LUME-EPICS

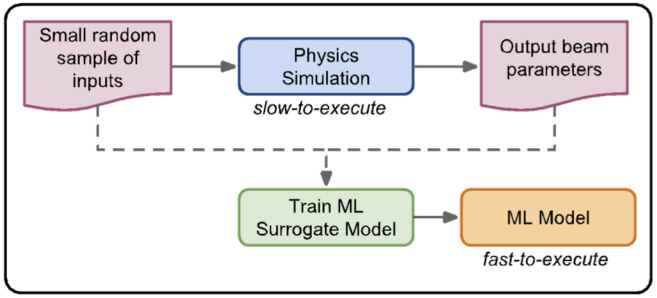
EPICS Collaboration Presentation, 7/9/21

Jacqueline Garrahan, Christopher Mayes, Hugo Slepicka,
Lipi Gupta, Auralee Edelen

U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

Background

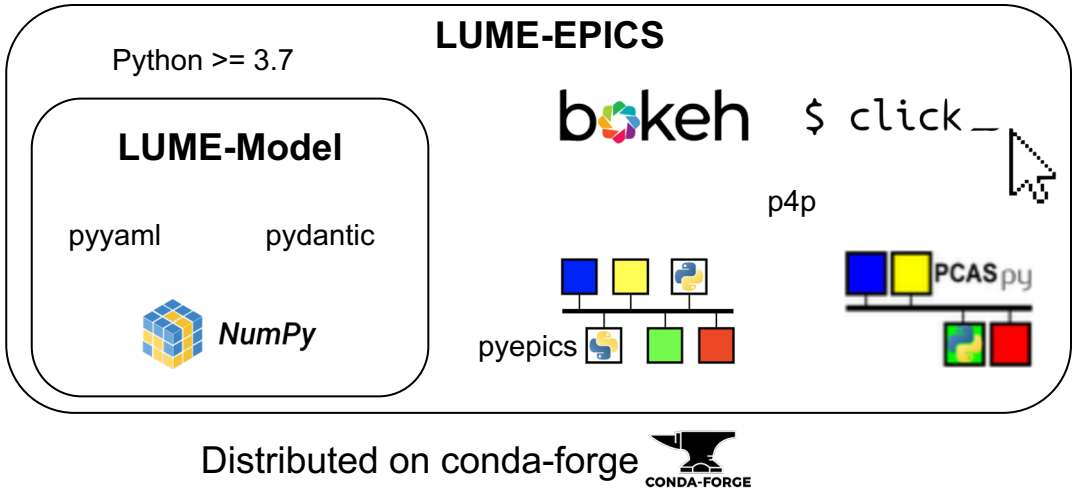


Typical surrogate modeling workflow

- LUME-EPICS and LUME-Model are members of the LUME project
- LUME project aims to wrap standard, developed electron/photon simulation codes with a common Python interface
- Surrogate models fall under this purview: ML surrogates trained on inputs and outputs of physics simulations lead to fast executing models, which may be used for tuning etc.
- NEED: Integration of surrogate models with the control system- execution on live variables, ability to surface outputs

C. E. Mayes, et al. (2021). Lightsource unified modeling environment (LUME), a start-to-end simulation ecosystem. In Proc. of IPAC (p. THPAB217).

LUME-EPICS (and LUME-model) Overview



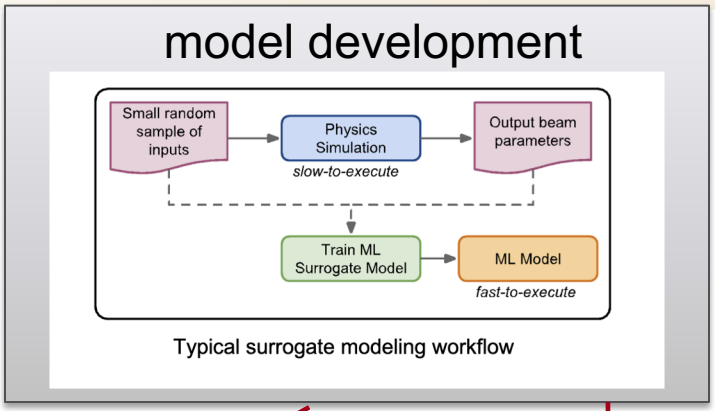
LUME-Model

- Base classes for guiding standardized development of surrogate model execution classes
- Variables classes with attribute type validation to enforce minimum data requirements

LUME-EPICS

- EPICS server (default both Channel Access and pvAccess, but configurable)
- Callbacks on input process variable update
- EPICS-based bokeh widgets for interface development
- Templated generation of displays

Components



Model-specific files
(eg. weights, architecture)

variable file

lume-epics model
- LUME-Model input variables accepted to evaluate method
- LUME-Model output variables returned

Following training, variables must be saved using the LUME-Model utility

update output variables

Model prediction callback on EPICS puts

LUME-EPICS server
- Input and output variables built with LUME-Model
- Configurable protocol (pvAccess/Channel Access)

put/get

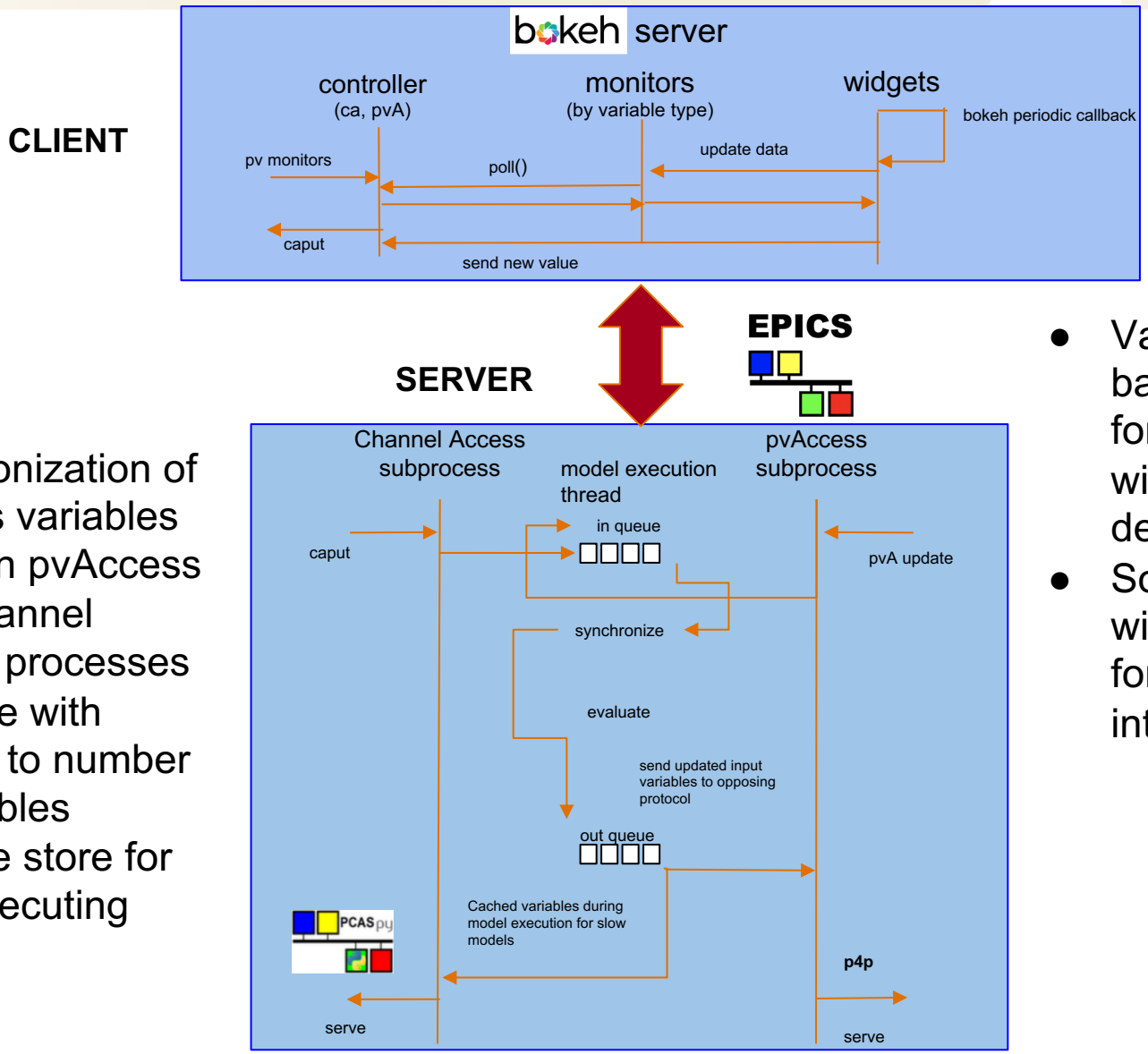
EPICS

LUME-EPICS client
- Bokeh widget tools initialized with LUME-Model variables
- Tiered controls:
  - Controller: Access process variables
  - Monitor: Formats data
  - Widgets: Development objects

Design features:

- Separable server and client tools
- Compatibility enforced by LUME-Model variable validation
- Extensible SurrogateModel execution class for high levels of customizability
- Templated class for Keras models

LUME-EPICS Application Structure



- Synchronization of process variables between pvAccess and Channel Access processes
- Scalable with respect to number of variables
- Variable store for slow executing models

- Variable type based monitors for continued widget development
- Some bokeh widgets adapted for easy integration

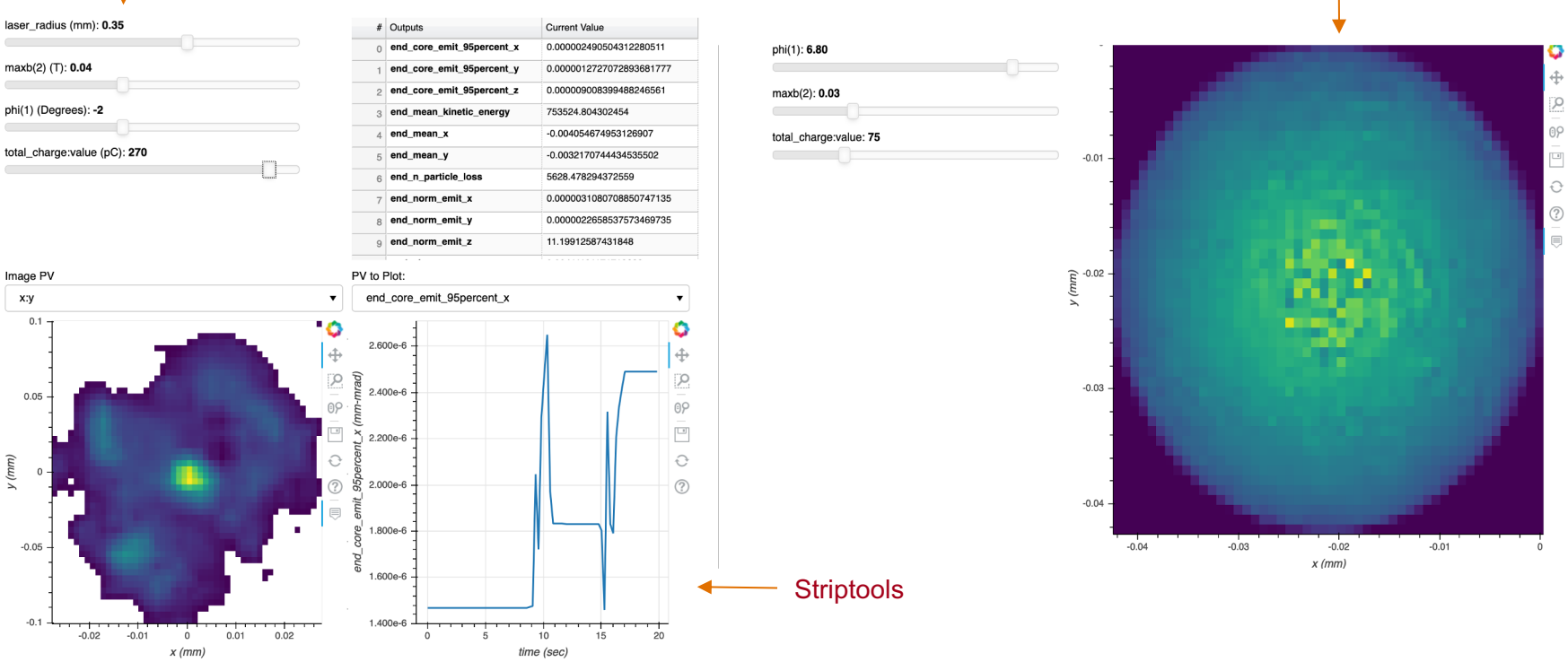
Applications: Neural network surrogate models

- Packaged neural network surrogate model of the LCLS cu injector and served using LUME-EPICS toolkit (credit: Lipi Gupta)

control sliders

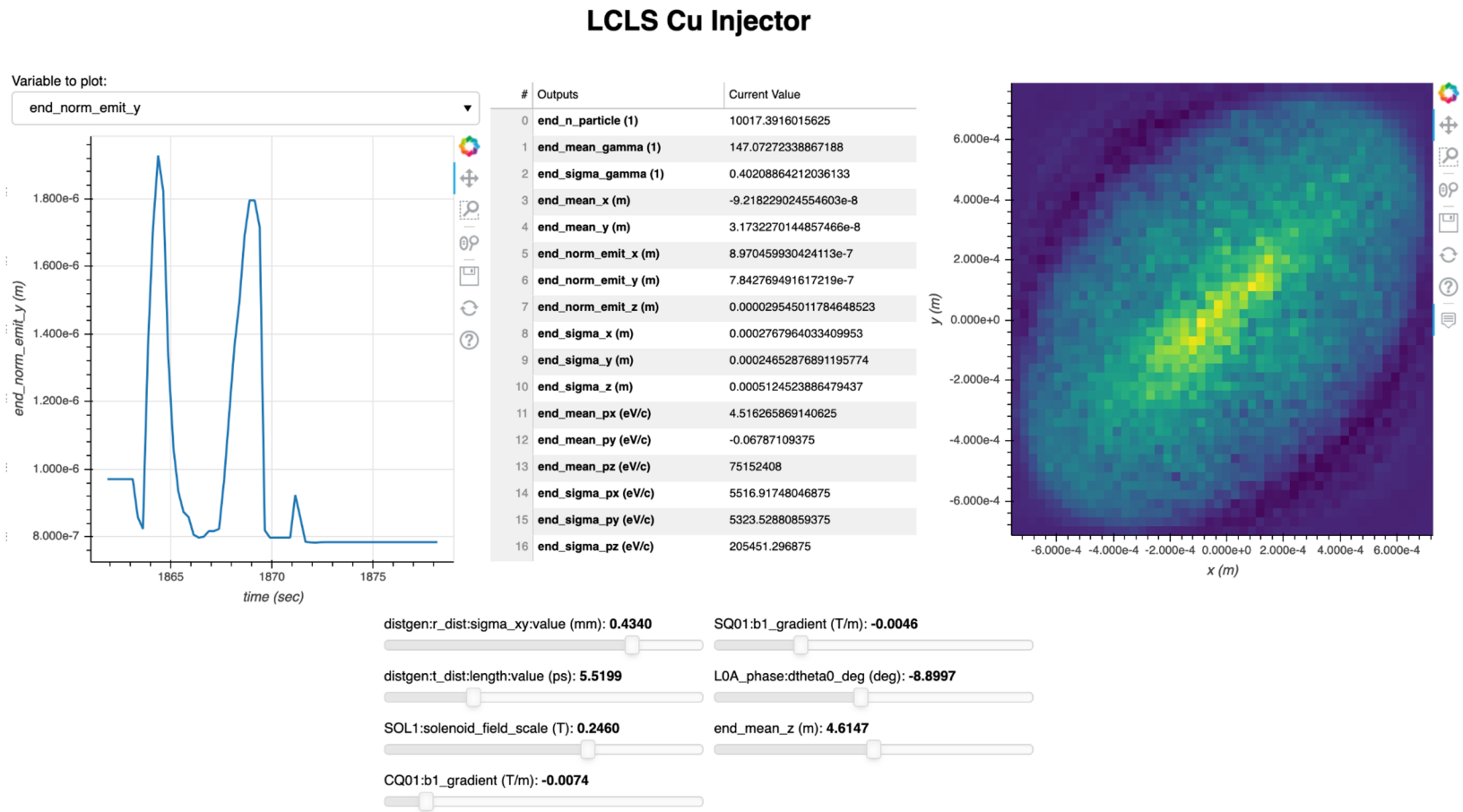
Rendered in a Jupyter Notebook:

Image plot widget



Applications: Neural network surrogate models

Served bokeh dashboard with controls:



Available here

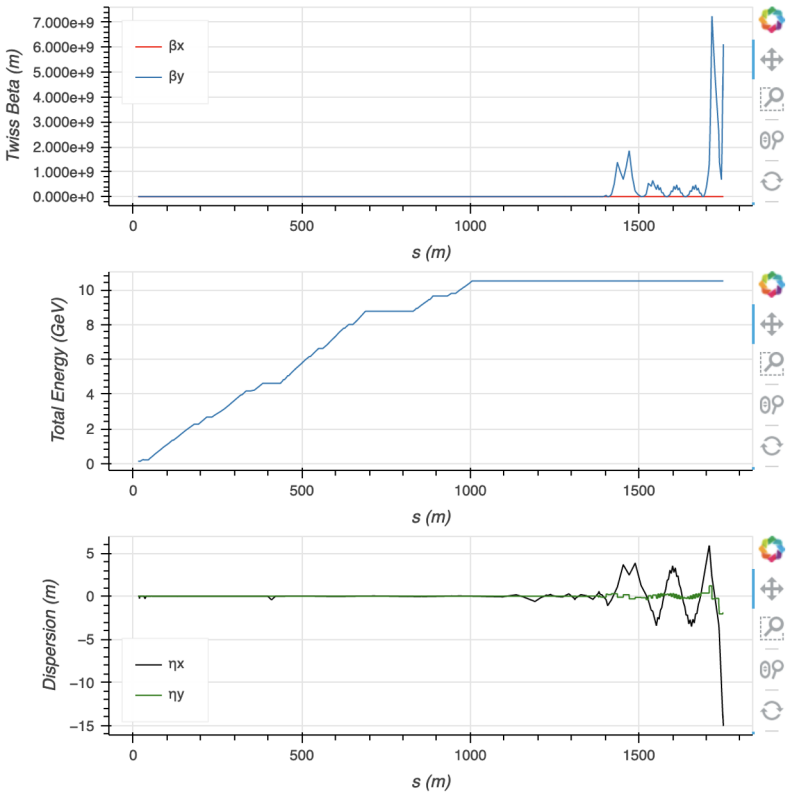
Applications: Bmad model execution with PyTao

LCLS copper HXR beamline model dashboard rendered locally with bridge to live accelerator PVs:

Array plot widget

Value table monitoring live model inputs

LUME LCLS cu_hxr model



Variable	Current Value
SBST:LI21:1:PHAS	-0.19599
SBST:LI22:1:PHAS	0.25324
SBST:LI23:1:PHAS	-0.0012646
SBST:LI24:1:PHAS	-0.0072327
SBST:LI25:1:PHAS	-0.085625
SBST:LI26:1:PHAS	0.41496
SBST:LI27:1:PHAS	8.6108
SBST:LI28:1:PHAS	0.0093842
ACCL:LI29:0:KLY_PDES	-34.027
ACCL:LI30:0:KLY_PDES	34.027
BMLN:LI21:235:MOTR	247.86
BMLN:LI24:805:MOTR	385.0
ACCL:LI21:1:L1S_S_PV	-20.071
ACCL:LI22:1:PDES	-36.284
ACCL:LI25:1:PDES	0.0
ACCL:LI21:1:L1S_S_AV	111.69
ACCL:LI21:180:L1X_S_AV	20.992
ACCL:LI21:180:L1X_S_PV	-159.84
KLYS:LI21:31:ENLD	205.45
KLYS:LI21:31:PHAS	-0.42025
KLYS:LI21:31:BEAMCODE1_STAT	1.0
KLYS:LI21:31:SWRD	0.0
KLYS:LI21:31:STAT	1.0

Development roadmap

- Very much in Beta development
- Immediate goals:
  - User acquisition
  - Varied applications
  - Stress test of documentation, etc.

Learn more

- LUME: https://www.lume.science/
- Surrogate model of the cu injector (Auralee Edelen): https://www.youtube.com/watch?v=1f42uRNfx18
- Dockerized LCLS cu injector model, served with Binder: (here)
- LUME-Model documentation: (https://slaclab.github.io/lume-model/)
- LUME-EPICS documentation: (https://slaclab.github.io/lume-epics/)

Questions/Comments?