tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

"A TinyML Approach to Deploy Reduced-Order Model of Complex Systems on Microprocessor"

Brenda Zhuang – Engineering Manager, MathWorks
Greg Coppenrath – Sr. Product Marketing Manager, MathWorks

July 18, 2023

TINY ML

www.tinyML.org

Thank you, **tinyML Strategic Partners**,
for committing to take tinyML to the next Level, together

Executive Strategic Partners

Qualcomm
AI research

Advancing AI research to make efficient AI ubiquitous

Power efficiency

Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization

Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning

Robust learning through minimal data, unsupervised learning, on-device learning

A platform to scale AI across the industry

Perception

Object detection, speech recognition, contextual fusion

Reasoning

Scene understanding, language understanding, behavior prediction

Action

Reinforcement learning for decision making

Edge cloud

Cloud

IoT/IIoT

Automotive

Mobile

Accelerate Your Edge Compute

SYNTIANT

Making Edge AI A Reality

www.syntiant.com

Platinum Strategic Partners

Renesas is enabling the next generation of AI-powered solutions that will revolutionize every industry sector.

renesas.com

DEPLOY VISION AI AT THE EDGE AT SCALE

Gold Strategic Partners

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Where what if becomes what is.

Witness potential made possible at analog.com.

Build the Future of tinyML

on arm

TINY ML TALKS webcast

EDGE IMPULSE

The Leading Development Platform for Edge ML

edgeimpulse.com

Driving decarbonization and digitalization. Together.

Infineon serving all target markets as
Leader in Power Systems and IoT

www.infineon.com

innatera

NEUROMORPHIC INTELLIGENCE FOR THE SENSOR-EDGE

www.innatera.com

Microsoft

The Right Edge AI Tools Can Make or Break Your Next Smart IoT Product

STMicroelectronics provides extensive solutions to make tiny Machine Learning easy

life.augmented

www.st.com/ai

Silver Strategic Partners

Join Growing tinyML Communities:

15.8k members in
49 Groups in 41 Countries

tinyML - Enabling ultra-low Power ML at the Edge

https://www.meetup.com/tinyML-Enabling-ultra-low-Power-ML-at-the-Edge/

3.8k members
&
12.7k followers

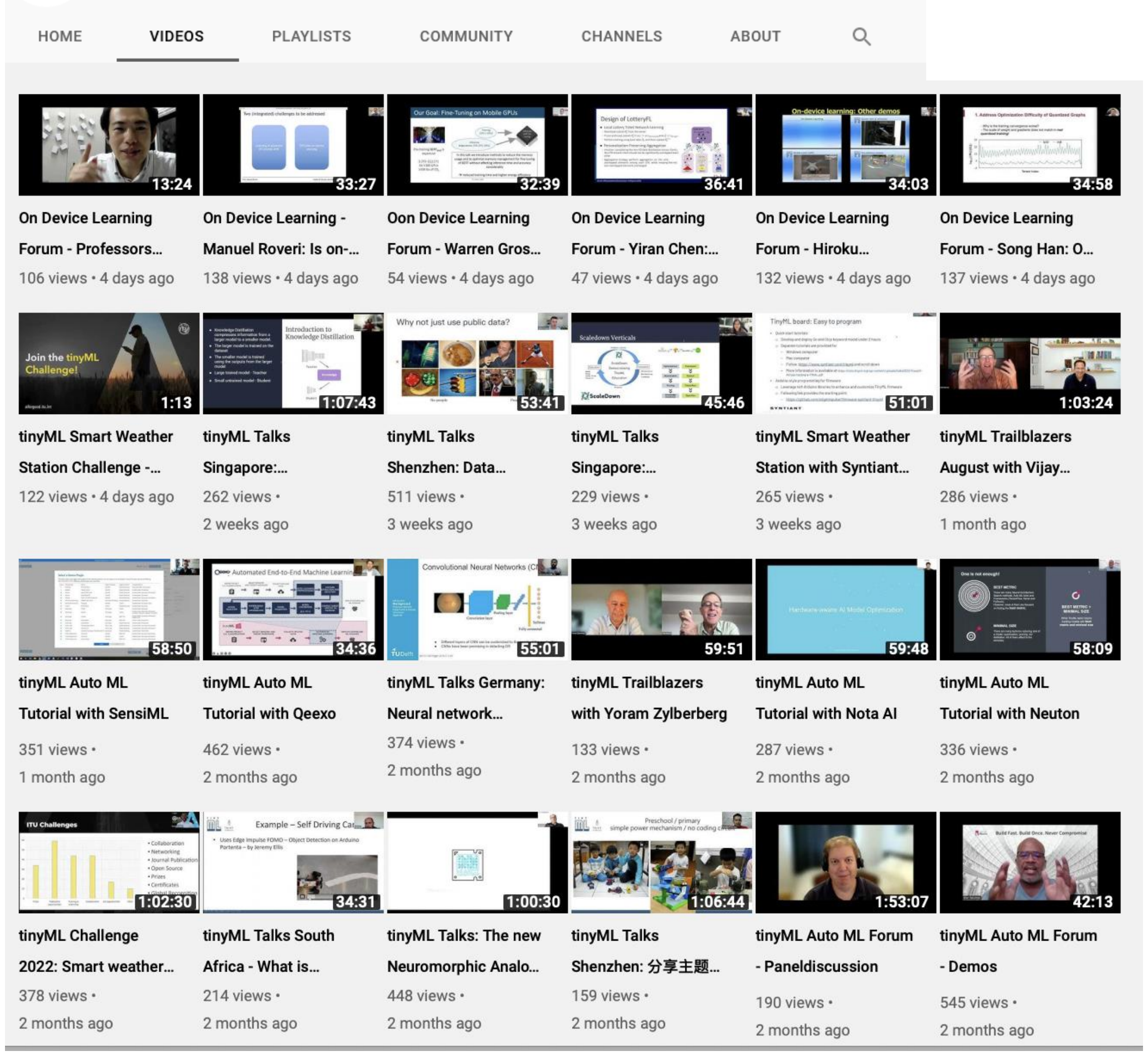
The tinyML Community

https://www.linkedin.com/groups/13694488/

TINY ML TALKS webcast

YouTube

10k subscribers, 607 videos with 354k views



Subscribe to
tinyML YouTube Channel
for updates and notifications
(including this video)

www.youtube.com/tinyML

Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums youtube.com/tinyml

Please use the Q&A window for your questions

Brenda Zhuang

Dr. Brenda Zhuang is a consulting engineer and engineering manager at MathWorks, where she leads a team responsible for software tools for automatic deployment of embedded applications, such as motor controls and deep learning, to microprocessors and FPGAs. Brenda joined MathWorks in 2007. She received her PhD from Boston University in Systems Engineering. She serves on the technical program committee in control theory, modeling and simulation.

Greg Coppenrath

Greg is the product marketing manager for Fixed-Point Designer and Deep Learning Toolbox Model Quantization Library. He has experience in the development of embedded systems and product development in the semiconductor industry. He received an MBA from Worcester Polytechnic Institute, an M.S. in Electrical Engineering from the University of Massachusetts Lowell, and received a B.S. in Electrical Engineering from WPI.

A TinyML Approach to Deploy Reduced-Order Model on Microprocessor

MathWorks®

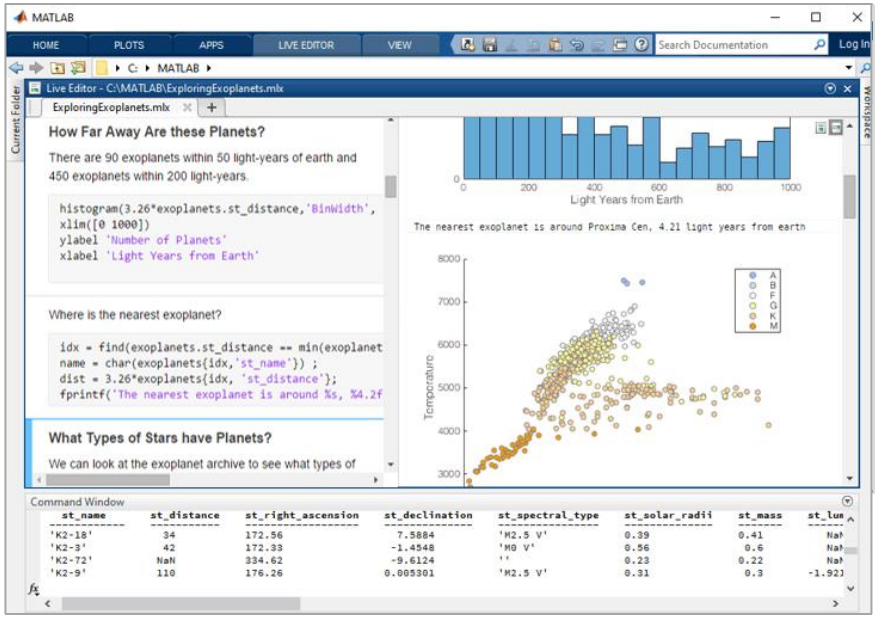
Meet the speakers today

Brenda Zhuang, PhD
Engineering Manager
MathWorks

Greg Coppenrath
Senior Product Manager
MathWorks

Our Products

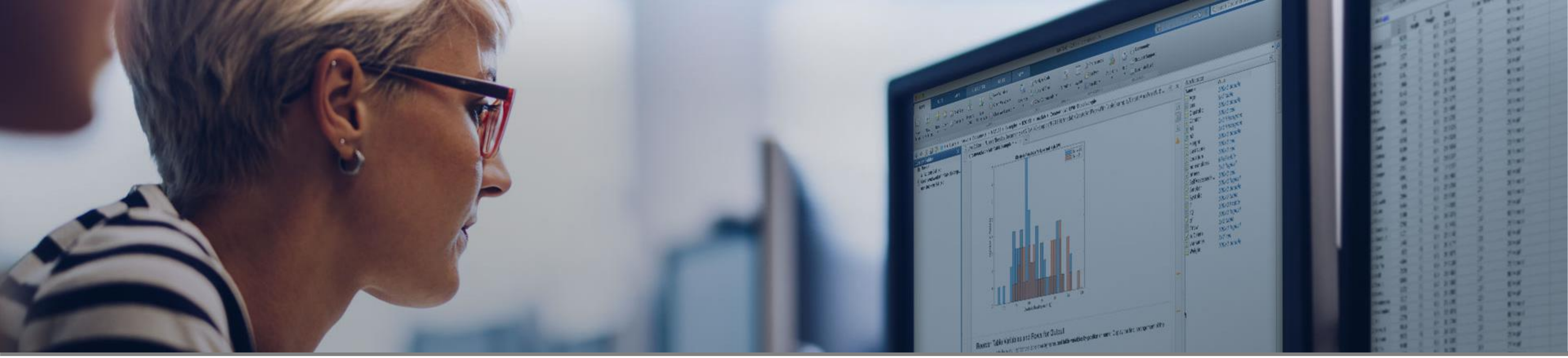
Computer Vision Toolbox
MATLAB
Simulink



MATLAB is a programming environment for algorithm development, data analysis, visualization, and numeric computation.

Simulink is a block diagram environment for designing, simulating, and testing systems.

More than 120 add-on products for specialized tasks.



Our Customers

Millions of engineers and scientists worldwide use MATLAB and Simulink.

5 million+
users in over 180 countries

100,000+
businesses, governments, and universities

All of the top 10 automotive and aerospace companies

Fortune: 2021 Global 500 auto companies
FlightGlobal: 2020 Top 100 aero companies*

*Excluding companies that are subject to embargos, sanctions, or other controls

MathWorks

MathWorks Today

6000+ staff
in 34 offices around the world

$1.25+ billion
in revenues

Privately held

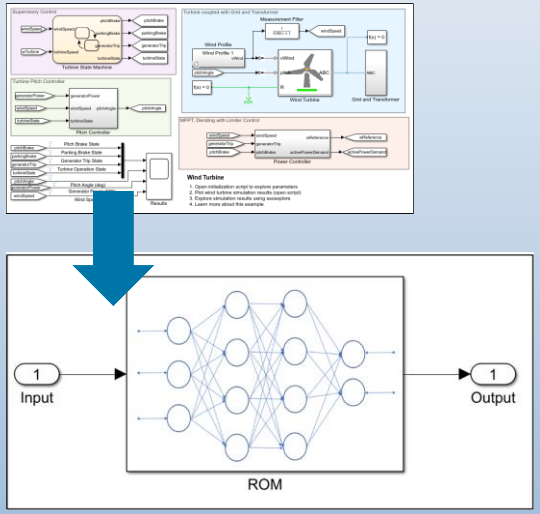
MathWorks

What’s New for Our Products

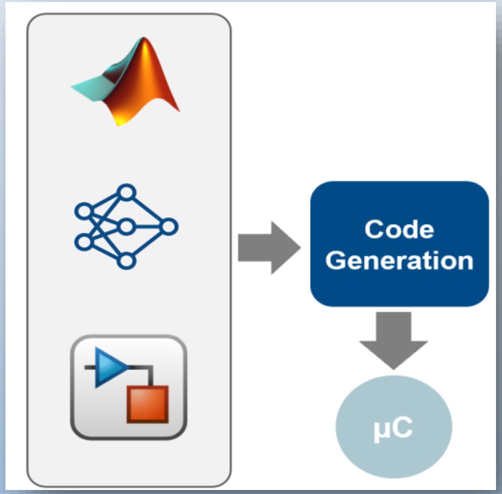
Megatrends changing our world

- Software in Everything
- Artificial Intelligence
- Autonomous Systems
- Digital Transformation
- Electrification of Everything

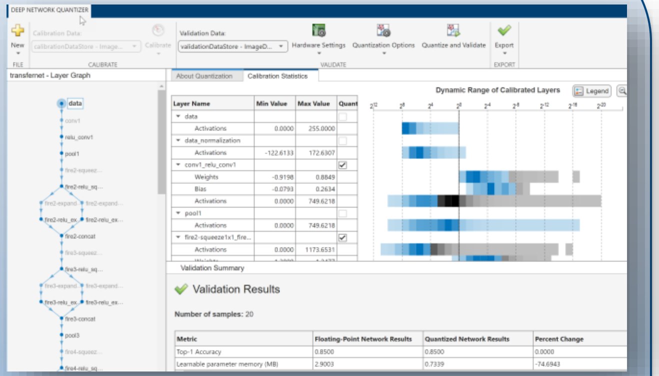
Today, we cover the workflow steps from model development using AI-driven methodology to compression for target deployment



Reduced-Order Model (ROM)



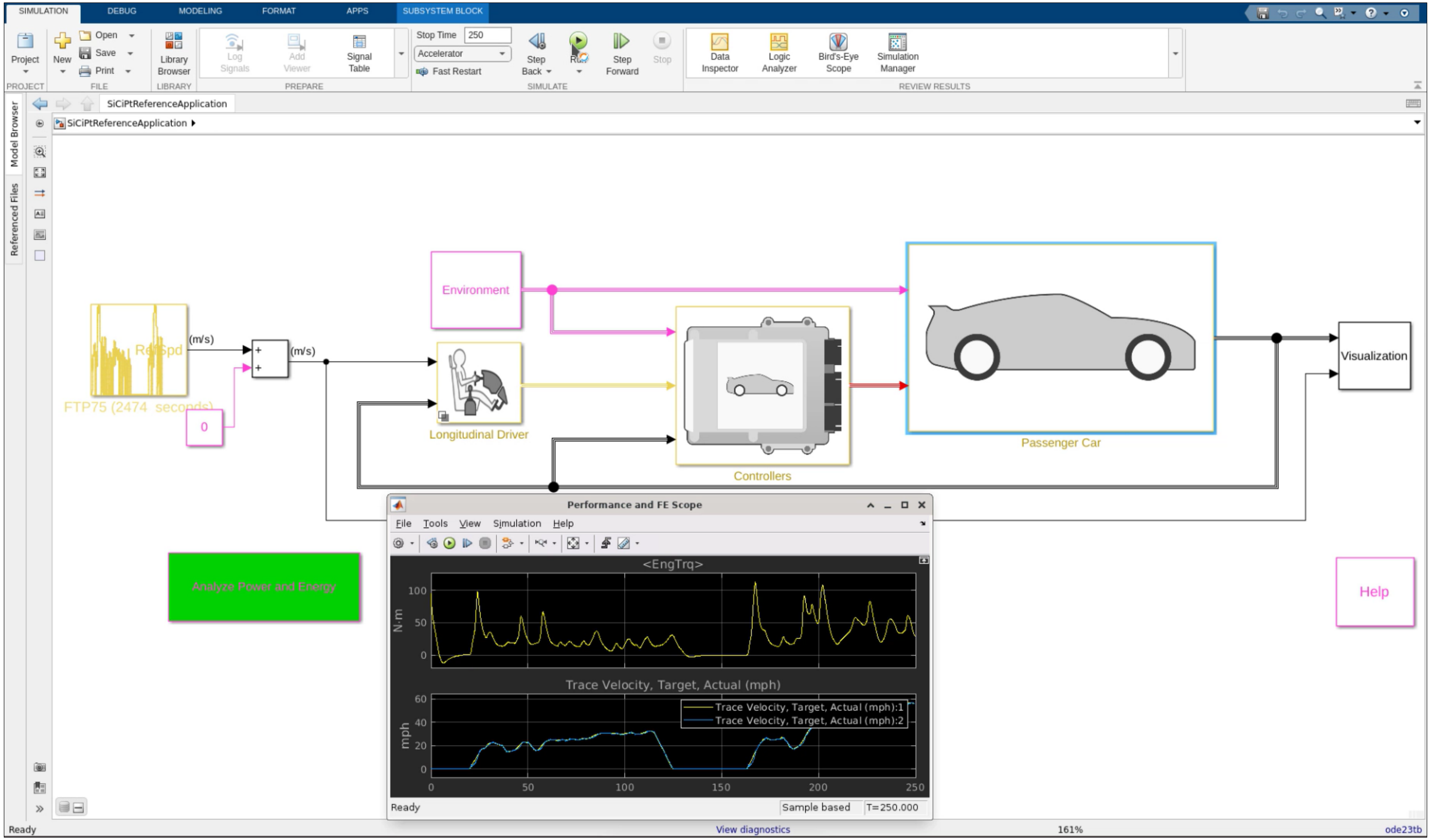
AI Model
Target Deployment



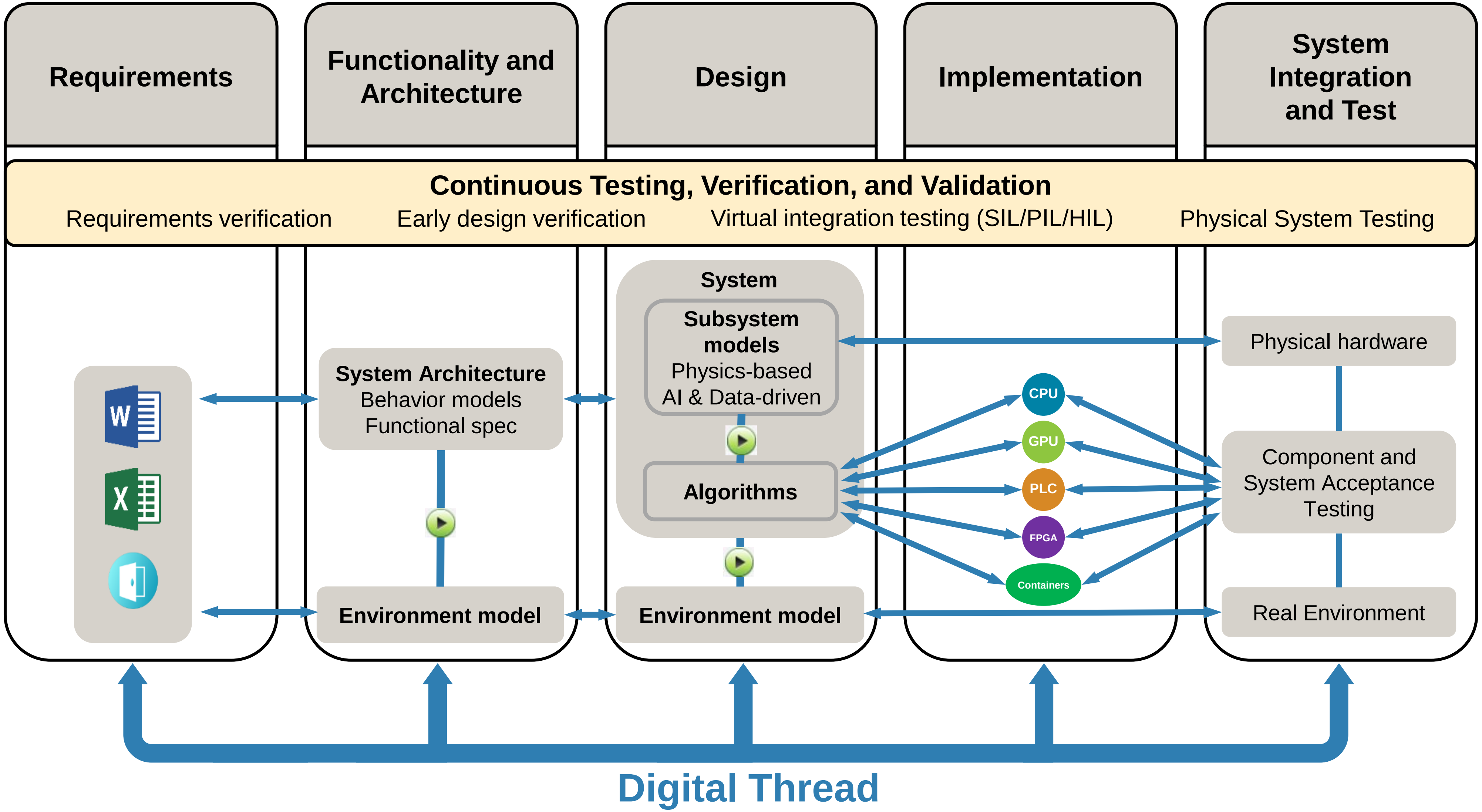
Model Compression

Example overview

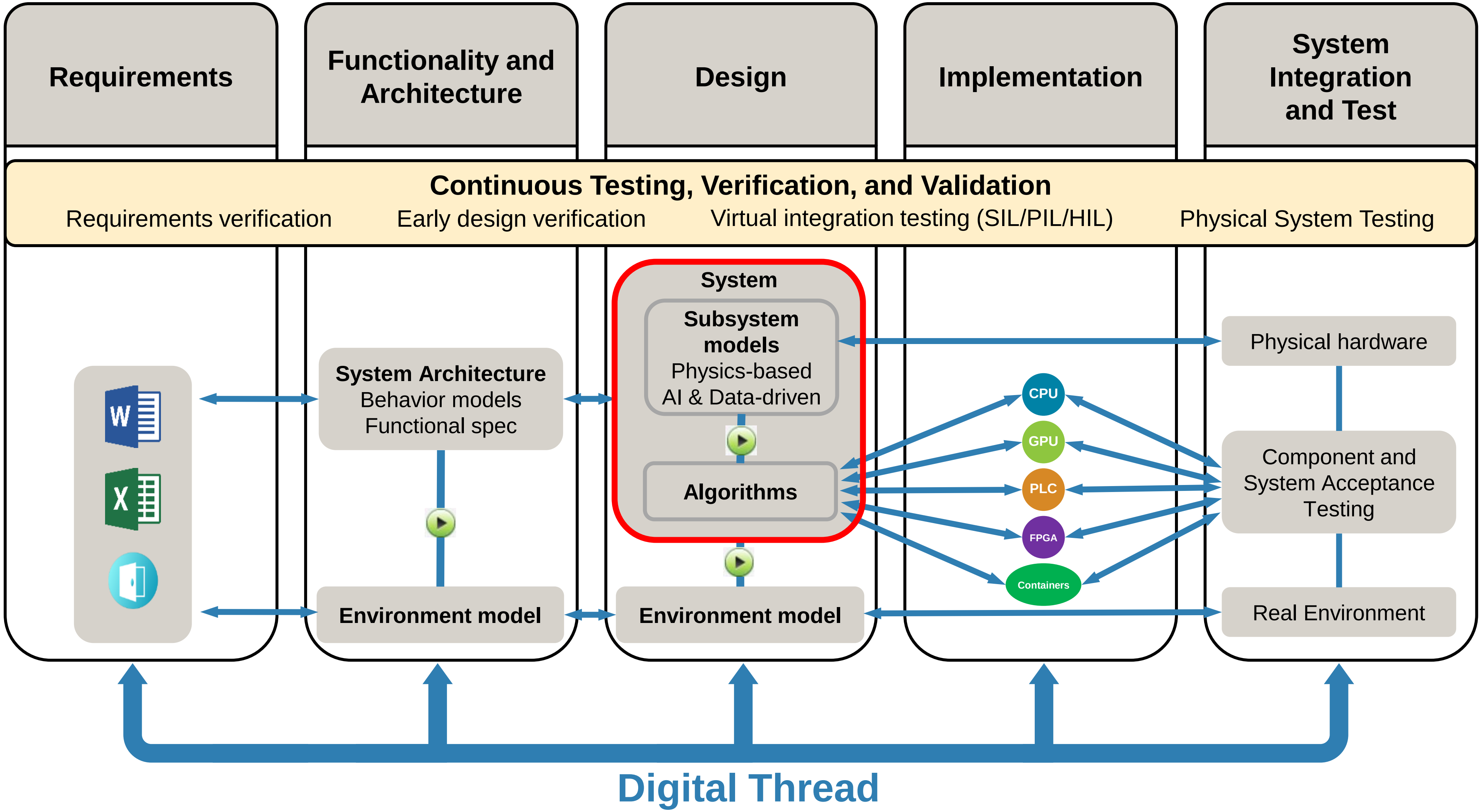
System-level simulation



Model-Based Design



Integrating AI into Model-Based Design



AI-driven system design

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

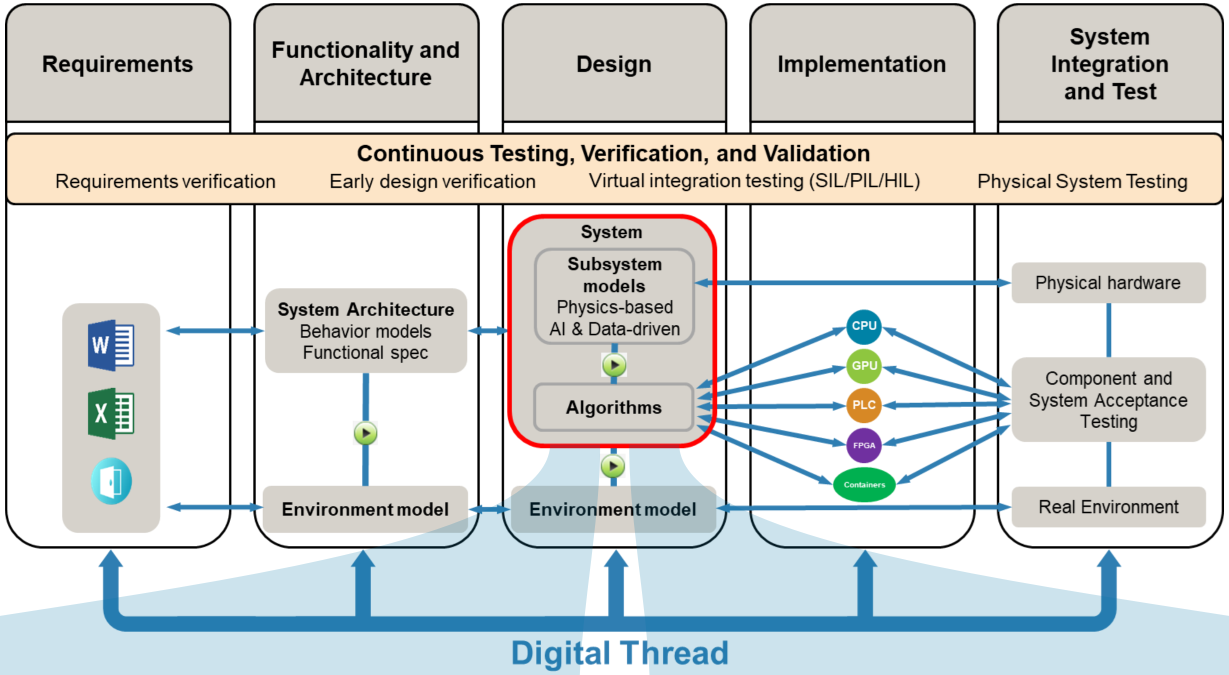
Simulation & Test

- Integration with complex systems
- System simulation
- System verification and validation

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

Integrate AI models into MBD for system-level simulation and code generation



AI for component modeling

- Speeding up desktop and HIL simulations
- Modeling component dynamics from data when first-principles models cannot be obtained

AI for algorithm development

- Virtual sensor modeling
- Sensor fusion
- Object detection

Reduced Order Modeling

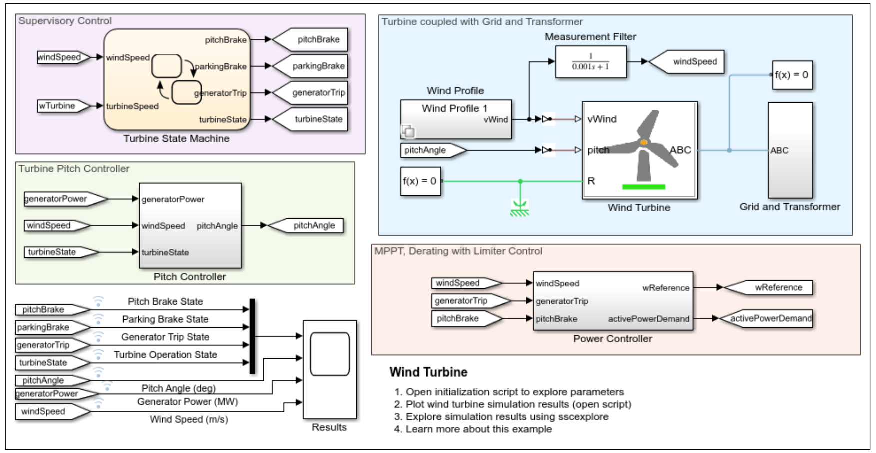
What

- Techniques to **reduce the computational complexity or storage requirement** of a computer model
- **Preserve the expected fidelity** within a controlled error

Why

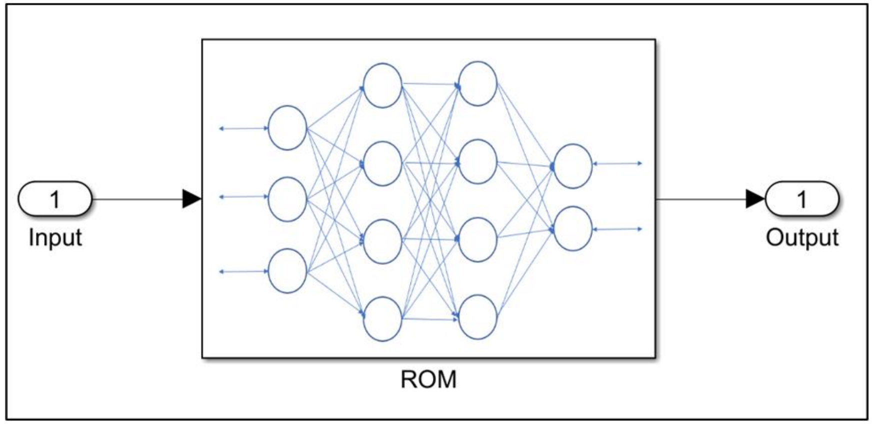
- Speed up system-level desktop simulation
- Hardware-in-the-loop testing
- Enable system-level simulation
- Develop virtual sensor, Digital twins
- Perform control design

High-fidelity model



Simulation time

Reduced-Order Model (ROM)

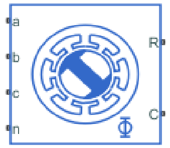


Reduced Order Modeling

How

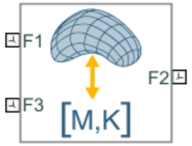
Reduced order model

Physics-based



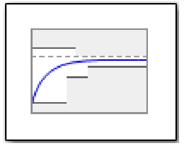
FEM-Parametrized PMSM

Model-based



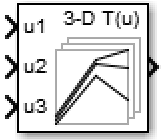
Reduced Order Flexible Solid

Linearization



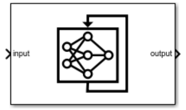
Data-driven

Static



Simulink, Curve Fitting Toolbox, Model-Based Calibration Toolbox

Dynamic



Simulink, Statistics and Machine Learning Toolbox, Deep Learning Toolbox, System Identification Toolbox

Focus today

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

BLACK BOX **GREY BOX** **WHITE BOX**

Data-driven models

Advantages

- May succeed when first-principles models are unavailable or challenging/impossible to find
- May reduce complexity, simulate faster
- Can leverage existing, measured data
- Do not require domain knowledge

Challenges

- Require a lot of data
- Are often not
  - interpretable, explainable
  - easily parameterizable in a physically meaningful way
- Cannot extrapolate well beyond training data

First-principles models

Advantages

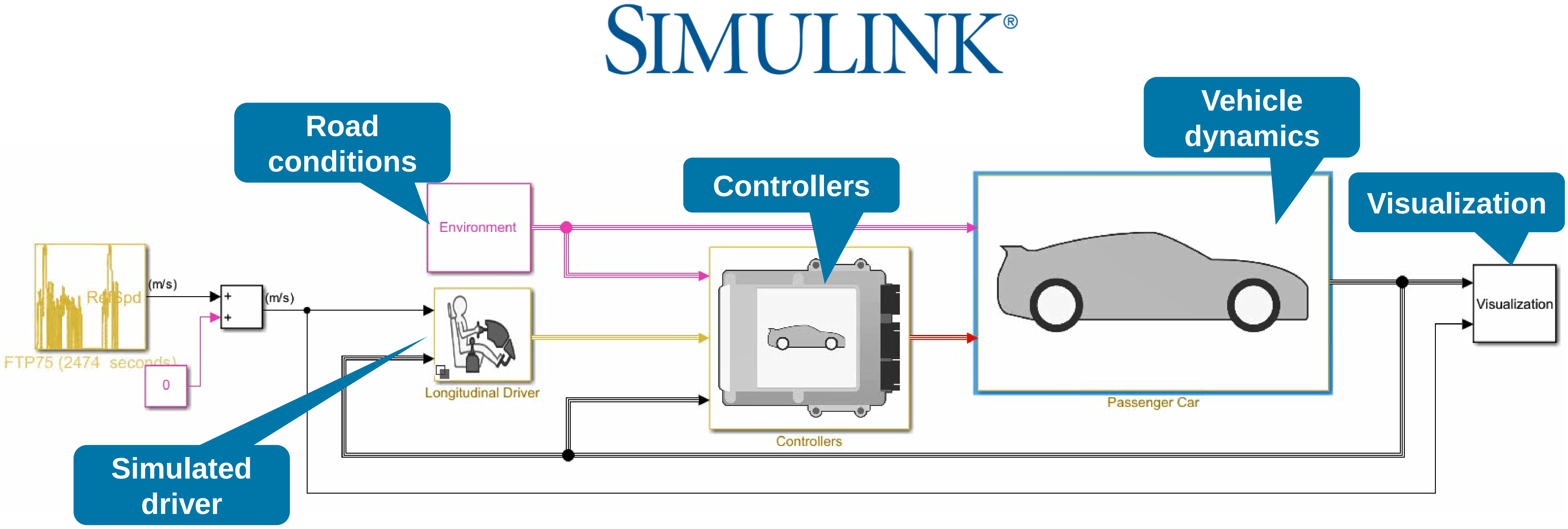
- May capture (global) parameterizable behaviors with low/high fidelity
- Have clear (explainable) physical meaning
- Do not require data engineering

Challenges

- Can be challenging/impossible to derive
- Require significant time for derivation
- Require expertise in the respective domain

Example overview

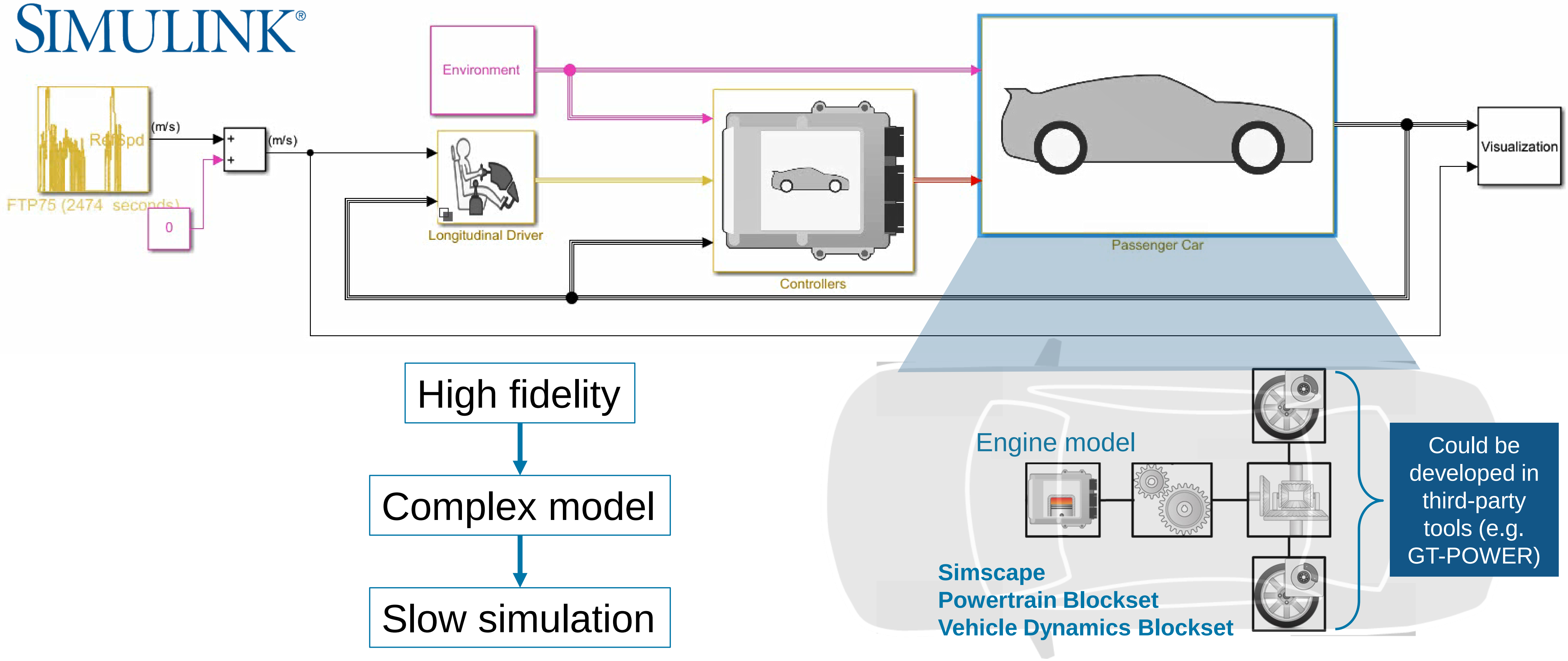
Replacing a first-principles engine model with an AI-based Reduced Order Model



Closed-loop control of vehicle speed

Example overview

Replacing a first-principles engine model with an AI-based Reduced Order Model

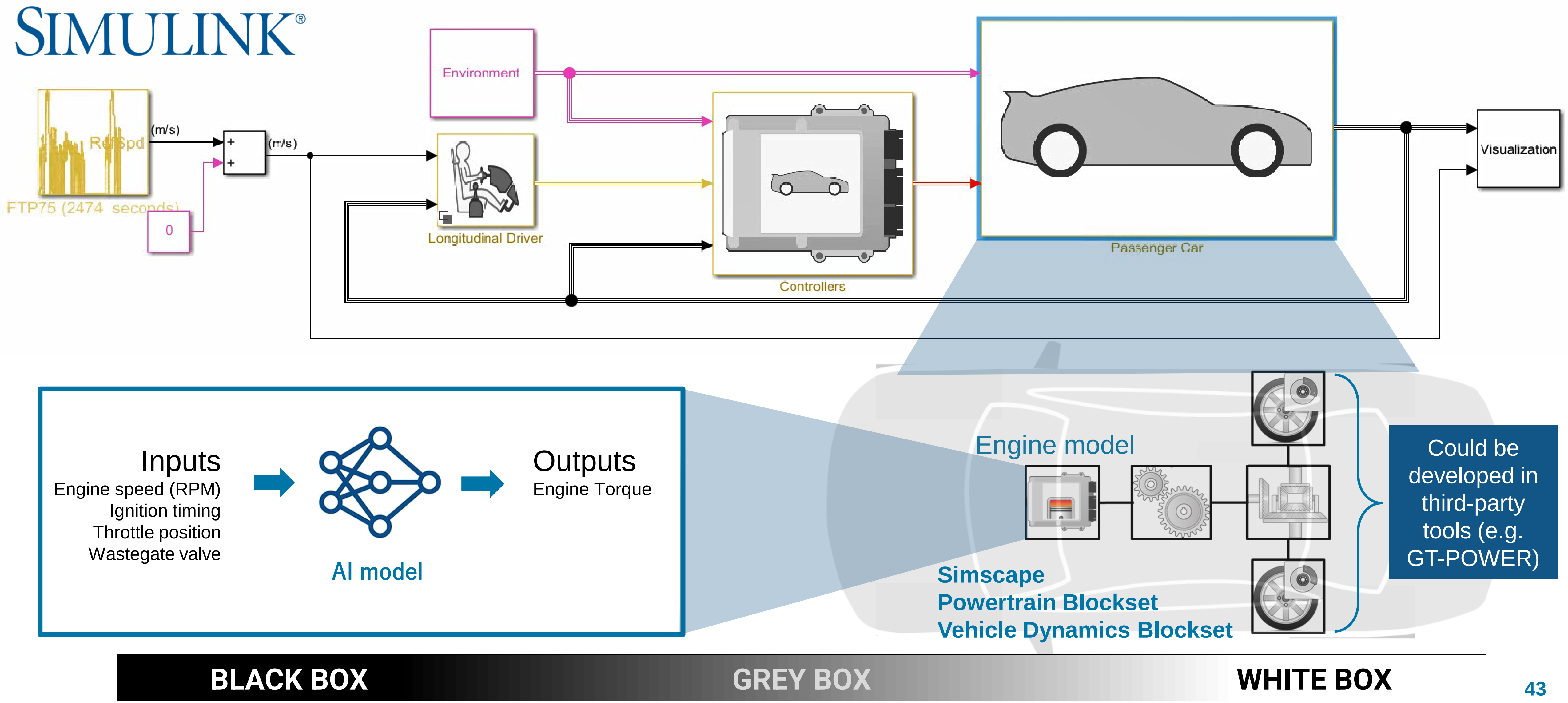


High fidelity → Complex model → Slow simulation

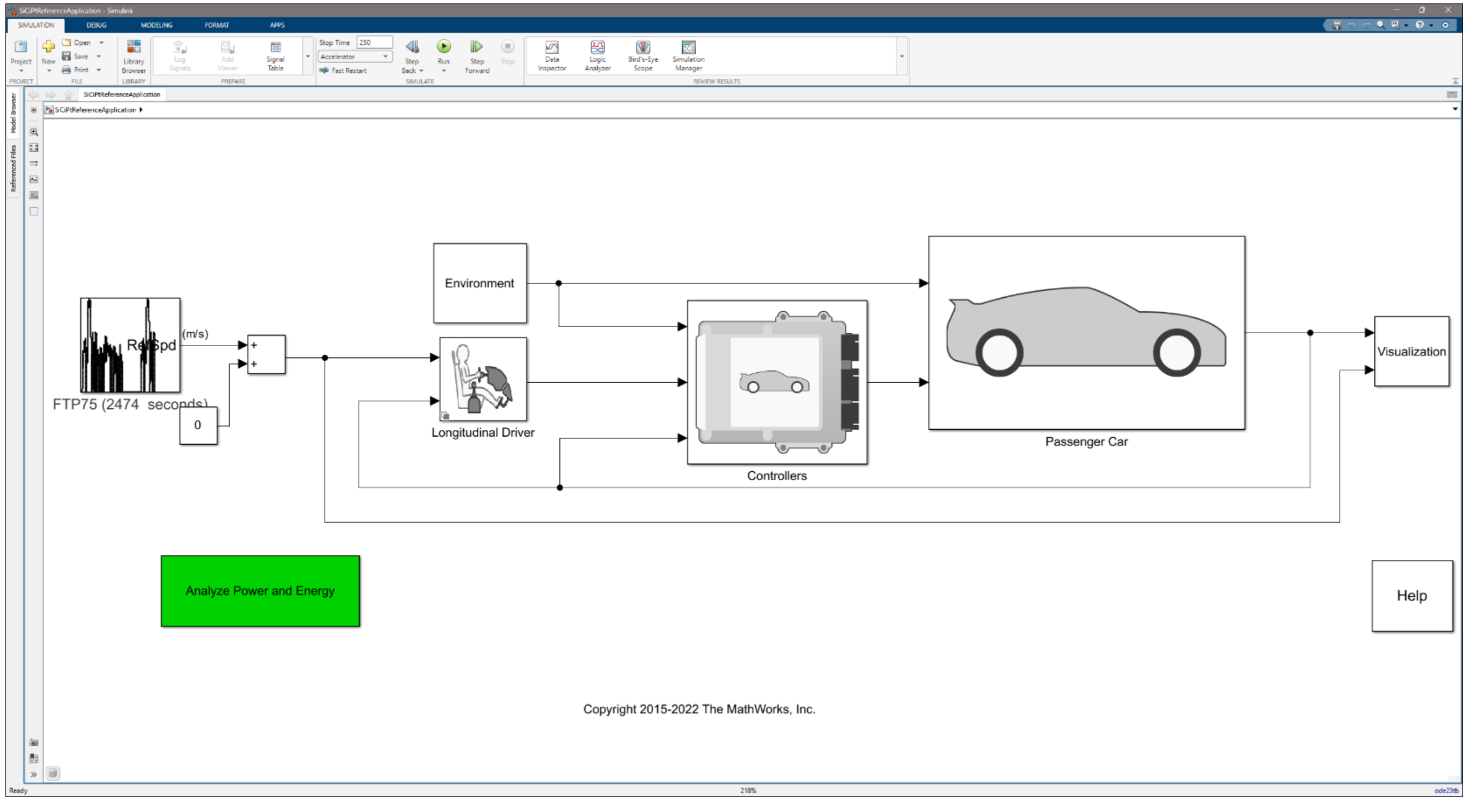
Could be developed in third-party tools (e.g. GT-POWER)

Example overview

Replacing a first-principles engine model with an AI-based Reduced Order Model



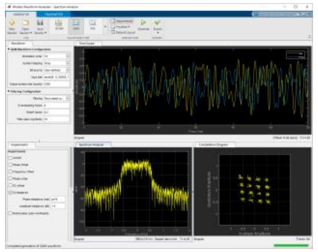
Generate synthetic data for training



Simulink/Simscape

Other techniques:

GANs



Wireless Waveform Generator

Unreal Engine®

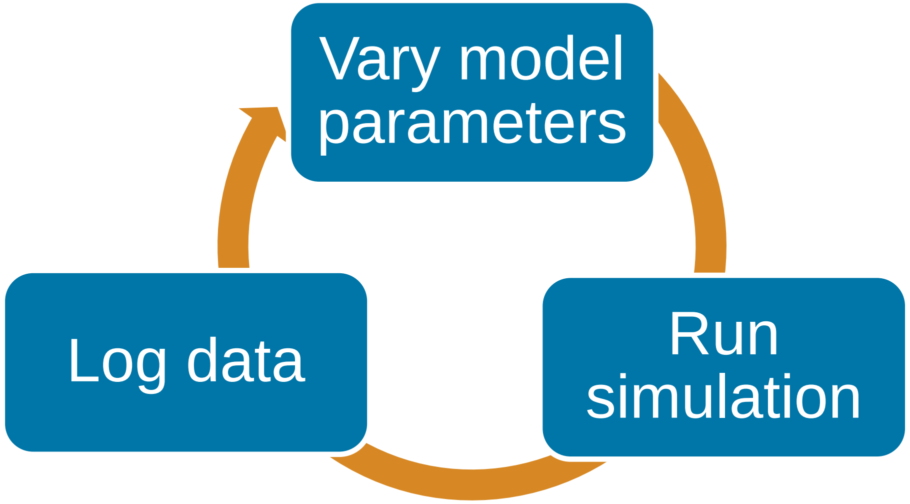
Data Preparation | AI Modeling | Simulation & Test | Deployment

Synthetic Data Generation

Design of Experiments

DoE = 512×3 table

	EngTrqReq	EngSpdR...	SpkAdvOfst
1	60	2000	-30
2	128	2500	15
3	94	2750	8
4	111	2875	-19
5	77	2625	-11
6	144	2125	4
7	85	2563	-21
8	119	3313	-28
9	68	2938	21

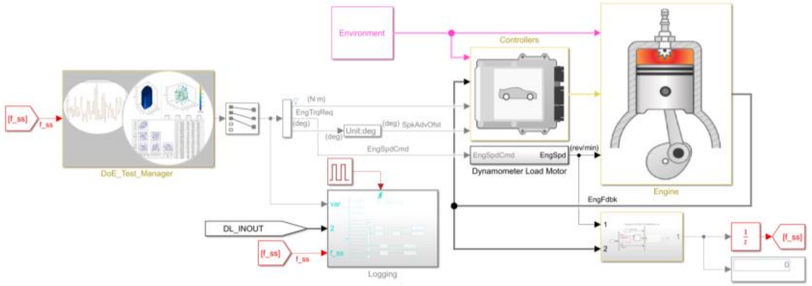


<u>Input features</u>

Engine speed (RPM)
Ignition timing
Throttle position
Wastegate valve

<u>Response</u>

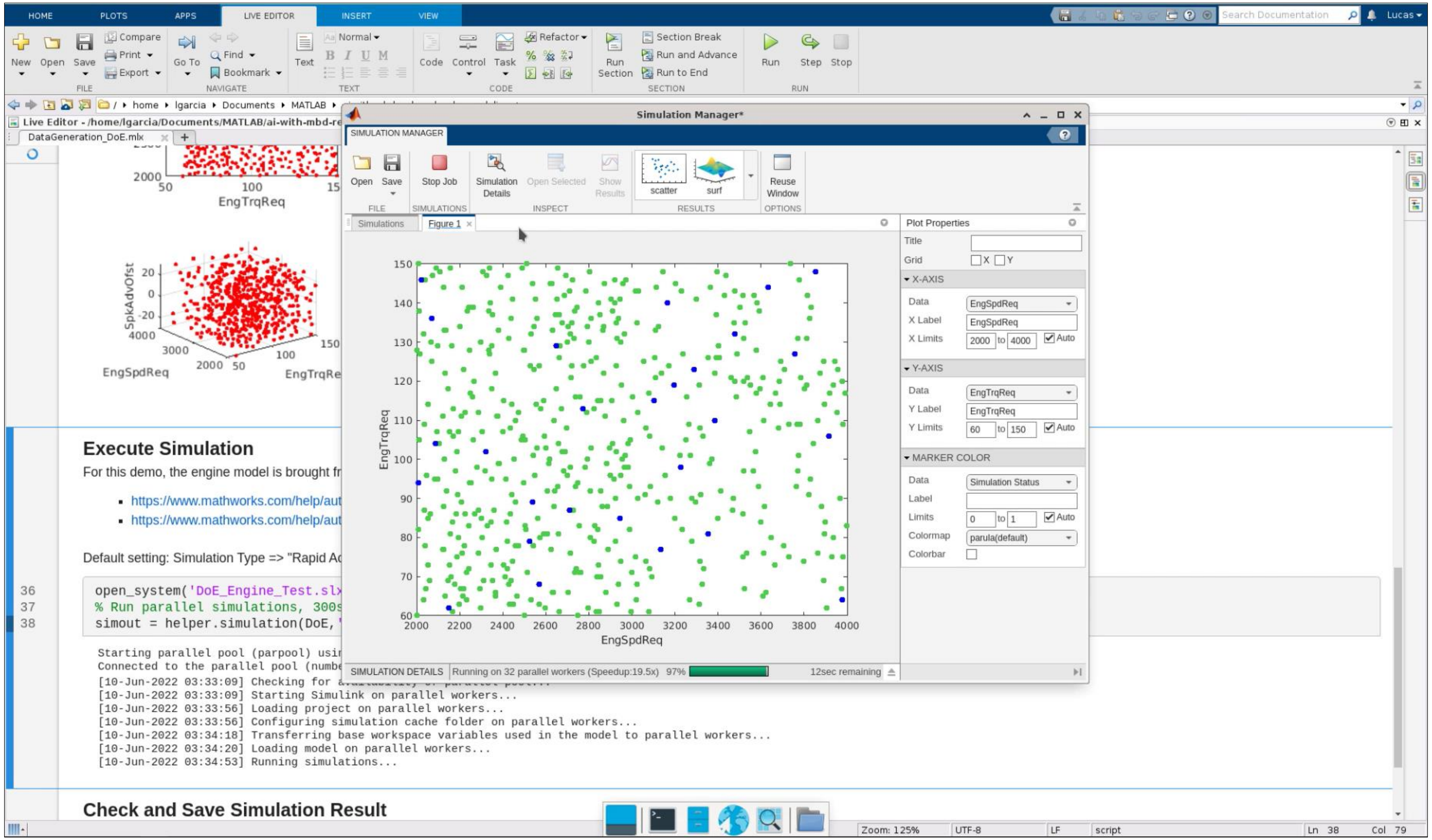
Engine Torque



Data Preparation | AI Modeling | Simulation & Test | Deployment

Synthetic Data Generation

Design of Experiments



Data Preparation | AI Modeling | Simulation & Test | Deployment

Data-driven ROM

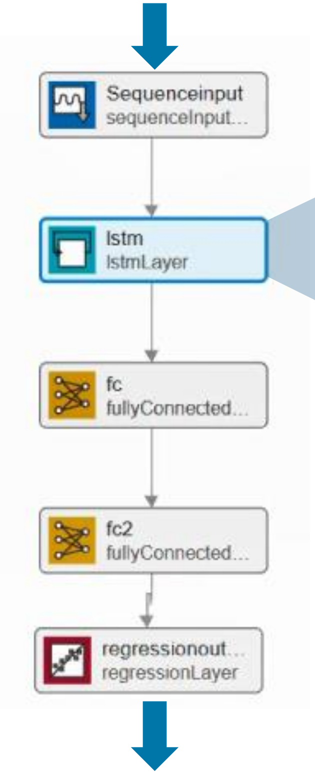
- Data-Driven ROM
  - Static
    - Look up table
    - Surface Fitting
    - etc.
  - Dynamic
    - LSTM
    - Neural State Space / Neural ODE
    - NLARX models
    - etc.

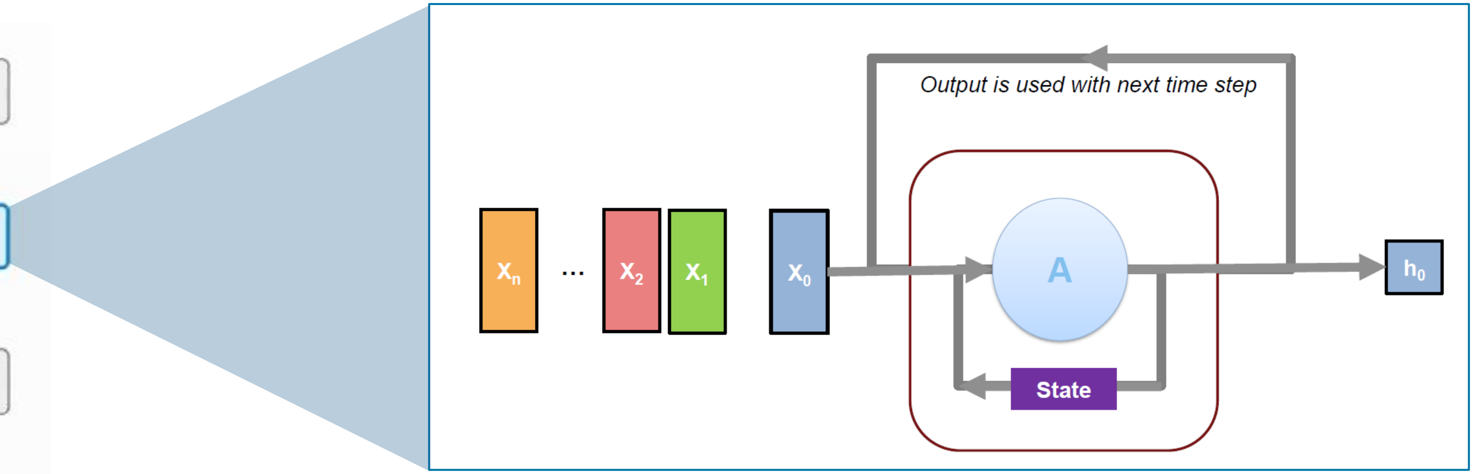
AI-based ROM using LSTMs

Capture time dependencies in time-series data

Inputs

- Engine speed (RPM)
- Ignition timing
- Throttle position
- Wastegate valve





LSTMs carry a memory cell (state) throughout

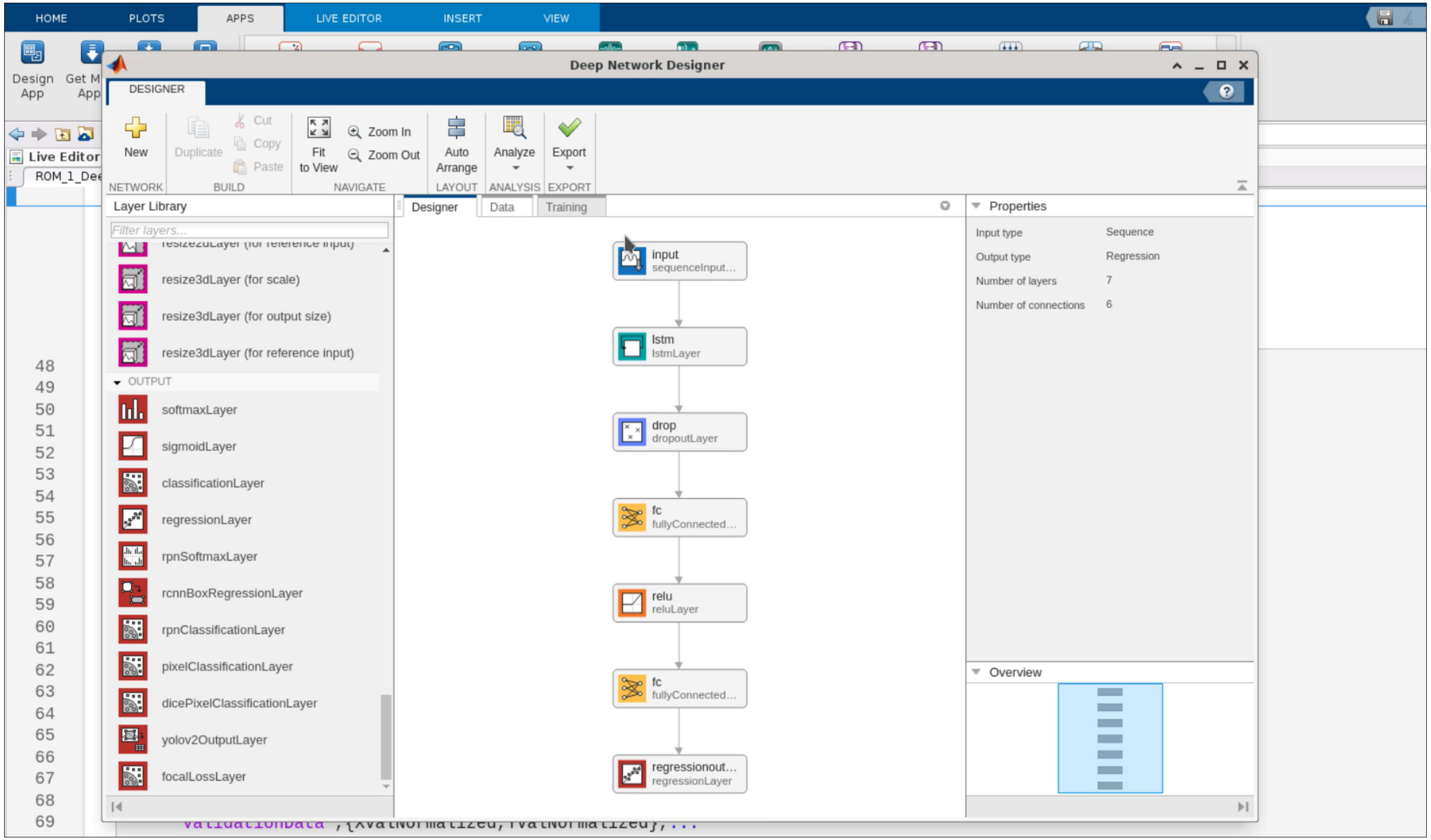
Outputs

- Engine torque

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

AI-based ROM using LSTMs

Capture time dependencies in time-series data

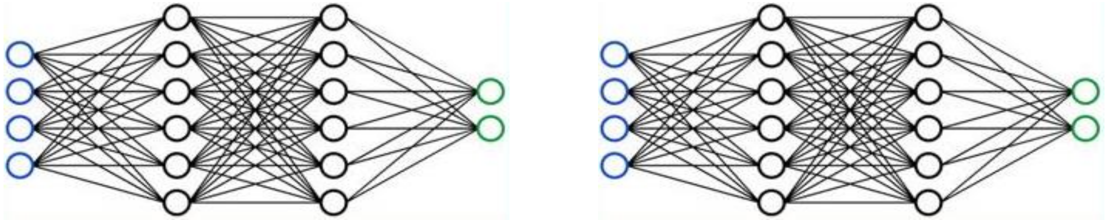


Data Preparation | **AI Modeling** | **Simulation & Test** | **Deployment**

AI-based ROM using Neural State Space

Create DL-based nonlinear state-space models without having to be a deep learning expert

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \end{cases} \Rightarrow$$



State Network (*f*) Output Network (*g*)

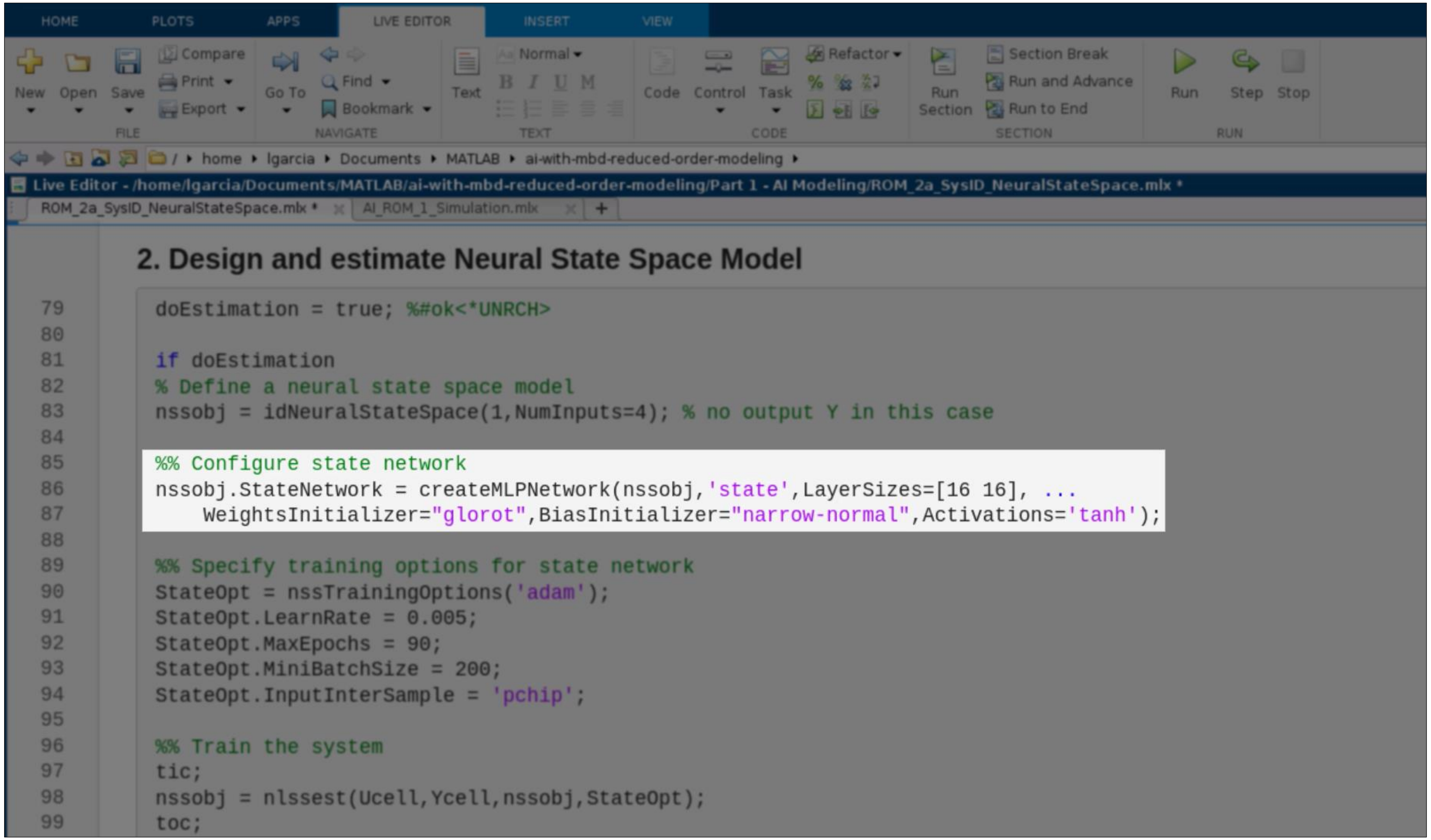
- The nonlinear state function f and nonlinear output function g are feedforward neural networks that learn from data

- Popularly known as Neural ODE in deep learning community

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

AI-based ROM using Neural State Space

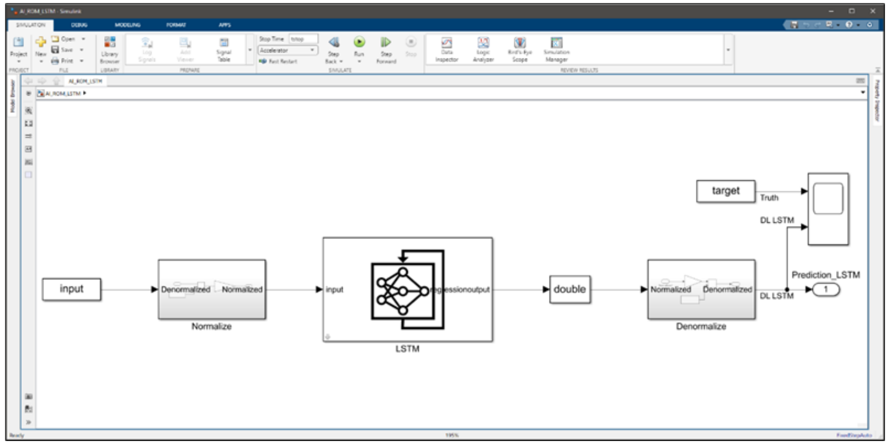
Create DL-based nonlinear state-space models without having to be a deep learning expert



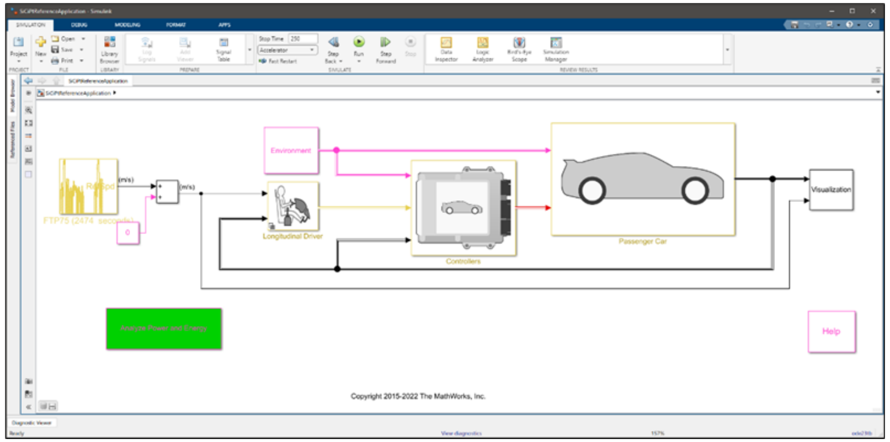
Data Preparation | **AI Modeling** | **Simulation & Test** | **Deployment**

Integrate your AI model for system-level simulation and test

Integration of trained AI model into Simulink

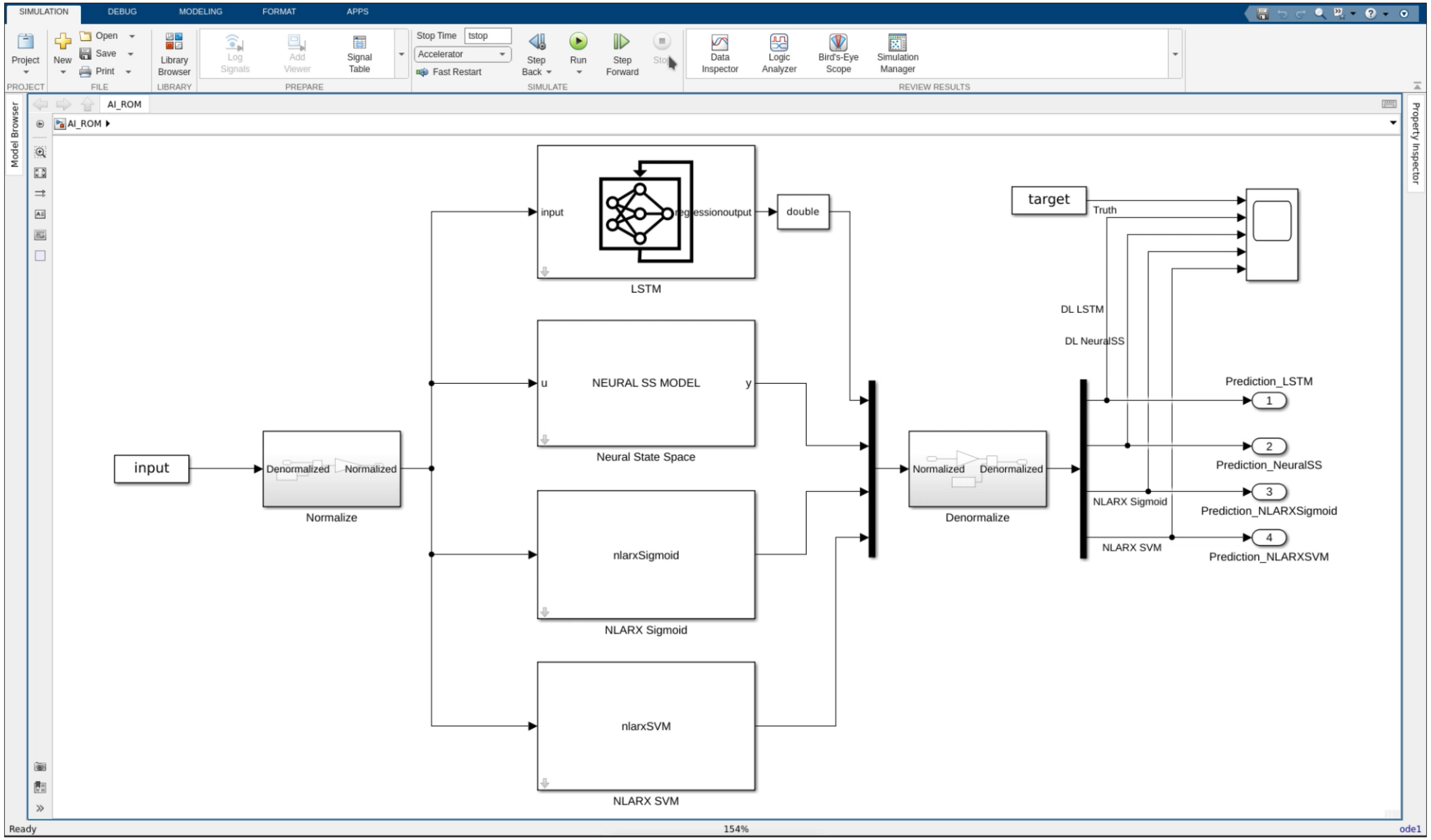


System-level simulation



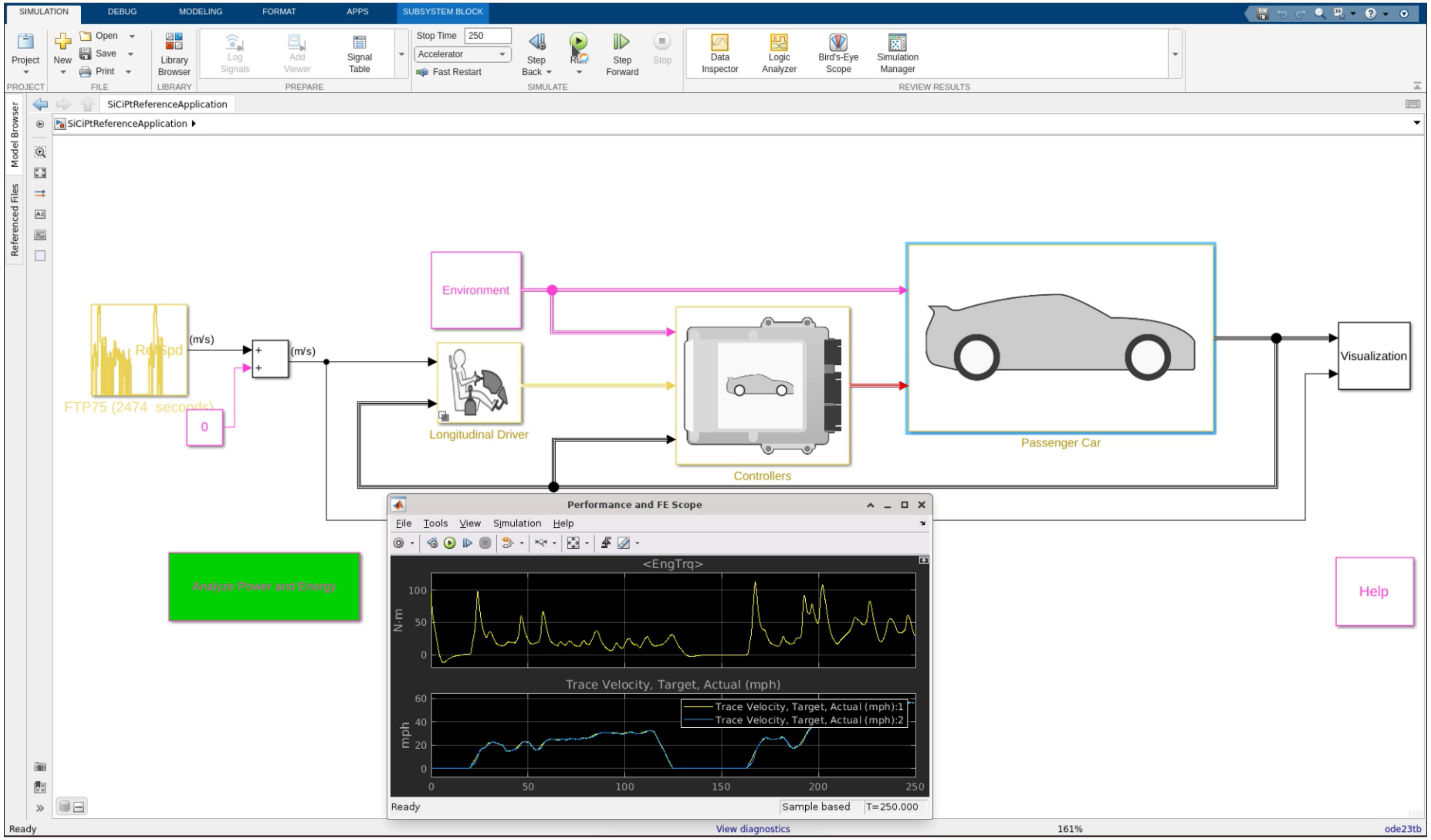
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

Integration of trained AI models into Simulink



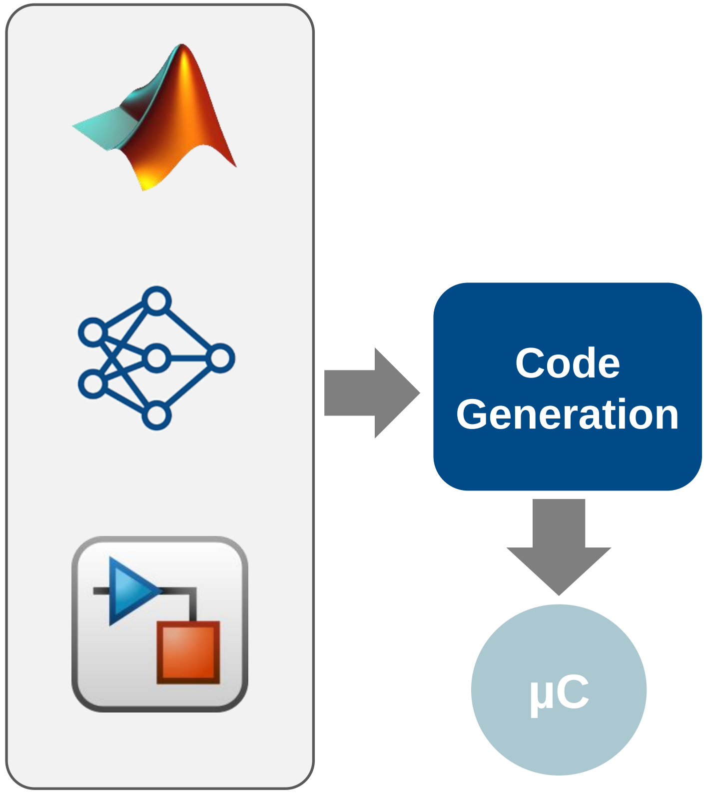
Data Preparation | AI Modeling | **Simulation & Test** | Deployment

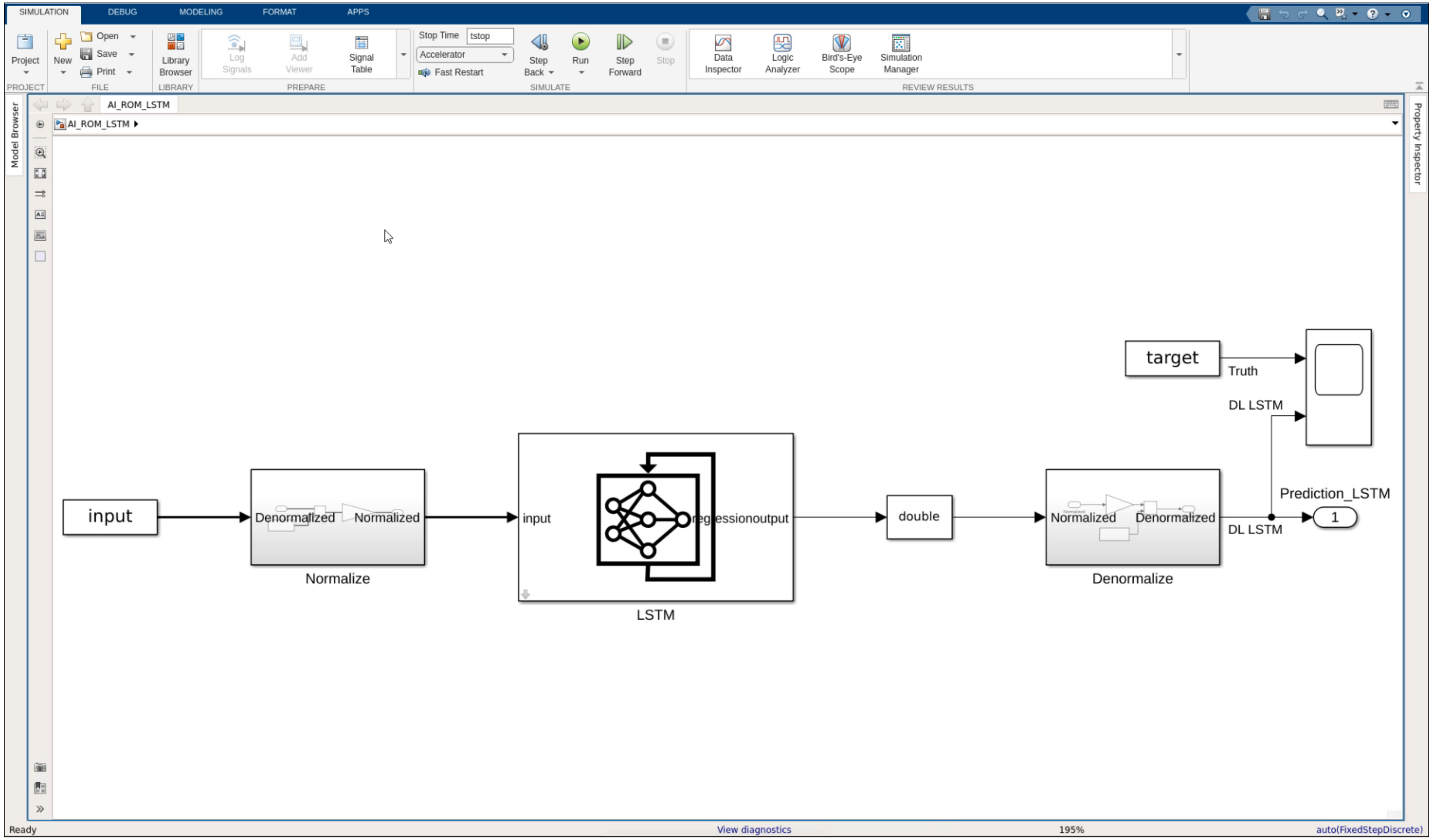
System-level simulation



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

Generate Library-Free C Code for Deep Learning Networks



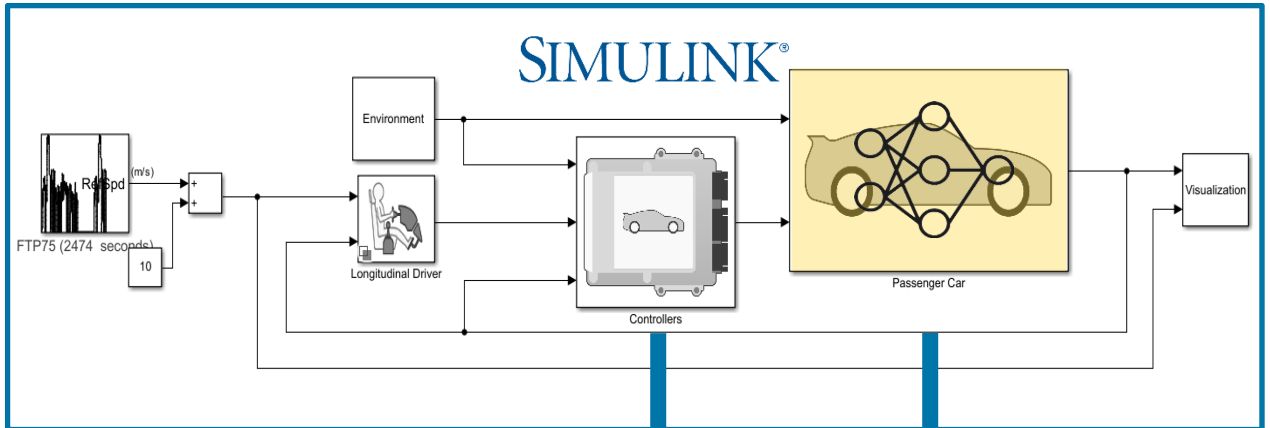


Data Preparation | AI Modeling | Simulation & Test | **Deployment**

Hardware-in-the-loop simulation

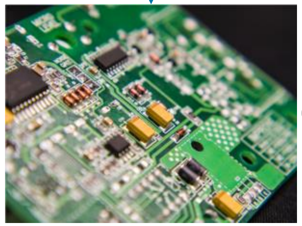
System-level integration and test

SIMULINK



Code generation from algorithm

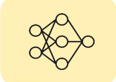
Code generation from plant model



Target platform

I/O

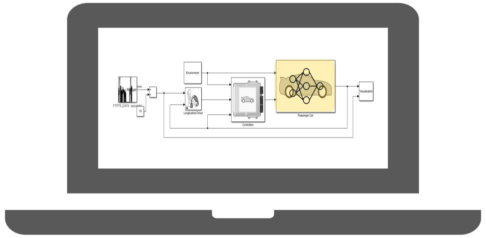
Real-time computer



Build & download models
Adjust parameters

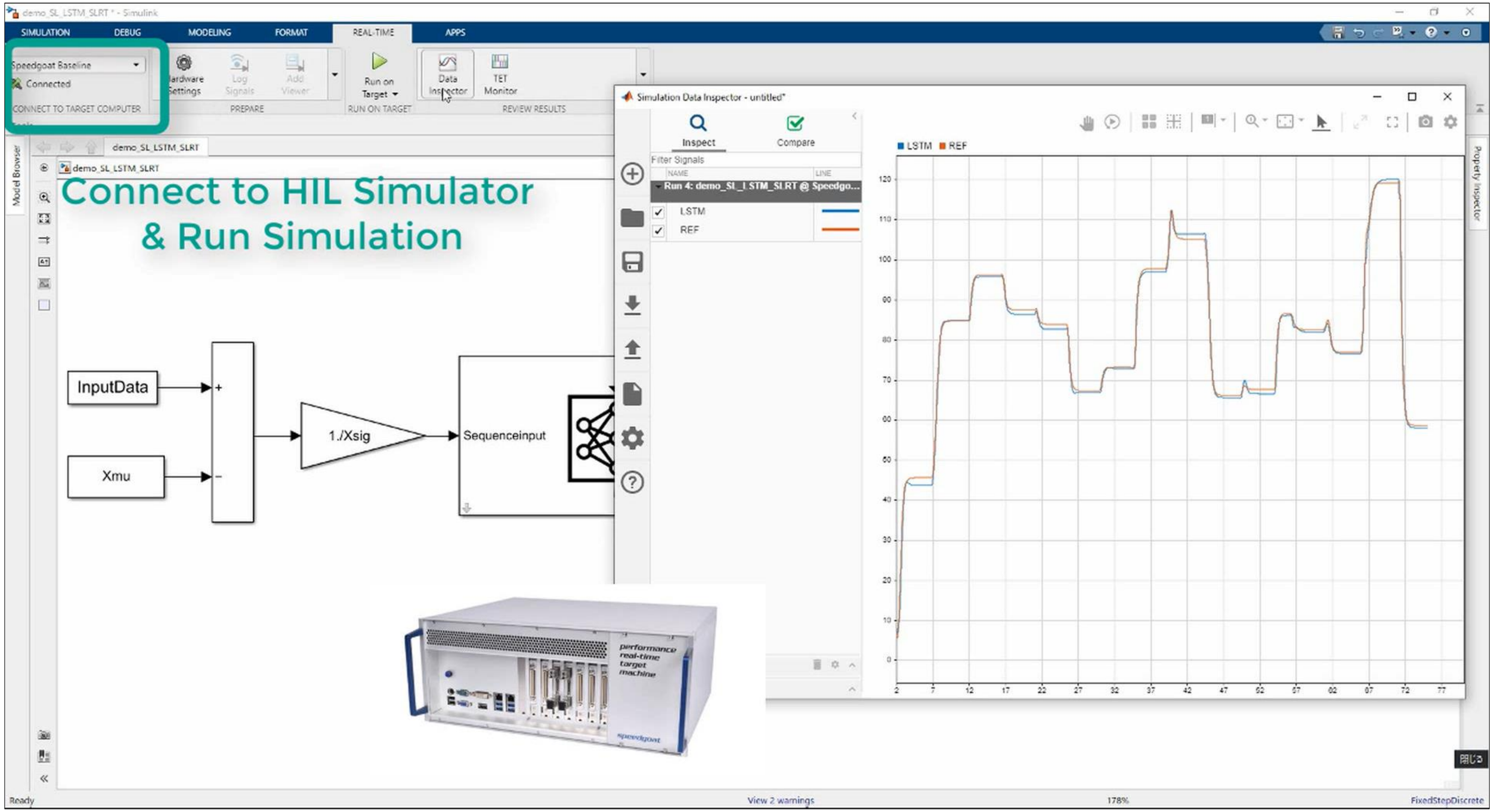
Monitor signals

SIMULINK



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

Hardware-in-the-loop simulation



Data Preparation | AI Modeling | Simulation & Test | Deployment

Manage AI tradeoffs for your system

	LSTM Long Short-Term Memory Network	**Neural SS** Neural State Space (Neural ODE)
Training Speed	Worse / Better*	Okay
Interpretability	Worse	Worse
Inference Speed	Okay	Better
Model Size	Okay	Better
Accuracy (RSME)	Better	Better

Results are specific to Vehicle Engine ROM example

Better Okay Worse

* *if trained using a GPU. Testing made with GPU NVIDIA A100*

Model compression bridges the gap between AI modelling and embedded deployment.

Data Preparation
- Data cleansing and preparation
- Human insight
- Simulation-generated data

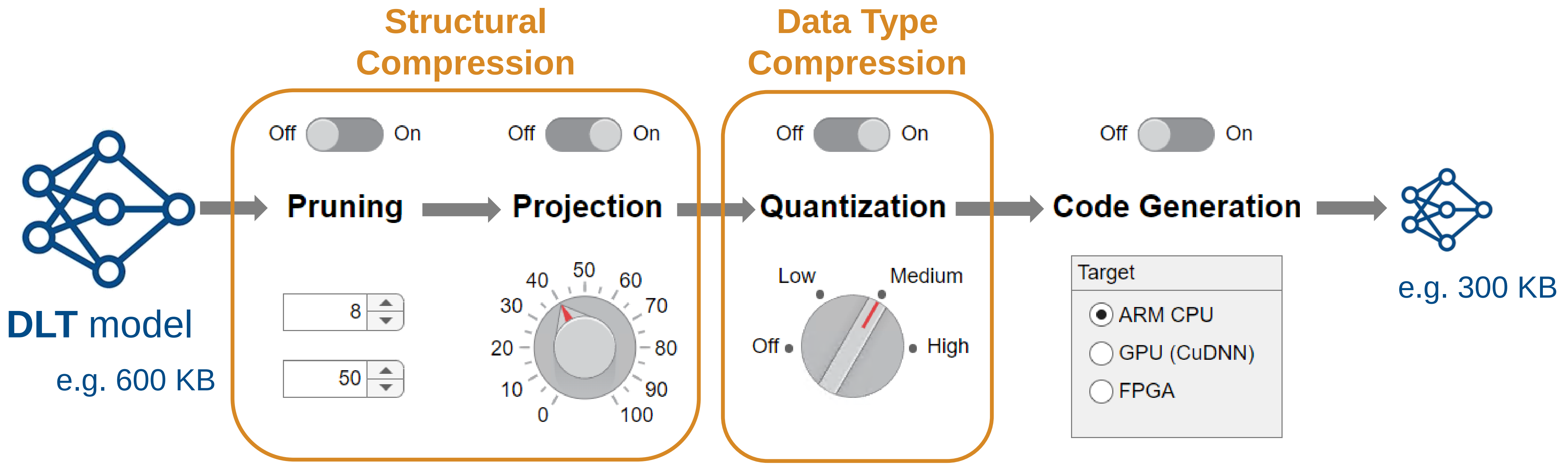
AI Modeling
- Model design and tuning
- Hardware accelerated training
- Interoperability

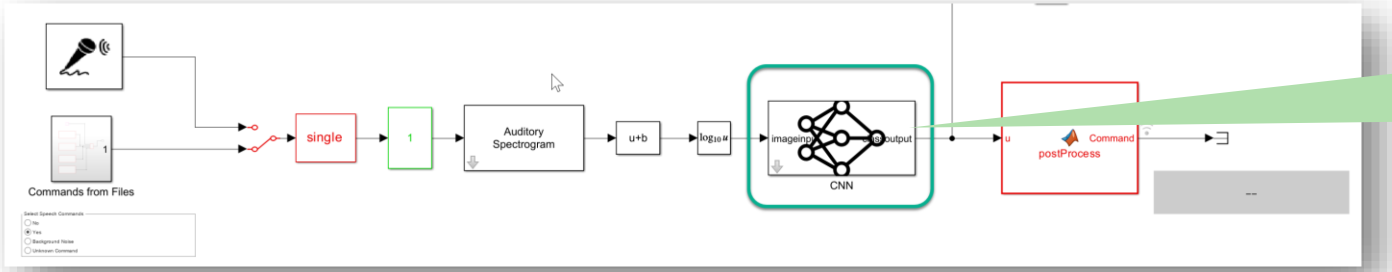
Con
- Pru
- Qu
- Iter

Simulation & Test
- Integration with complex systems
- System simulation
- System verification and validation

Deployment
- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

Problem Statement: Reduce model footprint and accelerate inference of DL models

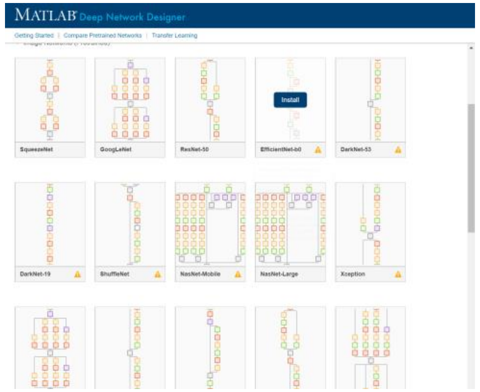




"model is 600kb and want to ***reduce it to smaller****. What to do?"*

Workflow steps to compress Deep Neural Nets

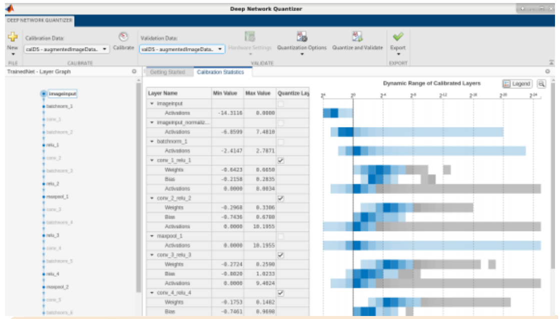
1 Deep Network Designer



2 Pruning

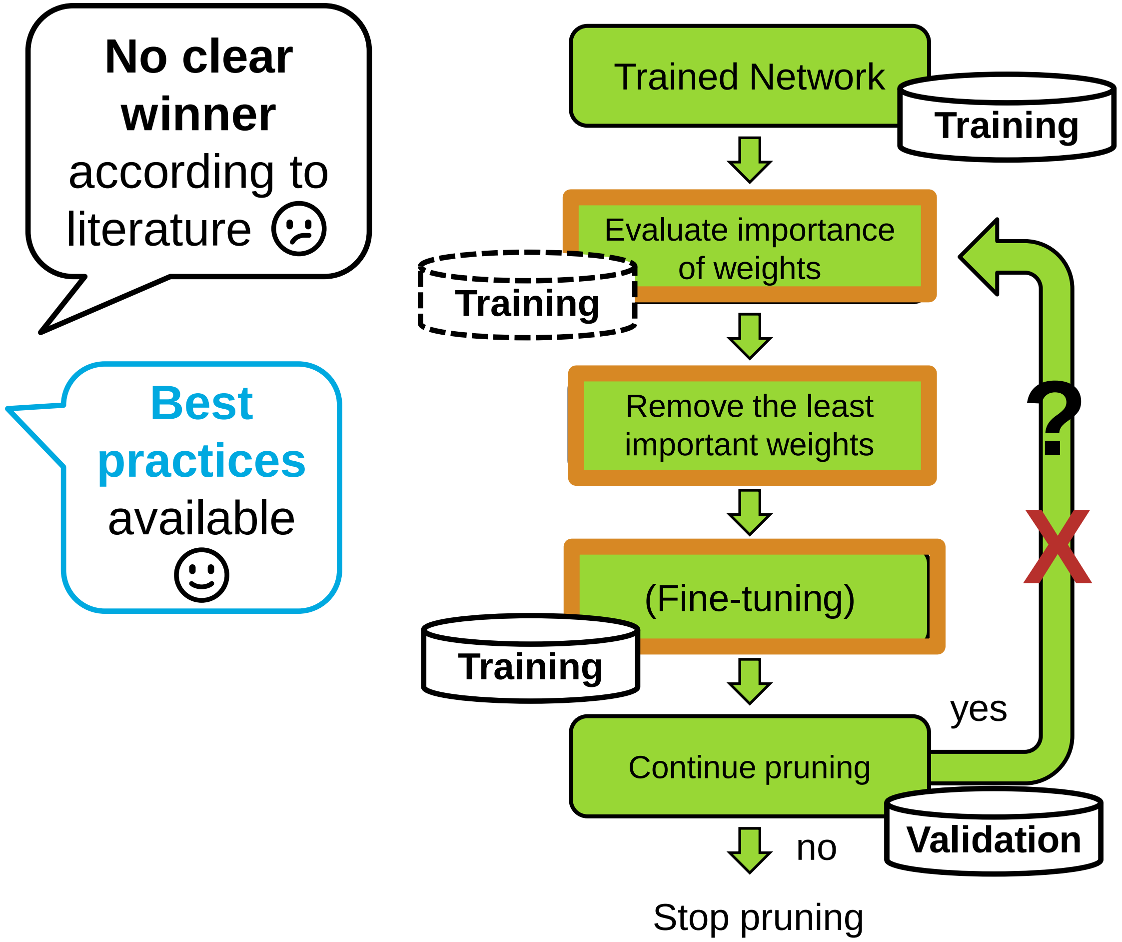
```
taylorPrunableNetwork(net)
```

3 Deep Network Quantizer



Understand Hardware Constraints → **Select Model** → **Simplify Model: Prune Deep Neural Network** → **Quantize Model Parameters** → **Deploy & Integrate**

Pruning algorithms follow a common process but can have lots of small variations

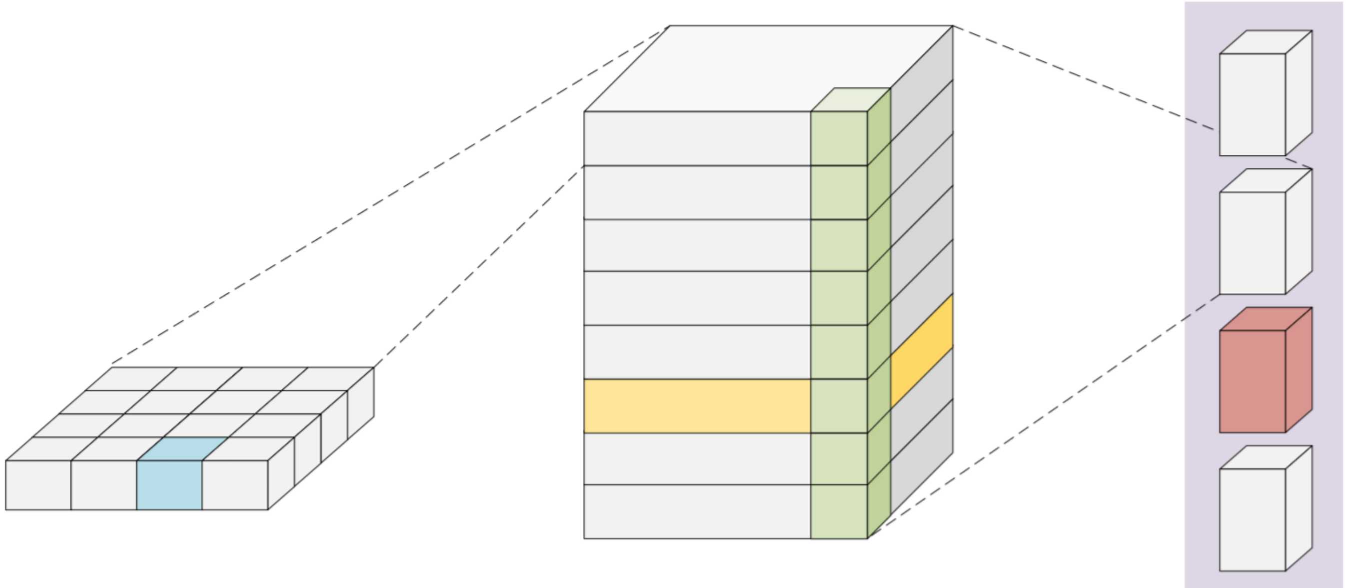


- Scoring
  - Absolute weight value
  - Gradient-based metric
  - Activations-based metric
- Pruning criteria
  - local (uniformly X% per layer)
  - global (X% across whole network)
- Fine-tuning, yes/no
- Scheduling
  - One-shot
  - Iteratively

Execution time vs. **effect** vs. **data required**

"Pruning Convolutional Neural Networks for Resource Efficient Inference"
Molchanov et al. 2017. https://arxiv.org/abs/1611.06440

Which parts of the network can be pruned?

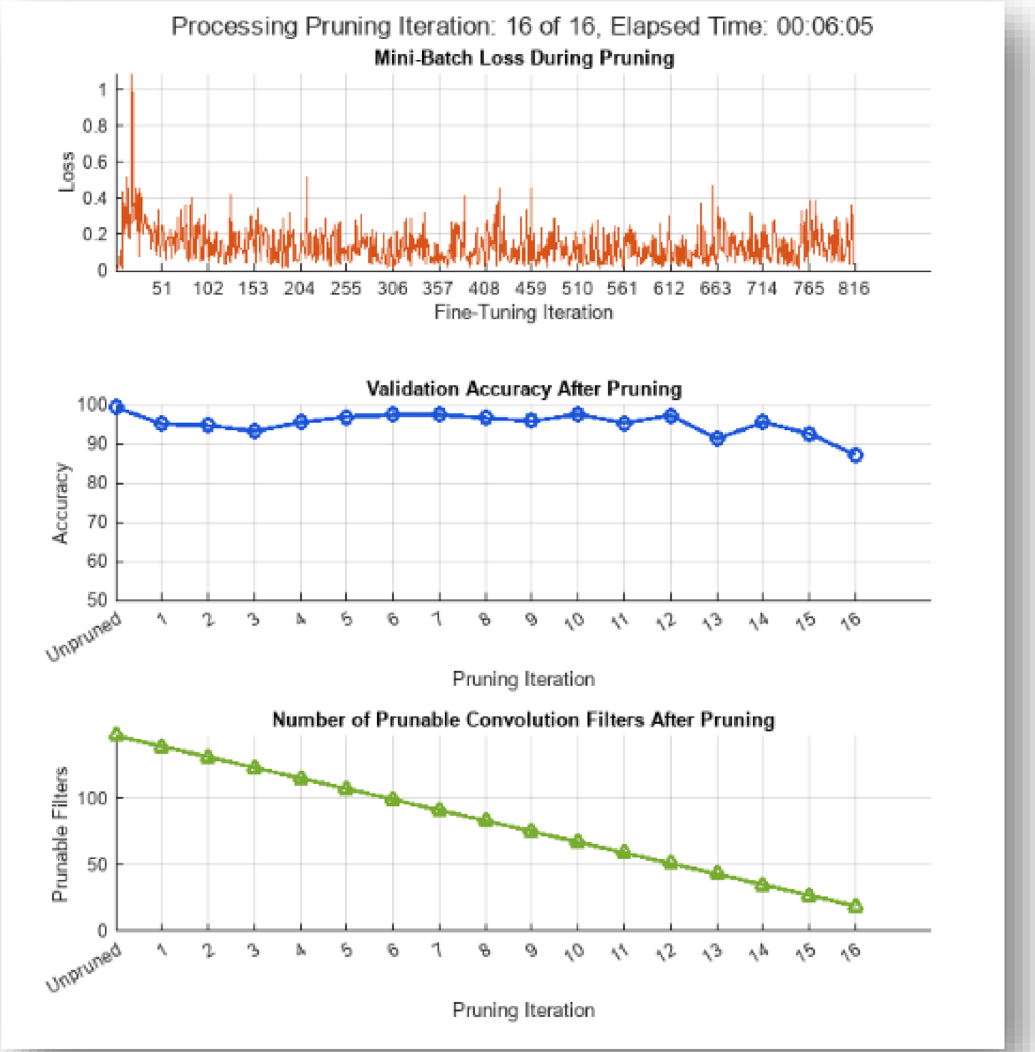


element-wise	channel-wise shape-wise filter-wise	layer-wise
individual connections	e.g. conv. filters,	
introduces sparsity	neurons in FC layer	
UNSTRUCTURED	**STRUCTURED**	

Pruning and Quantization for Deep Neural Network Acceleration: A Survey. Liang et al. https://arxiv.org/pdf/2101.09671.pdf

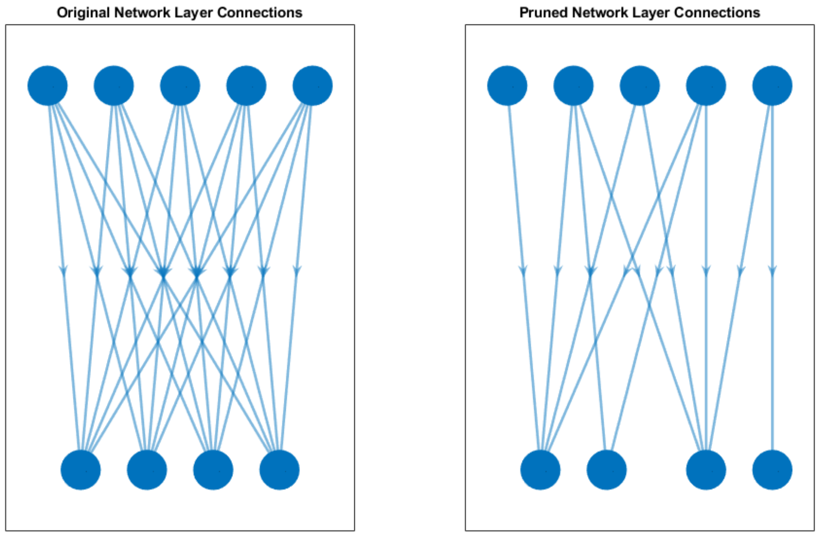
Taylor Pruning uses gradient score and eliminates number of filters in convolutional layers

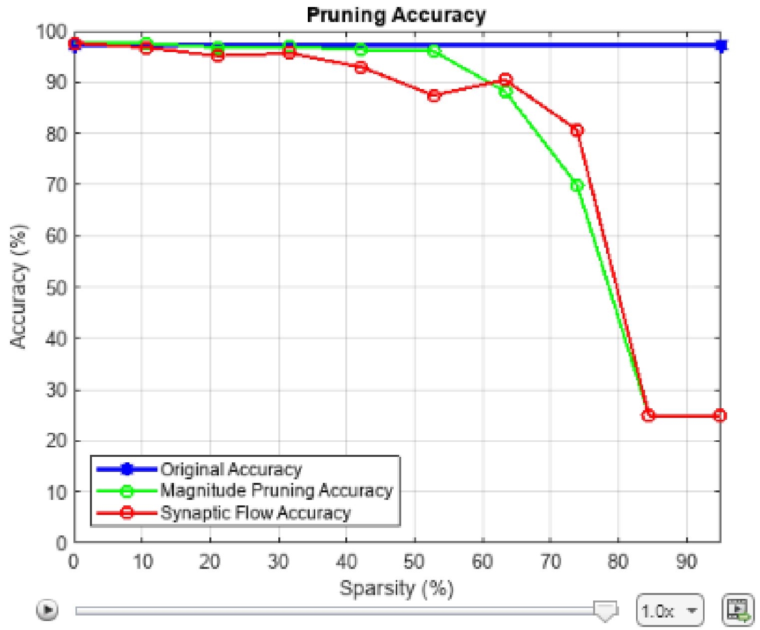
- Gradient-based method to estimate filter “importance” using first-order Taylor expansion
- Prune less important filters to reduce model size while maintaining predictive power
- **STRUCTURED** approach
- Fine-tune pruned model with data



Parameter Pruning zeros out lower score connections

- Calculate numerical scores to rank the connections in the network
- Iteratively remove less “important” connections
- **UNSTRUCTURED** approach





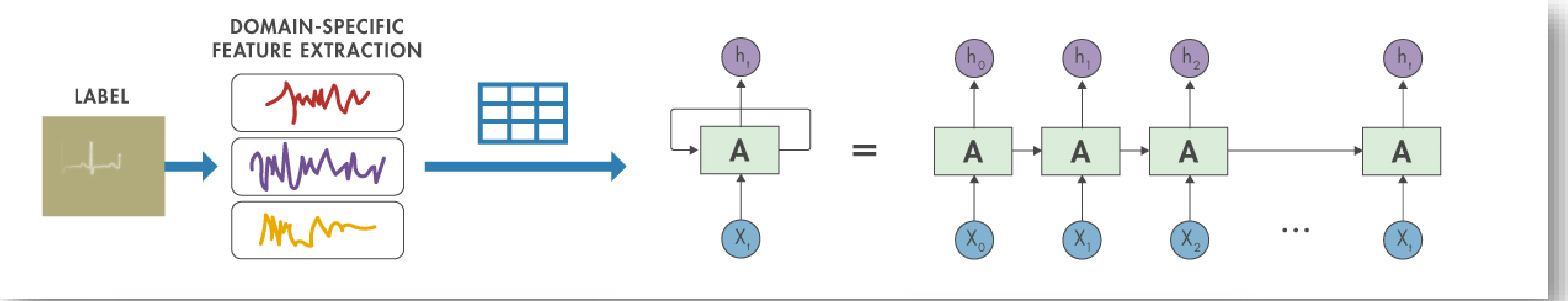
Examples are:

Magnitude Score

SynFlow Score: synaptic flow scores

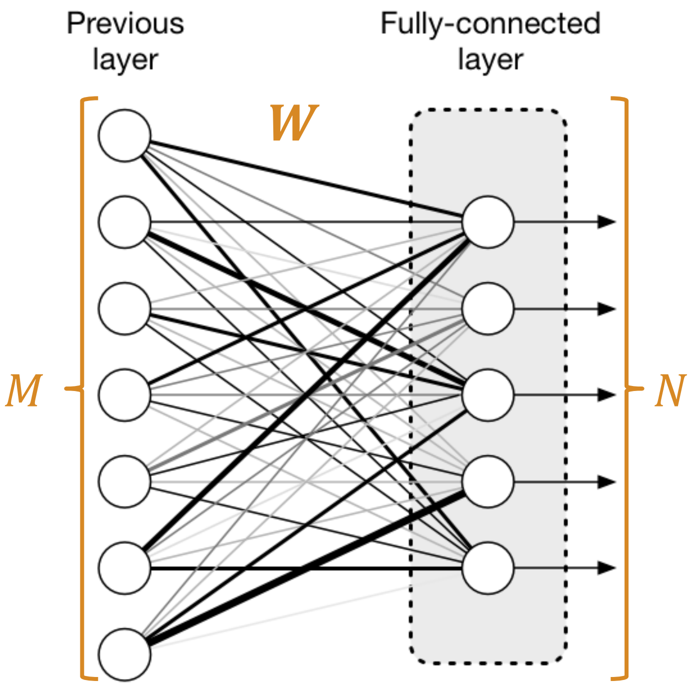
Structural pruning reduces problem dimensions via projection into subspace

For example, a LSTM Networks



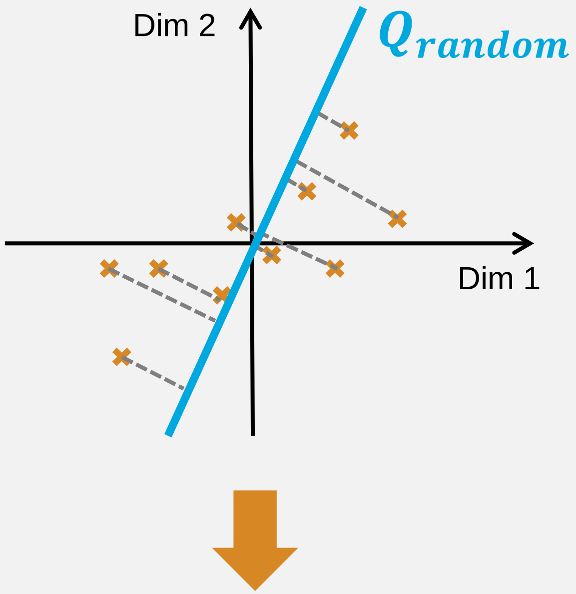
Two-step approach: Projection compression with neuron PCA

High-dimensional space of input and output neurons is underutilized



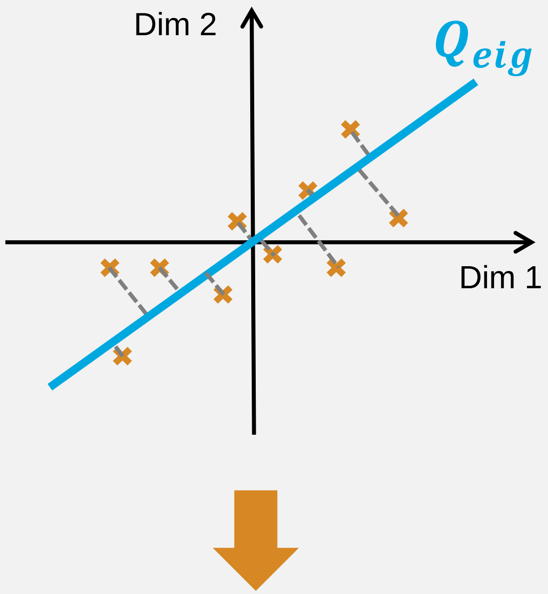
W is an N-by-M matrix

Dimensionality reduction via projection into subspace



projected layer

Minimize projection error via principal component analysis (PCA) of neurons



neuronPCA

Structural compression of LSTM layers to reduce model size

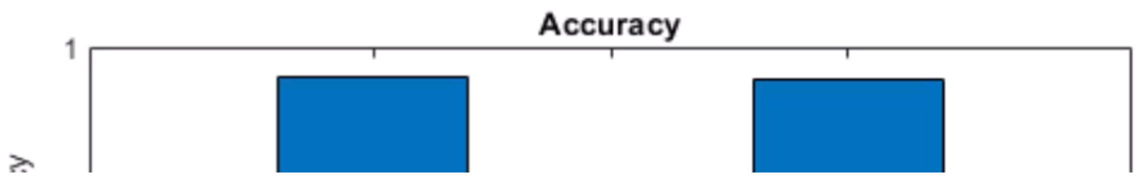
Compress Neural Network Using Projection

This example shows how to compress a neural network using projection and principal component analysis.

To compress a deep learning network, you can use *projected layers*. The layer introduces learnable projector matrices Q, replaces multiplications of the form Wx, where W is a learnable matrix, with the multiplication $WQQ^{\top}x$, and stores Q and $W' = WQ$ instead of storing W. Projecting x into a lower dimensional space using Q typically requires less memory to store the learnable parameters and can have similarly strong prediction accuracy. A projected deep neural network can also exhibit faster forward passes when run on the CPU or deployed to embedded hardware using library-free C or C++ code generation.

The `compressNetworkUsingProjection` function compresses a network by projecting layers into smaller parameter subspaces. For optimal initialization of the projected network, the function projects the learnable parameters of projectable layers into a subspace that maintains the highest variance in neuron activations. After you compress a neural network using projection, you can then fine-tune the network to increase the accuracy.

This chart shows the effect of projection and fine tuning on a trained network. In this case, the projected network has significantly fewer learnable parameters at the cost of classification accuracy. The fine-tuned projected network yields similar classification accuracy to the original network.



Results from compression of LSTM layers to reduce model size

Compress the network.

```
netProjected = compressNetworkUsingProjection(net,mbq);
```

```
Compressed network has 82.4% fewer learnable parameters.
Projected layers explain on average 96.6% of layer activation variance.
```

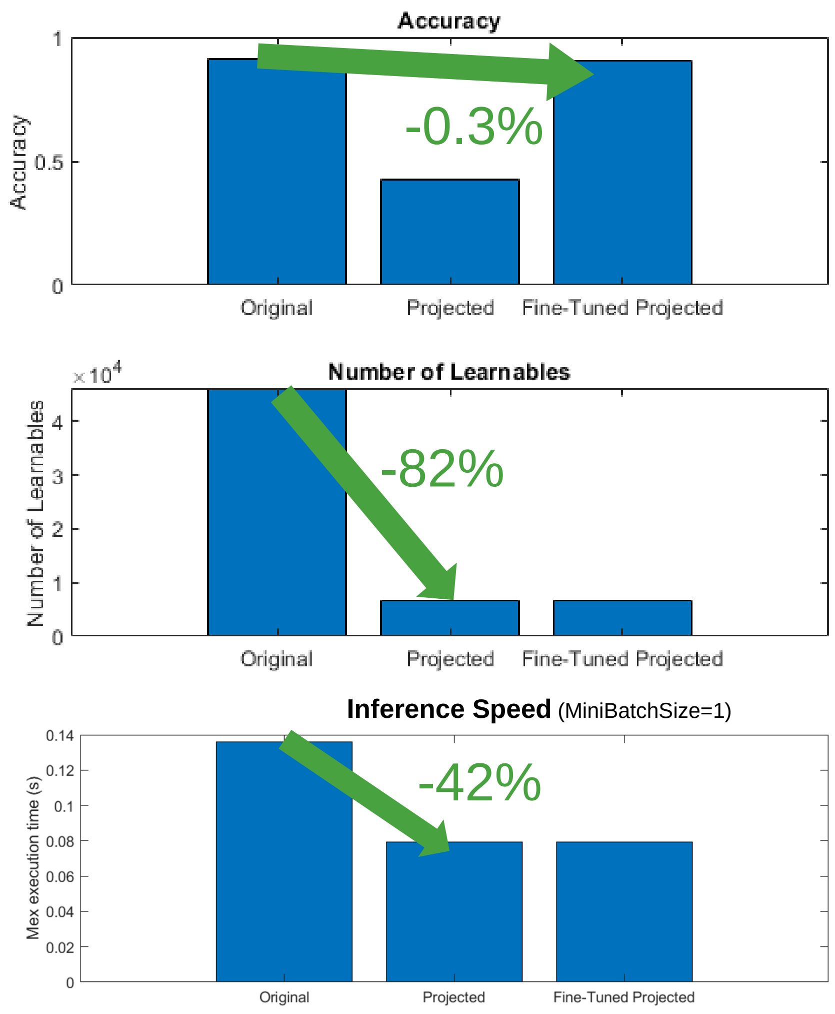
Optionally, precompute neuronPCA analysis for efficient experimentation:

```
npca = neuronPCA(netOriginal,mbqTrain,VerbosityLevel="steps");
```

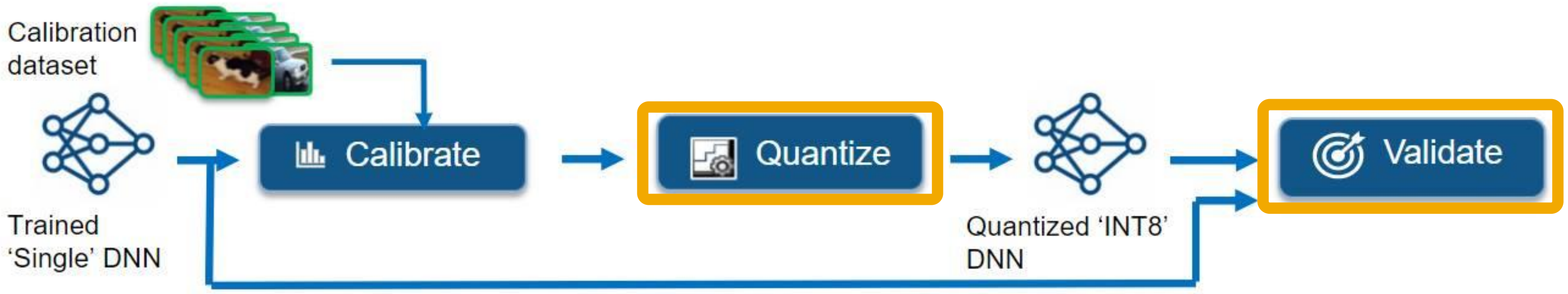
```
Computing layer activations and covariance matrices...
Computing eigenvalues and eigenvectors...
neuronPCA analyzed 1 layers: "lstm"
```

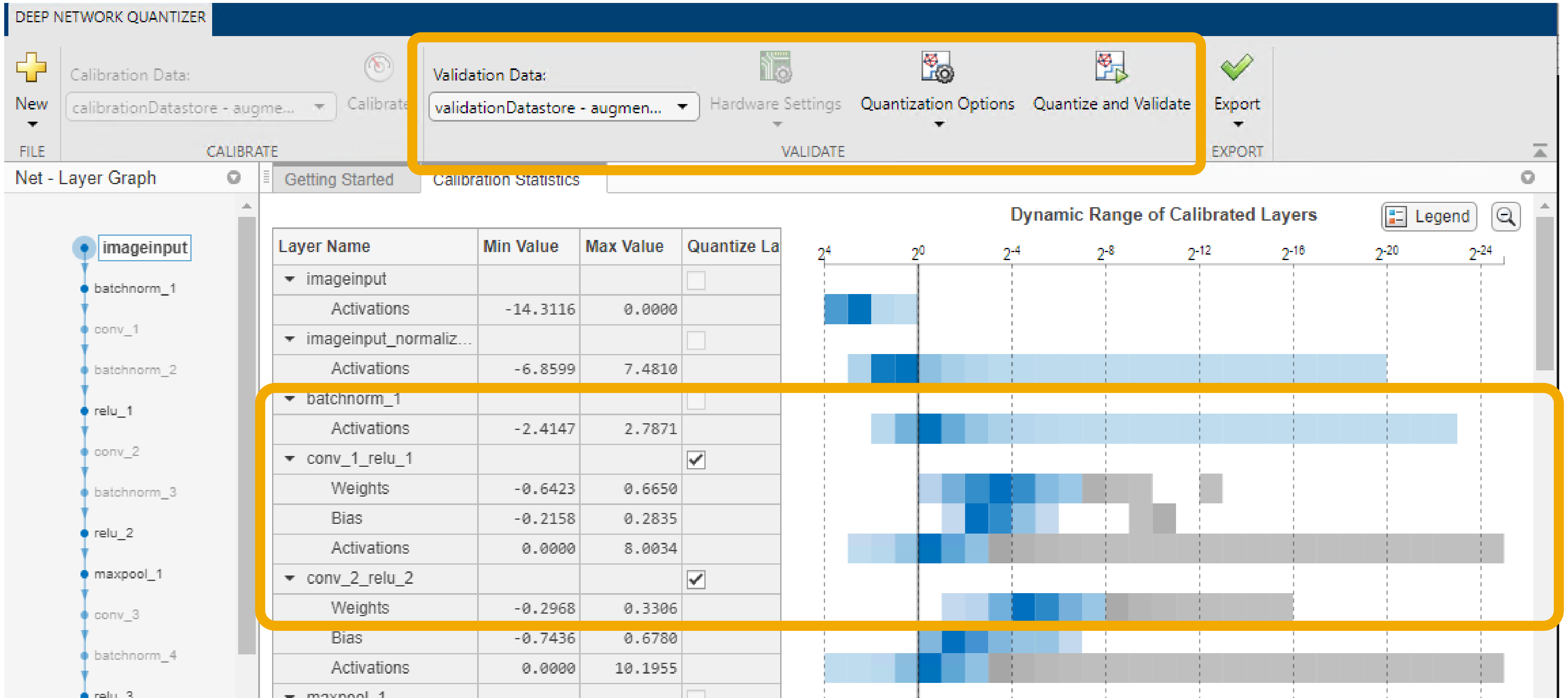
Doc Example: [Compress Neural Network Using Projection]

(Seq-2-One Classification on Japanese Vowels data set)

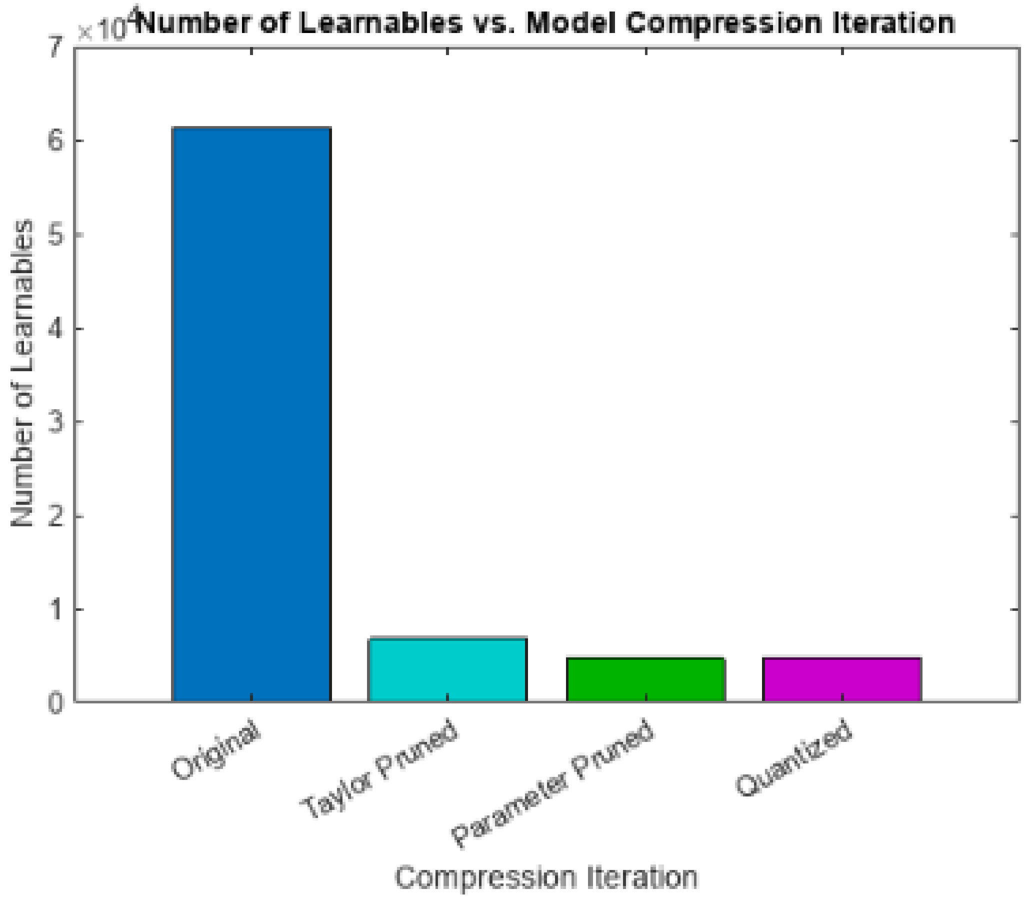


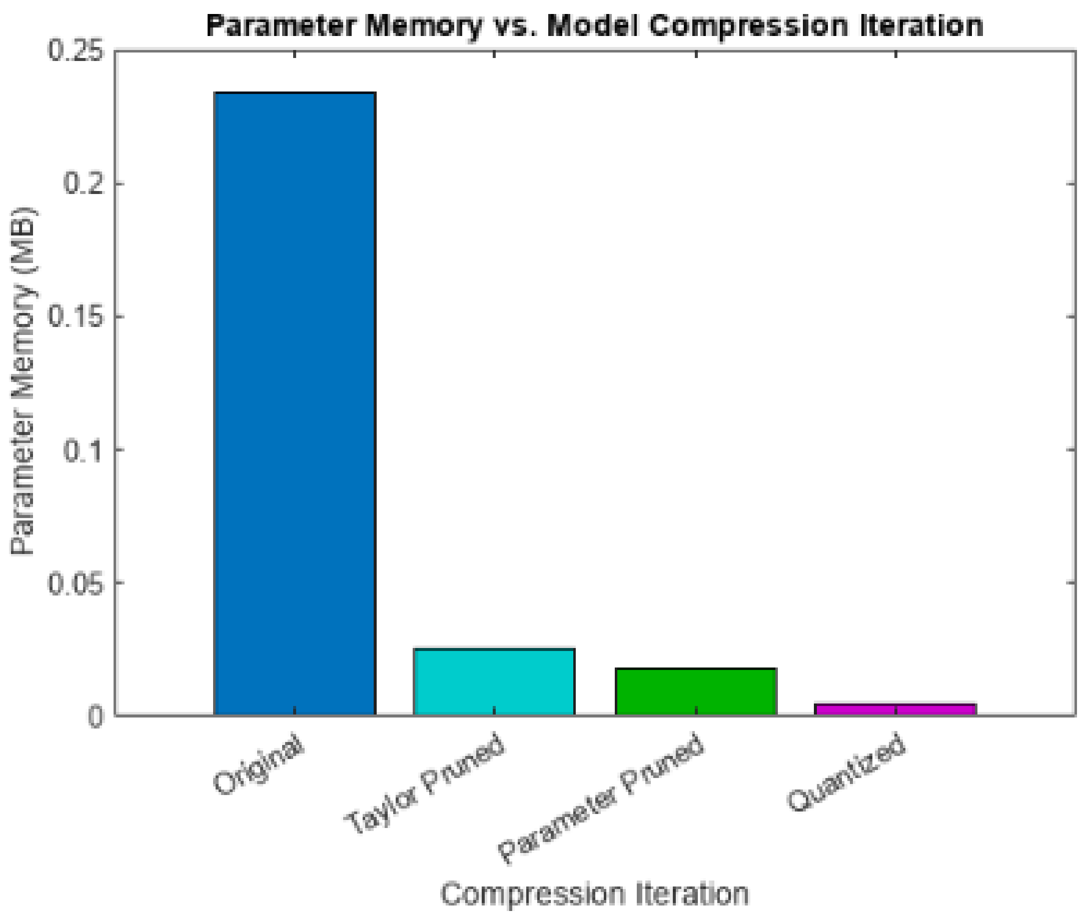
Deep Network Quantizer transparently applies quantization





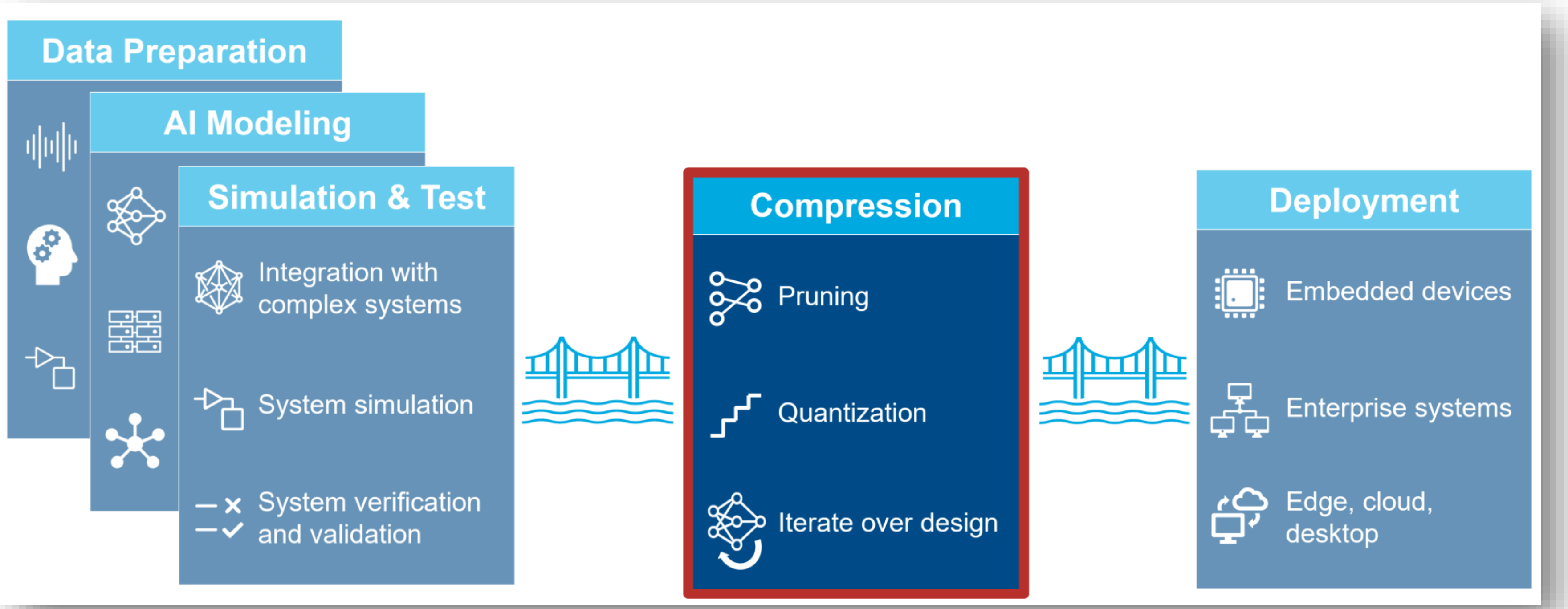
Reduce learnable parameters





Conclusion

Compress AI-based reduced-order engine model for deployment



- **Integrate trained AI model into Simulink** for system-level simulation together with first-principles components
- **Generate C code and perform HIL tests**
- **Deploy compressed TinyML model** to embedded target

Thank you!

Want to learn more?

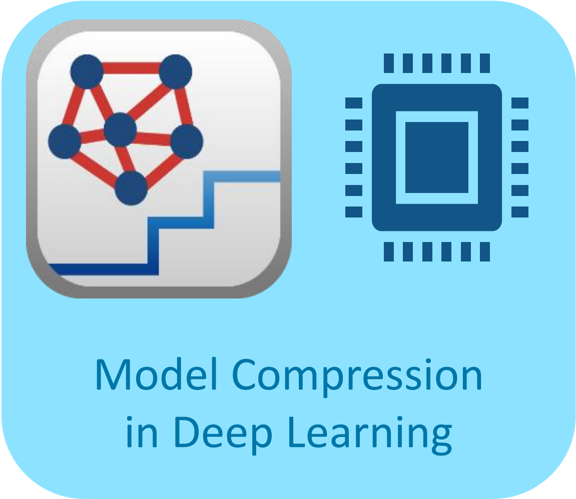
Quantization
https://www.mathworks.com/discovery/quantization.html

MATLAB for Deep Learning
https://www.mathworks.com/solutions/deep-learning.html

More applications and capabilities
https://www.mathworks.com/solutions.html

Brenda Zhuang
@Brenda-Zhuang

Greg Coppenrath
@GregCoppenrath



Copyright Notice

www.tinyml.org

Copyright Notice

www.tinyml.org