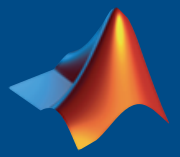
AI with Model-Based Design:

Virtual Sensor Modeling

MathWorks®

Agenda

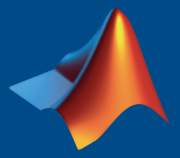
I. Introduction to AI with Model-Based Design
II. Deep dive of Virtual Sensor Modeling workflow
III. Resources for further learning



MathWorks®

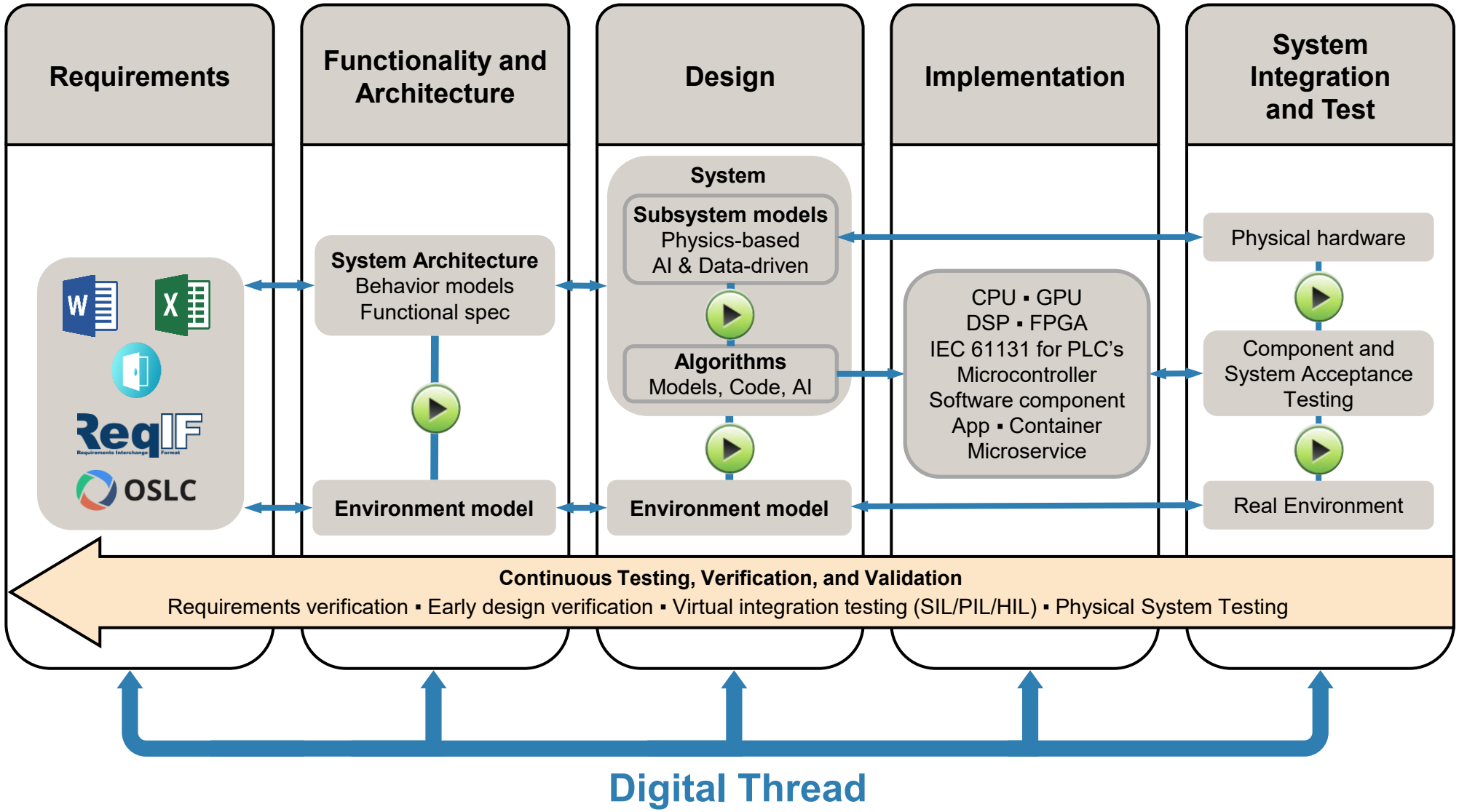
Agenda

I. Introduction to AI with Model-Based Design
II. Deep dive of Virtual Sensor Modeling workflow
III. Resources for further learning

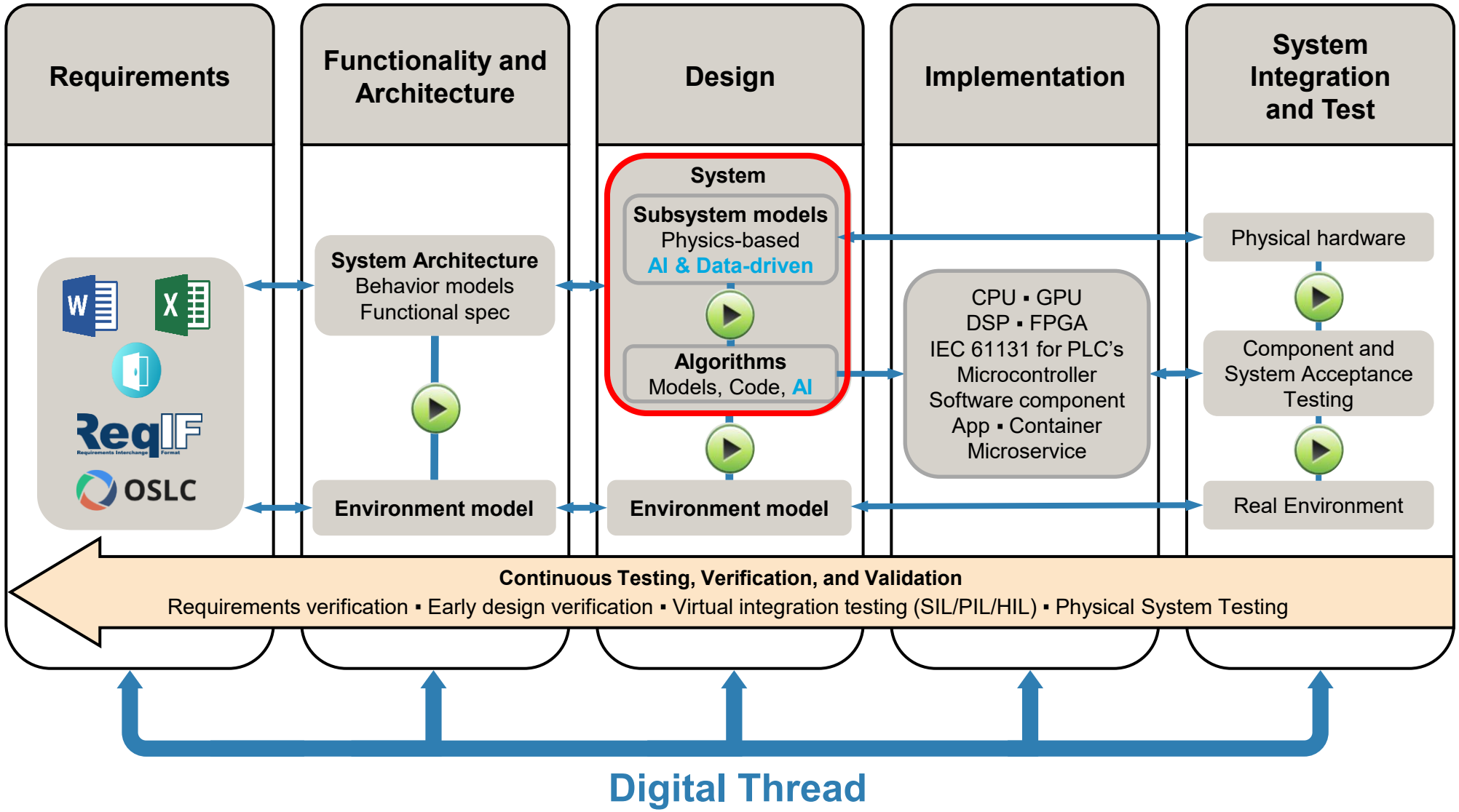


MathWorks®

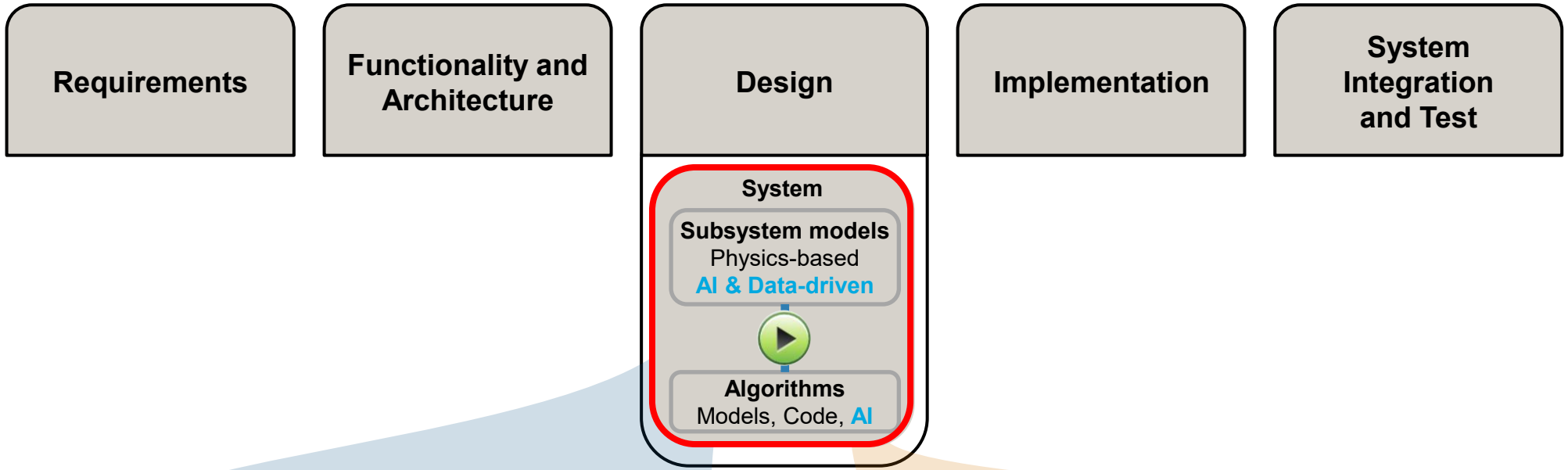
Model-Based Design



Integrating AI into Model-Based Design



AI for Component Modeling and Algorithm Development



AI for component modeling

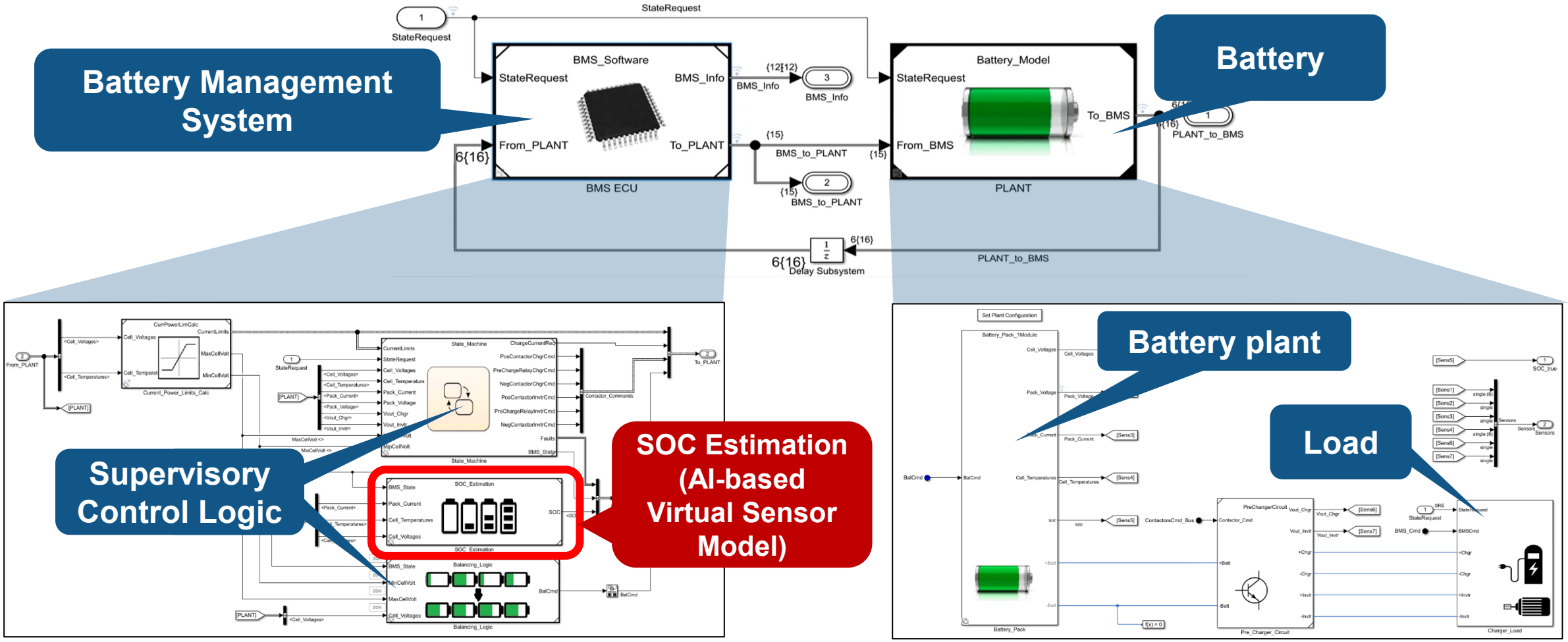
- System Identification
- Reduced Order Modeling (ROM)

AI for algorithm development

- Perception and Sensing **Virtual Sensors**
- Planning and Controls

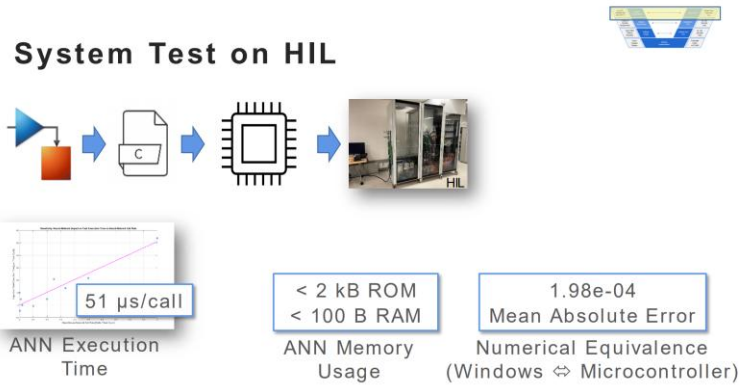
AI is often part of a larger system

Simulate and test all your components together in Simulink

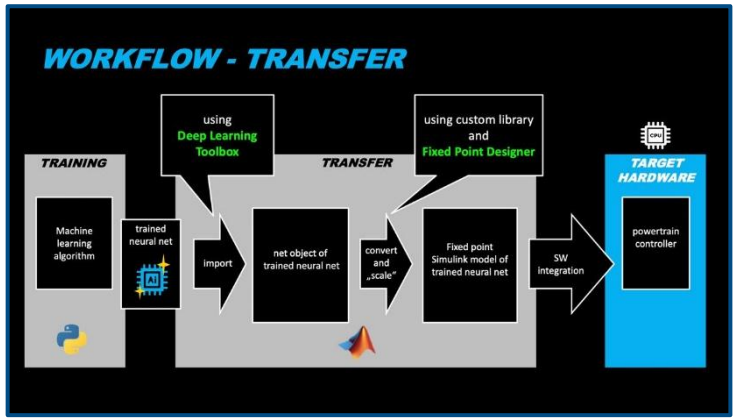


Virtual Sensor User Stories

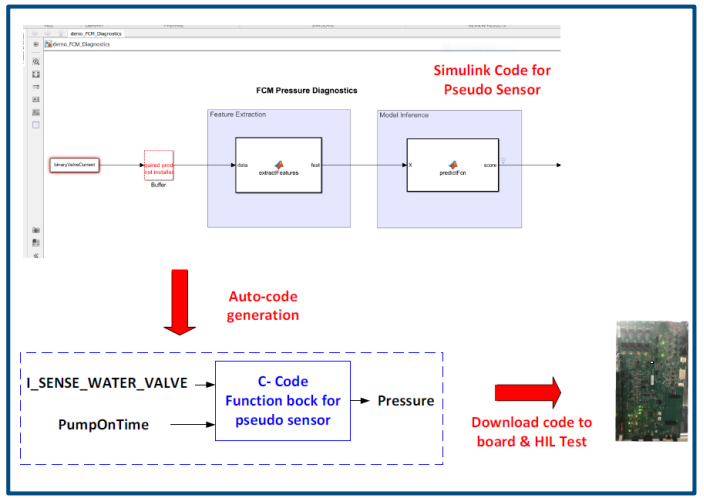
Gotion



Mercedes-Benz



Coca-Cola



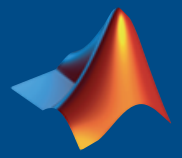
Gotion:
Battery Pack SOC Virtual Sensor with a Neural Network

Mercedes-Benz:
Engine Piston Virtual Pressure Sensors with Deep Neural Networks

Coca-Cola:
Beverage Dispenser Virtual Pressure Sensor with Machine Learning

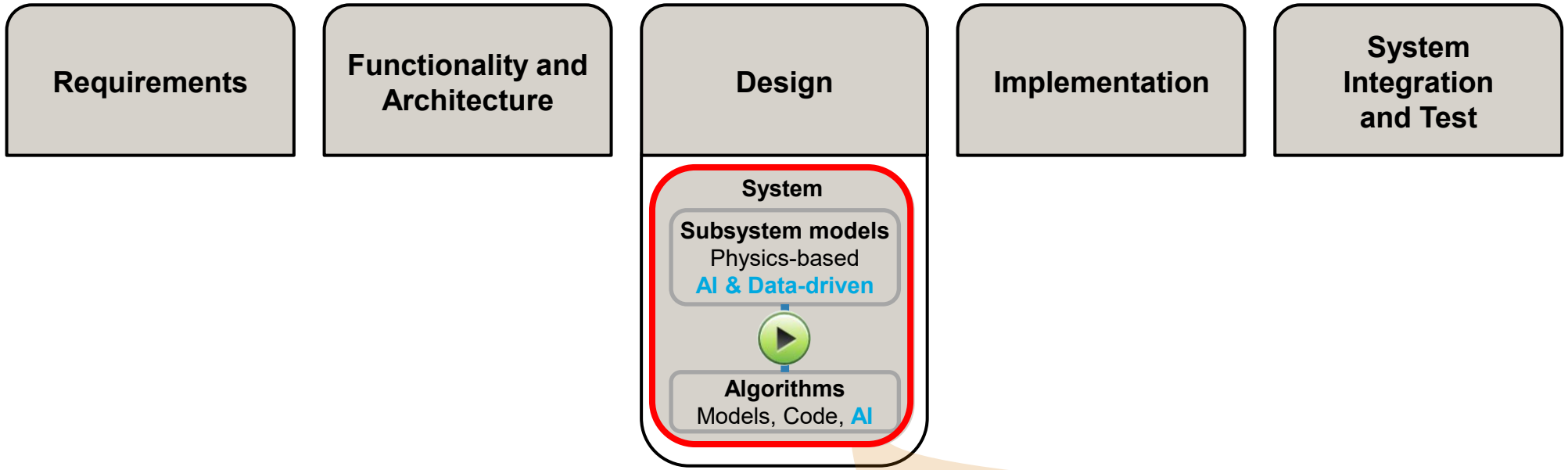
Agenda

I. Introduction to AI with Model-Based Design
II. Deep dive of Virtual Sensor Modeling workflow
III. Resources for further learning



MathWorks®

Focus today



AI for algorithm development

- Perception and Sensing **Virtual Sensors**
- Planning and Controls

A Virtual Sensor mimics a physical sensor using data from other measurements to estimate the quantity of interest

Why are Virtual Sensors relevant?

A **physical sensor** may be:

- Expensive
- Slow
- Noisy
- Unreliable
- Not feasible
- Unmanufacturable
- Degrading over time
- Requiring redundancy
- etc.

Data-driven vs. first-principles modeling

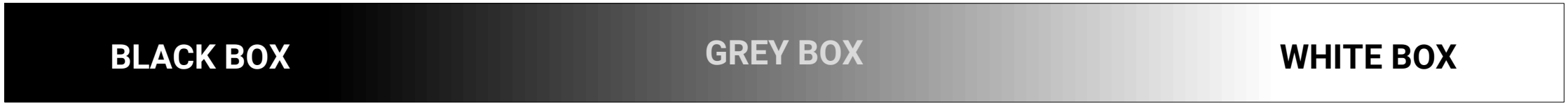
Data-driven models and first-principles models can co-exist

DATA-DRIVEN MODELS

Statistics, optimization, AI

FIRST-PRINCIPLES MODELS

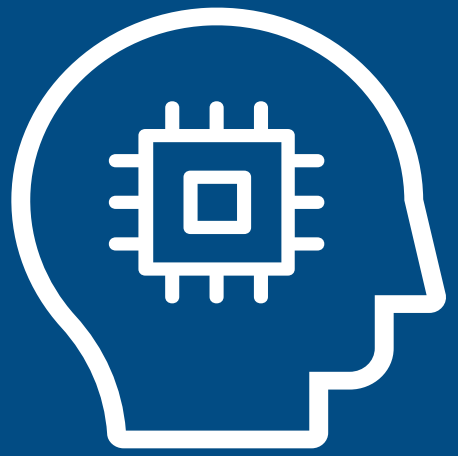
Physics, math, domain knowledge



The AI megatrend

ARTIFICIAL INTELLIGENCE

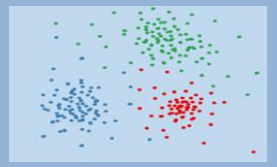
Any technique that enables machines to mimic human intelligence



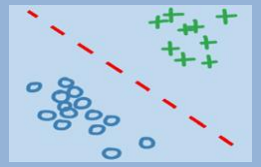
MACHINE LEARNING

Statistical methods that enable machines to “learn” tasks from data without explicitly programming

UNSUPERVISED LEARNING
(No Labeled Data)

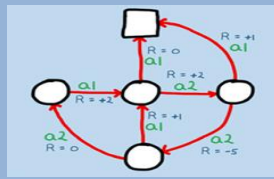


SUPERVISED LEARNING
(Labeled Data)



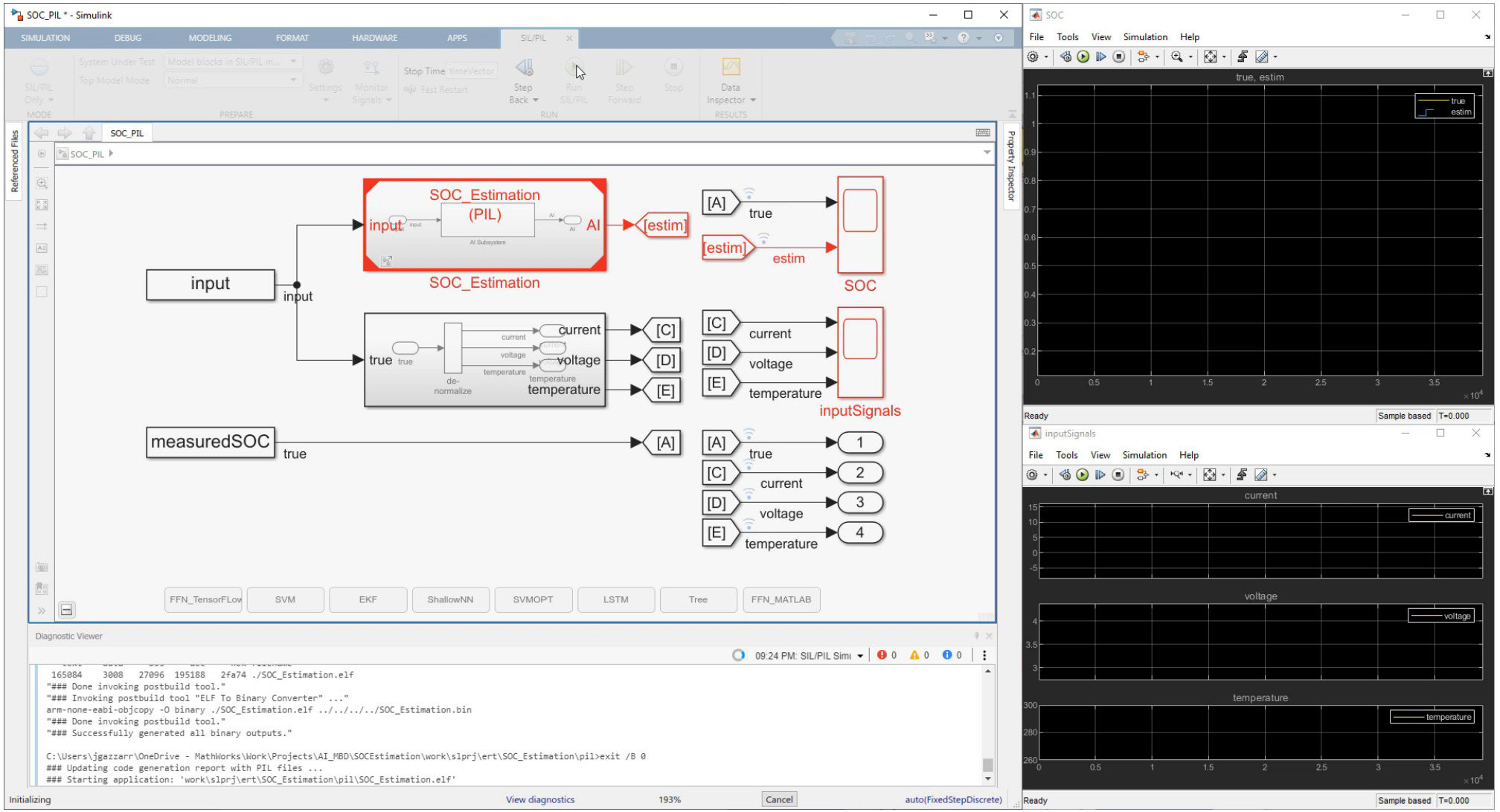
DEEP LEARNING
(Neural networks with many layers)

REINFORCEMENT LEARNING
(Interaction Data)

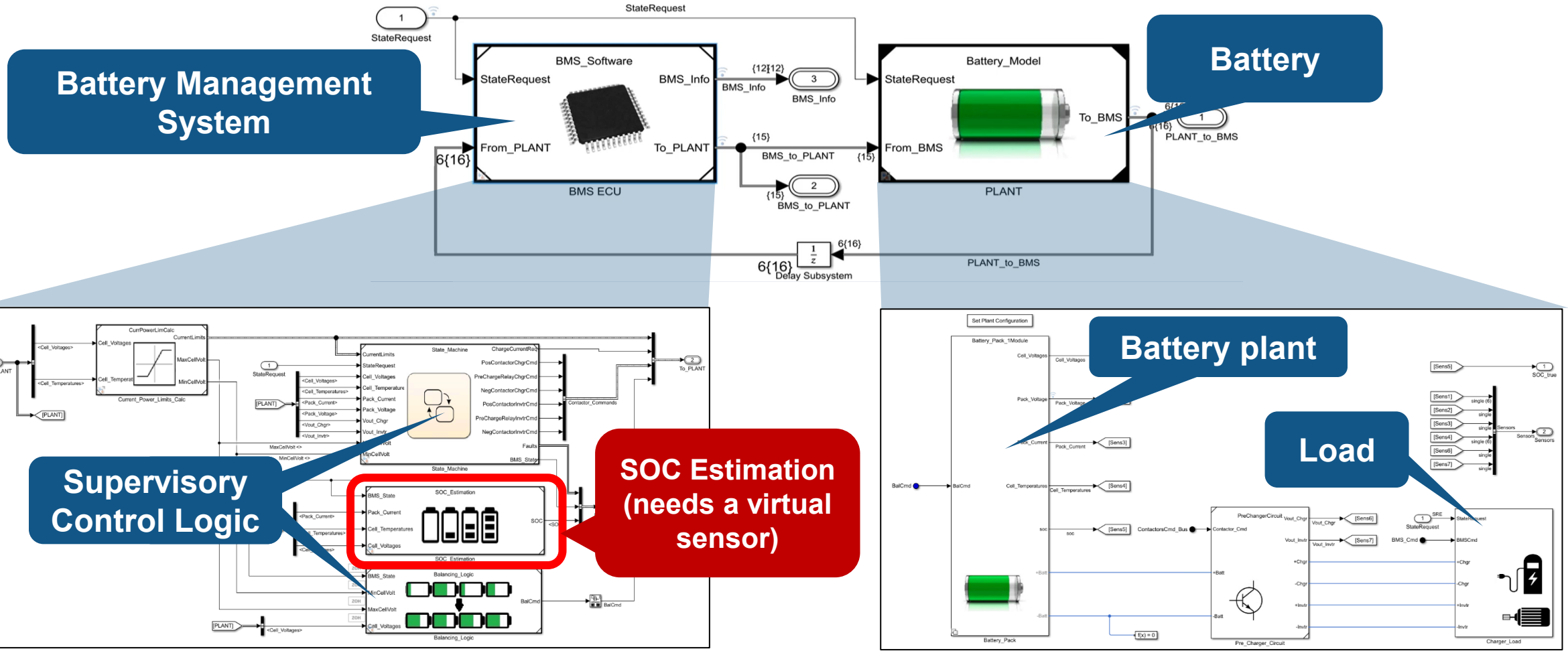


Virtual sensor for Battery State of Charge (SOC) estimation

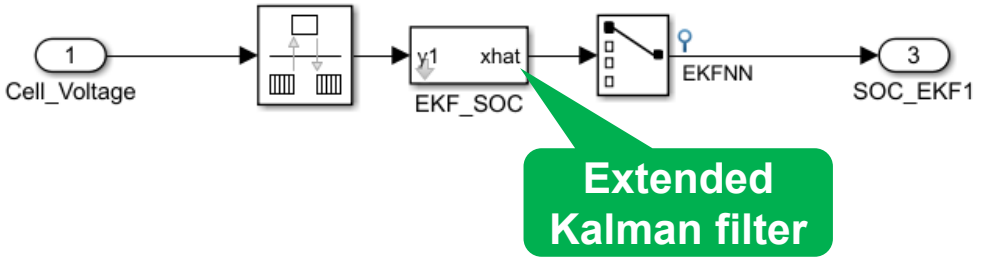
Using AI-based virtual sensors in System-level simulation

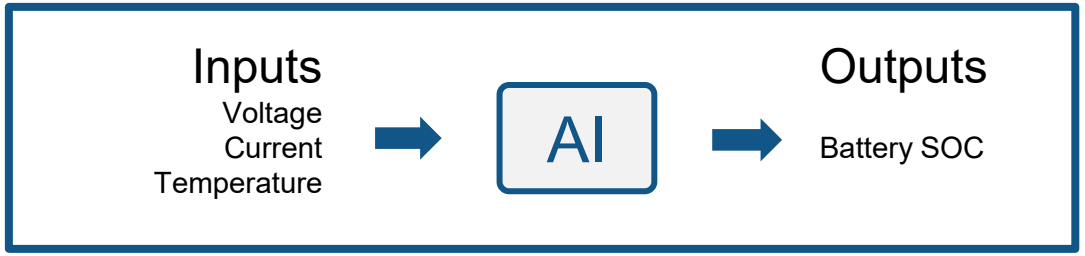


Having a physical sensor might not be feasible due to cost, manufacturing process, reliability, degradation, etc.



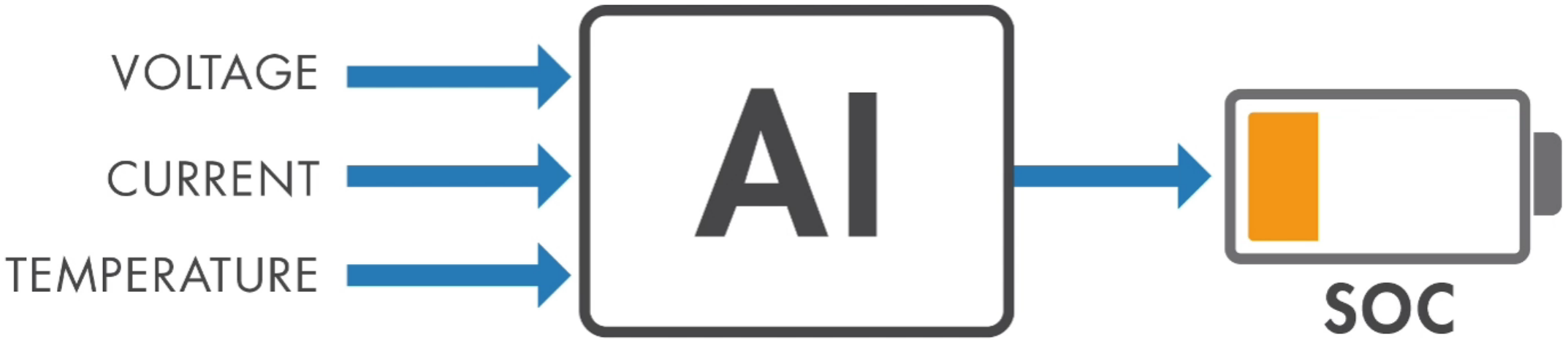
Virtual sensor for Battery State of Charge (SOC) estimation





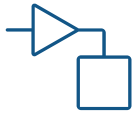
Feature	Kalman Filters	AI Methods
Nonlinearity	Requires EKF/UKF	Handles well
Data Requirements	Accurate model needed	Large datasets required
Proven History	Long history of reliable performance	Emerging and rapidly evolving

Battery State-of-Charge Estimation Using AI



Workflow for developing AI-based virtual sensors

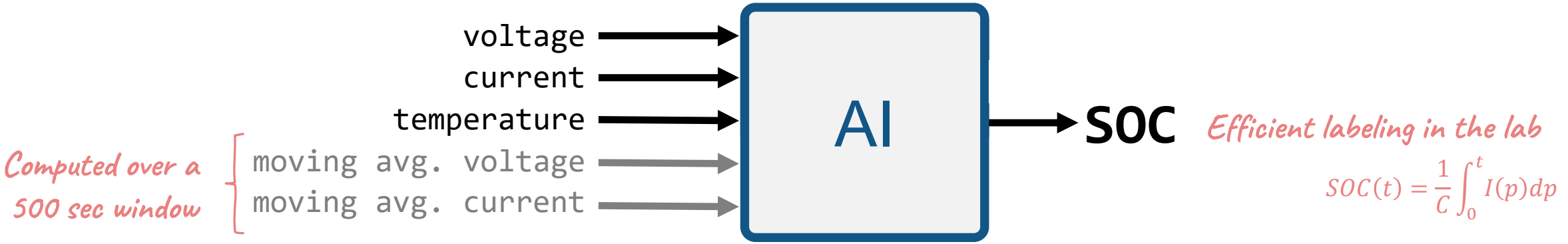





Data Preparation	AI Modeling	Simulation & Test	Compression	Deployment
Toolboxes for **domain-specific** pre/post-processing	**Low-code** AI Modeling and training through MATLAB Apps Model import from **PyTorch, TensorFlow, and ONNX**	**Simulink blocks for AI inference** make integration easy **Simulation-based** unit testing	**Model compression techniques** to reduce model size and accelerate inference	**Automatic code generation** for embedded targets

Data Preparation

Data source: McMaster University*



Predictors

Response

	1 Voltage	2 Current	3 Temperature	4 Moving Average Voltage	5 Moving Average Current	6 SOC
1	0.7510	0.3851	0.3031	0.7510	0.3851	0.2064
2	0.7510	0.3852	0.3046	0.7510	0.3851	0.2064
3	0.7510	0.3852	0.3061	0.7510	0.3852	0.2064
4	0.7510	0.3852	0.3076	0.7510	0.3852	0.2064
5	0.7510	0.3852	0.3091	0.7510	0.3852	0.2064
6	0.7510	0.3852	0.3106	0.7510	0.3852	0.2064
7	0.7510	0.3852	0.3120	0.7510	0.3852	0.2064
8	0.7510	0.3852	0.3135	0.7510	0.3852	0.2064
9	0.7510	0.3852	0.3150	0.7510	0.3852	0.2064
10	0.7510	0.3852	0.3165	0.7510	0.3852	0.2064

* https://data.mendeley.com/datasets/cp3473x7xv/3

AI Modeling – emerging and rapidly evolving

Type	Classic machine learning	Deep Learning		Hybrid Approach
		Feedforward neural network (FNN)	Recurrent neural network (RNN)	
Description	Learn patterns from data; often require feature engineering	Data flows in one direction through layers	Designed for sequential data, captures temporal dependencies	Combines AI and traditional methods, leverages the strengths of both approaches
Algorithms	Decision Trees, Random Forests, SVMs, etc.	FNN (also known as FFN)	Long-Short Term Memory (LSTM); Gated Recurrent Unit (GRU)	Kalman Filter + AI

AI Modeling – emerging and rapidly evolving

Type	Classic machine learning	Deep Learning	
		Feedforward neural network (FNN)	Recurrent neural network (RNN)
Description	Learn patterns from data; often require feature engineering	Data flows in one direction through layers	Designed for sequential data, captures temporal dependencies
Algorithms	Decision Trees, Random Forests, SVMs, etc.	FNN (also known as FFN)	Long-Short Term Memory (LSTM); Gated Recurrent Unit (GRU)

1 Train in MATLAB's **Machine Learning** Framework

2 Train in MATLAB's **Deep Learning** Framework

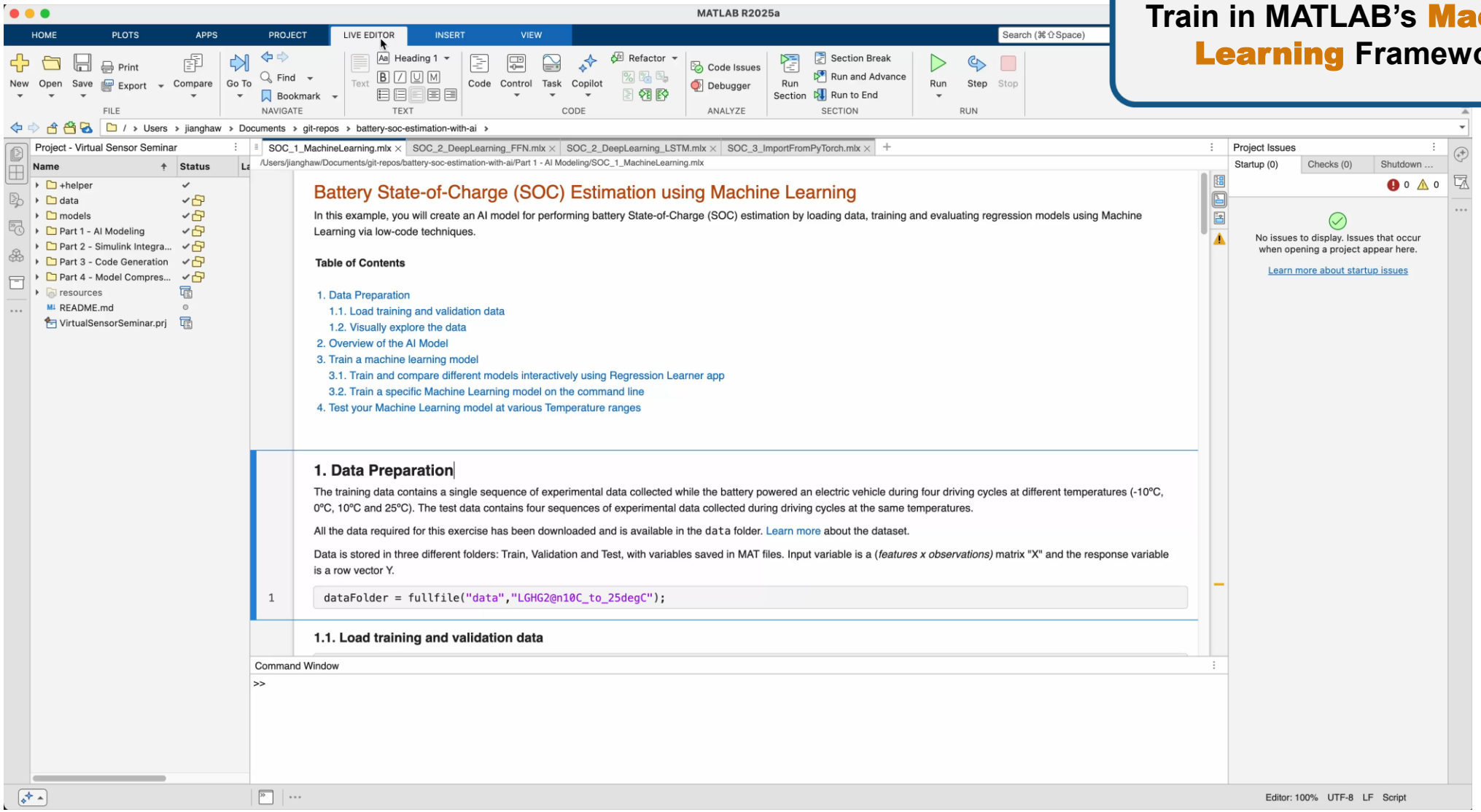
3 Import model from **TensorFlow or PyTorch**

Data Preparation | AI Modeling | Simulation & Test | Deployment

Train a regression tree model

1

Train in MATLAB's **Machine Learning** Framework

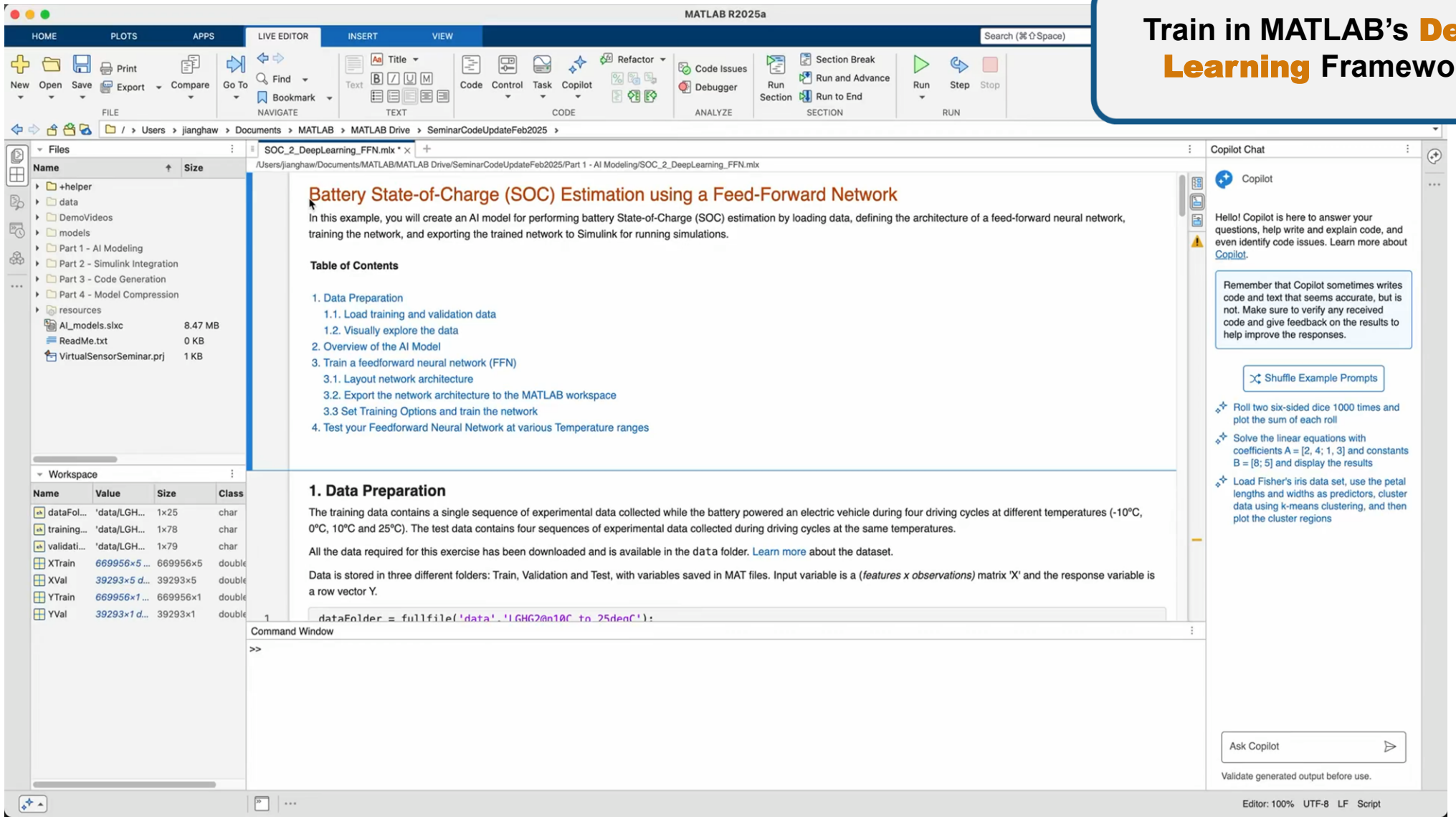


Data Preparation | **AI Modeling** | Simulation & Test | Deployment

Train a feedforward network network

2

Train in MATLAB's **Deep Learning** Framework

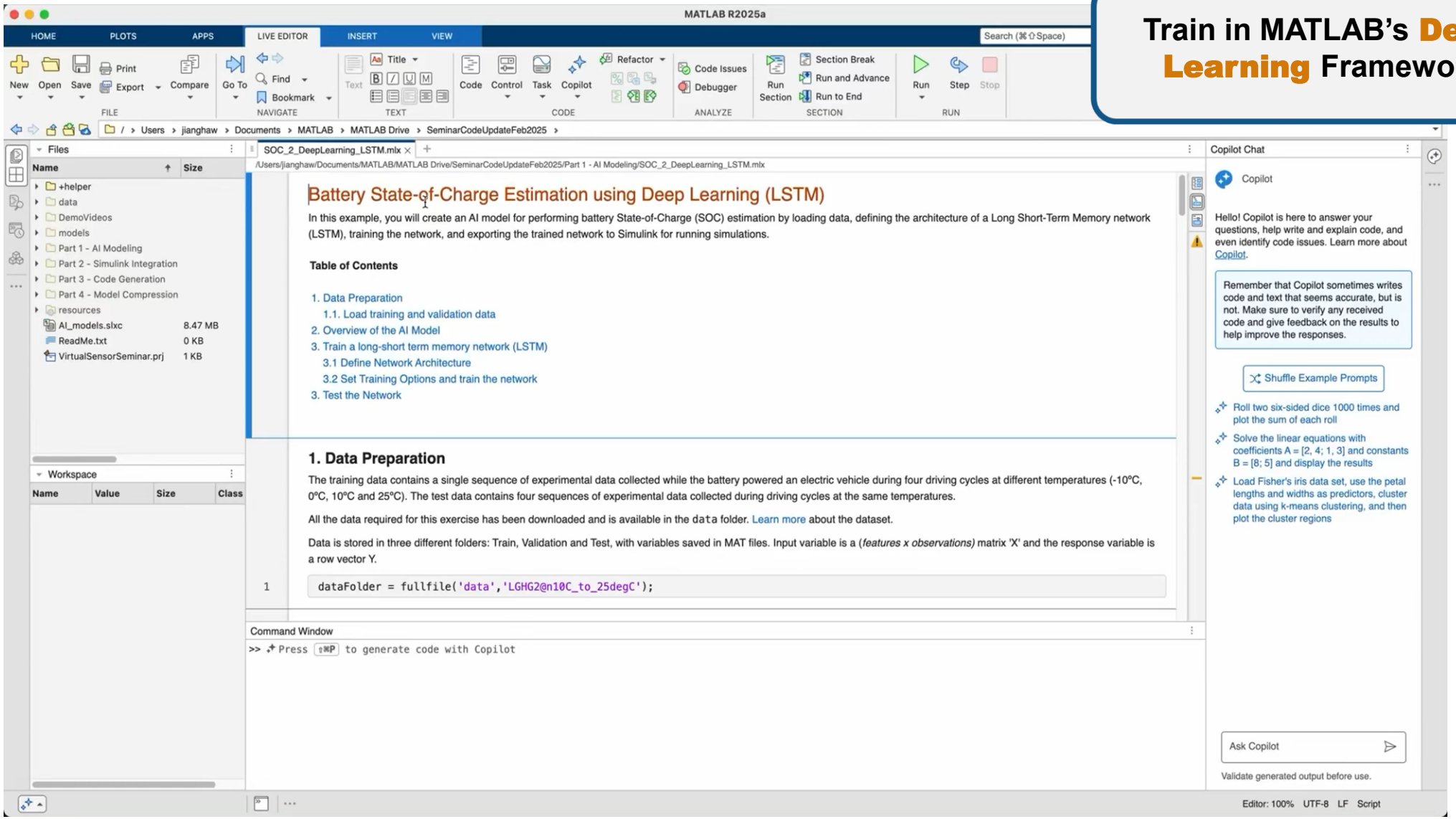


Data Preparation | **AI Modeling** | Simulation & Test | Deployment

Train an LSTM network

2

Train in MATLAB's Deep Learning Framework



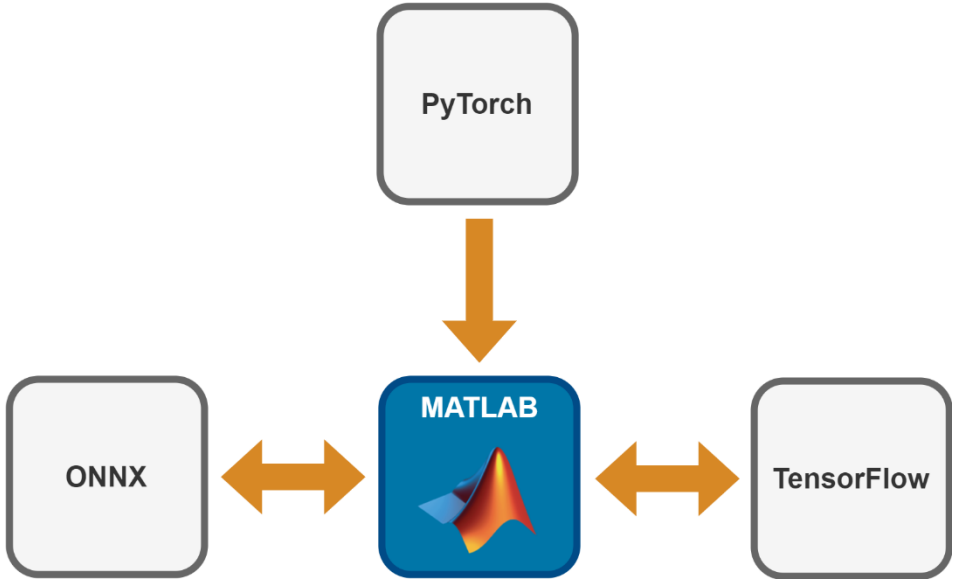
Data Preparation | **AI Modeling** | Simulation & Test | Deployment

Interoperate with Open-Source Frameworks

Framework Interoperability bridges the gap between data science, engineering and production

Import and Export models through:

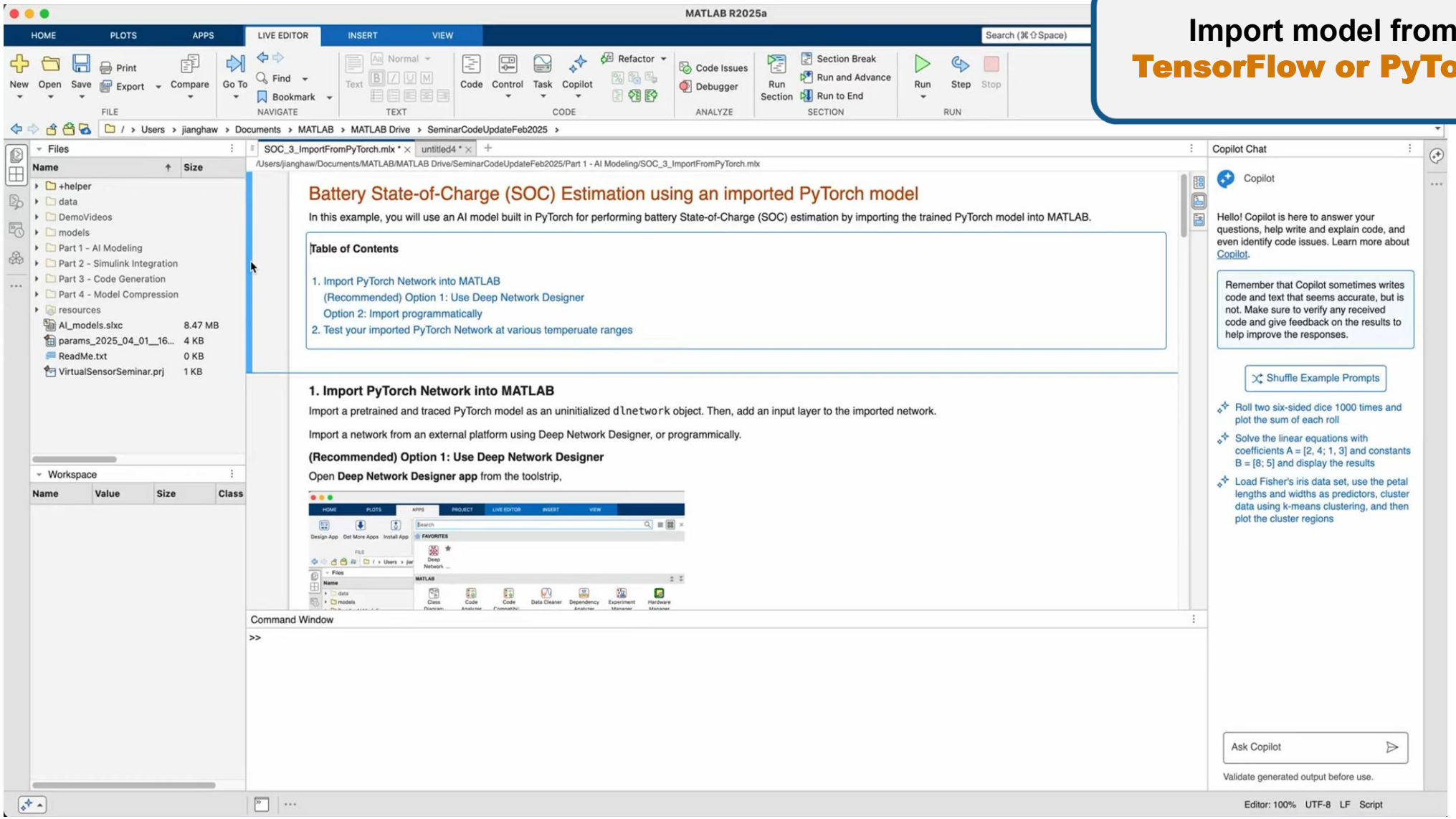
TensorFlow-Keras Import	R2017b
ONNX Converter (Import & Export)	R2018a
TensorFlow Converter (Import)	R2021a
TensorFlow Converter (Export)	R2022b
PyTorch Converter (Import)	R2022b
New functions for importing networks from ONNX and TensorFlow to dlnetwork	R2023b
Import networks from PyTorch and TensorFlow in Deep Network Designer	R2023b



Import a trained PyTorch model

3

Import model from
TensorFlow or PyTorch



Data Preparation | **AI Modeling** | Simulation & Test | Deployment

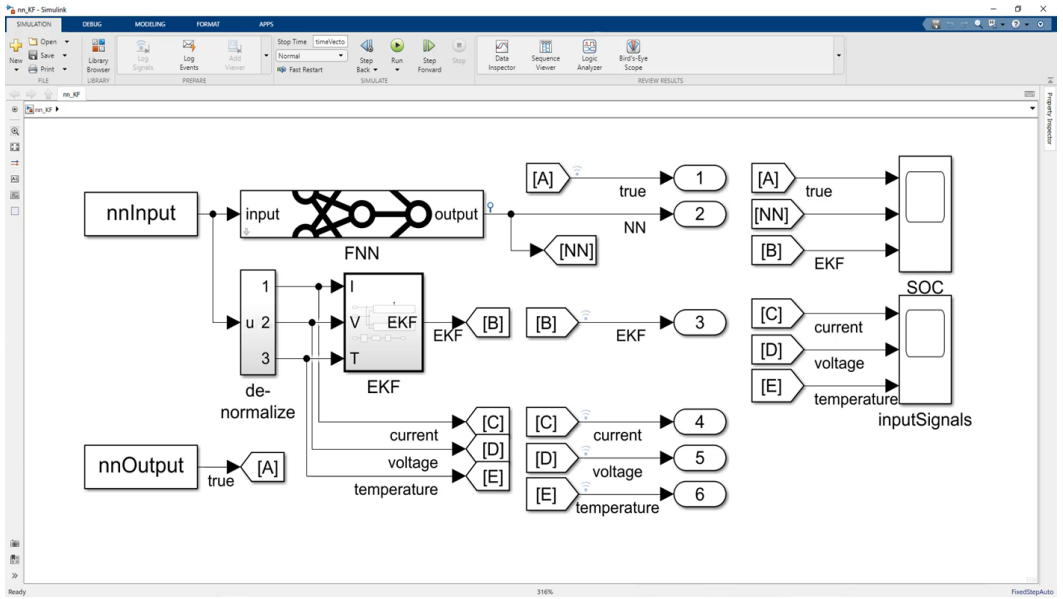
~~AI~~ Modeling is an iterative process

Each method has its own strengths and weaknesses. The choice often depends on the specific requirements and constraints of the application.

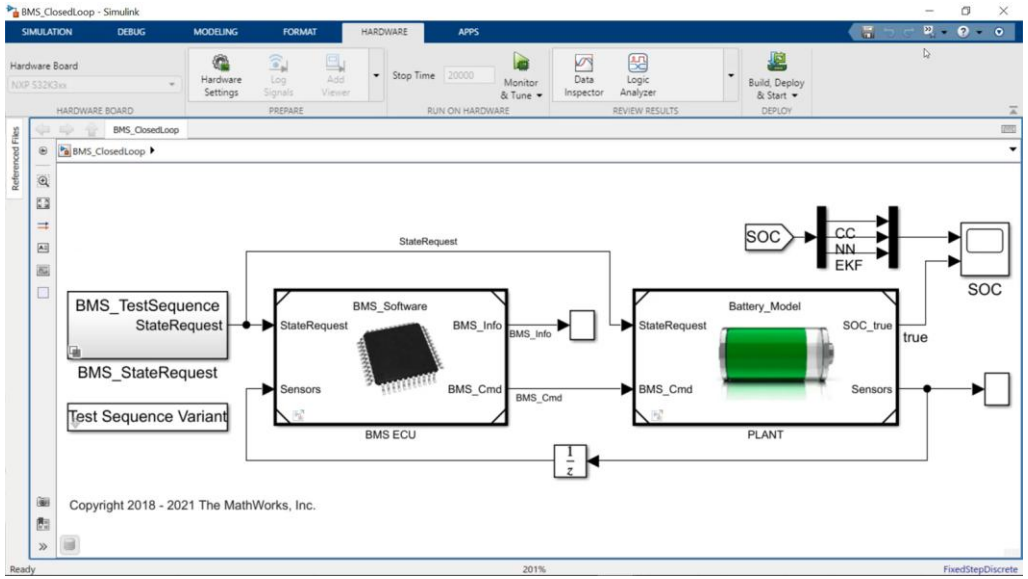
	Regression Tree	FFN	LSTM	GRU	Nonlinear ARX	Kalman Filter
Benefits	Simplicity and interpretability	Small model size, scales well with data, fast for inference	Suitable for sequence data, captures nonlinear dynamics well	Efficient with fewer parameters and run faster than LSTM	Provide physical insights, flexible choices of nonlinear estimators including AI	Real-time efficiency, can be combined with AI models to improve accuracy
Challenges	Feature engineering needed	Feature calculation and hyperparameter tuning needed	Computationally intensive, hyperparameter tuning needed	May not capture long dependencies as well as LSTM	Trial-and-error for the choice of nonlinear estimators	Highly dependent on the battery model, sensitive to system noise

Integrate your AI model for system-level simulation and test

Integration of trained AI model into Simulink

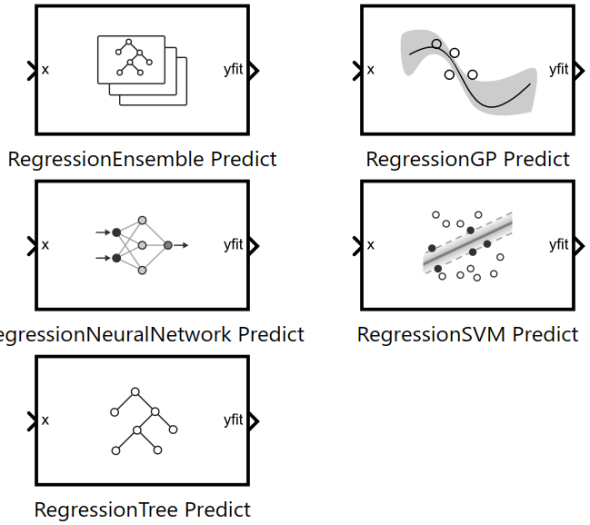


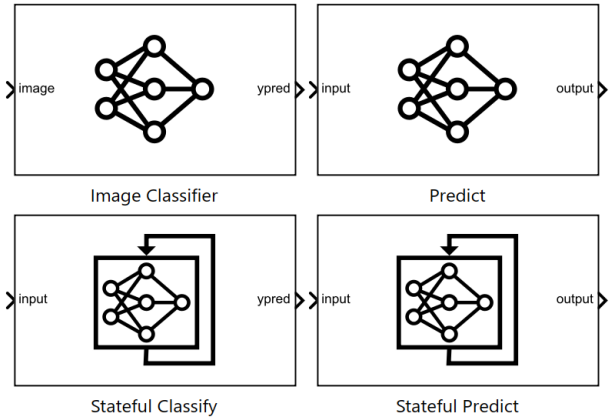
System-level simulation



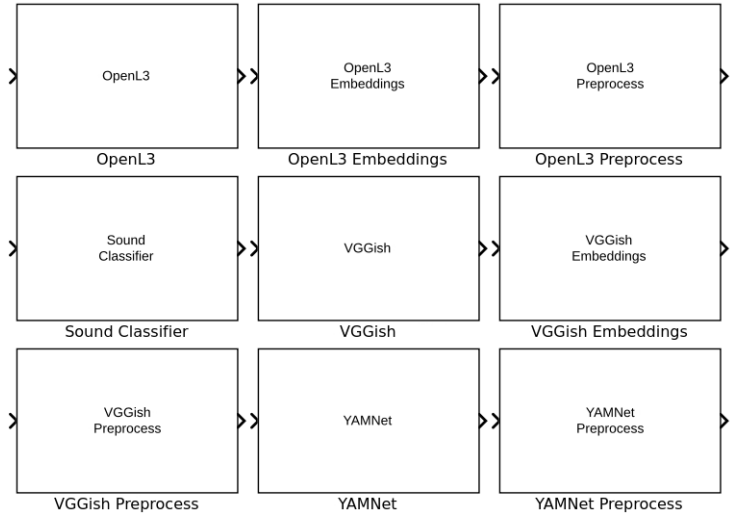
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

AI libraries in Simulink are expanding to include more AI blocks for more applications

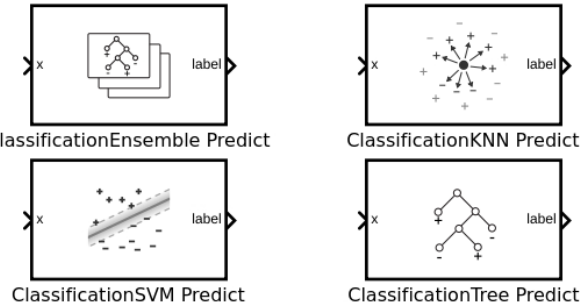




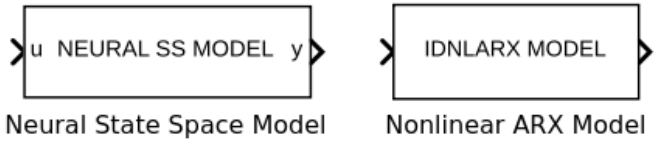
Deep Learning Toolbox



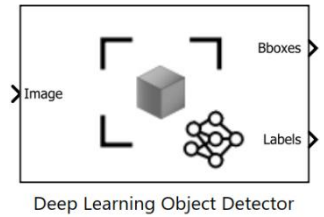
Audio Toolbox



Statistics and Machine Learning Toolbox



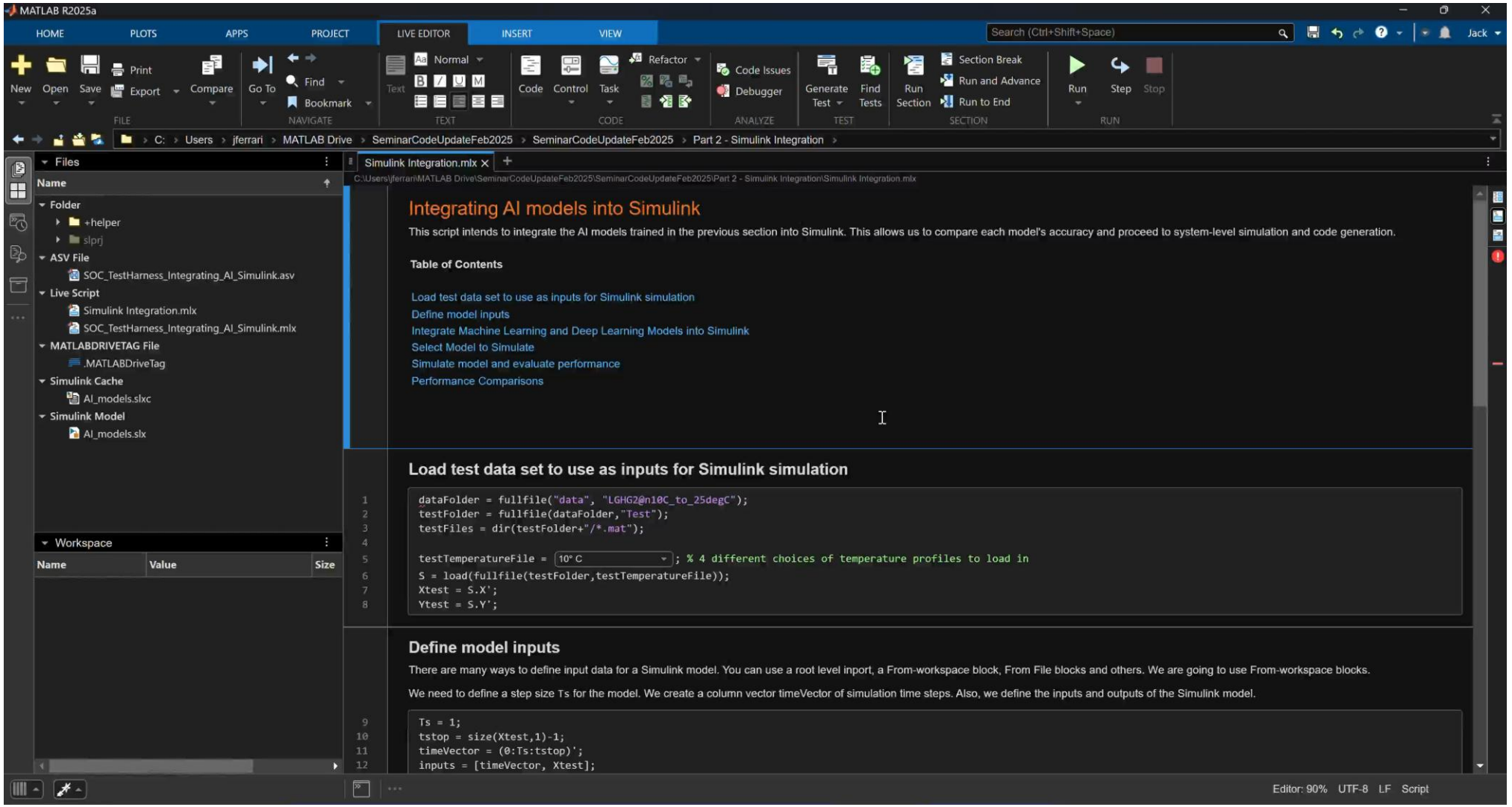
System Identification Toolbox



Computer Vision Toolbox

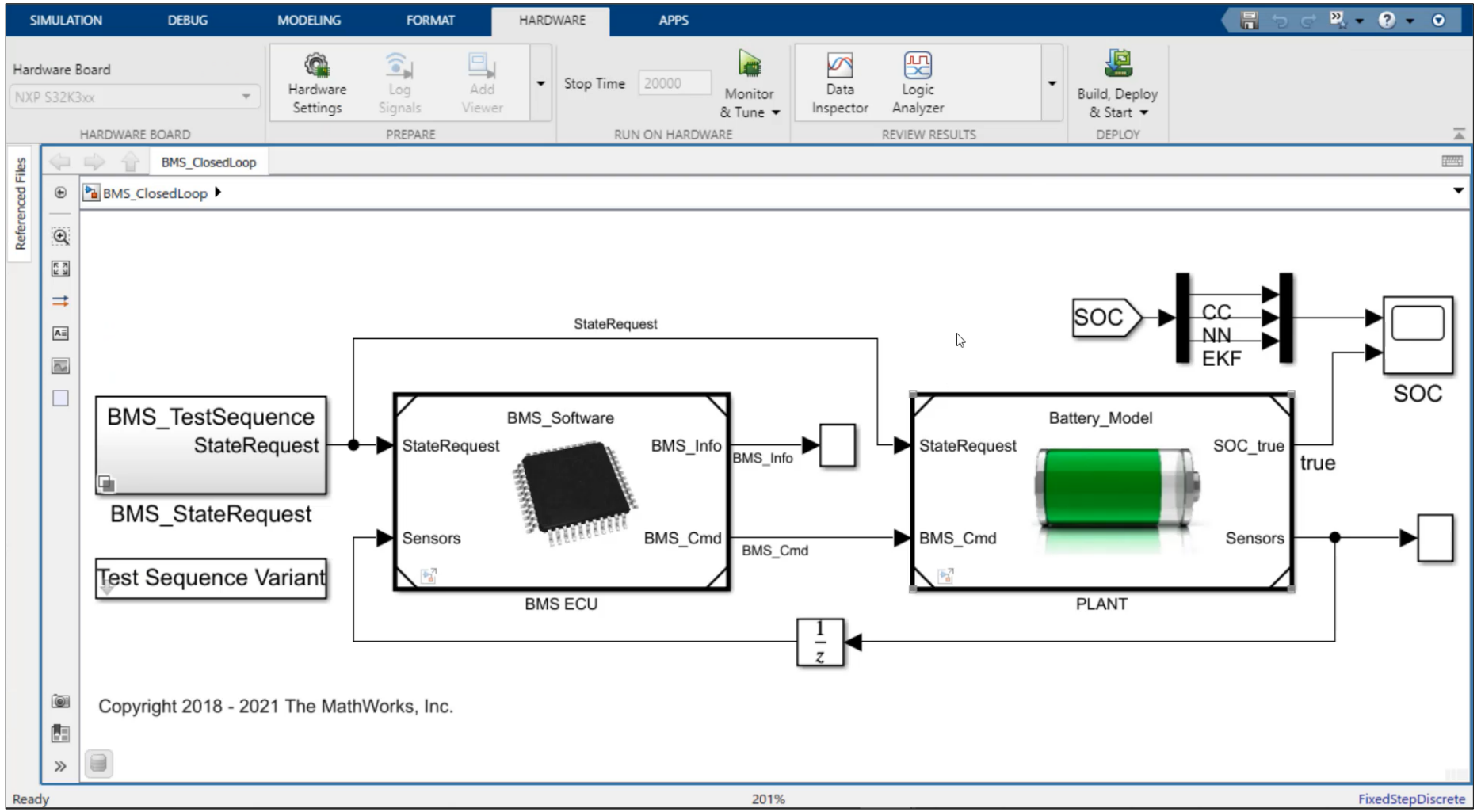
Data Preparation | AI Modeling | Simulation & Test | Deployment

Integration of trained AI models into Simulink



Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Closed-Loop System-Level Simulation



Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Deep learning layer blocks enable layer-level network modeling in Simulink

dlnetwork object

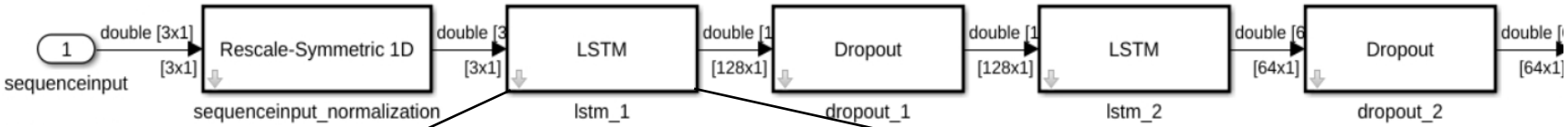
Deep Network Designer App

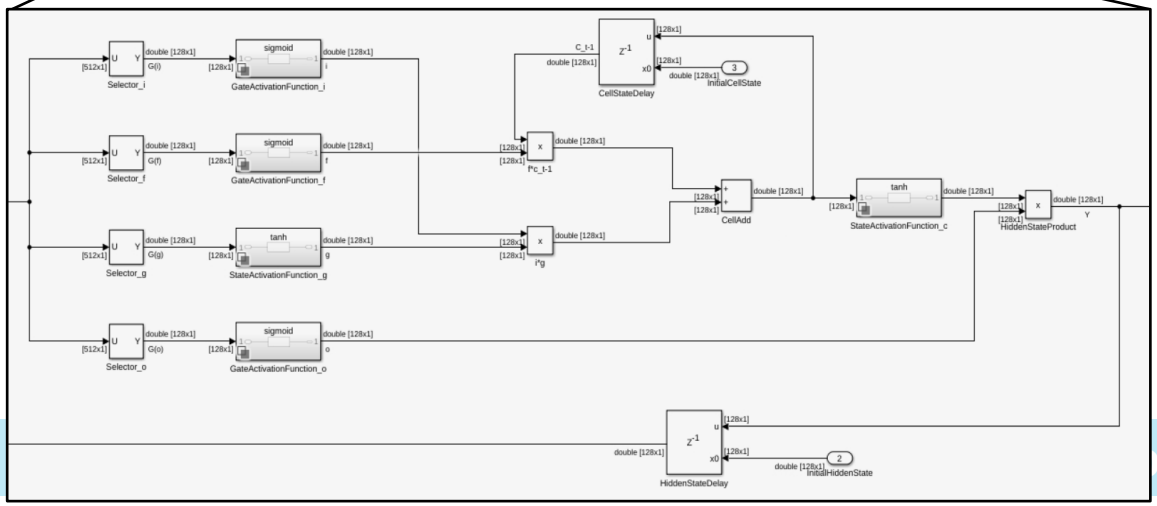
Simulate

Codegen & Deploy

C/C++ Code Generation

```
exportNetworkToSimulink(net)
```



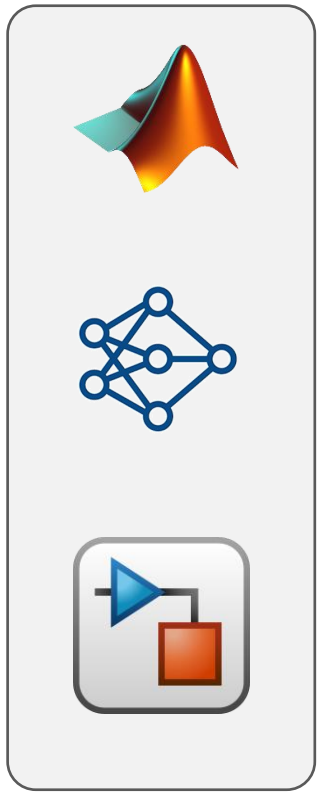


R2024b

Data Preparation

oyment

Deploy AI models to Host Machines and Embedded Hardware



Code Generation

CPU

GPU

MCU

FPGA

Any CPU
Plain C/C++ Source Code

Intel CPUs w/ MKL-DNN Accelerator Library

NVIDIA

NXP

life.augmented

Infineon

Texas Instruments

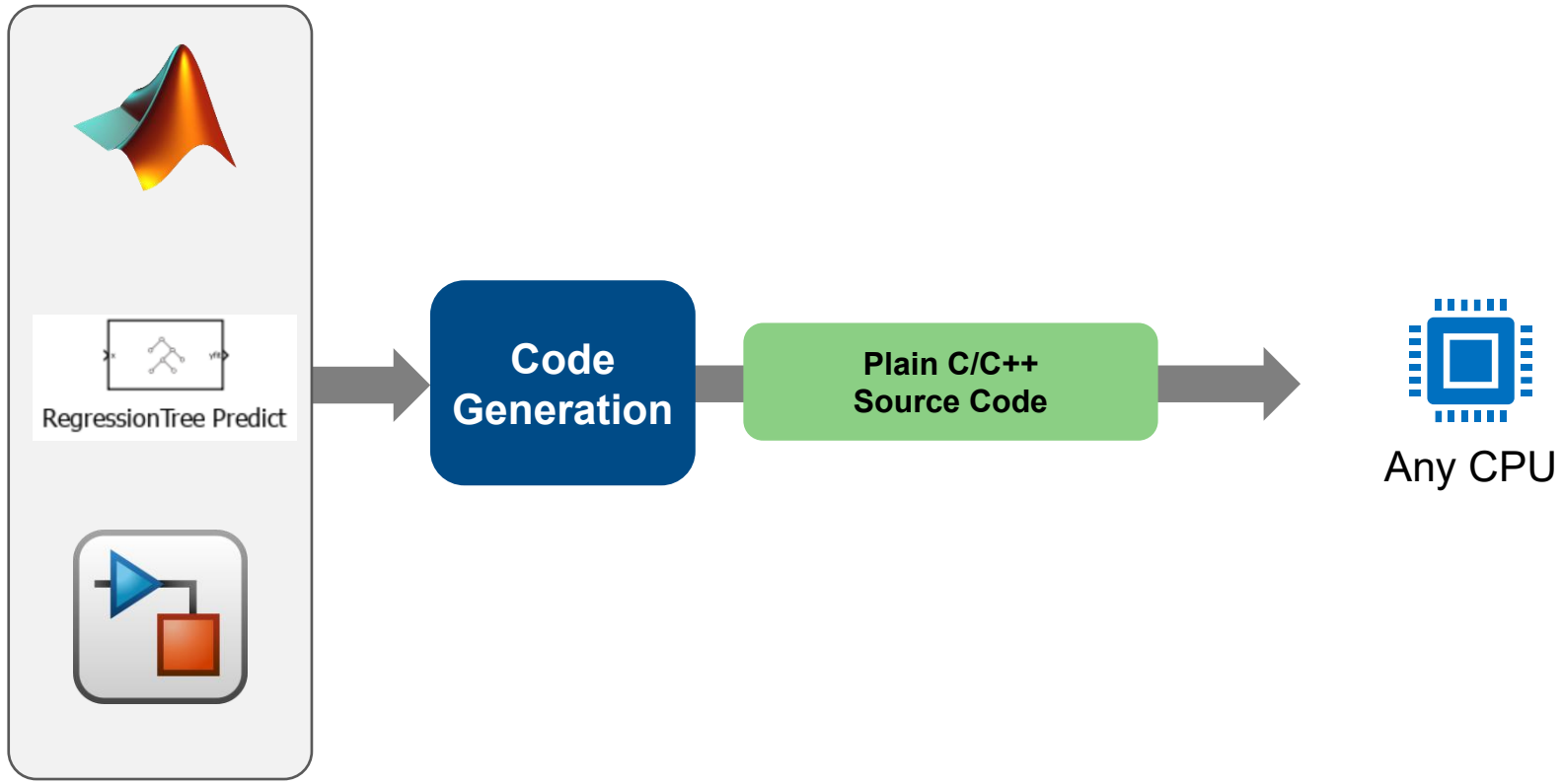
ZYNQ

altera
An Intel Company

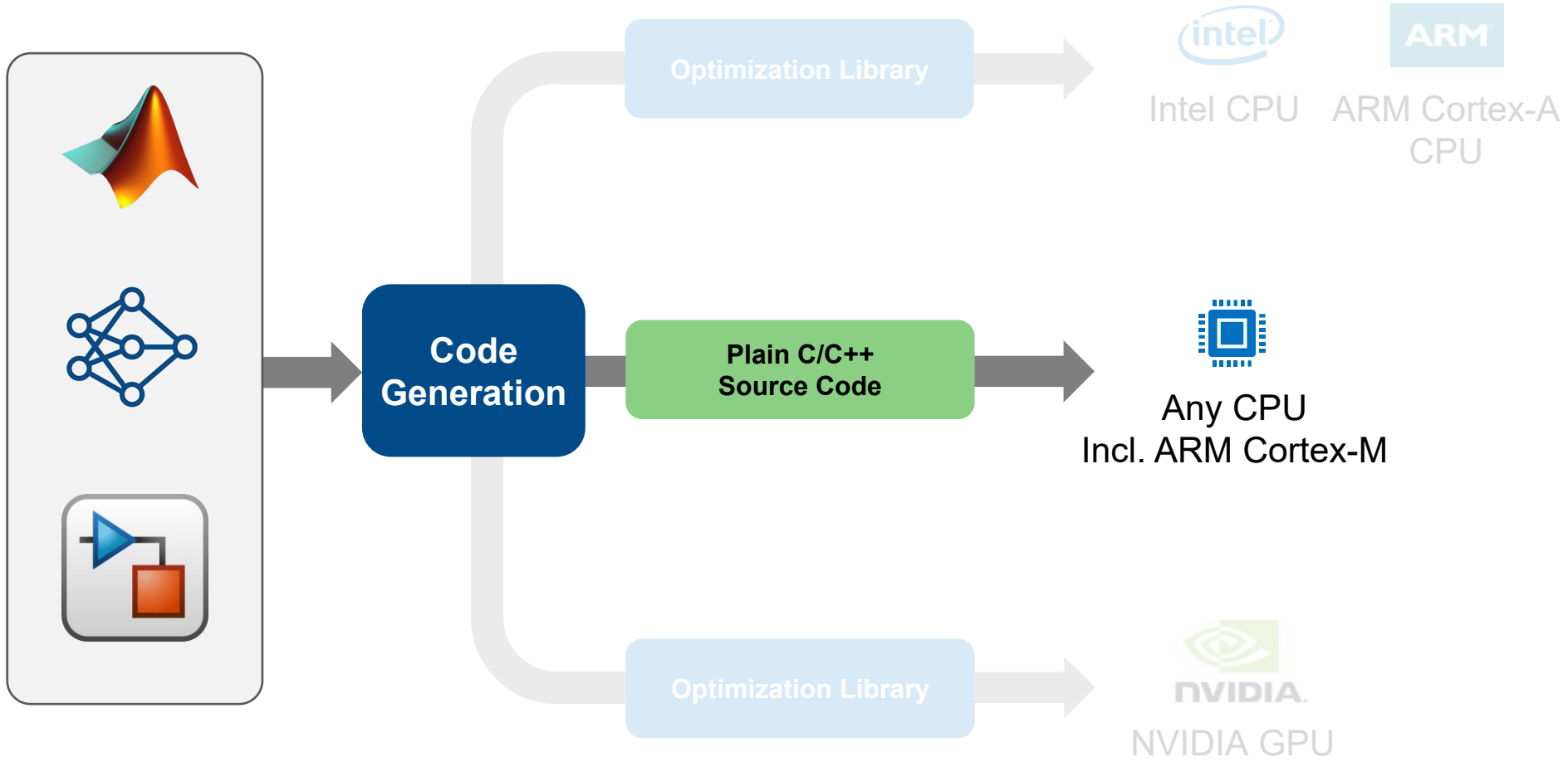
AMD XILINX

Data Preparation | AI Modeling | Simulation & Test | Deployment

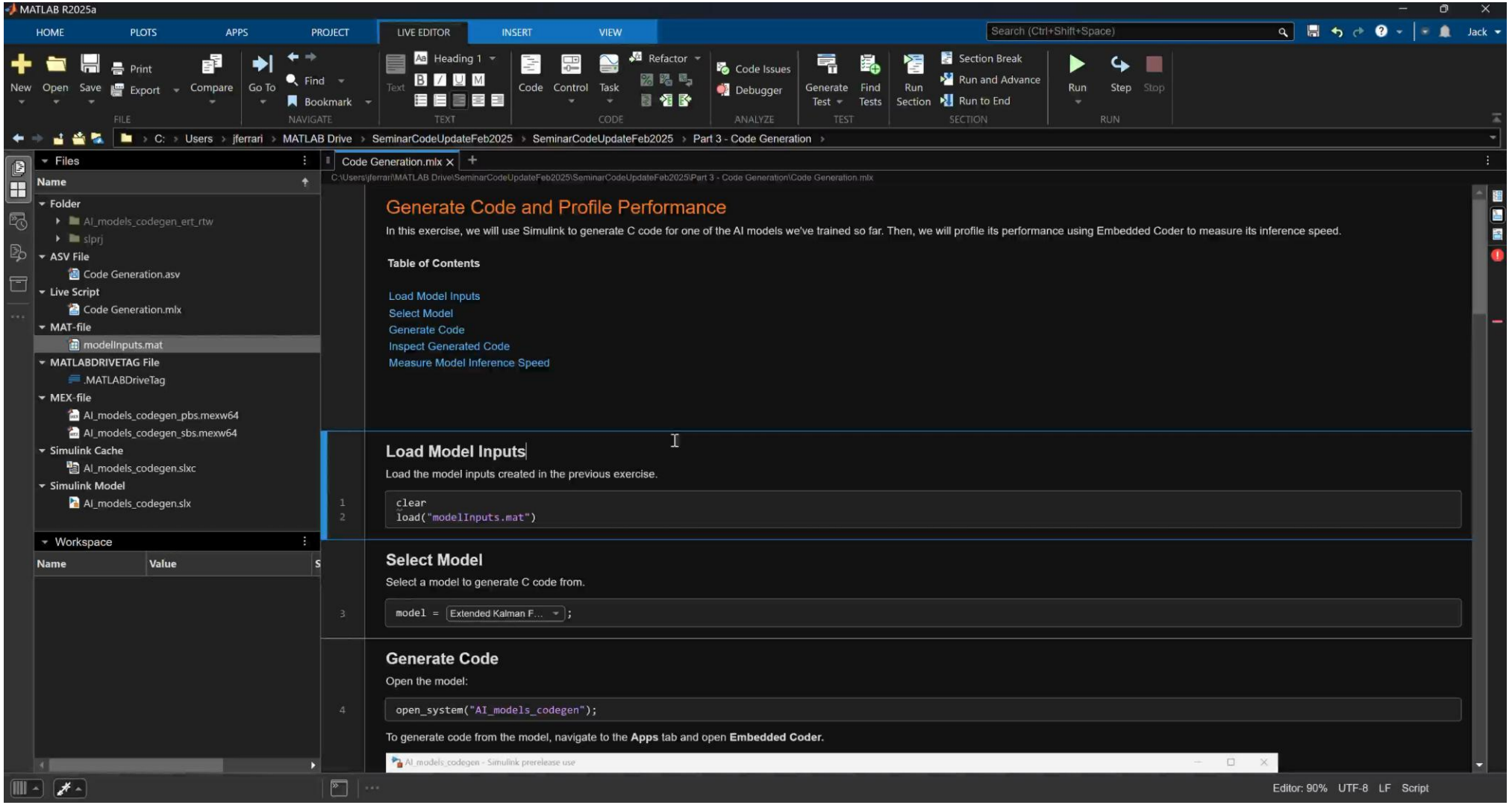
Use Embedded Coder to Generate Code for Machine Learning



Generate Target Independent C/C++ Code for Deep Learning Networks

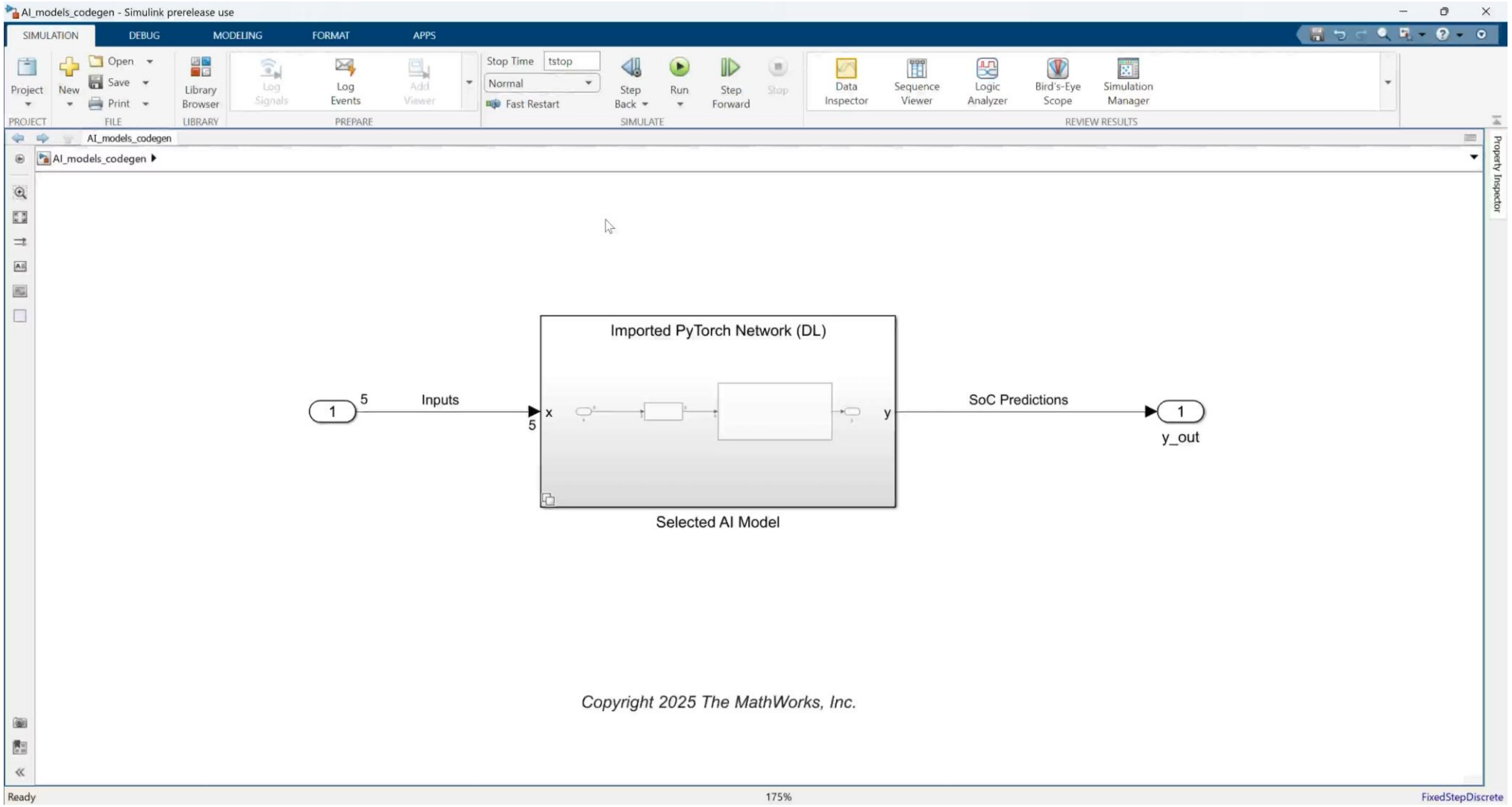


Generate Code

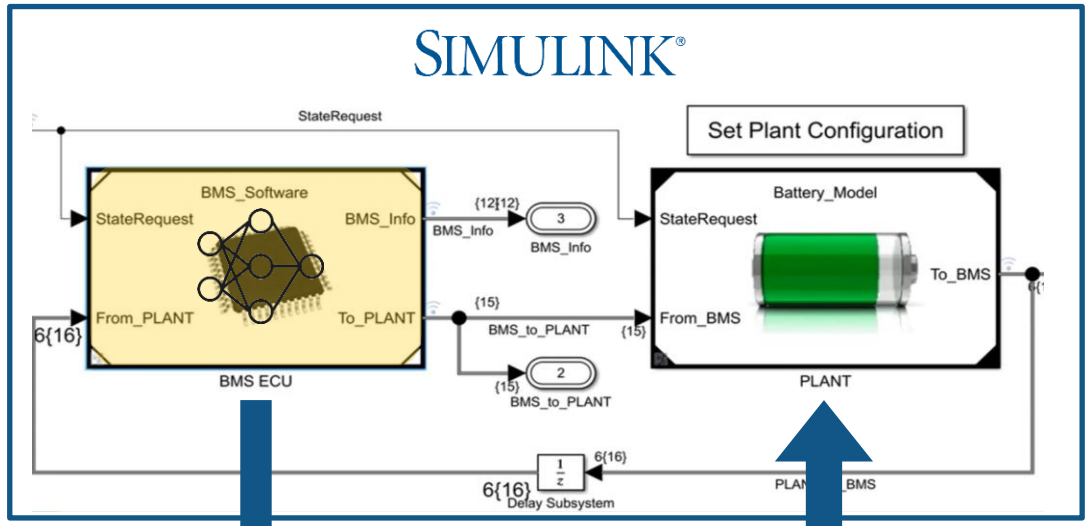


Data Preparation | AI Modeling | Simulation & Test | Deployment

Profile Performance



Processor-in-the-Loop Testing on ARM Cortex-M7 Processor



Code generation from algorithm

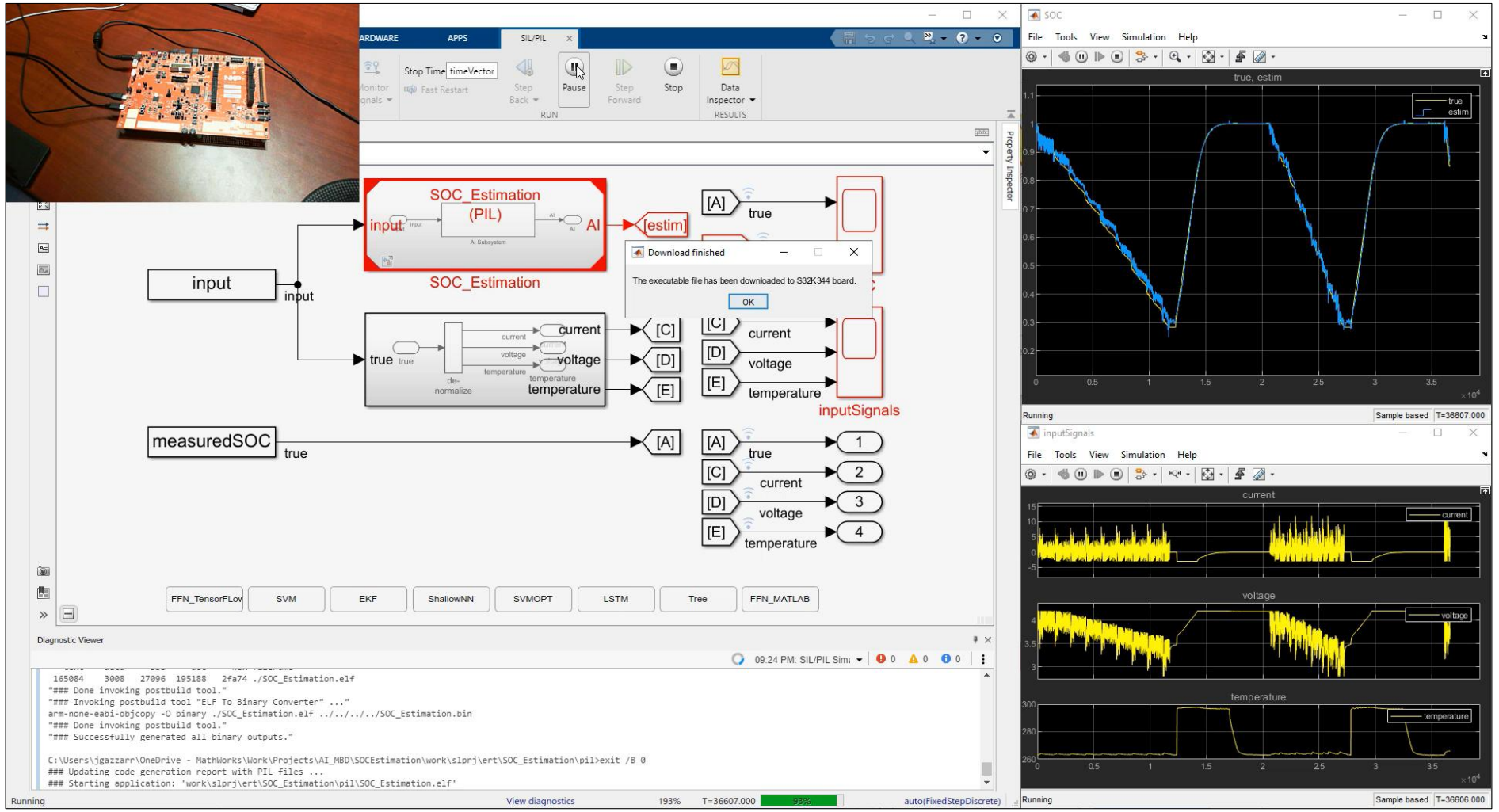
Deployed code communicates with simulated plant

NXP S32K344 board

Arm® Cortex®-M

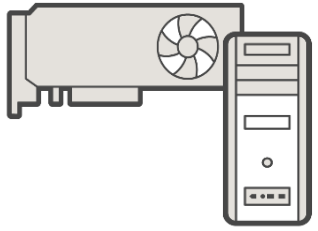
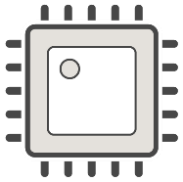
Data Preparation | AI Modeling | Simulation & Test | **Deployment**

Processor-in-the-Loop Testing on ARM Cortex-M7 Processor



Data Preparation | AI Modeling | Simulation & Test | **Deployment**

Embedded Devices Have Limited Resources

	Memory	Performance	Power
	32GB	TFLOPs	
	8GB	GFLOPs	
	1kB~1MB	MFLOPs	

Data Preparation | AI Modeling | Simulation & Test | Deployment

Manage AI tradeoffs for your system

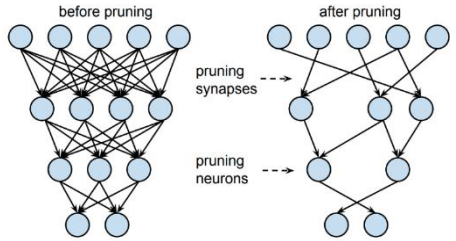
	EKF Extended Kalman Filter	**Tree** Fine Regression Tree	**FFN** 1-hidden layer Feedforward Network	**LSTM** Long Short-Term Memory Network
Preprocessing effort	🔵	🟡	🟡	🔵
Training Speed	N/A	🔵	🟡	🔴
Interpretability	🔵	🔵	🔴	🔴
Inference Speed	🟡	🔵	🔵🔵	🔴
Model Size	🔵	🟡	🔵	🔴
Accuracy (RSME)	🔵	🟡	🔵	🔵🔵

Results are specific to this Battery SOC Estimation example

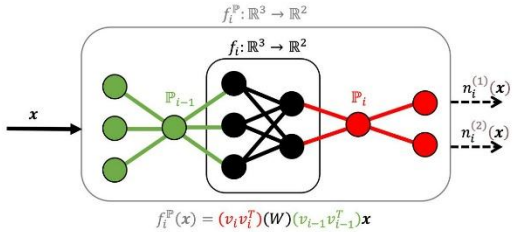
Much Better 🔵🔵 Better 🔵 Okay 🟡 Worse 🔴

Deep learning model compression in MATLAB

Structural Compression (Deep Learning Models)



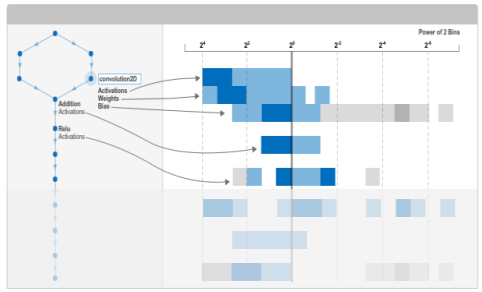
Pruning convolutional neural networks



Projection of deep neural networks

We'll take a closer look at the projection technique in today's seminar

Datatype Compression (Machine Learning + Deep Learning Models)

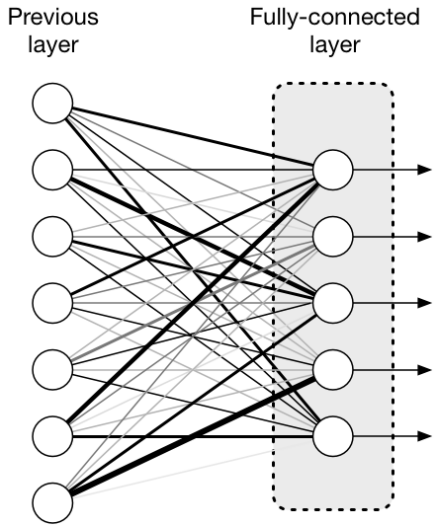


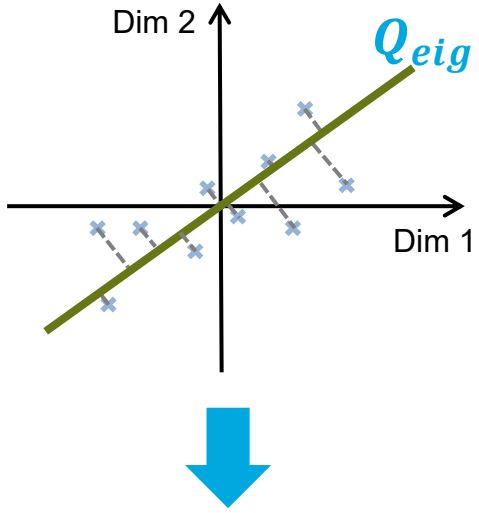
Quantization of network weights to lower precision datatypes (bfloat16, int8)

Data Preparation | AI Modeling | Simulation & Test | Deployment

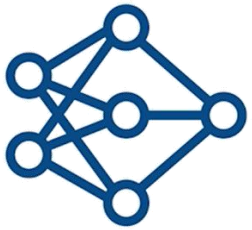
Projection is a structural compression technique that can help reduce the size of the model while retaining model accuracy

Assumption: High-dimensional space of input and output neurons is redundant

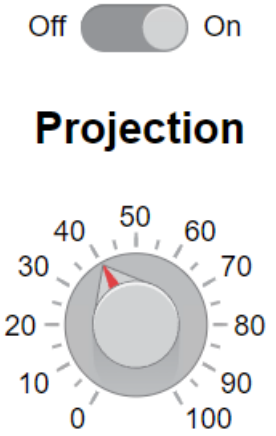




projected layer



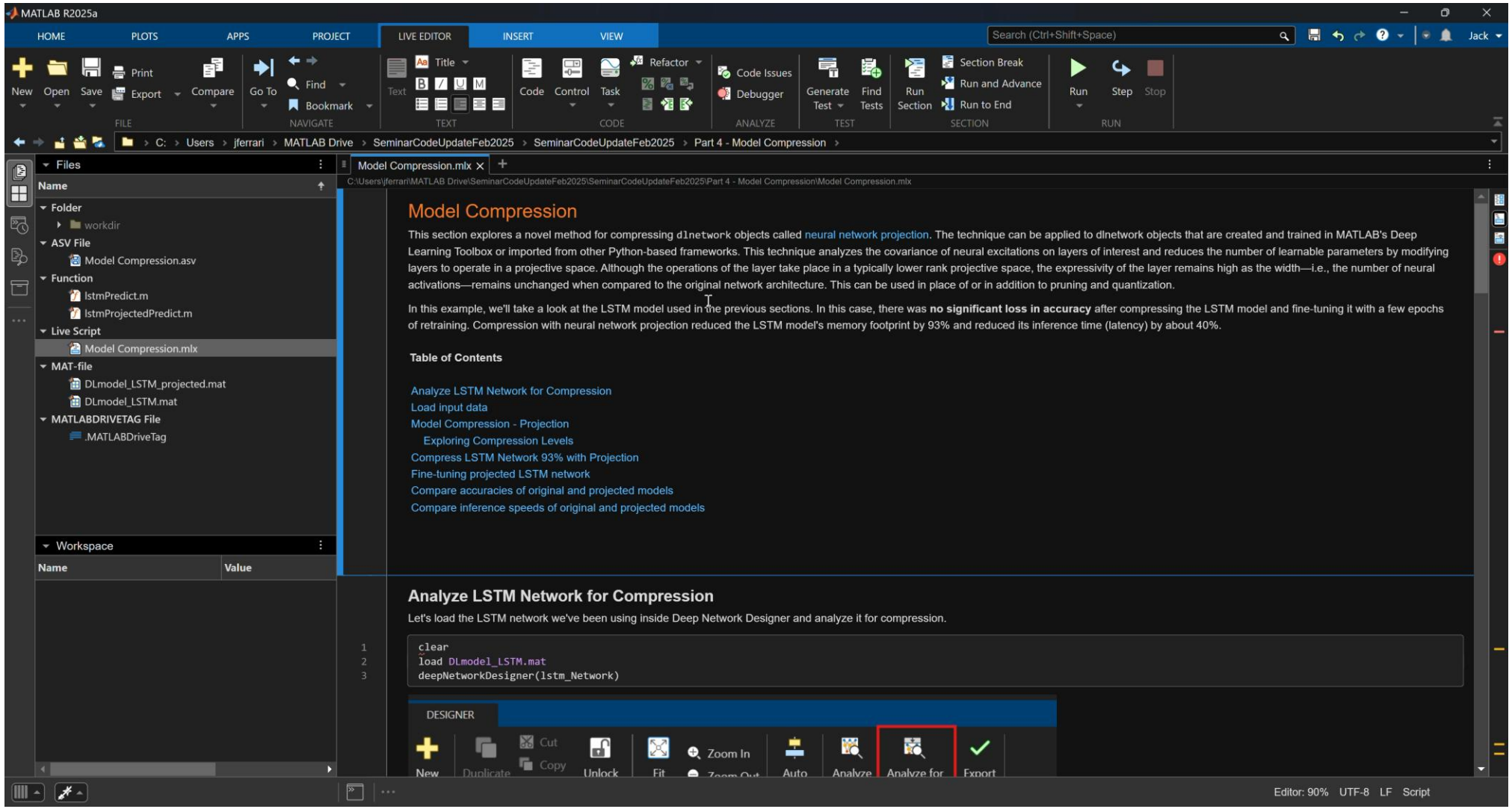
DLT model
e.g. 1MB



e.g. 600 KB

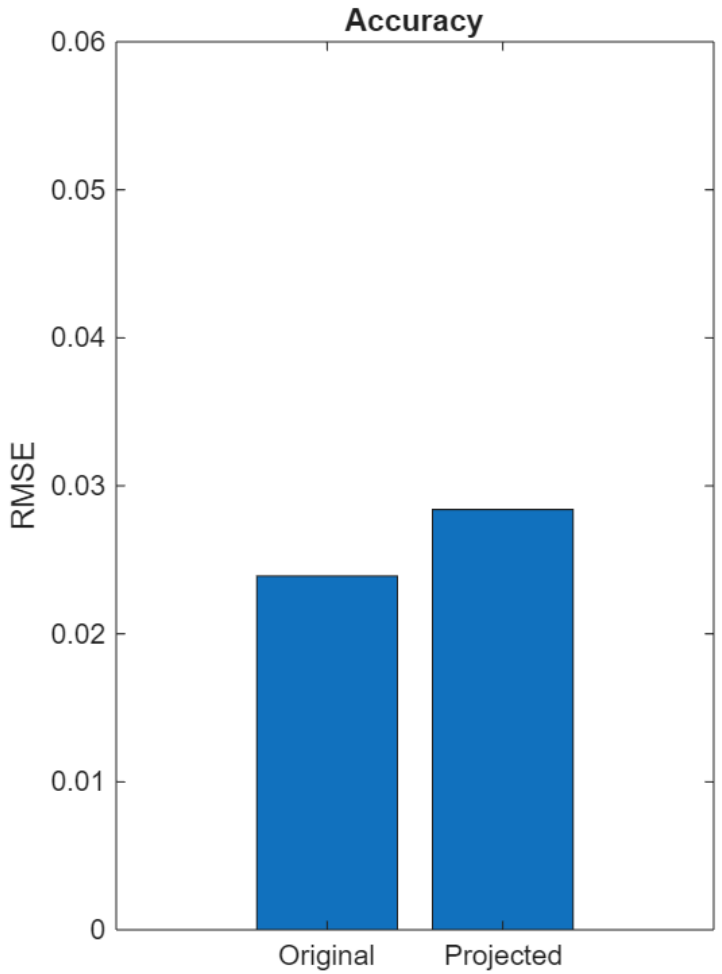
Projection finds optimal subspaces with which to perform layer operations, and can be tuned to meet specific compression requirements

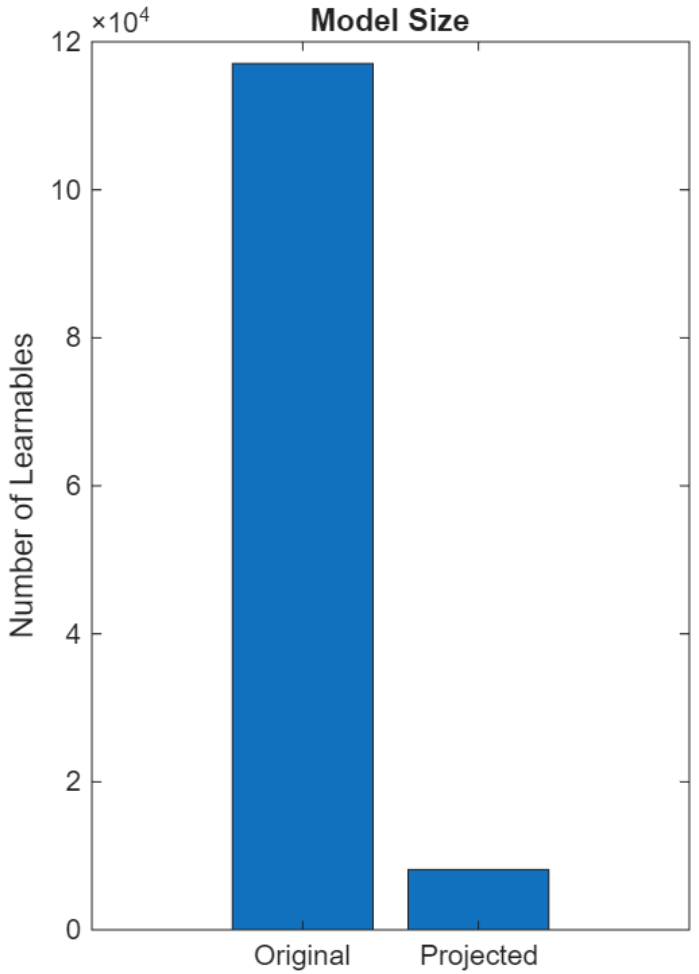
Compress LSTM Model with Projection

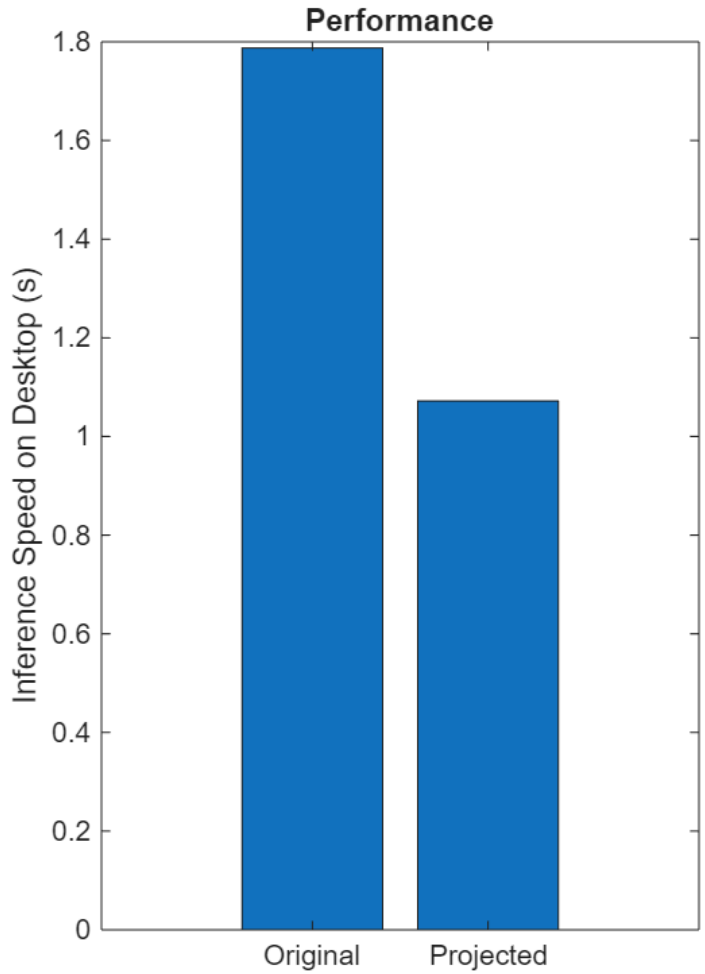


Data Preparation | AI Modeling | Simulation & Test | Deployment

Neural Network Projection Results







Manage AI tradeoffs for your system

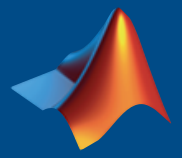
	EKF Extended Kalman Filter	**Tree** Fine Regression Tree	**FFN** 1-hidden layer Feedforward Network	**LSTM** Stacked Long Short-Term Memory Network	**LSTM*** * Compressed Stacked Long Short-Term Memory Network
Preprocessing effort	🔵	🟡	🟡	🔵	🔵
Training Speed	N/A	🔵	🟡	🔴	🔴
Interpretability	🔵	🔵	🔴	🔴	🔴
Inference Speed	🟡	🔵	🔵🔵	🔴	🟡
Model Size	🔵	🟡	🔵	🔴	🔵
Accuracy (RSME)	🔵	🟡	🔵	🔵🔵	🔵🔵

Results are specific to the Battery SoC Estimation Example

Much Better 🔵🔵 Better 🔵 Okay 🟡 Worse 🔴

Agenda

I. Introduction to AI with Model-Based Design
II. Deep dive of Virtual Sensor Modeling workflow
III. Resources for further learning



MathWorks®

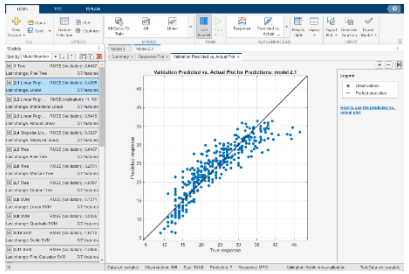
Topics we covered today

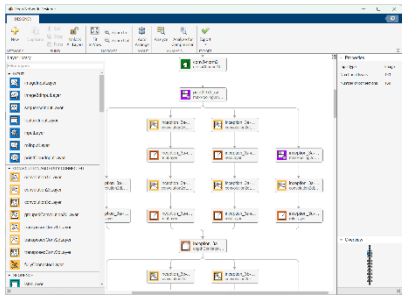
- AI modeling with machine learning and deep learning methods
- Python-MATLAB interoperability
- Integrate AI model into system-level simulation and test
- AI model compression
- Deploy AI models to embedded target

Why MATLAB & Simulink?

- Low-code tools for AI modeling
- Easily integrate AI models into Model-Based Design
- Model Compression, Verification, and Cross-Platform Code generation
- Interoperability of Open-Source frameworks

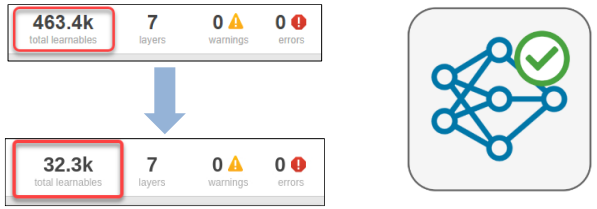
Low-code Apps

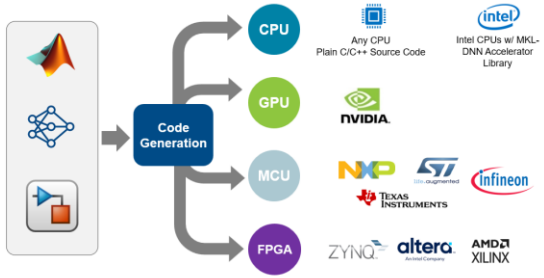




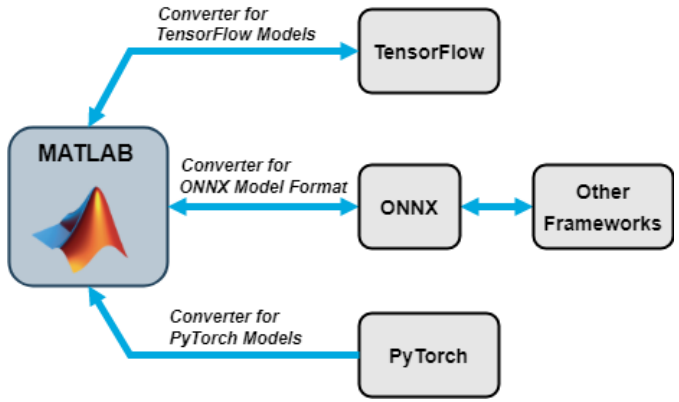
AI + Model-Based Design

Compression, Verification, Code Generation

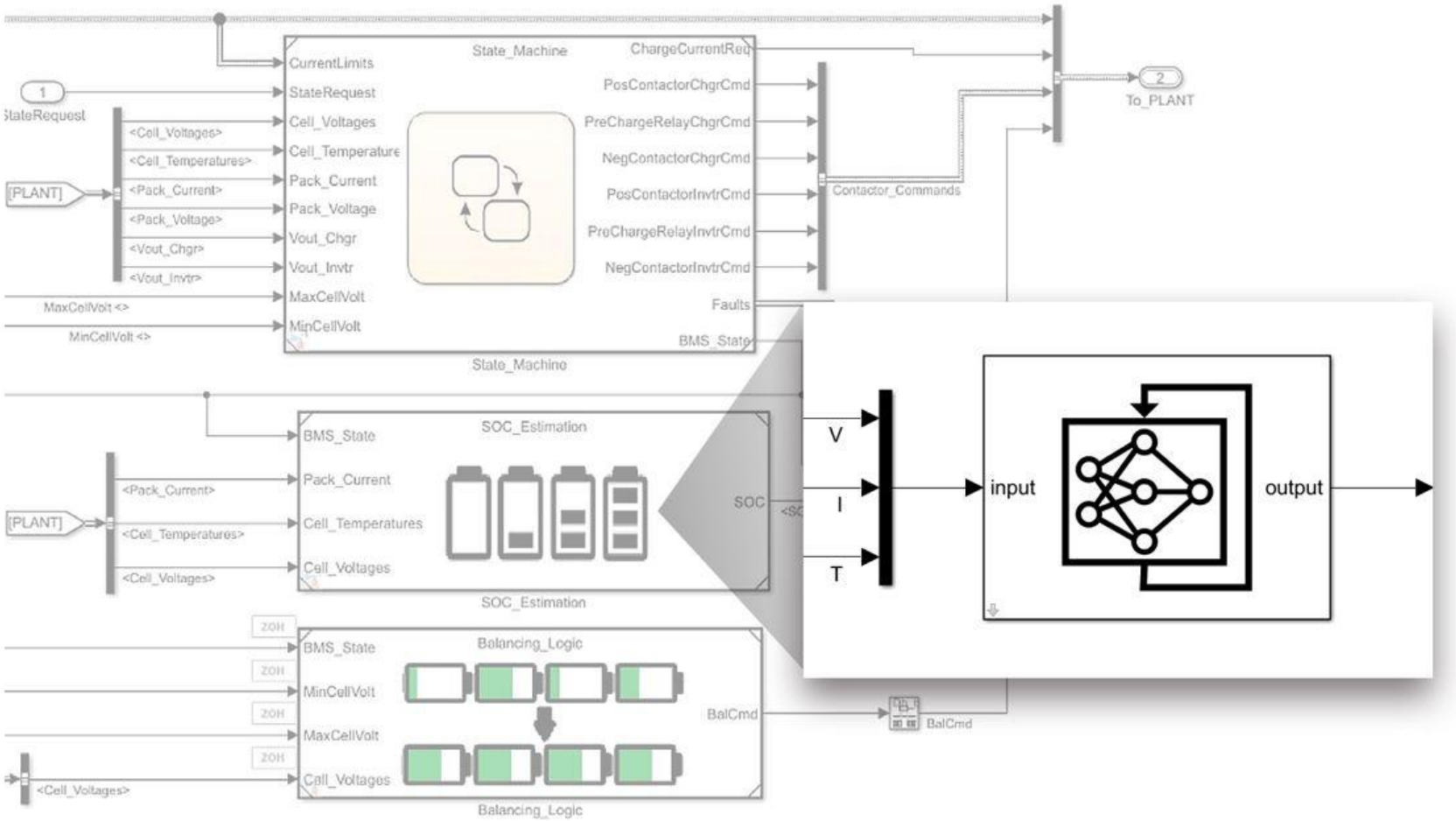




Interoperability



In summary, build a virtual sensor using AI and integrate into Simulink for system-level simulation and code generation



Topics we did not cover today

Model Comparison and Tuning

Explainable AI

AI Verification and Validation

And more… Let's follow up!

Resources for additional learning

- Other seminars and workshops
  - Reduced Order Modeling
  - Nonlinear system identification
  - Reinforcement Learning
  - Learning-based Controls
  - Hands-on virtual sensor modeling workshop
- AI with Model-Based Design
  - Solutions Page
- Self-paced trainings
  - Introduction:
    - Simulink onramp
    - Stateflow onramp
    - Simscape onramp
    - Machine learning onramp
    - Deep learning onramp
    - Reinforcement learning onramp
  - In-depth trainings:
    - Simulink
    - Machine learning
    - Deep learning
- Instructor-led trainings
  - AI, Data Science, and Statistics