EFFICIENTLY COMPOSING AND CONTROLLING HYBRID SIMULATIONS OF PDES AND ML MODELS

Kevin A. Brown¹, Kazutomo Yoshii¹, Elkin Cruz-Camacho², Christopher Carothers², Zhiling Lan³, Kai Shu⁴, Tanwi Mallick¹, Xin Wang³, Xiongxiao Xu⁴, Robert B. Ross¹ ¹Argonne National Laboratory, ²Rensselaer Polytechnic Institute, ³University of Illinois Chicago, ⁴Illinois Institute of Technology

BACKGROUND & CHALLENGES

- Combing the benefits of PDES and ML models requires orchestrating two independent and complex frameworks
- Parallel discrete event simulation (PDES) enable high-fidelity predictions
 - Complex models of physical processes
 - Orchestration of millions/billions of events
 - Computation cost limits scalability
- Machine learning (ML) models can provide faster predictions than PDES
 - Predict high-level behaviors without simulating lower-level activities
 - Require large amounts of training data cover a variety of conditions

GOAL

- Design a Director module to connect and dynamically orchestrate the activities PDES and ML models
- Create a simulation framework that uses the Director to control hybrid simulations of PDES and ML models
- Leverage the high-fidelity of PDES to simulate and explore novel, complex scenarios
- Leverage the speed of ML to predict behaviors of interest,

FRAMEWORK OVERVIEW

KRONOSframework



APPROACH AND DESIGN

Hybrid PDES/ML Simulation

Phase-specific Surrogates Application phases are annotated and ML models are trained/re-trained for only the current phase.

Use PDES When New Phase Starts PDES models generate training data for training/re-training model at the start of each phase.

Use Surrogate Once Trained Redirect PDES events to the Director to have the surrogates predict event durations.

The PDES toolkit is connected to external Python-based surrogate models using ZeroMQ (ZMQ) communication library

PDES and ML models may be modified independently

The Director manages interactions between PDES and surrogates

Director Module LP is ZMQ Clients

 Director is implemented as logical processes (LPs); moves training data and inferences between PDES and ML models



Online ML Model Management

- Surrogate server controls the ML models and processes Director requests
- Server supports online ML training
 - Director accumulates and sends bulk training data to server
 - Director can trigger model training and model accuracy testing et al.

Director Integration with Workload

- Workload LPs are paired with Director LPs
 - State variable to indicate current simulation mode (PDES/surrogate)
 - Workload event handlers for processing events from Director

EVALUATION



The Kronos Project is supported by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357 and the US National Science Foundation grant CCF-2109316.









