



Ian Kolaja Qualifying Exam

Machine Learning in PBR Measurement and Simulation

May 9th, 2024

Overview

Machine learning is valuable for Pebble Bed Reactor (PBR) operation because it enables faster and more accurate discharge fuel measurement and enhances core simulation capabilities

Outline

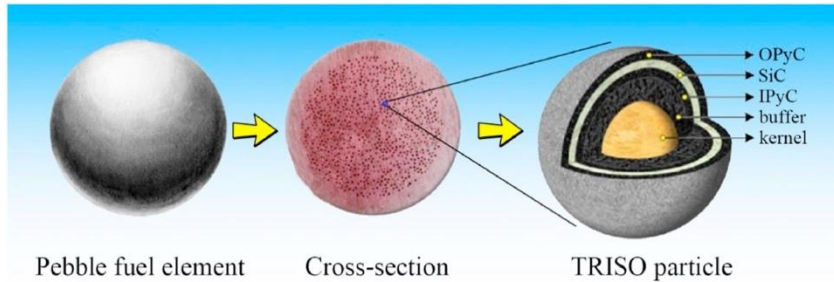
1. Introduction to PBR physics
2. Discharge pebble history and composition prediction
3. Predicting core state from discharge fuel
4. PEARLSim: Using ML for hyperfidelity core simulation



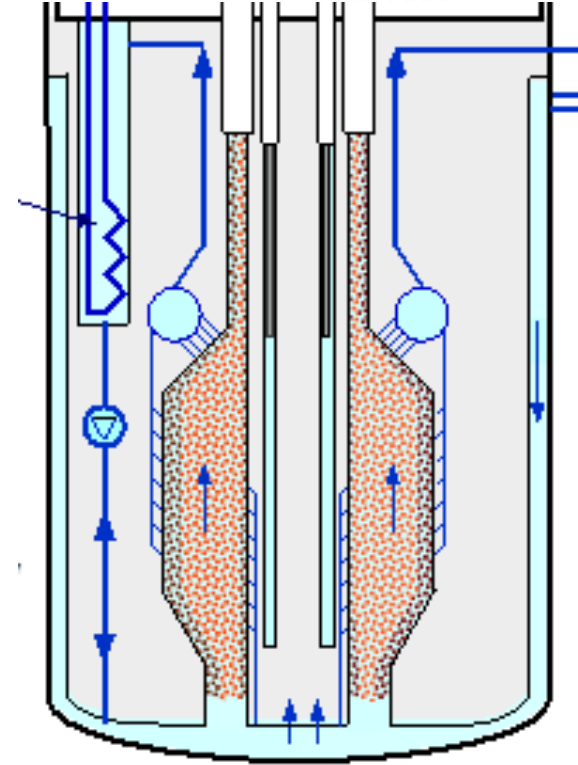
1) Introduction to Pebble Bed Reactor Physics

Pebble Bed Reactor Overview

- Use fuel “pebbles” instead of fuel rods
 - 10^4 – 10^5 pebbles in the core
 - Constant fuel circulation
- Features coolants other than water
 - Nonreactive gas (i.e. Helium)
 - Molten Salt (i.e. FLiBe)



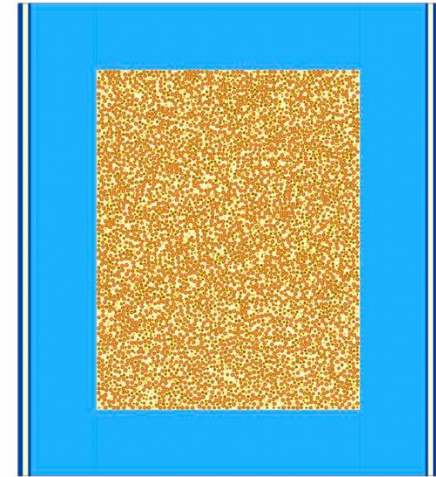
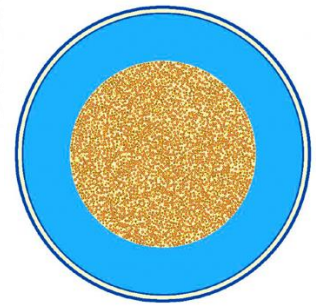
Schematic diagram of the pebble fuel, TRISO particle, and their relationship, Maolong Liu



Mk1 PB-FHR schematic

Design used: Generic FHR

- Benchmark model released by Kairos Power
- Relevant operating characteristics
 - 250,000 fuel pebbles (2cm radius)
 - Pebbles strictly make 8 passes
- Reactor physics insights
 - Neutron diffusion length much larger than pebble radius
 - Doesn't have conic regions
 - Relatively flat velocity profile



gFHR benchmark diagram



Questions on section?

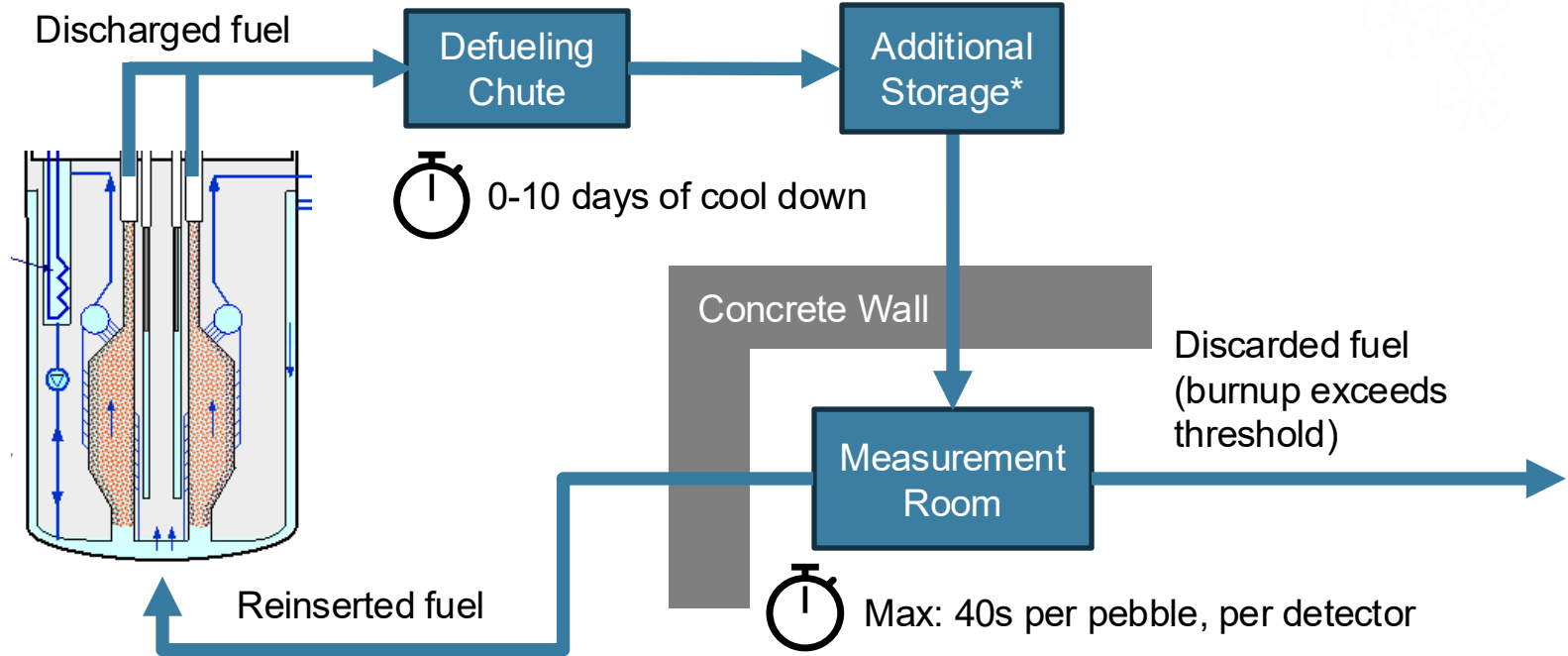


2) Discharge Pebble History and Composition Prediction

Pebble Assessment

- Objective: Simulate a realistic detected gamma spectrum from a gFHR fuel pebble and use ML to make predictions with it
- Challenges and Considerations:
 - About 2,300 pebbles leave the core a day, giving us about 40 seconds max to measure each pebble
 - Even with decay times of up to several days, pebbles emit on the order of 10^{13} gammas per second
 - For a detector, this could lead to serious dead time without prohibitively expensive shielding
- Can ML help overcome these challenges?

Hypothetical Burnup Measurement System

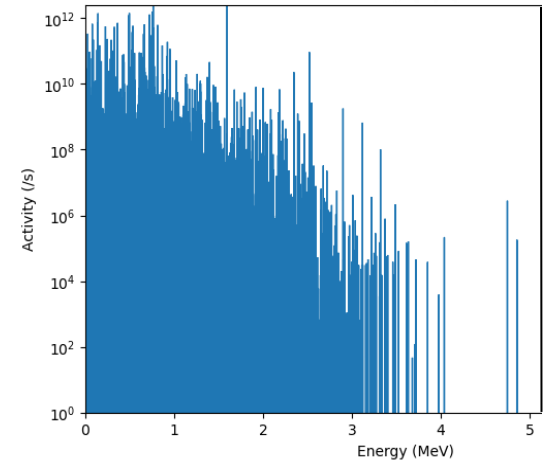




Detected Spectrum Generation

Emitted Gamma Source Generation

- Discharge compositions used to generate spectra
 - Unique, pebble-level data from HxF
- Different decay times between 0 and 10 days
 - 120 second variance applied
- Spectra produced in Serpent
 - Decay step (neutron mode)
 - Gamma source (photon mode)

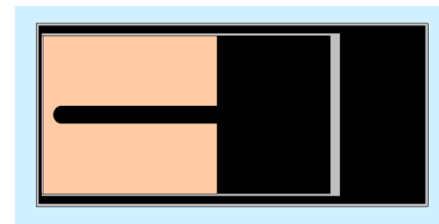
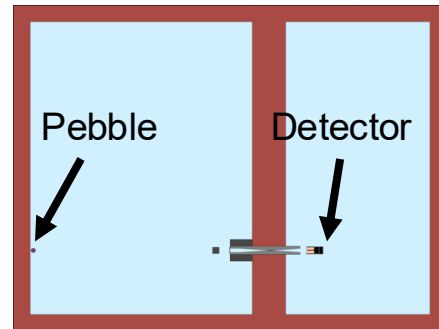
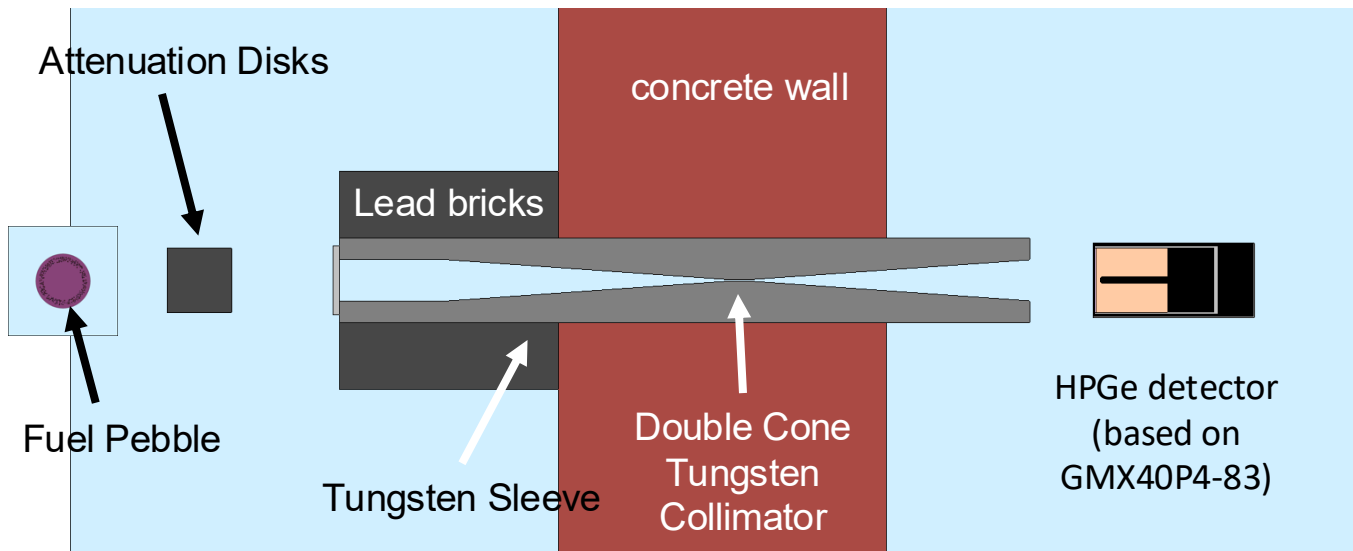


Detector Model (HPGe)



5 days of cooldown

Total pebble distance	250 cm
Collimator length	54 cm
Detector dimensions	6.06 x 6.69 cm
Average true count rate	220,000 cps

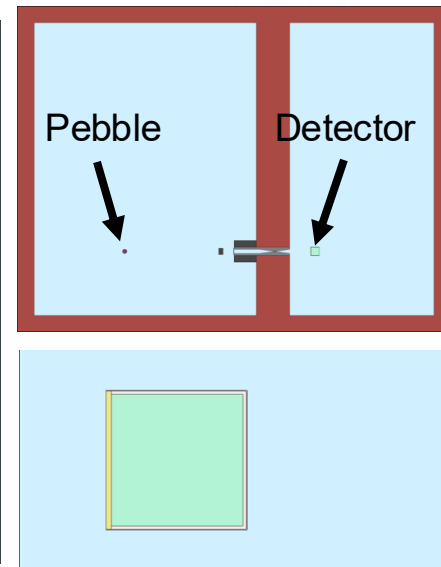
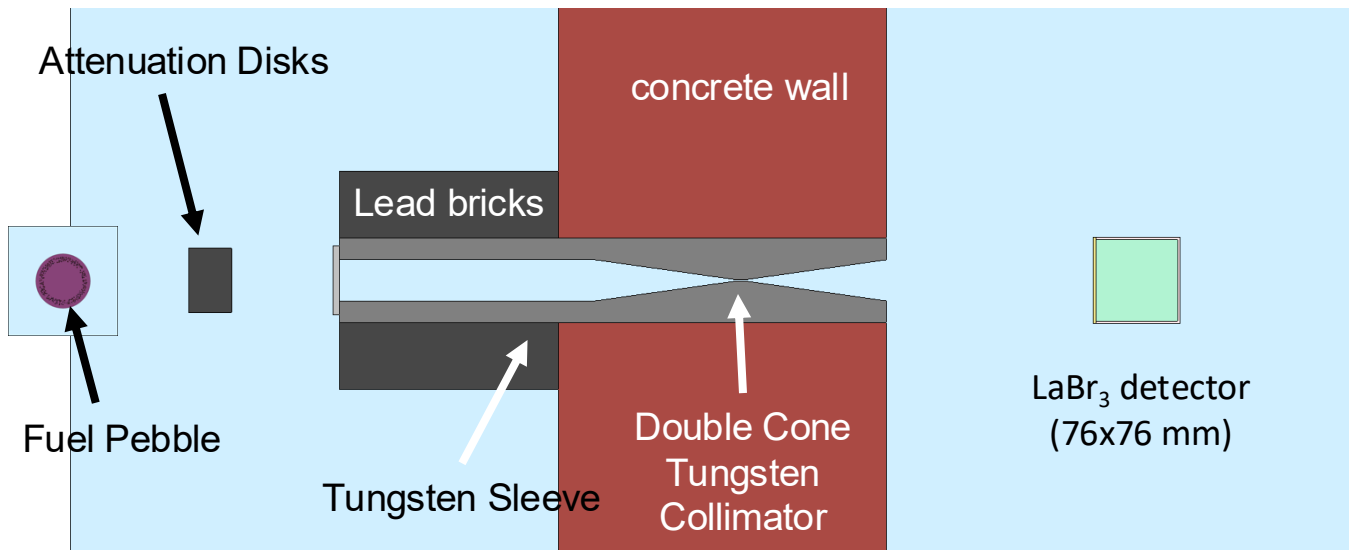


Detector Model (LaBr₃)



1 day of cooldown

Total pebble distance	170 cm
Collimator length	27 cm
Detector dimensions	7.6 x 7.6 cm
Average true count rate	870,000 cps

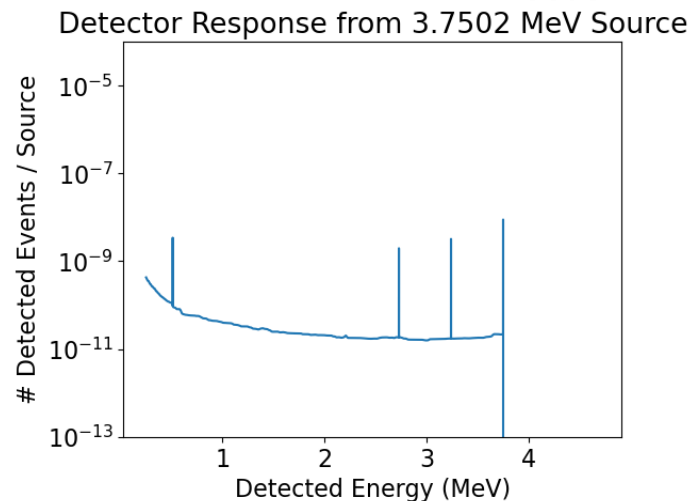
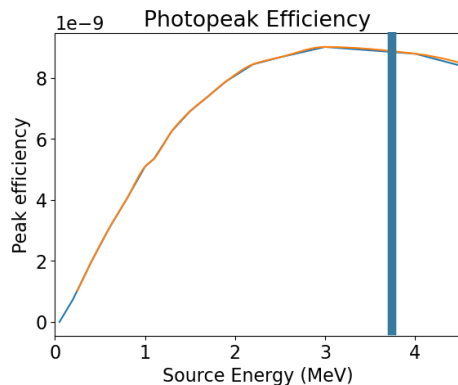


Detector Response Simulation: Tally

- Simulated at 16 different source energies
- Monoenergetic source sampled from pebble
- Pulse height tally in detector crystal
- Energy bins in the tally are divided to capture different reactions:
 - Coarse bins (20–50 keV wide) for continuum
 - Very fine bins for discrete energies (photopeak, pair production, single and double escape peaks)

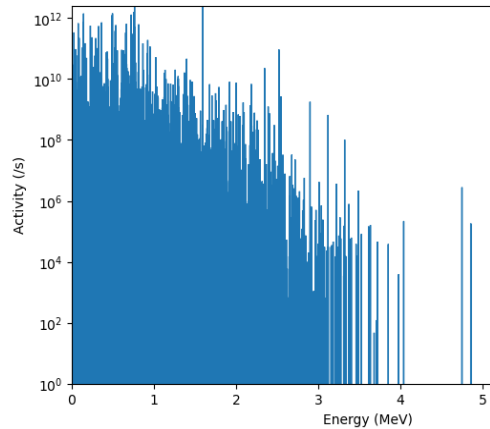
Detector Response Function

- 12000 channels from 0.25 MeV to 4.7 MeV
- Discrete peaks efficiencies are interpolated
- Continuum is fit with a RFR model

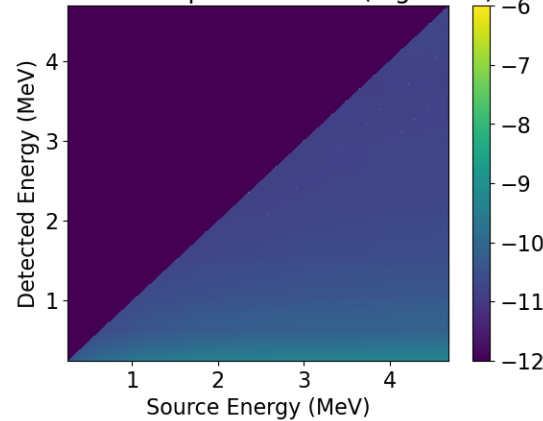


Detector Response Matrix

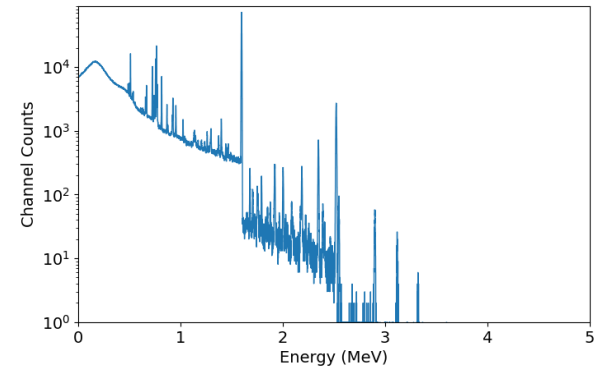
Emission Spectrum



Detector Response Matrix (log scale)



Average Incident Spectrum



Dead time and pile-up

40s measurement time with 0.4s standard deviation

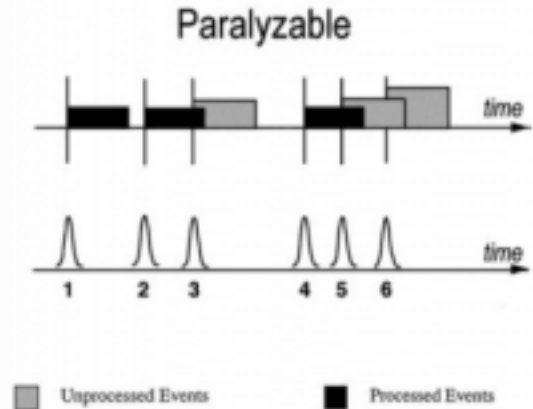
With average count rate R and measurement time t :

- Sample total counts $N \sim \text{Poisson}(Rt)$
- Sample N energies E from average spectrum
 - Gaussian broadened
- Sample N waiting times $t \sim \text{Exp}(1/R)$

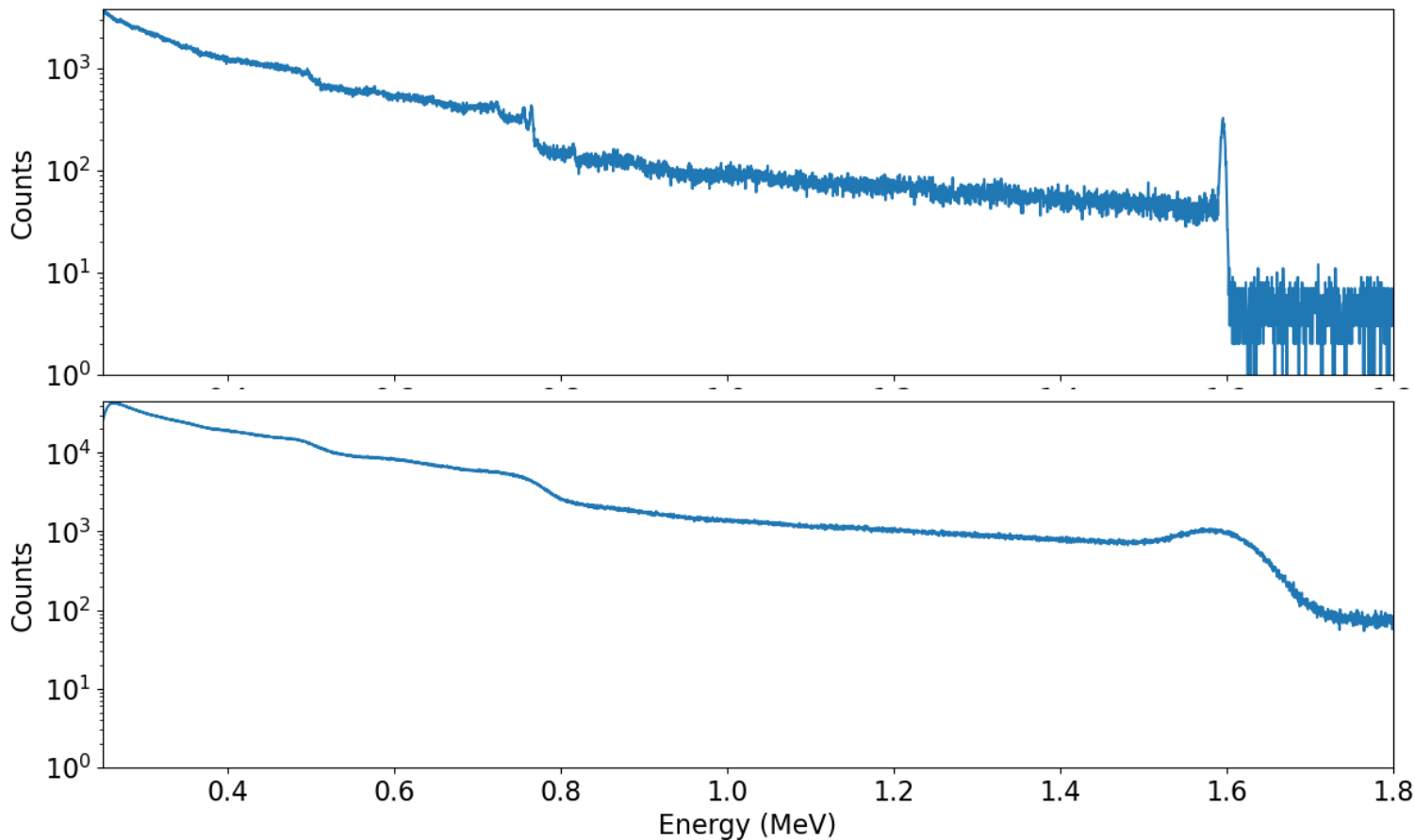
Events arriving:

- during peaking time are summed
- during dead time are rejected
- after are recorded normally

HPGe Dead Time	6 μs
LaBr3 Dead Time	0.016 μs



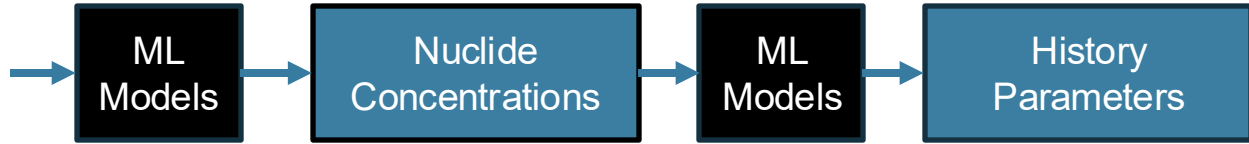
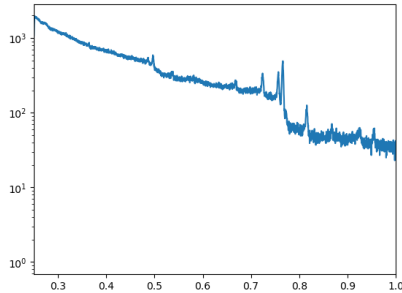
HPGe
(5d decay)



LaBr3
(1d decay)

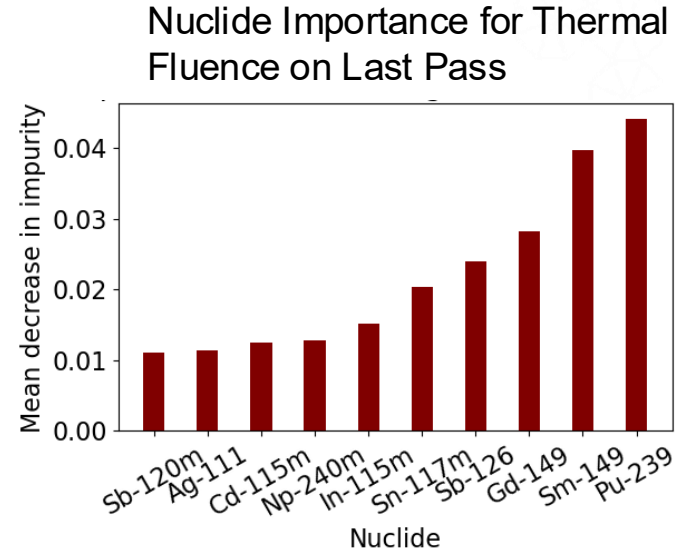
Pebble Prediction

Pebble Assessment: Two Layers



Nuclide Feature Importance (Ideal)

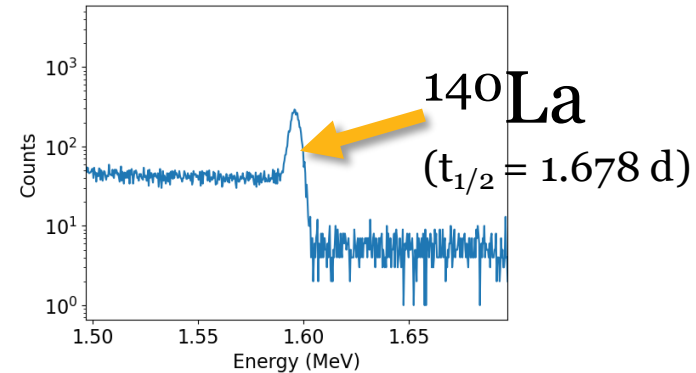
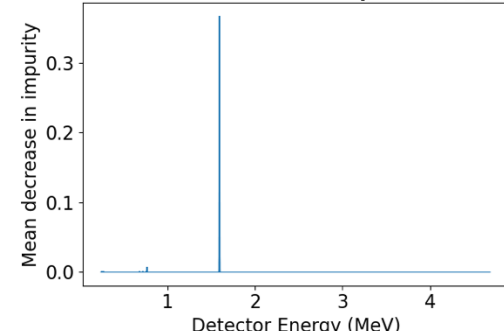
- A Random Forest Regressor (RFR) model is trained for each history parameter using *actual* concentrations from Serpent
- Informs which nuclides we need to predict
- Highlights physical trends and correlations



Spectrum Feature Importance

- For each nuclide, a RFR is fit using the actual spectral data
- The most important channels are determined
- Then, a NN model is trained and its hyperparameters optimized using only useful channels
- Rarely determined by that nuclide's own decay gammas

^{239}Pu Channel Importance



Final Model Training

- Neural Networks are optimized for each history parameter using the predicted important nuclides
- Number of hidden layers and layer sizes are coarsely optimized in large grid search

Random Forest Regressor Parameters

Max Depth	14
# Estimators	1000
Min. Feature Split	1
Bootstrapping	True

Neural Network Parameters

# Epochs	300
Batch Size	256
Activation	ReLU
Learning Rate	0.001
Network Sizes tried	(8, 2), (64, 8), (128, 16), (36)

History Metrics Performance

R^2 accuracy of model (HPGe – 40s measurement – 5d decay)

	LR on full Spectrum	NN on pred. nuclides	RFR on actual nuclides
%FIMA (burnup)	0.8021	0.9740	1.0000
Residence Time	0.8003	0.9671	0.9999
Passes	0.7974	0.9691	0.9999
Avg radial path on last pass	-1.9720	0.6643	0.9554
Thermal fluence on last pass	-3.9159	0.5576	0.8468

Baseline

Detected

Idealized

History Metrics Performance

R^2 accuracy of model (LaBr3 – 40s measurement – 1d decay)

	LR on full Spectrum	NN on pred. nuclides	RFR on actual nuclides
%FIMA (burnup)	0.7770	0.9856	1.0000
Residence Time	0.5851	0.9888	0.9999
Passes	0.5825	0.9848	0.9999
Avg radial path on last pass	-1.8001	0.8969	0.9554
Thermal fluence on last pass	-6.2175	0.8369	0.8468

Baseline

Detected

Idealized

Conclusions & Future Work

- Conclusions
 - Machine learning offers greater accuracy than linear regression
 - Timing resolution is more important than energy resolution
 - HPGe detectors may be unnecessarily expensive to run
- Future Work
 - Explore other detector choices & further optimize setup
 - Faster planar HPGe
 - Methods like pulse reconstruction or Compton suppression
 - Run different measurement times (multiples of 40s)
 - More extensive ML model optimization



Questions on section?



3) Predicting Core State from Discharge Fuel

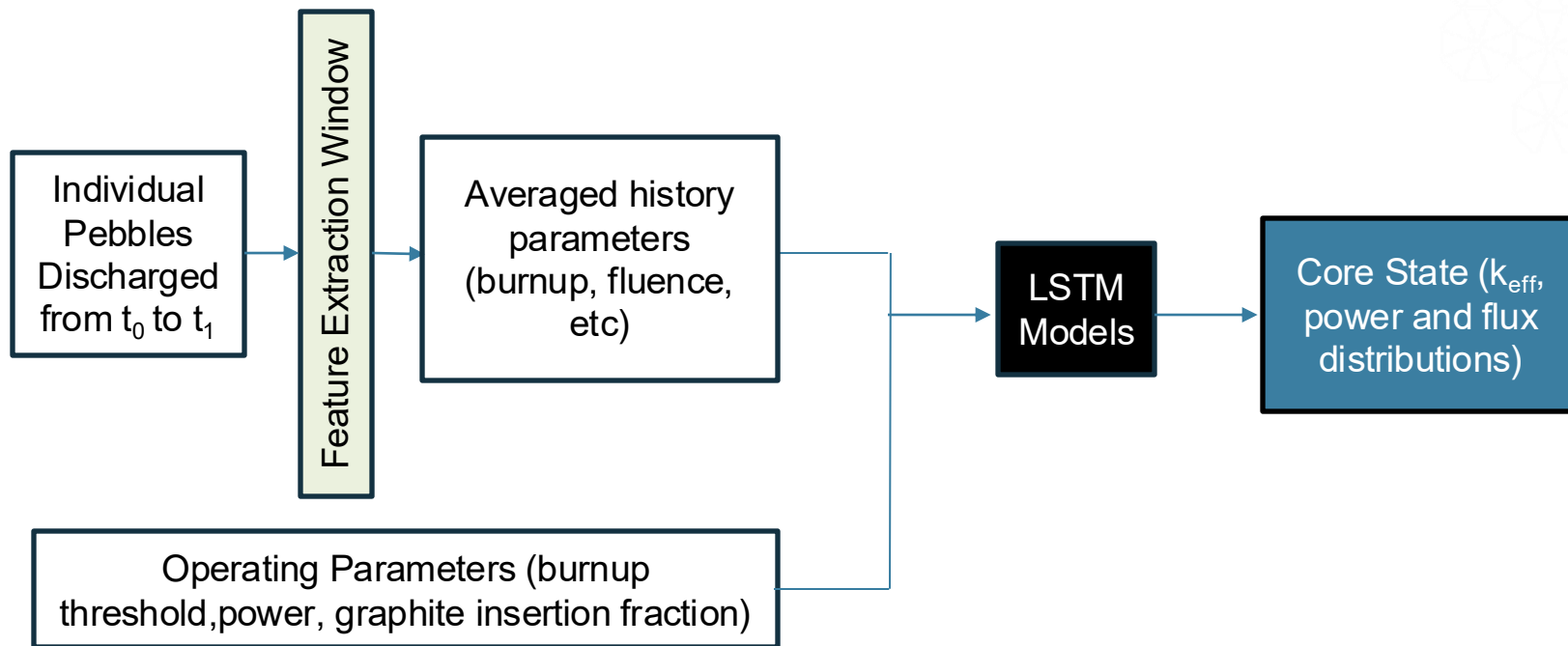
Core State Prediction

- Objective: Predict the overall core state using the current operating parameters and the predictions made from discharge pebbles
- Challenges and Considerations:
 - No in-core measurements possible
 - Latency: Changes to pebble insertion take at least 10 iterations/65 days to fully propagate through the core
 - Reaching new equilibria after operation change takes longer
 - Extracting meaning from thousands of discharge pebbles
 - Expensive to simulate long operating sequences in hyper-fidelity

Discharge Pebble Feature Extraction

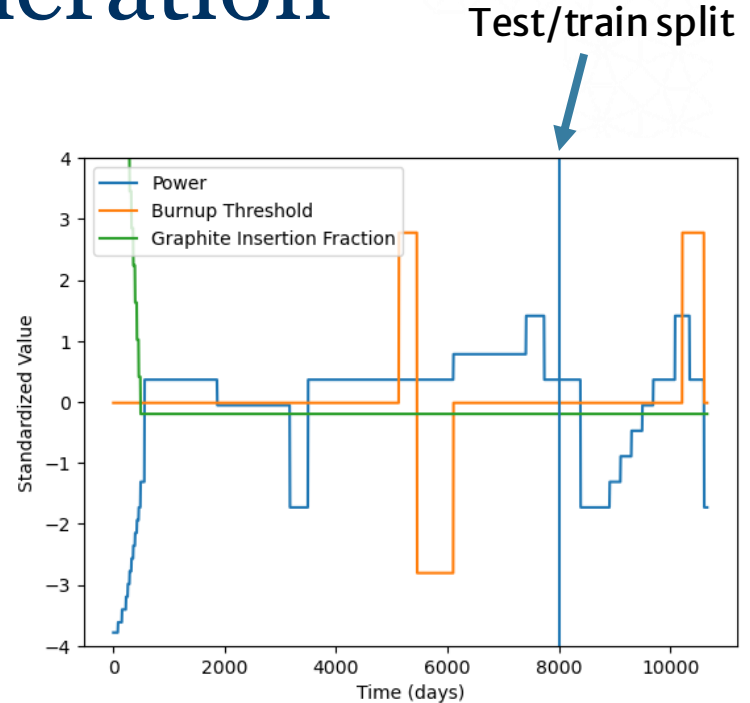
- Core simulators handle pebble movement in coarse steps
 - Exact order of pebbles coming out during a timestep largely meaningless
 - Using each discharge pebble as input would require massive networks
- Simple approach: Average the predicted features of all pebbles discharged during one time step
- Radial zone approach: Using its predicted average radial path, associate each pebble with a radial zone in the core, and take averages for each zone

Core State Prediction

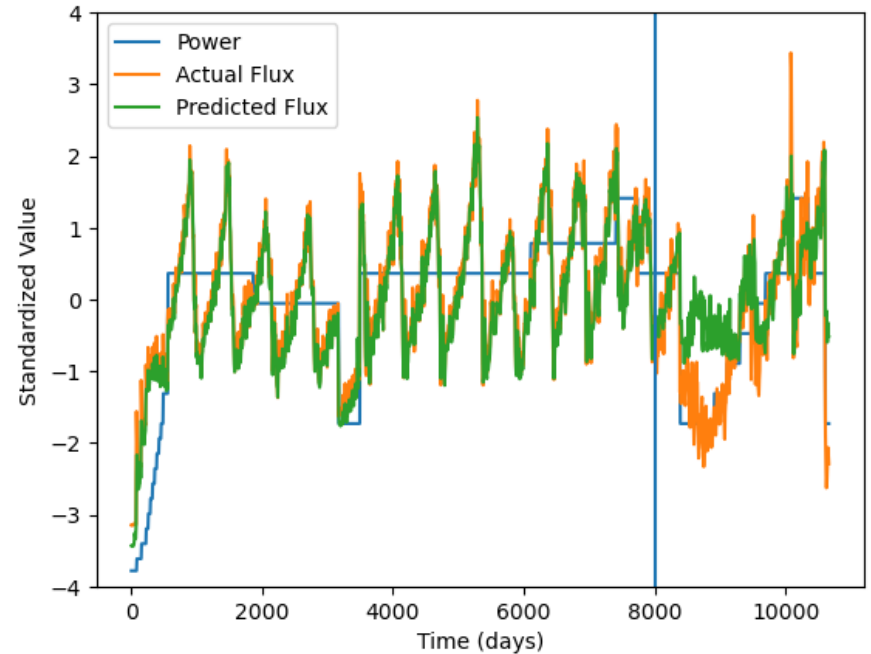
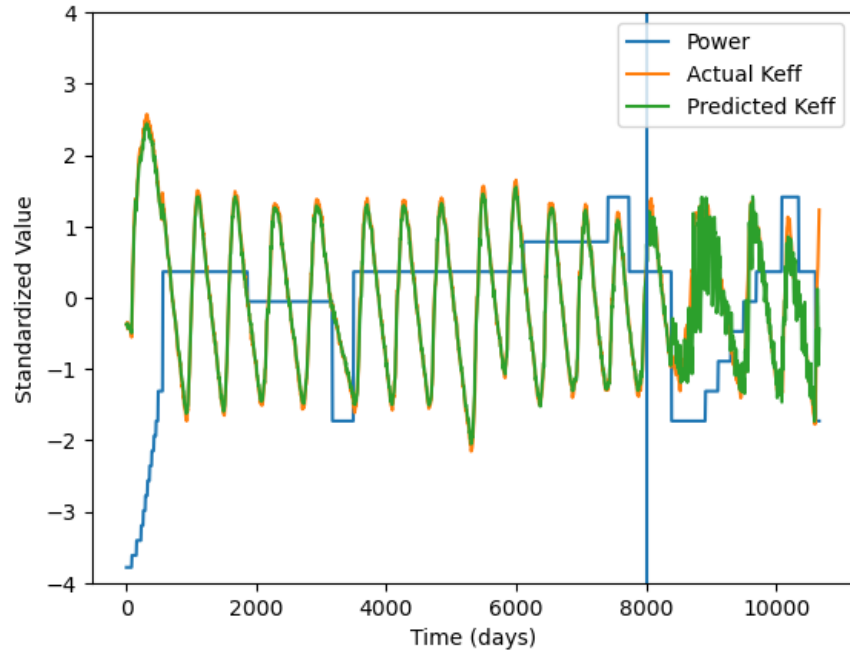


Time Series Data Generation

- Generating long operating sequences in HxF expensive
- For now, a zone-based core simulator is used (PEARLSim)
 - Explained in next section
- Features are extracted from zone-averaged pebbles
- Mock startup sequence generated with some power & burnup threshold perturbations

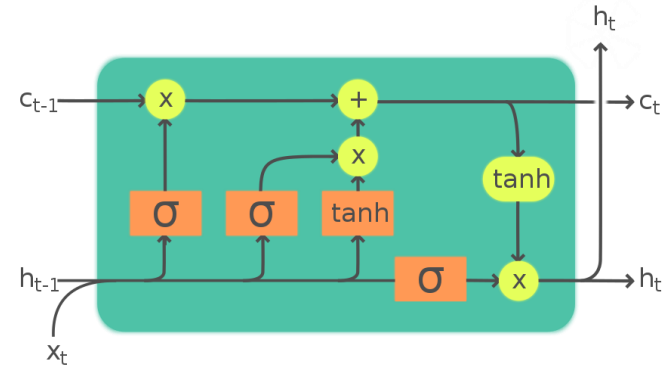


RFR single time step results



Sequence Prediction with LSTM

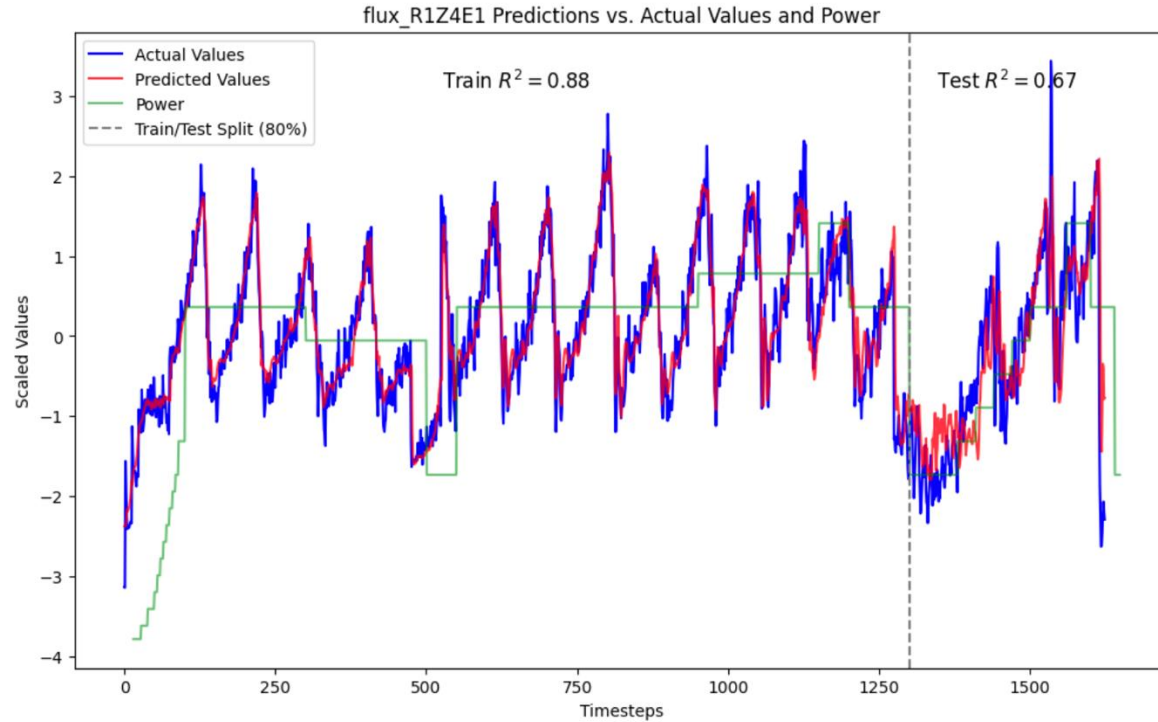
- Use of a Long Short-Term Memory (LSTM) network allows model to use input vectors from previous timesteps
- Better handles latency and oscillation of reactor response
- Additional hyperparameter to train in length of look-back window



Legend:

Layer	ComponentwiseCopy	Concatenate

LSTM results



Conclusions & Future Work

- Conclusions (early)
 - So far, little difference in performance between averaged and radial-zone-averaged features
- Future Work
 - Removing unphysical k_{eff} /flux oscillation in data generation
 - Apparent consequence of xenon building up in fresh fuel on timescale shorter than one time step
 - Including rod worth as target
 - Generating pebble data in hyperfidelity instead of zones
 - Forecasting core state X time steps into the future



Questions on section?

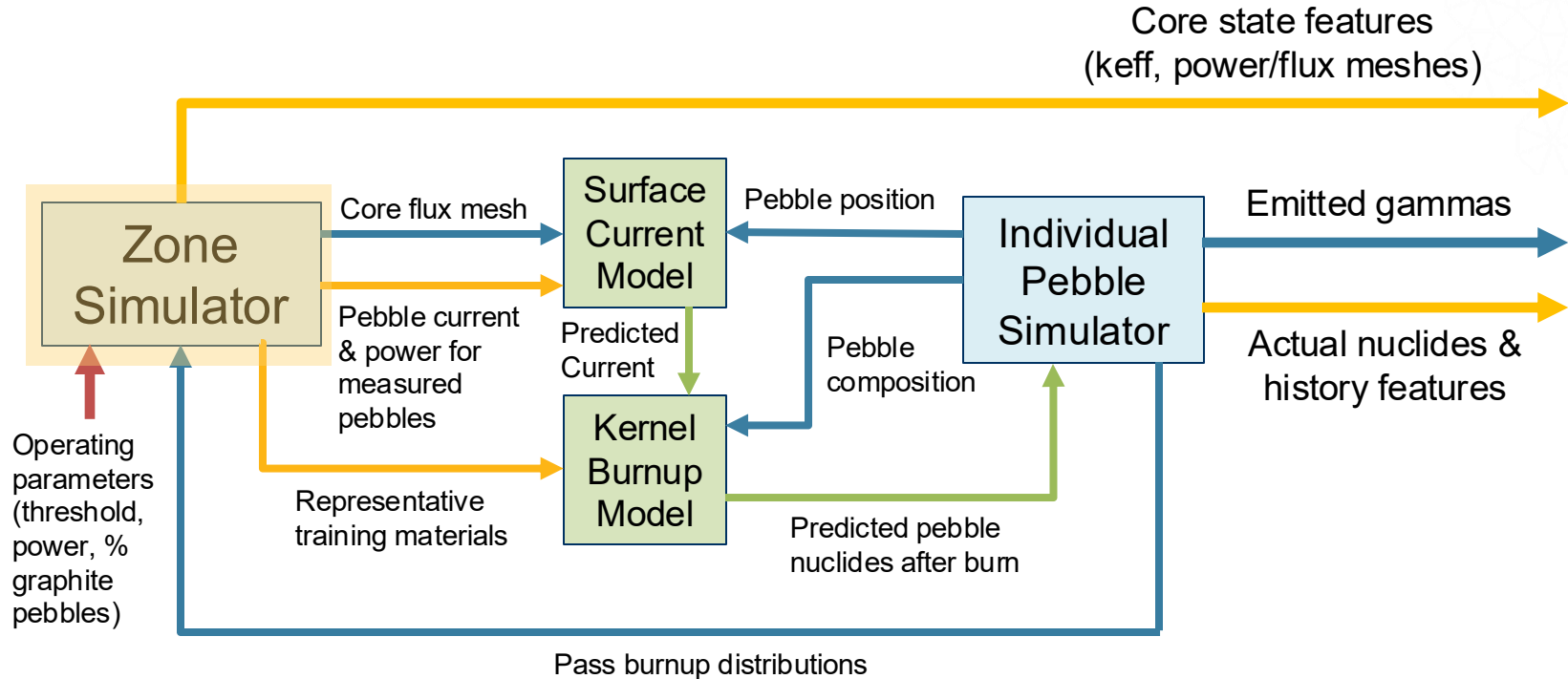


4) PEARLSim: Using ML for hyperfidelity core simulation

Hyperfidelity PBR Modeling

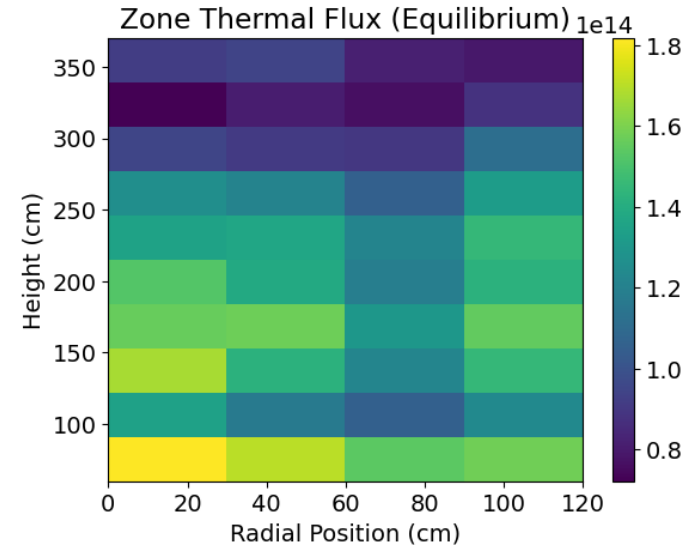
- Objective: Speed up hyperfidelity PBR calculations to generate longer training sequences for core state model
- Challenges and Considerations:
 - Simulating 250,000 unique pebbles computationally expensive
 - Very high memory footprint
 - Many particle histories needed for acceptable MC error
- Can ML be used to predict accurate pebble-wise flux and depletion coefficients to circumvent the need for hyperfidelity MC?

PEARLSim: Hybrid approach

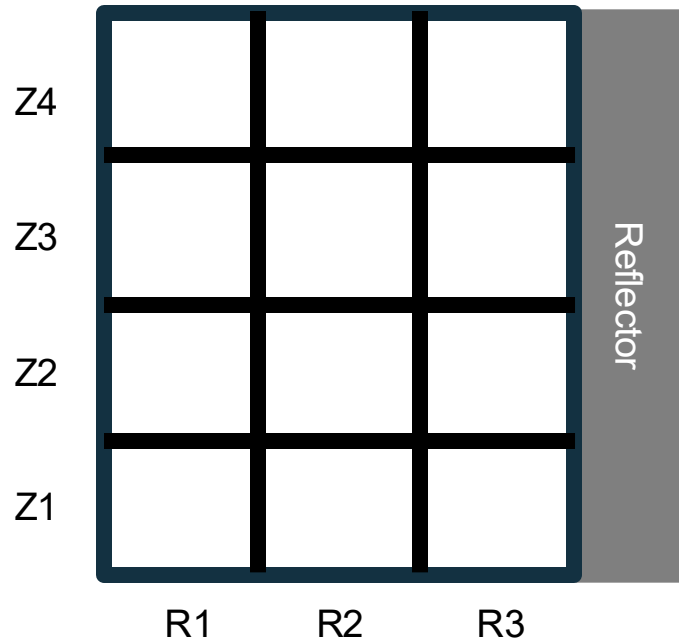


Zone Model for Core-Wide data

- Core volume divided into zones with similar flux
- Groups of pebbles are given average fuel compositions
 - ~400 materials instead of 250k
- Accurately captures core-wide properties like k_{eff} and the distribution of flux/power
- Can simulate 6.5 days of operation in 5-15 minutes



Zone-Based Core Simulator



- Each zone has a certain proportion of different fuel types
- Pebble positions in zone are randomly assigned a fuel type
- Average flux for each group computed for depletion

Zone Z1R3:

43% Graphite Pebbles

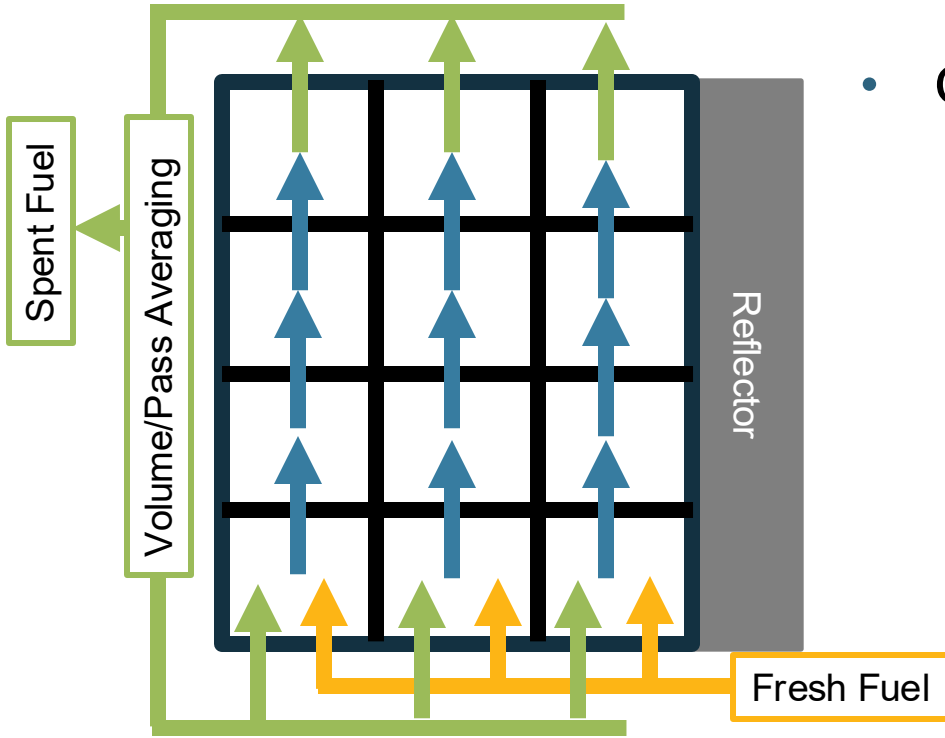
7% Group 1 Fuel Pebbles (FuelZ1R3G1)

7% Group 2 Fuel Pebbles

...

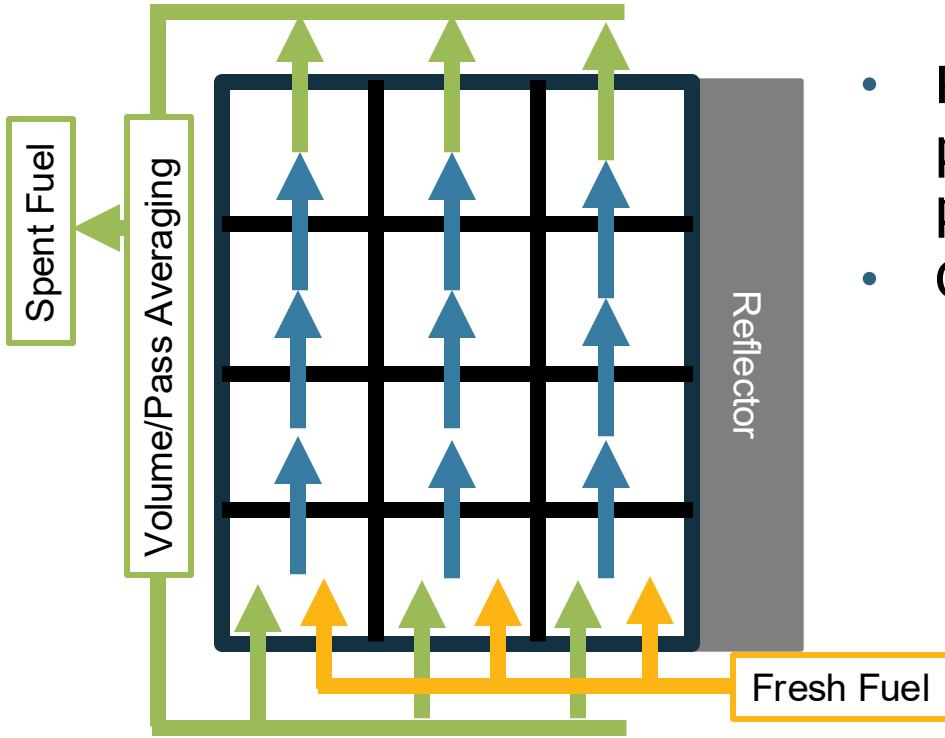
7% Group 8 Fuel Pebbles

Zone-Based Core Simulator



- On every iteration:
 - Material data and proportions are moved to zone above
 - Materials at the top are:
 - Potentially discarded based on burnup
 - Sorted in burnup groups and volume-averaged
 - Averaged materials and fresh fuel is reinserted at the bottom

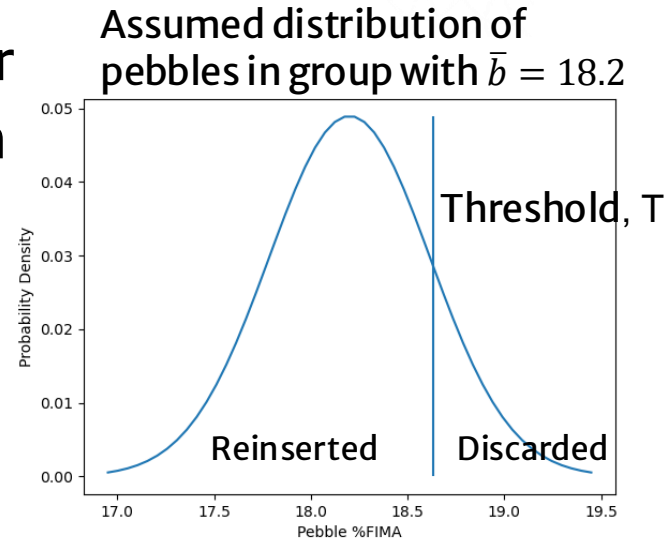
Zone-Based Core Simulator



- Reminder: changes in operating parameters take a while to fully propagate through the core
- Operating Parameters:
 - Power
 - Fuel insertion fractions
 - Burnup discard threshold
 - Burnup time step

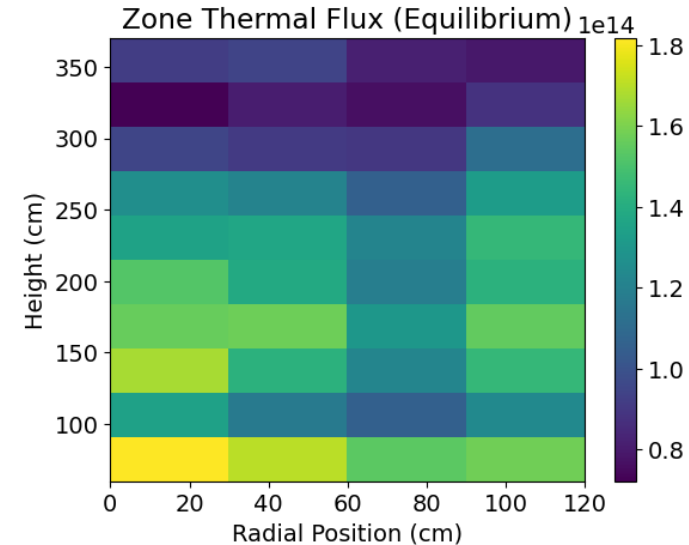
Pebble Discarding Criteria

- T, Threshold for %FIMA is set by operator
- Average burnup, $\bar{b}_g$, is calculated for each fuel group at the top of the core
- Pebble burnup $B \sim N(\bar{b}_g, \sigma_g)$
- Fraction of pebbles kept $\equiv P(B < T)$
- Standard deviation σ_g for group
 - Currently assumed from HxF result
 - Later will be acquired by hyperfidelity pebble model running in parallel

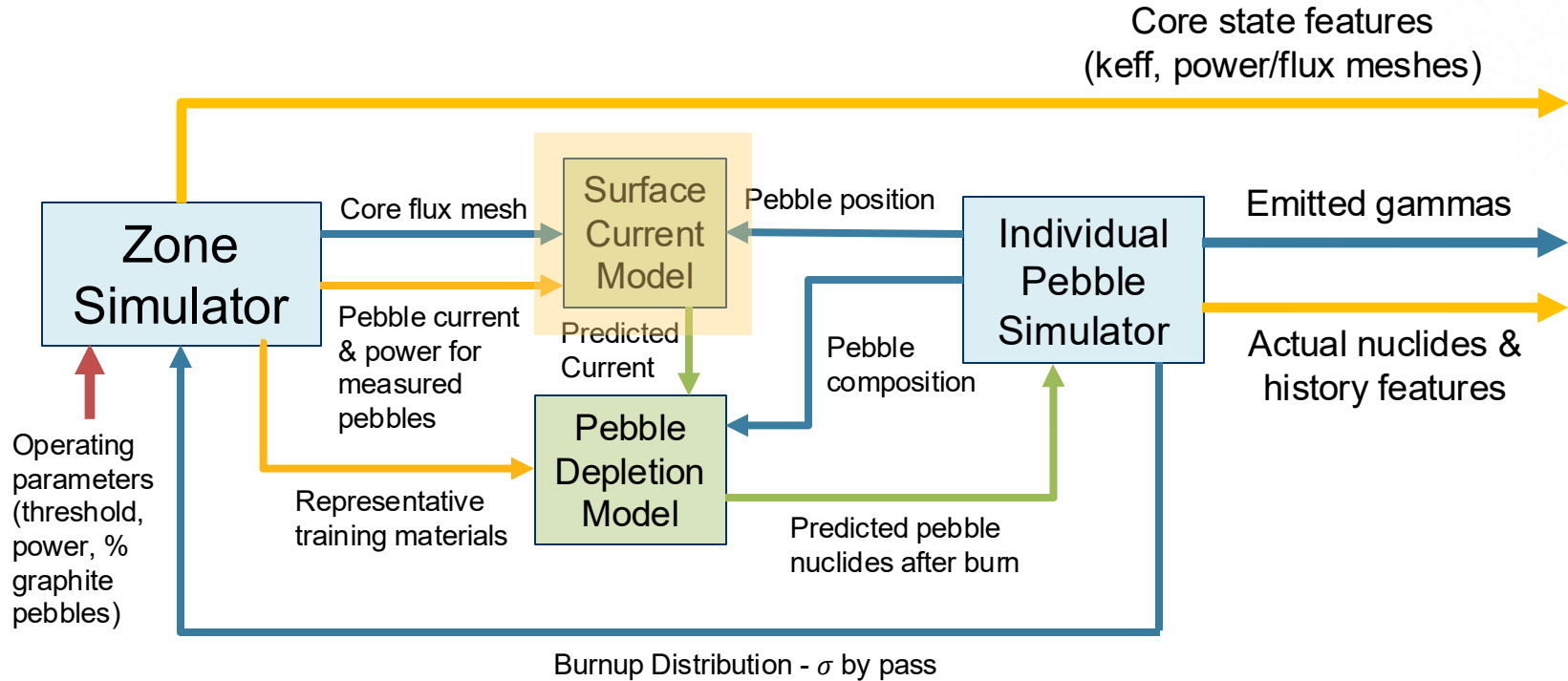


Zone-Based Core Simulator Output

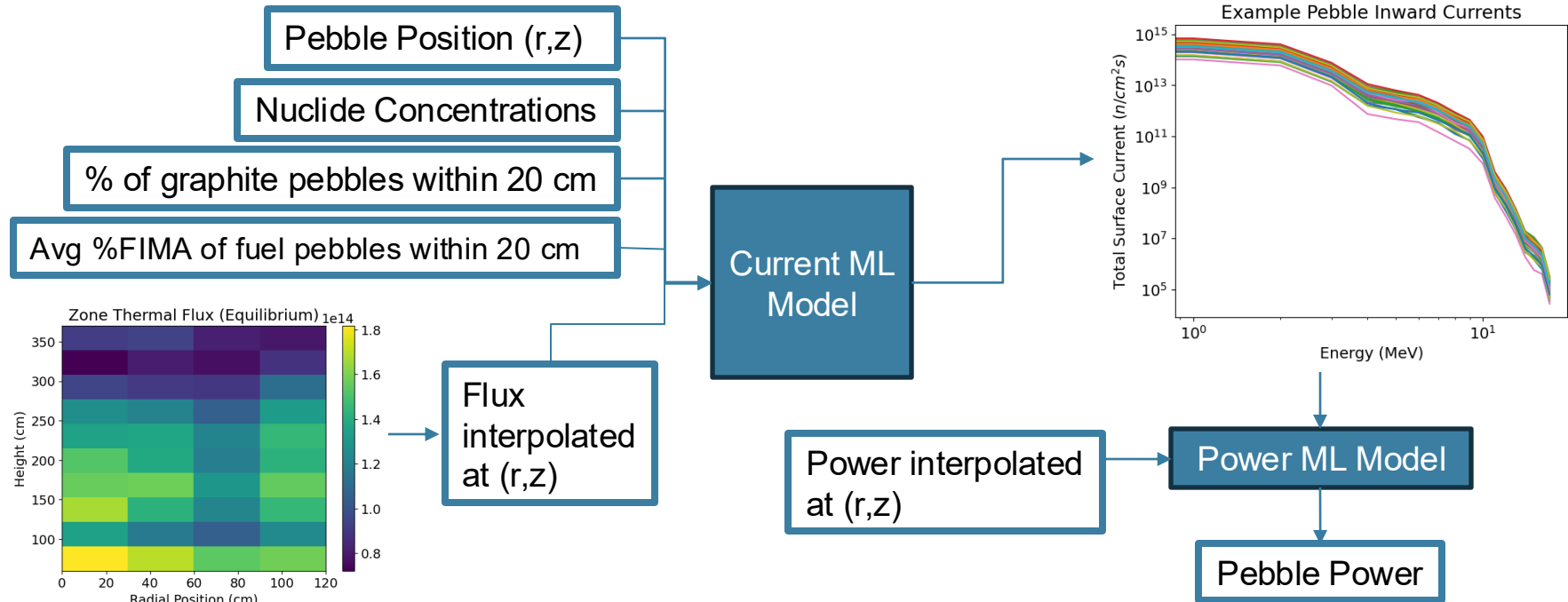
- Core-wide parameters as a function of time and operating inputs
 - Keff
 - Power profile by zone
 - Flux profile by zone, energy group
- Zone/burnup group averaged discharge compositions



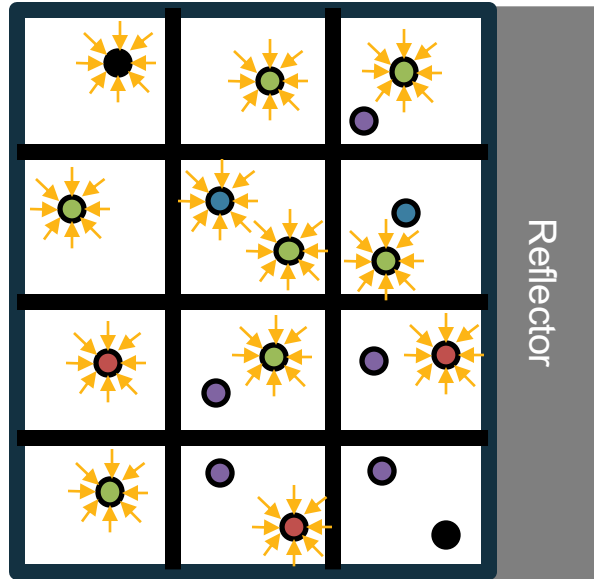
PEARLSim



Pebble Surface Current Model

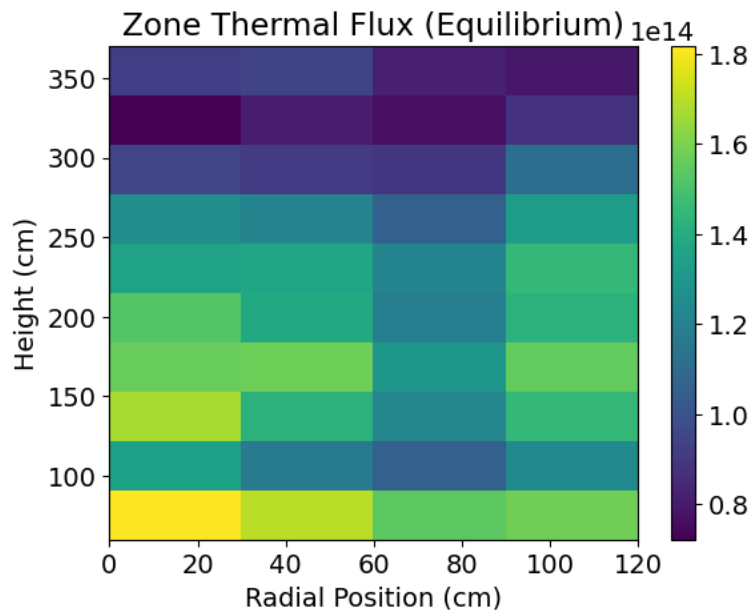


Current Model Data Generation



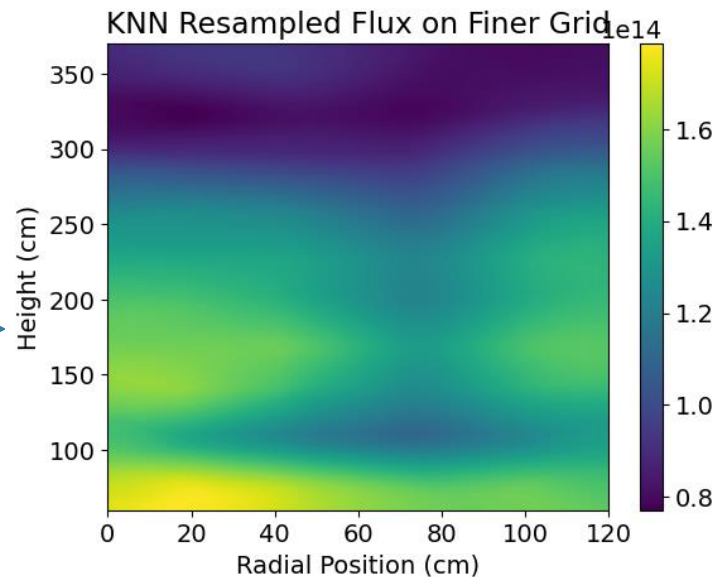
- Re-run select few of Serpent steps throughout operation history
- Add inward current tallies to outer surface of 5000 random pebbles
 - 18 energy groups
- Extract per pebble power from pebble bed
- Run with much higher numbers of particles (3000 cycles,)
- Takes 8 hours with 4 nodes (32 core)

Feature Extraction: Flux/Power mesh interpolation



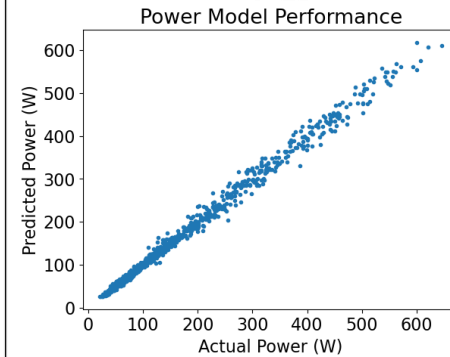
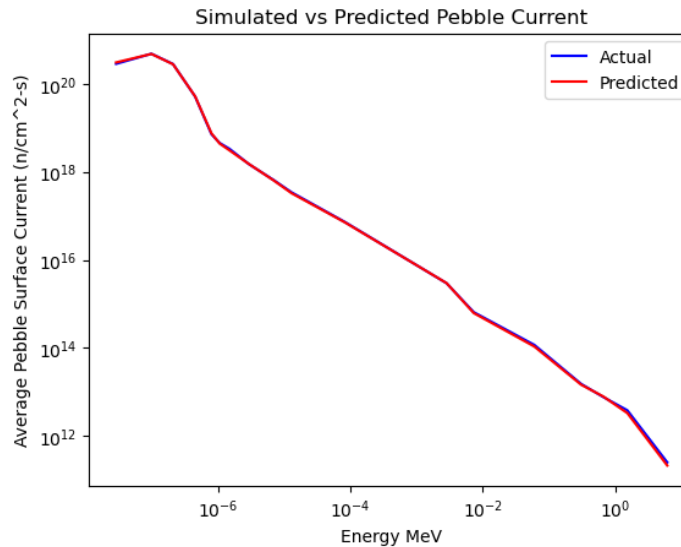
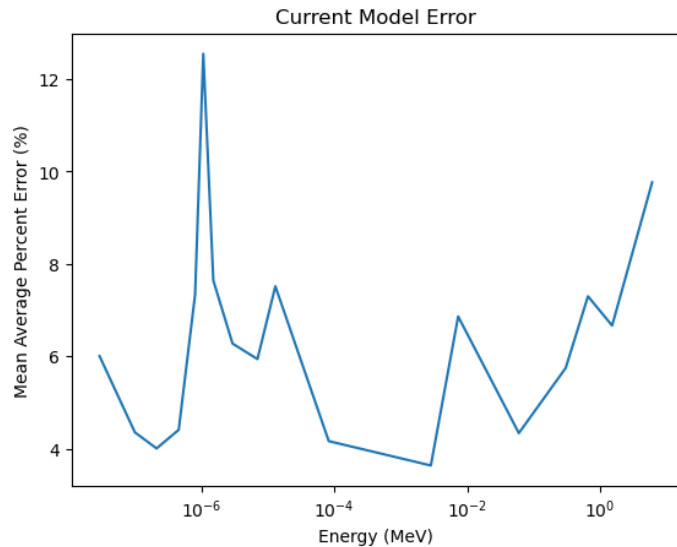
Raw data generated on each iteration

KNN interpolation at
each pebble (r, z)



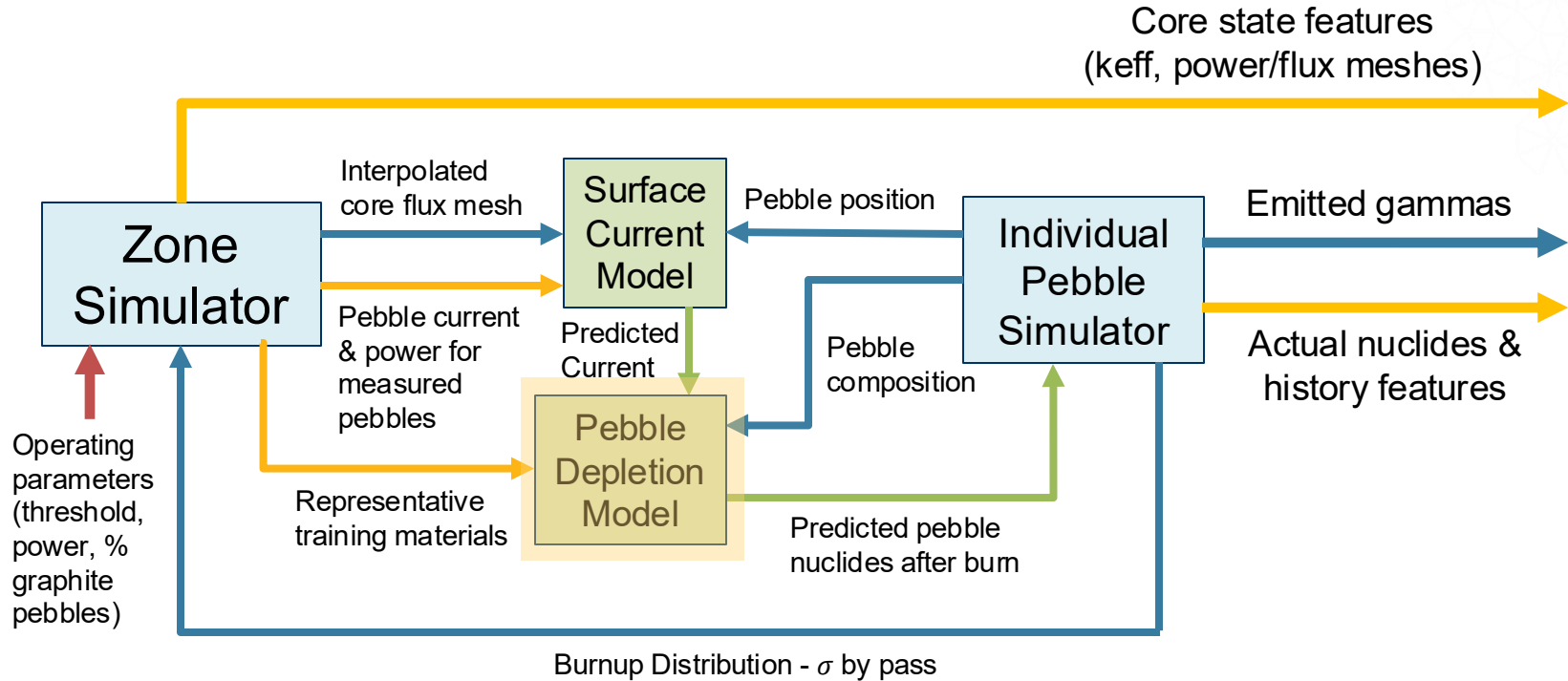
Single value feature computed for
each pebble

Pebble Surface Current Model Performance

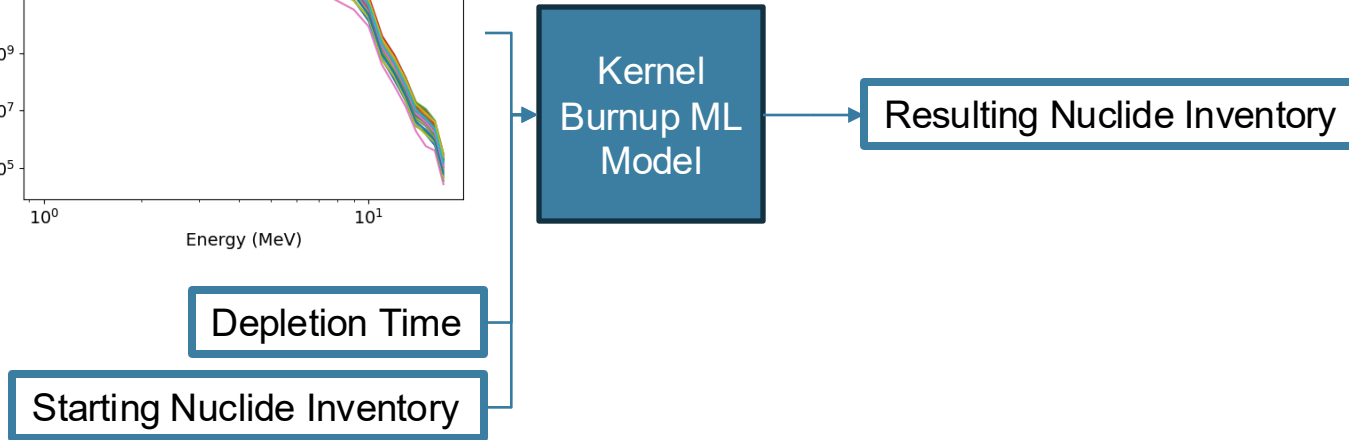
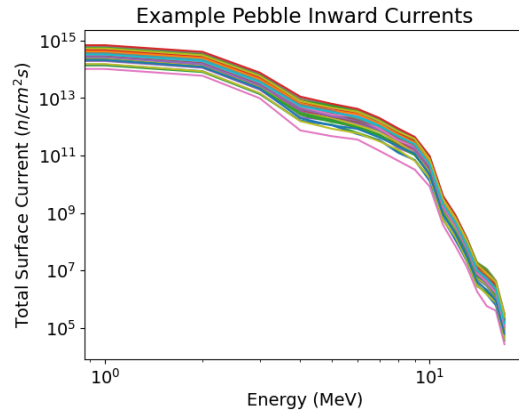


Power MAPE: 7.41%

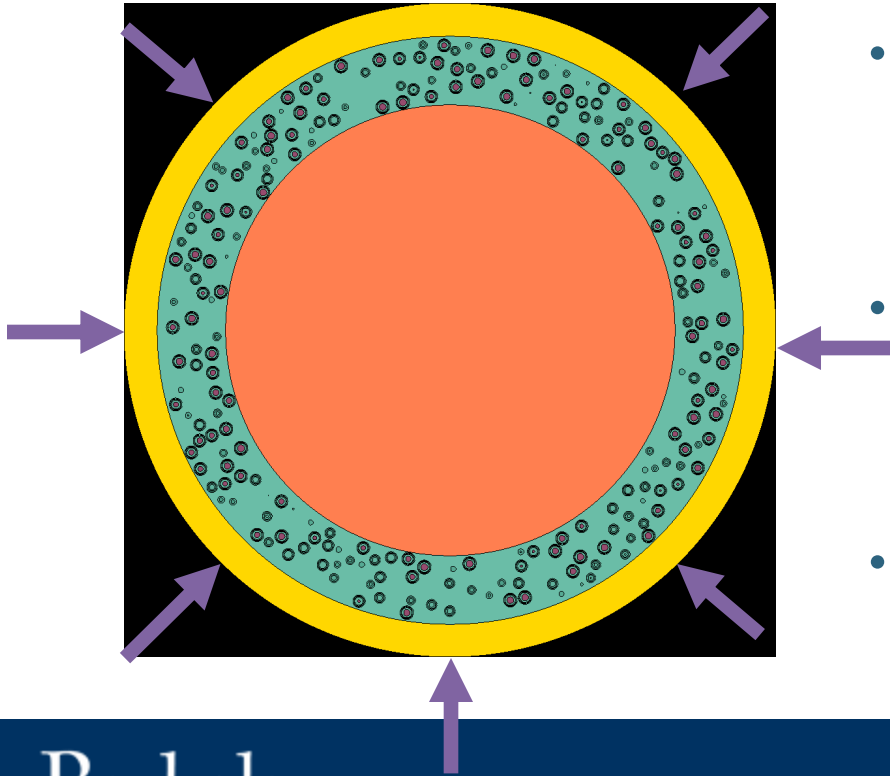
PEARLSim



Kernel Depletion Model

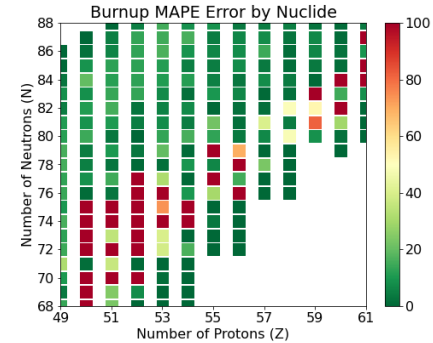
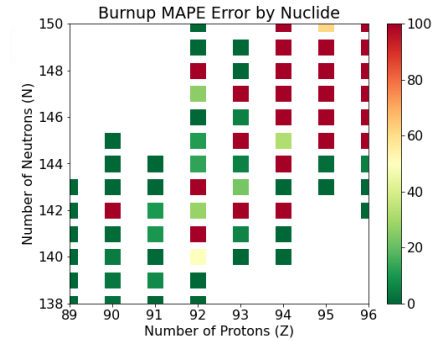
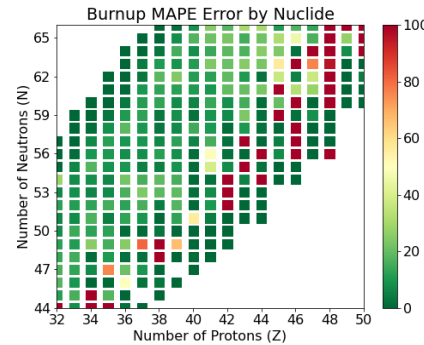
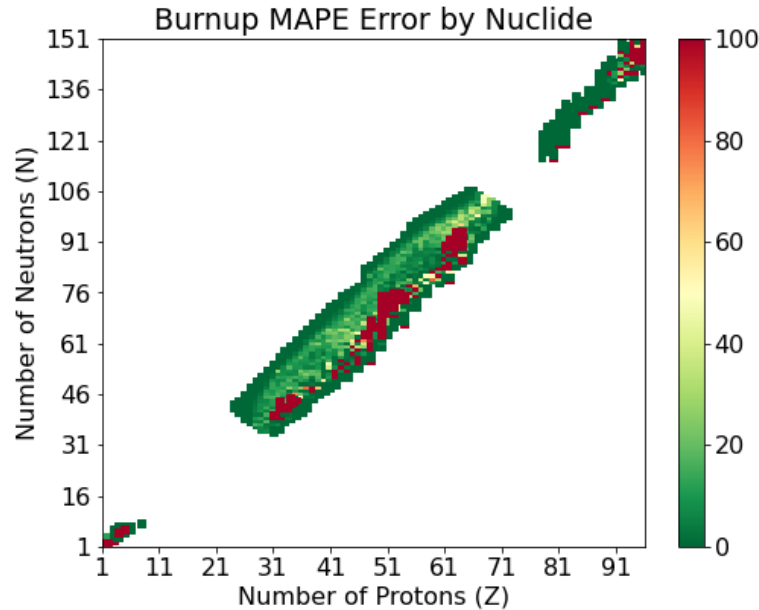


Data Generation

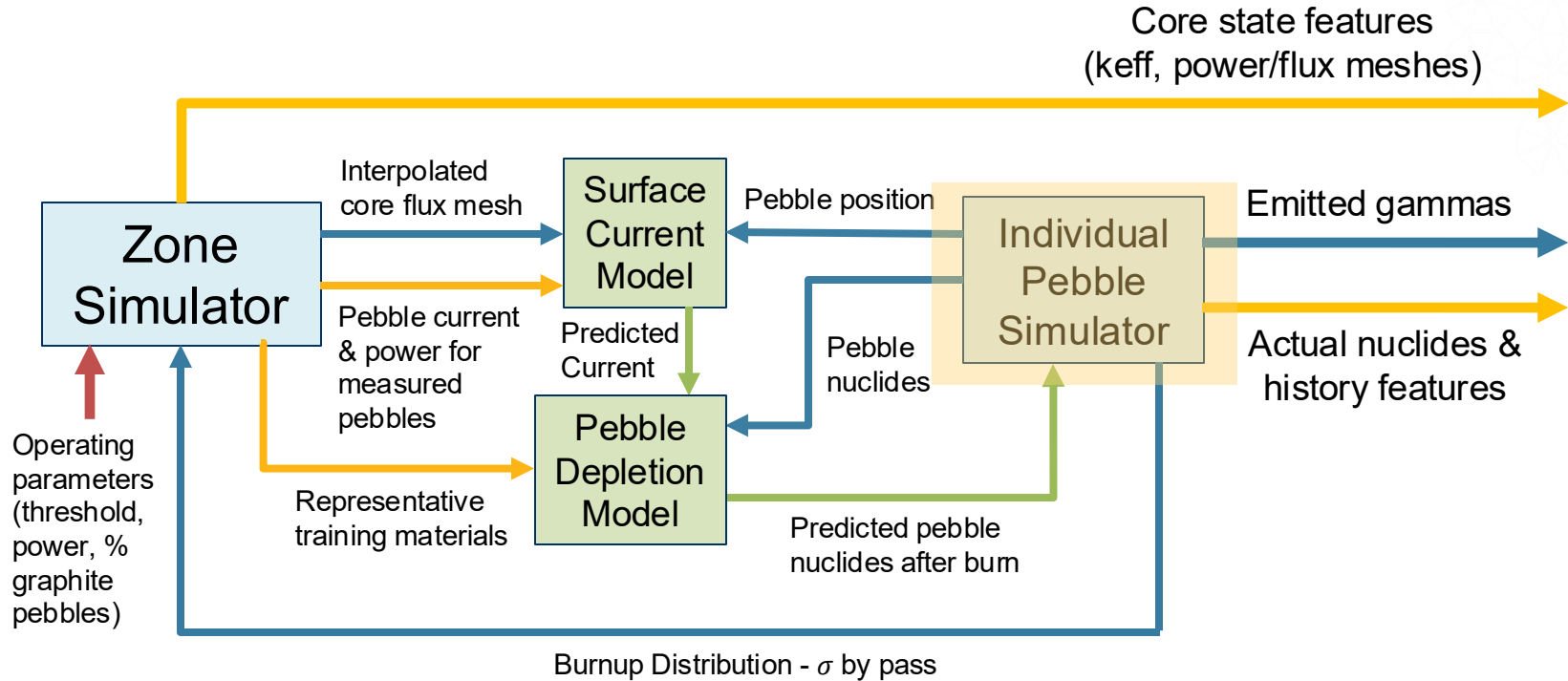


- Random materials are sampled from throughout the zone simulator's history to get starting concentrations
- A pebble with that material is modeled in Serpent in isolation with randomly selected current distributions and power values
- The final concentration after some depletion time is extracted

Depletion Model Performance

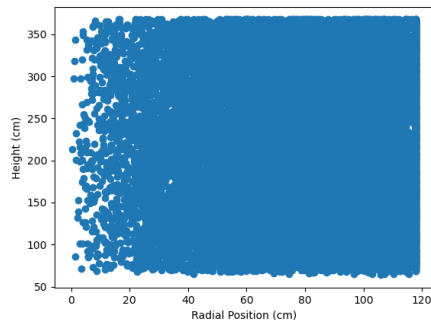


PEARLSim

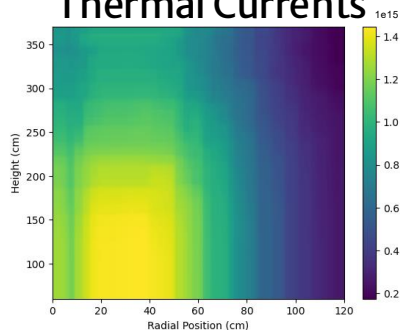


Individual Pebble Model

Pebble Positions



Predicted Pebble Thermal Currents



1. Update pebble locations according to velocity profile
2. Predict current and power for each pebble
3. Update nuclide inventories with depletion model
4. Save discharge pebble data
5. If enabled, provide burnup distribution data as feedback to zone simulator

Conclusions & Future Work

- Conclusions
 - Generating pebble-wise flux informed by zone simulator flux is possible
- Future Work
 - Current Model
 - Increase number of energy groups in current model
 - Depletion Model
 - Use ML instead to predict cross sections or very fine flux
 - Incorporate Bateman Equation solver from OpenMC
 - Explore potential accuracy gains
 - Much smaller substeps feasible
 - Predictor-corrector method that accounts for movement



Questions on section?

Final Conclusions

Machine learning is valuable for Pebble Bed Reactor (PBR) operation because it enables faster and more accurate discharge fuel measurement and enhances core simulation capabilities

Key Insights

- We can use ML to predict a lot of information from gamma spectra detected from fuel, even with a lot of noise or poor energy resolution
 - LaBr3 currently performs better with less shielding and few peaks
- These pebble predictions can be used with ML to predict core state
- We can use ML to generate hyperfidelity data faster

Thank you!

Acknowledgements

- You all for participating in this and your continued support
- Jaewon Lee and Joanna Szornel for insight on detector setup and modeling
- Chris Campbell for insight on detector setup realism and industry's approach
- Tatiana Siaraferas and Ludovic Jantzel for extensive collaboration on reactor physics and data generation
- Dino Bellugi and Jackson Temple for time series modeling with LSTM
- Berkeley Research Computing program at the University of California, Berkeley for providing the Savio computational cluster resource (supported by the UC Berkeley Chancellor, Vice Chancellor for Research and Chief Information Officer).