Online Resource 1 - Reproduction package

Article: Bounded affine barycentric-coordinate prediction for improving local extreme value forecasting

Journal: Nonlinear Dynamics

Authors: Naoki Fujiwara; Yoshito Hirata; Tohru Kawabe

Corresponding author: Naoki Fujiwara (s2620660@u.tsukuba.ac.jp)

Corresponding author affiliation: Degree Programs in Systems and Information Engineering, Graduate School of Science and Technology, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

Package version: 1.0 (30 October 2026)

Lyapunov-method revision: 1 November 2026 (finite-lag diagnostic and documentation)

Supplementary file: ESM_1.zip; extract it and work inside `Program/`.

This supplement contains the Python source code, fixed synthetic inputs, parameter settings, and compact numerical references for the article above. It is intended for publication alongside the article as Online Resource 1. `README.pdf` is the PDF version of this guide. `FILE_INDEX.md` describes the individual scripts; `METADATA.json` identifies this supplementary archive and its contents. No external code repository or separate code DOI is needed to run the distributed package.

Contents

- `code/`: prediction, evaluation, data generation, embedding selection, and checks.
- `config/`: frozen selection protocol and selected parameters.
- `data/toy/`: the exact synthetic inputs used in the experiments (seeds 0–99).
- `data/toy_extrapolation/`: input for the single-series extrapolation example.
- `reference_results/`: compact numerical reference values from the manuscript.
- `verification/file_manifest.json`: SHA-256 checksums of the distributed files.

The metadata at the start of this guide identifies the whole archive, including its numerical CSV files and frozen JSON configuration. Numerical input files retain their original formats and contents for reproducibility.

Flood observations are not distributed. See data/flood_dawson2005/README.md. Outputs are generated in `results/`; no virtual environment, cache, historical logs, saved plots, or manuscript-dependent figure assembly tools are included.

Setup and quick verification

Python 3.12 was used. The pinned dependency versions in `requirements.txt` were verified with Python 3.12.5 on macOS. Run the following from `Program/` after extracting the archive:

```
python3.12 -m venv .venv
source .venv/bin/activate
python -m pip install -r requirements.txt
python code/verify_submission.py
```

On Windows, activate the environment with `.venv\Scripts\activate` in Command Prompt or `.venv\Scripts\Activate.ps1` in PowerShell. The experiment commands are otherwise the same. These platforms have not been separately verified.

The check verifies file hashes and Python syntax, bounded affine extrapolation without clipping, rejection of non-finite predictions, the finite-lag neighborhood averaging, training-only input, and the two reference window lengths. It does not rerun all experiments or guarantee bit-for-bit agreement on every platform.

Toy-model prediction experiments

```
# Smoke run: one series in each of the eight conditions.
python code/run_selected_toy.py --include-exploratory --num-runs 1 --output-root results/smoke

# Manuscript settings: 100 series per condition.
python code/run_selected_toy.py --include-exploratory --output-root results/reproduction
```

Each condition runs direct and recursive prediction, with all-point and local-extreme evaluation, for weight bounds `[0,1]`, `[-1,2]`, and `[-2,3]`. `--grouprossler_0p` selects one condition. Without `--include-exploratory`, only the four Rössler conditions selected by FNS are run; Lorenz'63 uses the exploratory operational dimension reported in the manuscript.

System	Observation noise	m	tau	Evaluation samples
Rössler	clean	4	13	first 485
Rössler	0.1%, 0.5%, 1%	4	14	first 485
Lorenz'63	clean, 0.1%, 0.5%, 1%	3	16	first 105

Each input has 4500 training and 500 test samples. Noise experiments use the corresponding clean series and true test extrema for evaluation. Improvement is computed per series as `(MAE_base-MAE_new)/MAE_base*100`, followed by its mean and median across series. A one-series smoke run is not the published 100-series result. Recursive predictions are not clipped to the training range.

Extrapolation example

```
python code/extrapolation/reproduce_gaisou.py
```

Defaults are `m=4`, `tau=13`, `k=5`, and recursive 10-step prediction. The four target-point values are in reference_results/extrapolation_values.csv.

Parameter selection and supporting analyses

```
# Training-only MI/FNS analysis without flood observations; computationally expensive.
python code/delay_embedding_study.py --toy-only --config config/delay_embedding_study.json
```

Omit `--toy-only` only after supplying the separately obtained flood inputs in `data/flood_dawson2005/`. Selection uses training data only. Frozen selections are in `config/selected_parameters.json`; FNS did not select a dimension for Lorenz'63 or flood data in `m=1,...,12`.

`code/verify_fnn_embedding.py` provides supplementary conventional FNN analysis (see `--help` for settings). Its defaults are not the complete supplementary analysis reported in the manuscript. The `generate_*.py` scripts generate new synthetic data in `data/generated_toy/`. Use the distributed fixed inputs to reproduce the published comparisons: regenerating chaotic trajectories can produce different numerical values.

Finite-lag divergence diagnostic and evaluation windows

```
python code/lyapunov_01_Final_Lyapunov.py --system both
```

The historical entry-point filename is retained. The implementation is now `code/lyapunov_evaluation.py`, and all system settings are in `config/lyapunov_parameters.json`. The default without `--system` is Rössler. `--output-dir` or `LYAPUNOV_OUTPUT_DIR` sets the output directory. Use `--input` or `LYAPUNOV_INPUT` only with one explicitly selected system.

Only the first 4500 samples of each clean seed-0 CSV are used; these files have no header. Both neighbors and reference points must have successors through lag 50 within that training interval. For each reference point, Euclidean successor distances are averaged over its neighbors before taking the logarithm; the resulting logarithms are averaged equally over reference points. The slope is fitted over actual lags 6 through 15, inclusive.

System	Diagnostic m	Diagnostic tau	Radius epsilon	Theiler window
Rössler	3	6	0.14	10
Lorenz'63	3	3	0.08	10

System	Fitted slope per sample	Unrounded window	Rounded samples
Rössler	0.02372127	485.3418	485
Lorenz'63	0.10925145	105.3801	105

These are diagnostic embedding settings, separate from the prediction embeddings listed above. The window is `log(Delta/delta0)/lambda_fit`, with `delta0=1e-10`, `Delta=1e-5`, rounded by `floor(window+0.5)`. These scale parameters are not calibrated forecast errors. The prediction embeddings, forecasts, evaluation sample counts, and improvement-rate references were not changed by this method revision.

The diagnostic settings were selected retrospectively to keep the existing evaluation windows numerically consistent. The search used clean seed-0 training data only. It considered 1344 divergence curves and 87,360 fits, followed by a radius-only refinement. Numerical proximity to the previous values does not establish a parameter-invariant maximal Lyapunov exponent. The fitted slopes vary with embedding, radius, regression interval, and realization. Across the 100 fixed training series, their medians are 0.02469249 and 0.09654364, respectively. On seed 0, shifting the fit interval by one sample changes the slopes by up to about 7.7% and 8.5%; later-lag fits have smaller slopes. Treat 485 and 105 as fixed evaluation windows, not as guarantees of accurate forecasting.

The complete search and supporting checks can be regenerated from `Program/`:

```
python code/lyapunov_diagnostics/search_parameters.py
python code/lyapunov_diagnostics/validate_candidates.py
python code/lyapunov_diagnostics/validate_all_series.py
```

The last command applies the two fixed compatibility candidates to all 100 training series per system without retuning. These are exponent-diagnostic checks, not reruns of the prediction experiments. Search protocols are in `config/lyapunov_search_protocol.json` and `config/lyapunov_all_series_protocol.json`; compact references are `reference_results/lyapunov_100_series.csv`, `reference_results/lyapunov_100_series_summary.json`, `reference_results/lyapunov_sensitivity.csv`, and `reference_results/lyapunov_later_lags.csv`. Search outputs go to `results/lyapunov_diagnostics/`; the environment variable `LYAPUNOV_DIAGNOSTICS_OUTPUT_DIR` overrides that location. The search script also checks the neighborhood averaging against a direct implementation. The revision was checked with Python 3.12.5, NumPy 2.5.3, and SciPy 1.18.1; this is different from the original pinned verification environment above.

Flood benchmark

```
python code/real_data_flood/flood_recursive_hstep.py --data-dir /path/to/authorised/flood_csvs --output-dir
    results/flood
```

This is the exploratory legacy benchmark (`m=3`, `tau=5`, `k=5`, horizons 1–4). All nine observation variables are used as inputs; the last is the target. The script does not acquire data. The supplied files must match the processed inputs described in the data README, not simply any version of the source data.

Numerical reference and availability

reference_results/table_values.csv contains 208 audited improvement-rate entries from Tables 2–11 (192 toy-model entries and 16 aggregate flood entries). Table 1 selections are recorded in `config/selected_parameters.json`. These compact references are not raw observations or per-time-step flood predictions. Regenerated experiment plots may differ in presentation from the final assembled manuscript figures.

Permission to publish research using the flood data was obtained by a co-author, as confirmed by the authors. The authors do not have permission to redistribute the observations. Enquiries about access should be directed to Christian W. Dawson; see the original data reference in the data README.

The authors' code and fixed synthetic inputs are supplied in this supplementary archive. Code availability does not supply the separately held flood observations. The source citation and access route for those observations are in `data/flood_dawson2005/README.md`.

No separate software or synthetic-data reuse licence has been specified in this package. Do not infer an MIT, BSD, or other software licence from the availability of the files. Third-party flood observations are not part of this distribution.