USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Voting-Bloc Entropy: A New Metric for DAO Decentralization

Andres Fabrega, *Cornell University;* Amy Zhao, *IC3;* Jay Yu, *Stanford University;* James Austgen, *Cornell Tech;* Sarah Allen, *IC3 and Flashbots;* Kushal Babel, *Cornell Tech and IC3;* Mahimna Kelkar, *Cornell Tech;* Ari Juels, *Cornell Tech and IC3*

https://www.usenix.org/conference/usenixsecurity25/presentation/fabrega-entropy

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.

USENIX Security '25 Artifact Appendix: Voting-Bloc Entropy: A New Metric for DAO Decentralization

Andrés Fábrega[1], Amy Zhao[3], Jay Yu[5], James Austgen[2], Sarah Allen[3,4], Kushal Babel[2,3], Mahimna Kelkar[2], Ari Juels[2,3]

[1] Cornell University [2] Cornell Tech [3] IC3 [4] Flashbots [5] Stanford University

A Artifact Appendix

This appendix describes how to use the artifacts included in *Voting-Bloc Entropy: A New Metric for DAO Decentralization*. The core contribution of our paper is Voting-Bloc Entropy (VBE), which is a new framework for computing the degree of decentralization in DAOs. At a high-level, VBE consists of two main steps:

1. Computing a *clustering function* over the utility functions of the members of the DAO. This step results in groups—or *blocs*—of DAO members with aligned incentives and interests, as determined by the similarity of their utility functions.

2. Computing an *entropy metric* over the distribution of voting power (i.e., DAO tokens) across the resulting blocs.

Since VBE is a framework, it can be instantiated with any clustering function and entropy metric.

Note that utility functions are so-called latent variables, which cannot be directly measured. So, to compute VBE in practice, we need to use some on-chain, observable data to estimate the utility functions of the members of the DAO, such as voting history. This leads to the "observable" VBE (oVBE) framework, which computes the clustering step using some relevant dataset that represents utility functions.

A.1 Abstract

The main component of our artifact is the *oVBE toolkit*, which is a library for computing various instantiations of oVBE, supporting some of the most popular clustering algorithms (e.g., K-means, hierarchical clustering, DBSCAN, etc) and entropy functions (e.g., min-entropy, Shannon entropy, etc). Using our library, users can compute the oVBE of a DAO, and understand its degree of decentralization or perform VBE-based governance experiments.

At a high-level, to use our toolkit, users simply input a data set of observable data, and select appropriate parameters for clustering and entropy. While in principle our toolkit is compatible with any dataset, most users will probably be interested in voting history as the input data. As such, our artifact also includes scripts for easily scraping the voting history of DAOs from the blockchain, as well as for preparing it in a format that is compatible with the rest of the VBE toolkit.

In the rest of this appendix, we describe the general use of our artifact, as well as how to use it specifically to reproduce the experiments of our paper.

A.2 Description & Requirements

Our artifact relies on fairly standard software dependencies, which can run on typical consumer laptops. We summarize the requirements in this section.

A.2.1 Security, privacy, and ethical concerns

There are no security or data privacy risks associated with evaluating our artifact. In particular, we have provided API keys for the reviewers, so that they do not need to provide personal information to register with the services that our toolkit leverages.

A.2.2 How to access

The artifact stable repository can be found at `https://zenodo.org/records/14675832?token=eyJhbGciOiJIUzUxMiJ9.eyJpZCI6ImY3ODk0ZGF%5B%E2%80%A6%5DIQ1CNdjCqd1D4NC7ldSWtrYyVog-sJusoe9Ma2tfu82NEJv2sJo4UO8bpOBcpEw`.

Note that, as described further below, API keys for Tally and Snapshot are necessary to use some parts of our artifact.

A.2.3 Hardware dependencies

Our artifact does not require any special hardware features.

A.2.4 Software dependencies

Our artifact requires Python 3.11 (at least), and installation of the Python libraries specified in file `https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/requirements.txt`.

We tested our artifact on MacOS and Ubuntu, but believe it should run on any platform that can install the aforementioned Python libraries. If using Ubuntu, additional dependencies need to be installed running "sudo apt install python3-venv libpq-dev", as described in https://github.com/DAO-Decentralization/VBE/tree/main/VBE-data/README.md.

A.2.5 Benchmarks

To reproduce our experiments, we have made available our data in the form of a read-only database, the details of which can be found at https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/.env.example.

A.3 Set-up

A.3.1 Installation

To install our artifact, follow the instructions in https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/README.md followed by the instructions in https://github.com/DAO-Decentralization/VBE/blob/main/VBE-library/README.md.

A.3.2 Basic Test

Using the example included in https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/data_setup/dao_input.csv, run "python tally_api.py" or "python snapshot_api.py". This should successfully output files into VBE-data/data_output/ folder if selecting the option to save as CSV. This tests the basic functionality of our data processing scripts

Using the resulting data output, one can then test the oVBE toolking by following the instrictions in https://github.com/DAO-Decentralization/VBE/tree/main/VBE-library to run either vbe_parameters.py or run_vbe.py for a single or full instance of calculations.

A.4 Evaluation workflow

This section describes how to use and evaluate our artifact. The high-level workflow of our artifact proceeds as follows:

- First, use the scripts in the VBE-data directory to acquire governance data for DAOs of interest, which will be the input to the VBE library. This data will be scraped using Tally and Snapshot, for which API keys are required. The scrapped data will be saved either in a database endpoint specified by the user, or output as a CSV file. Note that this step is optional, as users that already have governance data can directly use the oVBE toolkit.
- Use the scripts in VBE-library to compute oVBE based on some input data (which can potentially be the output of the prior step). In particular, to reproduce the results of our paper, we have made available a read-only database with the data used by our results.

A.4.1 Major Claims

The results in our paper that rely on our artifact are the following:

1. Average minimum entropy VBE values for DAOs listed as follows in Figure 1: Arbitrum (0.9254), Optimism (0.9929), Gitcoin (0.8536), Nouns DAO (0.8806), and Uniswap (0.7435). This is proven in Experiment 4 (E4) where VBE is calculated using default parameters across all organizations and proposals in windows.

2. Voter participation, Nakamoto Coefficient and Gini Index from Figure 1 are from Experiment 3 (E3) to aggregate proposal and DAO-level information.

A.4.2 Experiments

Experiment #1: pull Snapshot data. This step is used to pull information from the Snapshot API for all DAOs listed in the input file.

1. Preparation: Steps for dependencies and configuration can be found in https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/README.md. Make sure to have a Snapshot API key in the .env file.

2. Execution: run "python snapshot_api.py", specify whether to save locally or to a database, and view outcomes in the database tables or in data_output/ folder.

3. Results: Depending on the data_input.csv specified, expect to see the corresponding data for DAOs, proposals, and voter information.

Experiment #2: pull Tally data. This step is used to pull information from the Tally API for all DAOs listed in the input file.

1. Preparation: Steps for dependencies and configuration can be found in https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/README.md. Make sure to have a Tally API key in the .env file.

2. Execution: run "python tally_api.py", specify whether to save locally or to a database, and view outcomes in the database tables or in data_output/ folder.

3. Results: Depending on the data_input.csv specified, expect to see the corresponding data for DAOs, Proposals, and Voter information.

Experiment #3: aggregate data analytics. This step is used to pull information from database or CSVs and turn them into aggregate level statistics

1. Preparation: As a prerequisite, have data pulled from Snapshot or Tally as per the two experiments above.
2. Execution: run "python data_analytics.py", specify whether to save locally or to a database, and view aggregated data in the database tables or in data_output/ folder.
3. Results: The aggregate data includes information like the voter participation rate, number of proposals, percentiles for voting power, and voter choices.

Experiment #4: compute oVBE. This is the main experiment in our artifact, which computes oVBE on an input database or CSV file.

1. Preparation: Follow steps outlined in the VBE-library README file. If using a database connection, make sure it is properly configured in the .env file.
2. Execution: run python run_vbe.py, specify whether to save locally or to a database, and view outcomes in the database tables or in data_output/ folder.
3. Results: Results are output to the data_output/ folder, where the main information for VBE is saved to dao_vbe.csv or the dao_vbe table in the database.

The database endpoint containing the data used for our experiments (which can be specified in the .env file) can be found in https://github.com/DAO-Decentralization/VBE/blob/main/VBE-data/.env.example.

A.5 Notes on Reusability

Edit the run_vbe.py to have a different model, different parameters. Can edit load_data.py to accommodate different input data, and featurize information for clustering and VBE calculation. Add more scripts in the future (beyond Snapshot and Tally) to bring in other sources for governance data. Use vbe_parameters.py to run a wide range of clustering and entropy calculations - this (and utils.py) can be configured in the future to include more options not listed. Rds_readonly.py can also be used as a starting point to read and manipulate data in the database. If needed, custom queries can be written to extract specific information.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.