Interfere: Studying Intervention Response Prediction in Complex Dynamic Models

D. J. Passey [1] and **Peter J. Mucha** [2]

1 University of North Carolina at Chapel Hill, United States **2** Dartmouth College, United States

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The vision of Interfere is simple: What if we used high quality scientific models to benchmark causal prediction tools? Randomized experimental data and counterfactuals are essential for testing methods that attempt to infer causal relationships from data, but obtaining such datasets can be expensive and difficult. Mechanistic models are commonly developed to simulate scenarios and predict the response of systems to interventions across economics, neuroscience, ecology, systems biology and other areas (Baker et al., 2018; Banks et al., 2017; Brayton et al., 2014; Izhikevich & Edelman, 2008) . Because these models are painstaking calibrated with the real world, they have the ability to generate diverse and complex synthetic counterfactual data that are characteristic of the real processes they emulate. Interfere offers the first steps towards this vision by combining (1) a general interface for simulating the effect of interventions on dynamic simulation models, (2) a suite of predictive methods and cross validation tools, and (3) an initial benchmark data set of dynamic counterfactual scenarios.

Three-dimensional trajectories of sixty scenarios simulated with the Interfere package. All models depicted are either differential equations or discrete time difference equations. The trajectory in blue represents the natural behavior of the system and the red depicts how the system responds to an intervention. Many of the models pictured have more than three dimensions (in such cases, only the three dimensions of the trajectory with the highest variance are shown). These sixty scenarios make up the Interfere Benchmark 1.1.1 for intervention response prediction which is available online for download.

Figure 1: Three-dimensional trajectories of sixty scenarios simulated with the Interfere package. All models depicted are either differential equations or discrete time difference equations. The trajectory in blue represents the natural behavior of the system and the red depicts how the system responds to an intervention. Many of the models pictured have more than three dimensions (in such cases, only the three dimensions of the trajectory with the highest variance are shown). These sixty scenarios make up the Interfere Benchmark 1.1.1 for intervention response prediction which is available online for download.

Statement of Need

Over the past twenty years the scientific community has experience the emergence of multiple frameworks for identifying causal relationships in observational data (Imbens & Rubin, 2015; Pearl, 2009; Wieczorek & Roth, 2019). The most influential frameworks are probabilistic and, while is not a necessary condition for identifying causality, historically a linear relationship was often assumed. However, when attempting to anticipate the response of complex systems in the medium and long term, a linear approximation of the dynamics is insufficient. Therefore, scientists have increasingly begun to employ non-linear techniques for causal analysis e.g. (Runge, 2022). Still, there are relatively few techniques that are able to fit causal dynamic nonlinear models to data. Because of this, we see an opportunity to bring together the insights from recent advancements in causal inference with historical work in dynamic modeling and simulation.

In order to facilitate this cross pollination, we focus on a key causal problem — predicting how a complex system responds to a previously unobserved intervention — and designed the Interfere package for benchmarking tools aimed at intervention response prediction. The dynamic models contained in Interfere present challenges for causal inference that can likely only be addressed with the incorporation of mechanistic assumptions alongside probabilistic tools. As such, the Interfere package presents an opportunity for cross pollination between the causal inference community and the modeling and simulation community.

Usage

The Interfere package is designed around four tasks: (1) Simulation, (2)intervention, (3) forecasting method optimization and (4) intervention response prediction. The following section will describe each task in detail alongside example code.

1. Simulation.

The models implemented in the Interfere package are mainly stochastic differential equations simulated with Ito's method (e.g. $d\mathbf{X} = A\mathbf{X} + d\mathbf{W}$) or difference equations (e.g. $x[n+1] = 0.25x[n] - 0.5x[n-1]$), simulated via initial conditions and stepping forward in time. Each dynamic model class included in the Interfere package has a simulate method. To run a simulation, the package requires an array of equally spaced time values and an initial conditions or past observations. For example:

```
import numpy as np
import interfere
import optuna

# Set up simulation parameters
initial_cond = np.random.rand(3)
t_train = np.arange(0, 10, 0.05)
dynamics = interfere.dynamics.Belozyorov3DQuad(sigma=0.5)

# Generate trajectory
sim_states = dynamics.simulate(t_train, initial_cond)
```

Simulation of System: The natural, uninterupted evolution of a chaotic system studied in (Belozyorov, 2015) with the addition of a small amount of stochastic noise. For simplicity, we've let $x = x_1$, $y = x_2$ and and do not plot x_0 for.

Figure 2: Simulation of System: The natural, uninterupted evolution of a chaotic system studied in (Belozyorov, 2015) with the addition of a small amount of stochastic noise. For simplicity, we've let $x = x_1$, $y = x_2$ and and do not plot x_0 for.

2. Intervention

Next, we can take exogenous control of x by pinning it to $\sin(t)$ and simulate the response of y. The resulting simulation reveals how the behavior of the system is altered by this particular intervention. See 3 for an example.

```
# Time points for the intervention simulation
test_t = np.arange(t_train[-1], 15, 0.05)
```

```python
# Intervention initialization
intervention = interfere.SignalIntervention(iv_idxs=1, signals=np.sin)

# Simulate intervention
interv_states = dynamics.simulate(
    test_t,
    prior_states=sim_states,
    intervention=intervention,
)
```

System trajectory with intervention: The figure above demonstrates the effect that taking exogenous control of $x(t)$ by via do$(x(t) = \sin(t))$ has on y. The intervention (black) and response (blue), depict a clear departure from the natural evolution behavior of the system.

Figure 3: System trajectory with intervention: The figure above demonstrates the effect that taking exogenous control of $x(t)$ by via do$(x(t) = \sin(t))$ has on y. The intervention (black) and response (blue), depict a clear departure from the natural evolution behavior of the system.

3. Optimization

Interfere offers tools to optimize forecasting methods for time series prediction. By using Interfere's cross validation objective function along with a hyperparameter optimizer (Optuna), it is possible to compare multiple hyperparameter setting on multiple folds of time series data. To simplify this process, every Interfere forecasting method comes with sensible preset hyperparameter ranges for the optimizer to explore.

```python
# Select the SINDy method for hyperparameter optimization.
method_type = interfere.SINDy

# Create an objective function that aims to minimize cross validation error
# over different hyper parameter configurations for SINDy
cv_obj = interfere.CrossValObjective(
    method_type=method_type,
    data=sim_states,
    times=t_train,
    train_window_percent=0.3,
    num_folds=5,
    exog_idxs=intervention.iv_idxs,
)

# Run the study using optuna.
study = optuna.create_study()
study.optimize(cv_obj, n_trials=25)

# Collect the best hyperparameters into a dictionary.
best_param_dict = study.best_params
```

4. Intervention Response Prediction

Using the best parameters from the hyperparameter optimization run, we can fit the forecasting method to all the data that occurred prior to the intervention, treating the states we plan to manipulate as exogenous. This way, the method expects to be given exogenous data about the intervention variable(s). After fitting to the unperturbed system, we forecast the intervention response by treating the desired intervention as an exogenous input signal to include in the forecast.

```python
# Initialize SINDy with the best perfoming parameters.
method = interfere.SINDy(**study.best_params)

# Use an intervention helper function to split the pre-intervention data
# into endogenous and exogenous columns.
Y_endog, Y_exog = intervention.split_exog(sim_states)

# Fit SINDy to the pre-intervention data.
method.fit(t_train, Y_endog, Y_exog)

# Use the inherited interfere.ForecastingMethod.simulate() method
# To simulate intervention response using SINDy
pred_traj = method.simulate(
    test_t, prior_states=sim_states, intervention=intervention
)
```

Forecasting Intervention Response: Example of forecasting the response of the Belozyorov system to a sinusoidal intervention. Here, the intervention consists of taking exogenous control of $x(t)$ (black). The ground truth response, $y(t)$ for $t > 10$ is plotted in blue. Here, an equation discovery algorithm, SINDy (Brunton et al., 2016) is fit to the data that occurs prior to the intervention, and makes an attempt to predict the intervention response (red curve).

Figure 4: Forecasting Intervention Response: Example of forecasting the response of the Belozyorov system to a sinusoidal intervention. Here, the intervention consists of taking exogenous control of $x(t)$ (black). The ground truth response, $y(t)$ for $t > 10$ is plotted in blue. Here, an equation discovery algorithm, SINDy (Brunton et al., 2016) is fit to the data that occurs prior to the intervention, and makes an attempt to predict the intervention response (red curve).

Primary Contributions

The Interfere package provides three primary contributions to the scientific community.

Example experimental setup possible with Interfere: Comparing intervention response prediction for deterministic and stochastic versions of the same system. Can stochasticity help reveal associations between variables?

Figure 5: Example experimental setup possible with Interfere: Comparing intervention response prediction for deterministic and stochastic versions of the same system. Can stochasticity help reveal associations between variables?

1. Dynamically Diverse Counterfactuals at Scale

The "dynamics" submodule in the Interfere package contains over fifty dynamic models. It contains a mix of linear, nonlinear, chaotic, continuous time, discrete time, stochastic, and deterministic models. The models come from a variety of disciplines including economics, finance, ecology, biology, neuroscience and public health. Each model inherits the from the Interfere BaseDynamics type and gains the ability to take exogenous control of any observed state and to add measurement noise. Most models also gain the ability to make any observed state stochastic where magnitude of stochasticity can be controlled by a simple scalar parameter or fine tuned with a covariance matrix.

Because of the difficulty of building models of complex systems, predictive methods for complex dynamics are typically benchmarked on less than ten dynamical systems (Brunton et al., 2016; Challu et al., 2023; Pathak et al., 2018; Prasse & Van Mieghem, 2022; Vlachas et al., 2020). As such, Interfere offers a clear improvement over current benchmarking methods for prediction in complex dynamics.

Most importantly, Interfere is built around interventions: the ability to take exogenous control of the state of a complex system and observe the response. Imbuing scientific models with general exogenous control is no small feat because models can be complex and are implemented in a variety of ways. Thus Interfere offers the ability to produce multiple complex dynamic *counterfactual scenarios* at scale. This unique feature enables large scale evaluation of dynamic causal prediction methods—tested against systems with properties of interest to scientists.

2. Cross Disciplinary Forecast Methods

A second contribution of Interfere is the integration of dynamic *forecasting* methodologies from deep learning, applied mathematics and social science. The Interfere "ForecastingMethod" class is expressive enough to describe, fit and predict with multivariate dynamic models and intervene on the states of the models during prediction. This cross disciplinary mix of techniques affords new insights into the problem of intervention response prediction.

3. Opening Up Intervention Response to the Scientific Community

The third major contribution of Interfere is that it poses the intervention response problem—a highly applicable question, to the broader community. The Interfere Benchmark 1.1.1 has the potential provide simple comprehensive evaluation of computational methods on the intervention response problem and therefore streamline future progress towards correctly anticipating how complex systems will respond to new scenarios.

Related Software and Mathematical Foundations

Predictive Methods

The Interfere package draws extensively from the Nixtla open source ecosystem for time series forecasting. Nixtla's NeuralForecast proves three of the methods that are integrated with Interfere's interface and StatsForecast provides one of the methods (Azul Garza, 2022; Olivares et al., 2022). Nixtla also provided the inspiration for the cross validation and hyperparameter optimization workflow. Interfere also integrates with predictive methods from the PySINDy and StatsModels packages (Kaptanoglu et al., 2022; Seabold & Perktold, 2010). An additional reservoir computing method for global forecasts comes from (Harding et al., 2024). Hyperparameter optimization is designed around the Optuna framework (Akiba et al., 2019).

Finding forecasting methods to integrate with Interfere was difficult due to the (1) lack of multivariate dynamic forecasting methods (2) lack of dynamic methods that allow exogenous variables (3) the fact that many methods only offer a fixed forecast window do not implement recursive prediction.

Dynamic Models

See the table below for a full list of dynamic models with attributions that are currently implemented in the Interfere package. The dynamic models in were implemented directly from mathematical descriptions except for two which adapt existing simulations from the PyClustering package (Novikov, 2019).

Dynamic Model Class	Description and Source	Properties
Arithmetic Brownian Motion	Brownian motion with linear drift and constant diffusion (Øksendal, 2005)	Stochastic, Linear

Dynamic Model Class	Description and Source	Properties
Coupled Logistic Map	Discrete-time logistic map with spatial coupling (Lloyd, 1995)	Nonlinear, Chaotic
Stochastic Coupled Map Lattice	Coupled map lattice with stochastic noise (Kaneko, 1991)	Nonlinear, Stochastic, Chaotic
Michaelis Menten	Model for enzyme kinetics and biochemical reaction networks (Srinivasan, 2022)	Nonlinear, Stochastic
Lotka Voltera SDE	Stochastic Lotka-Volterra predator-prey model (Hening & Nguyen, 2018)	Nonlinear, Stochastic
Kuramoto	Coupled oscillator model to study synchronization (Rodrigues et al., 2016)	Nonlinear, Stochastic
Kuramoto Sakaguchi	Kuramoto model variant with phase frustration (Sakaguchi & Kuramoto, 1986)	Nonlinear, Stochastic
Hodgkin Huxley Pyclustering	Neuron action-potential dynamics based on Hodgkin-Huxley equations (Hodgkin & Huxley, 1952)	Nonlinear
Stuart Landau Kuramoto	Coupled oscillators with amplitude-phase dynamics (Cliff et al., 2023)	Nonlinear, Stochastic
Mutualistic Population	Dynamics of interacting mutualistic species (Prasse & Van Mieghem, 2022)	Nonlinear
Ornstein Uhlenbeck	Mean-reverting stochastic differential equation (Gardiner, 2009)	Stochastic, Linear
Belozyorov 3D Quad	3-dimensional quadratic chaotic system (Belozyorov, 2015)	Nonlinear, Chaotic
Liping 3D Quad Finance	Chaotic dynamics applied in financial modeling (Liping et al., 2021)	Nonlinear, Chaotic
Lorenz	Classic chaotic system describing atmospheric convection (Lorenz, 2017)	Nonlinear, Chaotic
Rossler	Simplified 3D chaotic attractor system (Rössler, 1976)	Nonlinear, Chaotic
Thomas	Chaotic attractor with simple structure and rich dynamics (Thomas, 1999)	Nonlinear, Chaotic
Damped Oscillator	Harmonic oscillator with damping and noise (Classical linear model)	Linear, Stochastic
SIS	Epidemiological model (Susceptible-Infected-Susceptible) (Prasse & Van Mieghem, 2022)	Nonlinear, Stochastic
VARMA Dynamics	Vector AutoRegressive Moving Average for time series modeling (Hamilton, 2020)	Linear, Stochastic
Wilson Cowan	Neural mass model for neuronal population dynamics (Wilson & Cowan, 1972)	Nonlinear
Geometric Brownian Motion	Stochastic model widely used in financial mathematics (Black & Scholes, 1973)	Nonlinear, Stochastic
Planted Tank Nitrogen Cycle	Biochemical cycle modeling nitrogen transformation in aquatic systems (Fazio & Jannelli, 2006)	Nonlinear
Generative Forecaster	Predictive forecasting models trained on simulation, then used to generate data (Written for Interfere)	Stochastic
Standard Normal Noise	IID noise from standard normal distribution (Cliff et al., 2023)	Stochastic
Standard Cauchy Noise	IID noise from standard Cauchy distribution (Cliff et al., 2023)	Stochastic

Dynamic Model Class	Description and Source	Properties
Standard Exponential Noise	IID noise from standard exponential distribution (Cliff et al., 2023)	Stochastic
Standard Gamma Noise	IID noise from standard gamma distribution (Cliff et al., 2023)	Stochastic
Standard T Noise	IID noise from Student's t-distribution (Cliff et al., 2023)	Stochastic

Acknowledgements

This work was supported by NSF GRFP.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

Azul Garza, C. C., Max Mergenthaler Canseco. (2022). *StatsForecast: Lightning fast forecasting with statistical and econometric models*. PyCon Salt Lake City, Utah, US 2022. https://github.com/Nixtla/statsforecast

Baker, R. E., Peña, J.-M., Jayamohan, J., & Jérusalem, A. (2018). Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology Letters*, *14*(5), 20170660. https://doi.org/10.1098/rsbl.2017.0660

Banks, H. T., Banks, J. E., Bommarco, R., Curtsdotter, A., Jonsson, T., & Laubmeier, A. N. (2017). Parameter estimation for an allometric food web model. *International Journal of Pure and Apllied Mathematics*, *114*(1). https://doi.org/10.12732/ijpam.v114i1.12

Belozyorov, V. Y. (2015). Exponential-algebraic maps and chaos in 3D autonomous quadratic systems. *International Journal of Bifurcation and Chaos*, *25*(04), 1550048.

Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, *81*(3), 637–654.

Brayton, F., Laubach, T., & Reifschneider, D. (2014). The FRB/US model: A tool for macroeconomic policy analysis. *FEDS Notes*, *2014-04*, 03.

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *113*(15), 3932–3937. https://doi.org/10.1073/pnas.1517384113

Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., & Dubrawski, A. (2023). NHITS: Neural Hierarchical Interpolation for Time Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(6), 6989–6997. https://doi.org/10.1609/aaai.v37i6.25854

Cliff, O. M., Bryant, A. G., Lizier, J. T., Tsuchiya, N., & Fulcher, B. D. (2023). Unifying pairwise interactions in complex dynamics. *Nature Computational Science*, *3*(10), 883–893. https://doi.org/10.1038/s43588-023-00519-x

Fazio, R., & Jannelli, A. (2006). Mathematical and numerical modeling for a bio-chemical aquarium. *Applied Mathematics and Computation*, *174*(2), 1370–1383.

Gardiner, C. (2009). *Stochastic methods* (Vol. 4). Springer Berlin Heidelberg.

Hamilton, J. D. (2020). *Time series analysis*. Princeton university press.

Harding, S., Leishman, Q., Lunceford, W., Passey, D. J., Pool, T., & Webb, B. (2024). Global forecasts in reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *34*(2), 023136. https://doi.org/10.1063/5.0181694

Hening, A., & Nguyen, D. H. (2018). Stochastic lotka–volterra food chains. *Journal of Mathematical Biology*, *77*, 135–163.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4), 500.

Imbens, G. W., & Rubin, D. B. (2015). *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction* (1st ed.). Cambridge University Press. https://doi.org/10.1017/CBO9781139025751

Izhikevich, E. M., & Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, *105*(9), 3593–3598. https://doi.org/10.1073/pnas.0712231105

Kaneko, K. (1991). Coupled map lattice. In *Chaos, Order, and Patterns* (pp. 237–247). Springer.

Kaptanoglu, A. A., Silva, B. M. de, Fasel, U., Kaheman, K., Goldschmidt, A. J., Callaham, J., Delahunt, C. B., Nicolaou, Z. G., Champion, K., Loiseau, J.-C., Kutz, J. N., & Brunton, S. L. (2022). PySINDy: A comprehensive python package for robust sparse system identification. *Journal of Open Source Software*, *7*(69), 3994. https://doi.org/10.21105/joss.03994

Liping, C., Khan, M. A., Atangana, A., & Kumar, S. (2021). A new financial chaotic model in Atangana-Baleanu stochastic fractional differential equations. *Alexandria Engineering Journal*, *60*(6), 5193–5204.

Lloyd, A. L. (1995). The coupled logistic map: A simple model for the effects of spatial heterogeneity on population dynamics. *Journal of Theoretical Biology*, *173*(3), 217–230.

Lorenz, E. N. (2017). Deterministic Nonperiodic Flow 1. In *Universality in Chaos, 2nd edition* (pp. 367–378). Routledge.

Novikov, A. V. (2019). PyClustering: Data mining library. *Journal of Open Source Software*, *4*(36), 1230. https://doi.org/10.21105/joss.01230

Øksendal, B. K. (2005). *Stochastic differential equations: An introduction with applications* (6th ed., corrected third printing 2005). Springer. ISBN: 978-3-540-04758-2

Olivares, K. G., Challú, C., Garza, A., Canseco, M. M., & Dubrawski, A. (2022). *NeuralForecast: User friendly state-of-the-art neural forecasting models.* PyCon Salt Lake City, Utah, US 2022. https://github.com/Nixtla/neuralforecast

Pathak, J., Hunt, B., Girvan, M., Lu, Z., & Ott, E. (2018). Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Physical Review Letters*, *120*(2), 024102. https://doi.org/10.1103/PhysRevLett.120.024102

Pearl, J. (2009). *Causality* (2nd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511803161

Prasse, B., & Van Mieghem, P. (2022). Predicting network dynamics without requiring the knowledge of the interaction graph. *Proceedings of the National Academy of Sciences*, *119*(44), e2205517119. https://doi.org/10.1073/pnas.2205517119

Rodrigues, F. A., Peron, T. K. D., Ji, P., & Kurths, J. (2016). The Kuramoto model in complex networks. *Physics Reports*, *610*, 1–98.

Rössler, O. E. (1976). An equation for continuous chaos. *Physics Letters A*, *57*(5), 397–398.

Runge, J. (2022). *Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets*. arXiv. https://doi.org/10.48550/arXiv.2003.03685

Sakaguchi, H., & Kuramoto, Y. (1986). A soluble active rotater model showing phase transitions via mutual entertainment. *Progress of Theoretical Physics*, *76*(3), 576–581.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *9th Python in Science Conference*.

Srinivasan, B. (2022). A guide to the Michaelis–Menten equation: Steady state and beyond. *The FEBS Journal*, *289*(20), 6086–6098.

Thomas, R. (1999). Deterministic chaos seen in terms of feedback circuits: Analysis, synthesis," labyrinth chaos". *International Journal of Bifurcation and Chaos*, *9*(10), 1889–1905.

Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., & Koumoutsakos, P. (2020). Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, *126*, 191–217. https://doi.org/10.1016/j.neunet.2020.02.016

Wieczorek, A., & Roth, V. (2019). Information Theoretic Causal Effect Quantification. *Entropy*, *21*(10), 975. https://doi.org/10.3390/e21100975

Wilson, H. R., & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, *12*(1), 1–24.