H., Rudin, C., & Shaposhnik, Y. (2021). Understanding how dimension reduction tools work: An empirical approach to ceciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization. *Journal of Machine Learning Research*, *22*(201), 1–73. http://jmlr.org/papers/v22/20-1061.html