
ML ↦ P
Mapping Machine Learning to Physics


ML
P

Mapping Machine Learning to Physics (ML2P)

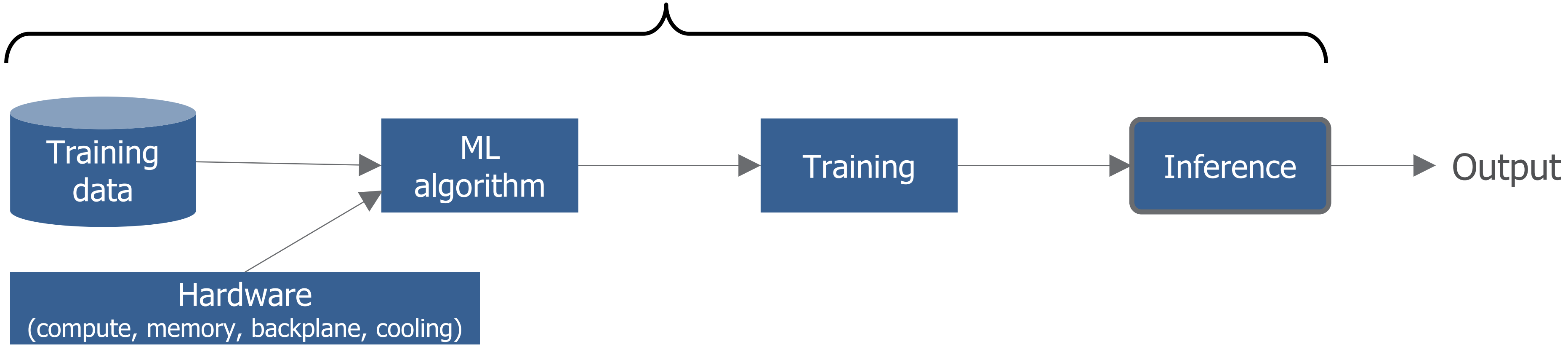
Small Program (6.1, 24 months)

Bernard McShea
Program Manager, Information Innovation Office

Preserve, Tune, Optimize

August 2025

Redefine power consumption to be a 'first-class citizen' throughout the machine learning life cycle



Definition: Energy-aware ML is optimized for efficiency with respect to power for the lifespan of the ML (e.g., dataset selection to model inference) while retaining performance (e.g., A/P/R/F1 scores)

ML2P enables granular power considerations at any point in the model's life cycle. For example, some DoD applications may be strictly focused on inference power.

A – Accuracy / P – Precision / R – Recall / F1 score – harmonic mean of precision and recall
ML – Machine Learning

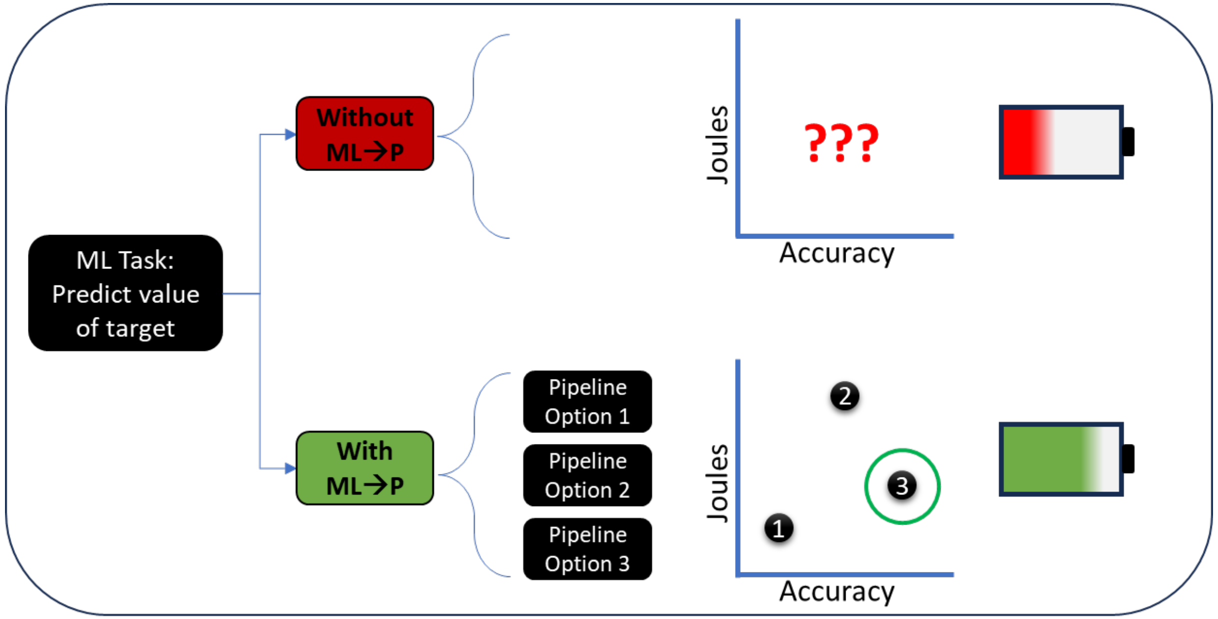
H1: Mapping Machine Learning to Physics (ML2P)

Hypothesis: Power consumption and performance of ML models on existing hardware can be improved by preserving local energy semantics and tuning energy-performance objective function enabling energy aware ML

Machine learning at the edge* in a resource-constrained battlefield

Today:
- Adversary adapts, diminishing relevance of pretrained models
- High operational tempo forces operators to do more with less
- Jammed communications limit remote model update/resupply

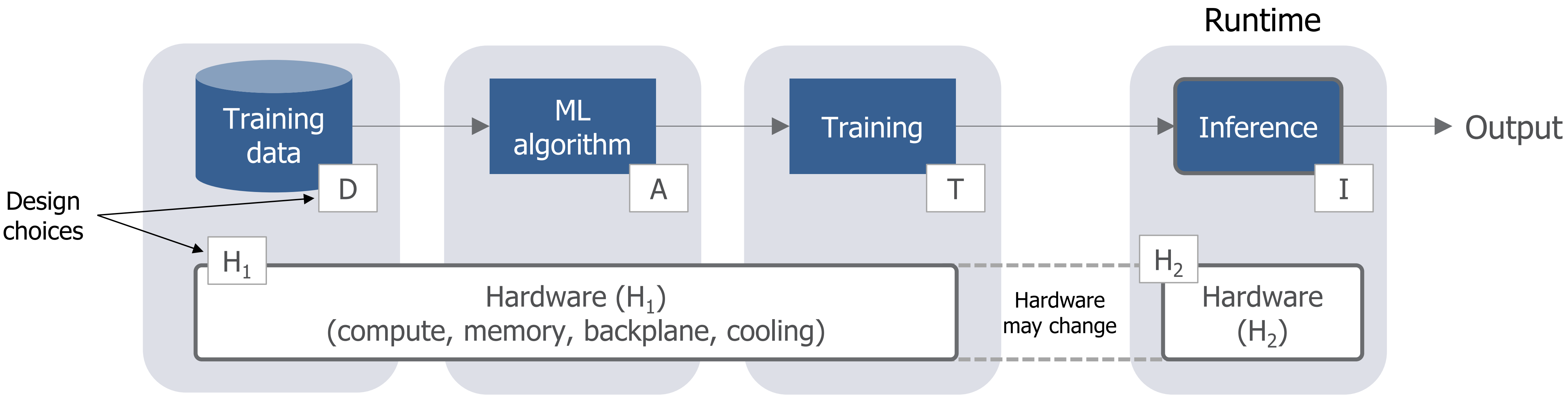
What is needed:
- Extend warfighters existing capability (+range)
- Provide theoretical basis for creating energy aware ML software and hardware



ML2P will map ML performance to Joules of energy associating ML to physics

ML – Machine Learning
*Edge – computing that is located close to where the data is generated

H2: Today, all previous hardware and software design choices are discarded after each step in the pipeline



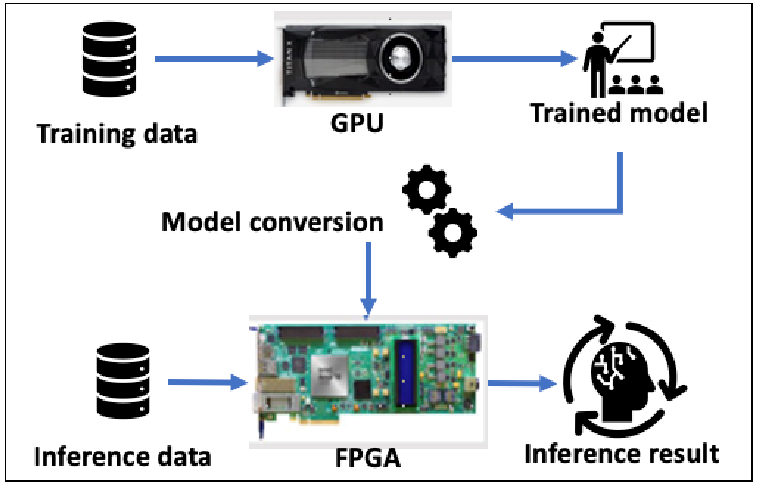
- Optimization is focused only on model performance
- Optimization is isolated in each step
- Information needed for optimization and hardware design is discarded

D – data
H – hardware T – training
A – algorithm I – inference

H2: Today, we are missing a principled way to construct software to fully utilize available hardware

We are missing a principled way to inform switching between manufacturers

Emerging research: Liu et al., [1] demonstrated a ***7x decrease in inference time*** by converting a trained GPU model while retaining accuracy



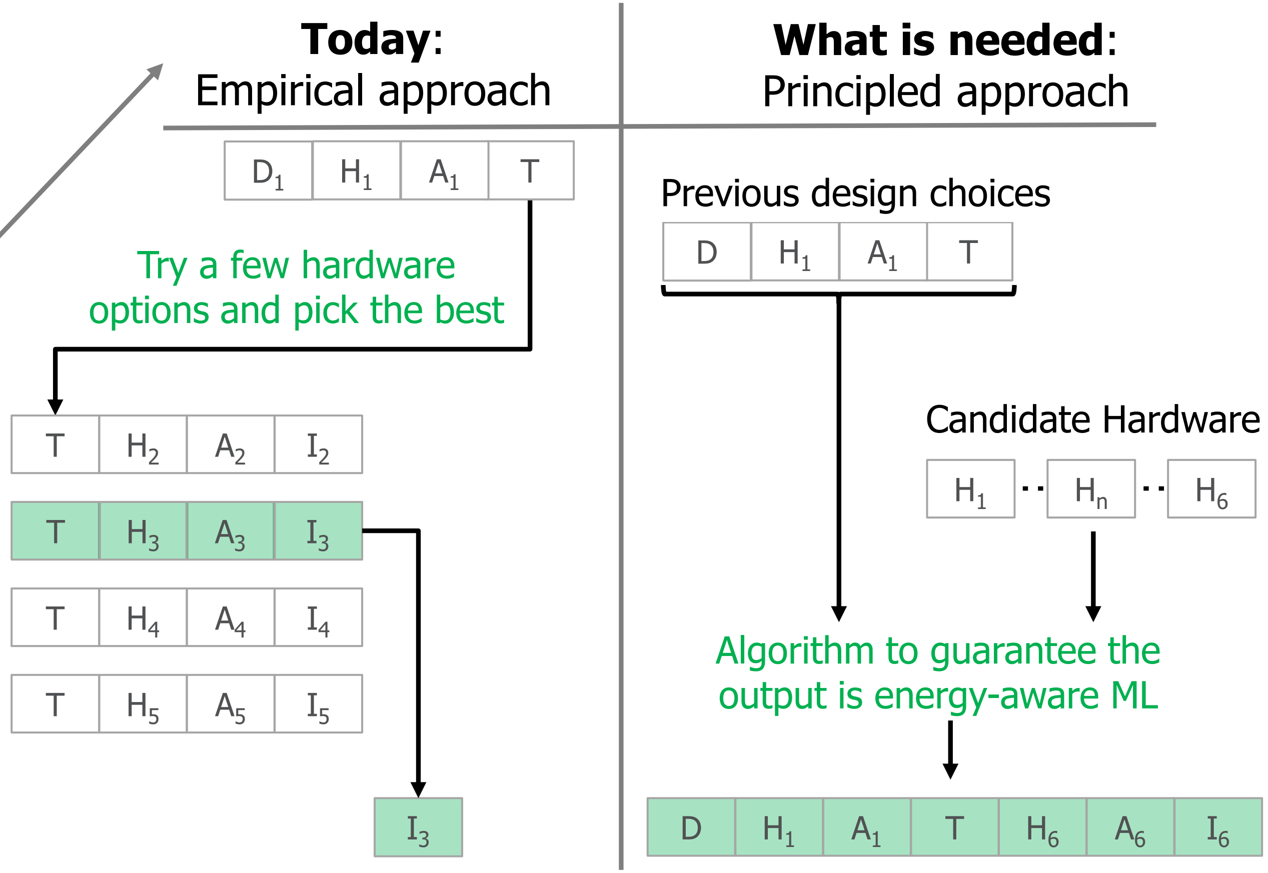
GPU: NVIDIA TITAN

Empirical

FPGA: Intel Arria 10

Hardware	**CPU** E5-1620	**GPU** TitanXp	**FGPA** Arria10
Average Inferencing Time (microseconds)	**14.786 μs**	**9.565 μs**	**1.35 μs**

[1] Liu, Xu, et al. (2020) Journal of Ambient Intelligence and Humanized Computing.



Successful **empirical** findings are not sufficient to inform **principled** construction as previous design information discarded is not fully rediscovered

GPU – Graphical Processing Unit
FPGA – Field-Programmable Gate Array
ML – Machine Learning
μs – microseconds

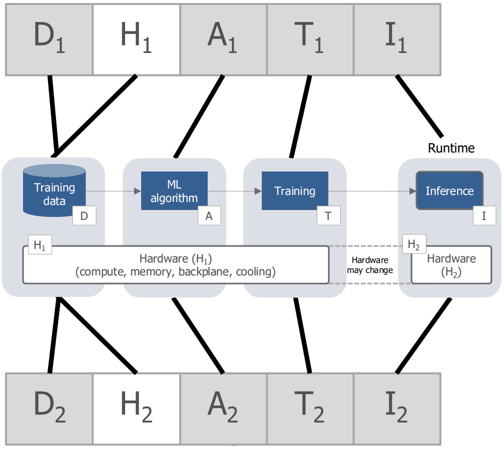
D – data
H – hardware
A – algorithm
T – training
I – inference

H2: Today, we are missing a principled way to compare multiple HW ML designs

Industry discards critical elements of the machine learning process used to generate the TOPS/Watt score **invalidating comparison**

"Lies, Damn Lies, and TOPS/Watt" [1]

GPU
NVIDIA TITAN



FPGA
Intel Arria 10

Key

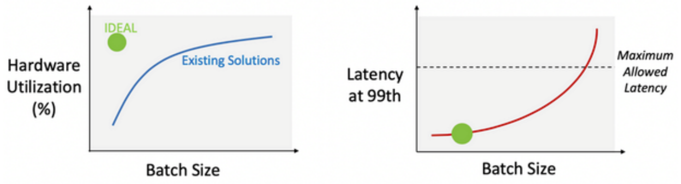
- Discarded previous design choices
- TOPS/Watt

Other factors discarded:
temperature, nominal voltage

[1] Geoff Tate. (2019) https://semiengineering.com/lies-damn-lies-and-tops-watt/

Gaming of TOPS/Watt metrics

- **Selective Operation Counting**: Vendors may count each Multiply-Accumulate operation (MAC) as two operations—one for the multiplication and one for the addition.
- **Unspecified and Optimistic Operating Conditions**
- **Overstated Utilization Rates**: The reported number of operations often assumes 100% utilization of all processing units, which is rarely the case.
- **Batch Size Manipulation**



Batching improves HW utilization but increases latency

TOPS/Watt: count of trillions of arithmetic operations a processor can perform per second for each watt of power used.

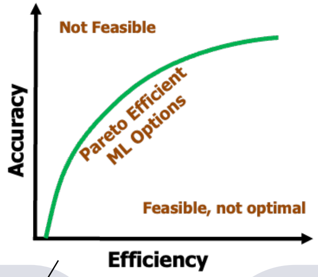
TOPS – Tera-Operations/Second
GPU – Graphical Processing Unit
FPGA – Field-Programmable Gate Array
ML – Machine Learning
HW – Hardware

D – data
H – hardware
A – algorithm

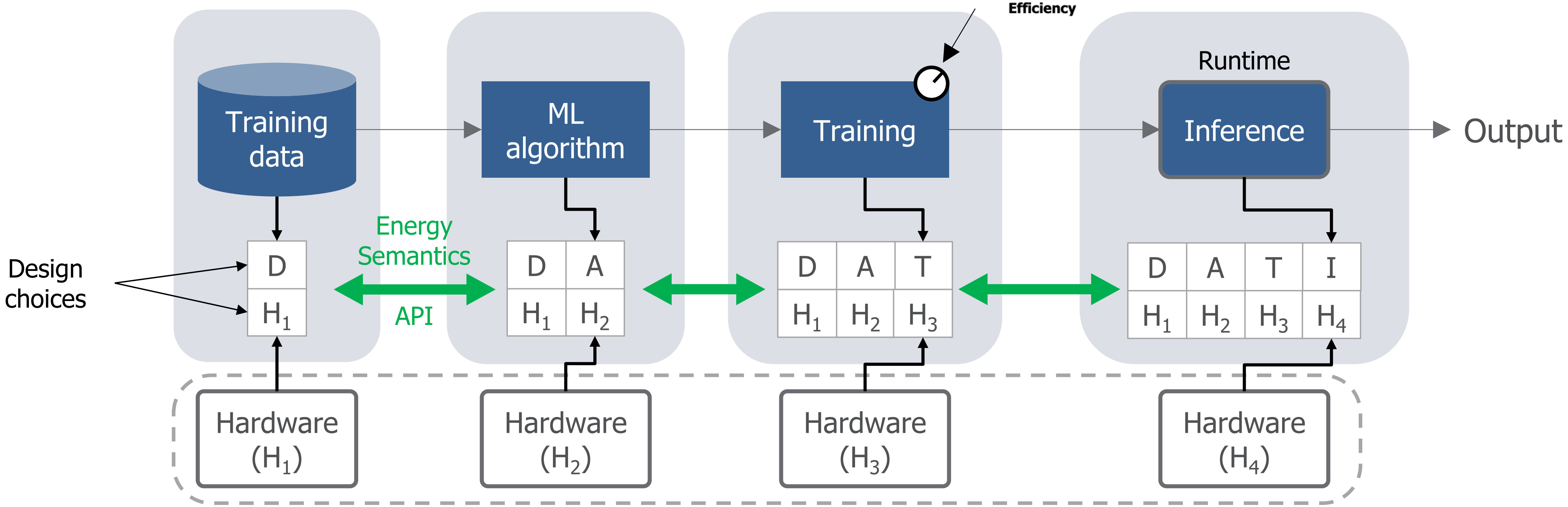
T – training
I – inference

H3: ML2P - preserve, tune, optimize

Preserve local energy semantics
Tune training energy performance objective function



Tunable objective function based on energy use



- Optimization can be tuned between efficiency and performance
- Energy semantics can enable energy-aware ML
- Hardware design can be informed by energy semantics

D – data
H – hardware T – training
A – algorithm I – inference ML – Machine Learning

H3: ML2P - preserve, tune, optimize

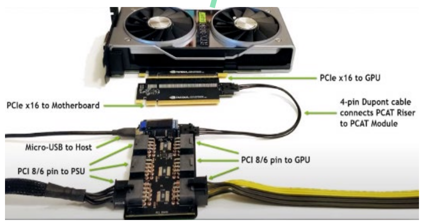
Preserve local energy semantics → **Tunable** training energy performance objective function → **Optimizing** for energy aware ML

Discover and **preserve** local energy semantics

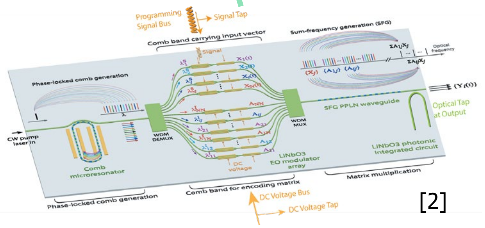
D → T → I

Take accurate granular measurement power preserving semantic information enabling multi-objective optimization with respect to power and A/P/R/F1

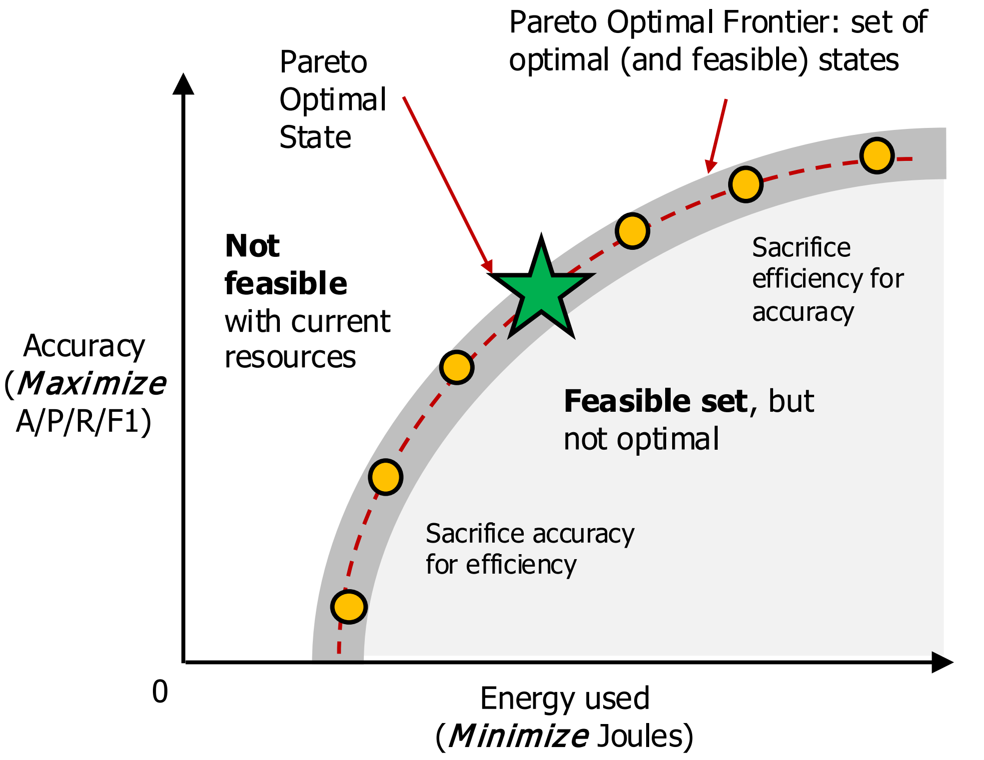
A_1	. . .	A_j	. . .	A_n
H_1		H_j		H_n



[1]


[2]

Develop **tunable** objective functions

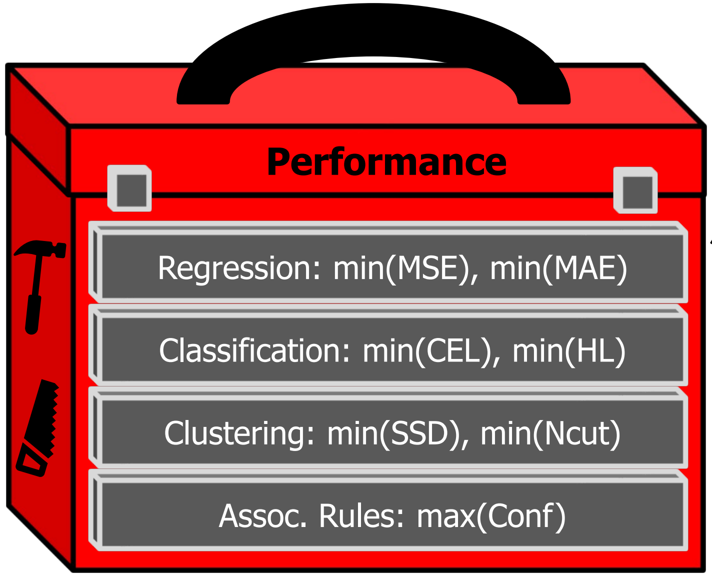


ML2P will construct energy-aware ML with **optimized** power (J) and performance (A/P/R/F1) for a given task (e.g., clustering, classification) and candidate hardware for a point in the model's life cycle.

A – Accuracy / P – Precision / R – Recall / F1 score – harmonic mean of precision and recall
ML – Machine Learning
J – Joule

ML-relevant objective functions

Today objective functions maximize performance (A/P/R/F1)



Let B(x) be one or more existing objective function covering the objective space

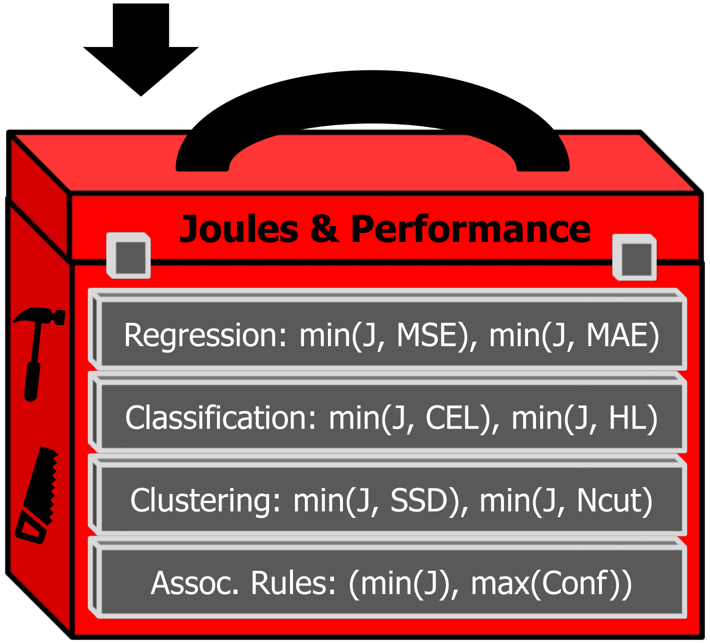
(e.g., Mean Squared Error, Huber Loss, Cross Entropy Loss, KL Divergence)

$$\min_{y \in (J, B(x))} B'(y)$$

Let $B'(y)$ be a multi-objective function that minimizes over power (J) and loss.

ML2P objective functions minimize power (J) and maximize performance (A/P/R/F1)

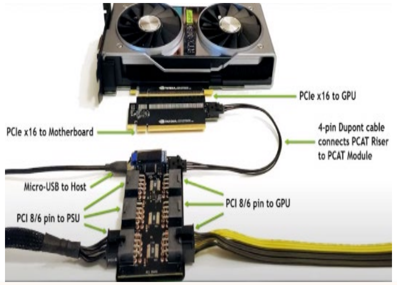
(J)



Note: $B'(y)$ functions which covers the maximum space of the total ML relevant objective function space are favored.

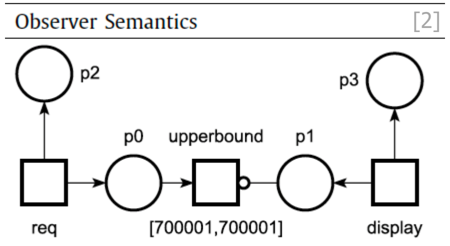
H6: Tasks and deliverables

ML2P will produce code, algorithms and documentation describing power measurement and energy semantics for machine learning



[1]

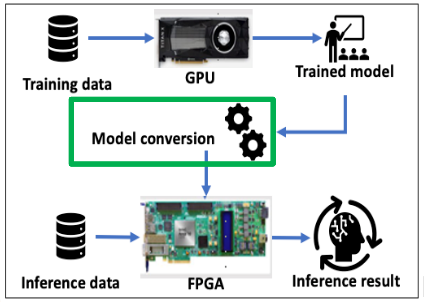
Develop Forensic hardware instrumentation (direct power measurements)



MMC Formula

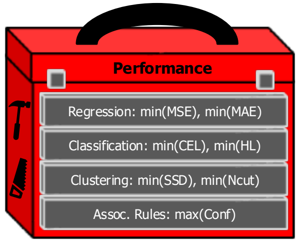
$$\text{mmc}(\neg p2 \wedge p3) == 0$$
$$\text{mmc}(\neg p0 \wedge p1) == 0$$
$$\text{mmc}(\neg(\neg p2 \wedge p3) \wedge \neg(\neg p0 \wedge p1)) \neq 0$$

Create machine readable ES-ML

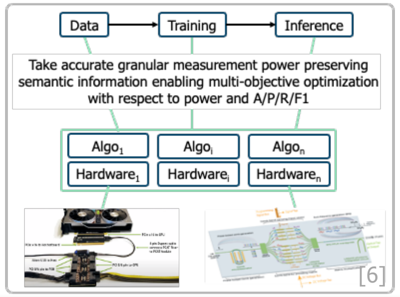


[3]

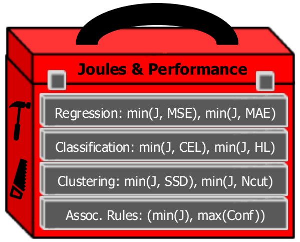
Convert trained models to different hardware types

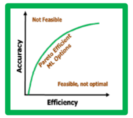


Select objective functions

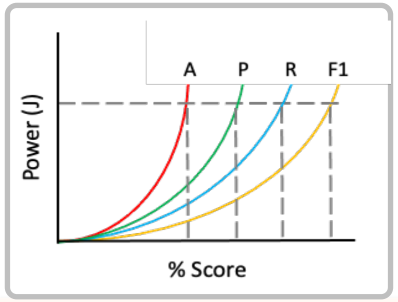


Explore multi-objective optimization via ES-ML





Explore multi-objective optimization (Joules, A/P/R/F1)



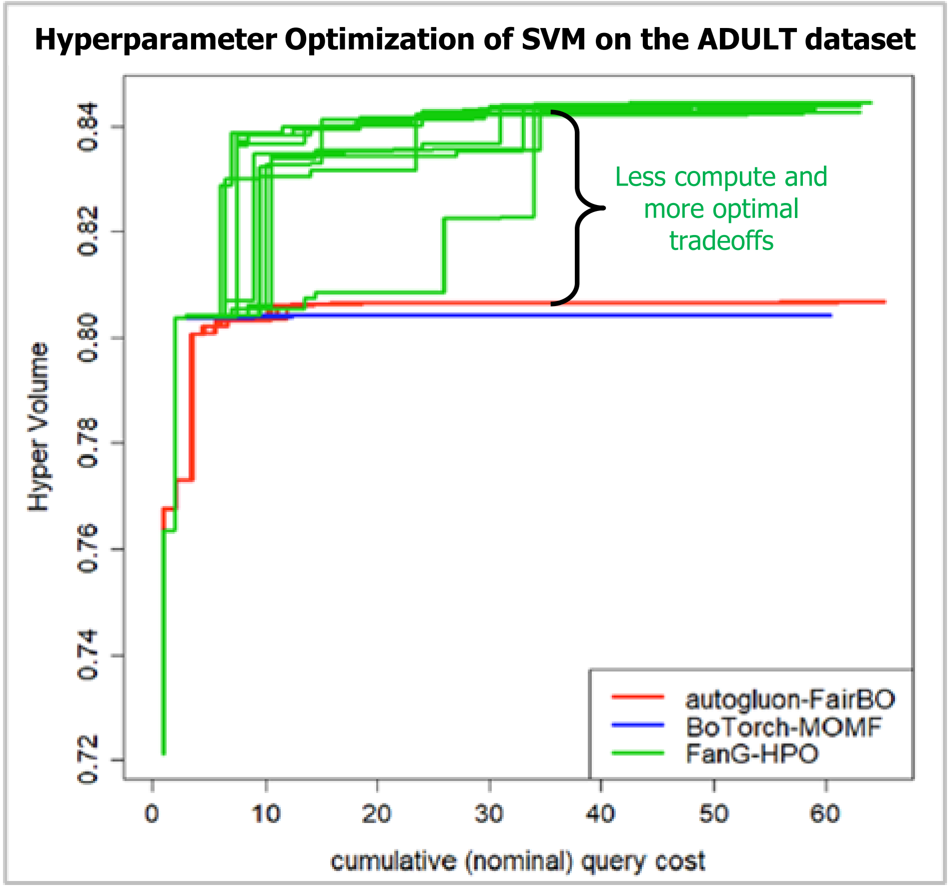
Discover algorithms for constructing energy-aware ML covering the lifespan of a model

FY26 **FY27** **FY28**

A – Accuracy / P – Precision / R – Recall / F1 score – harmonic mean of precision and recall
ES-ML – Energy Semantics for Machine Learning

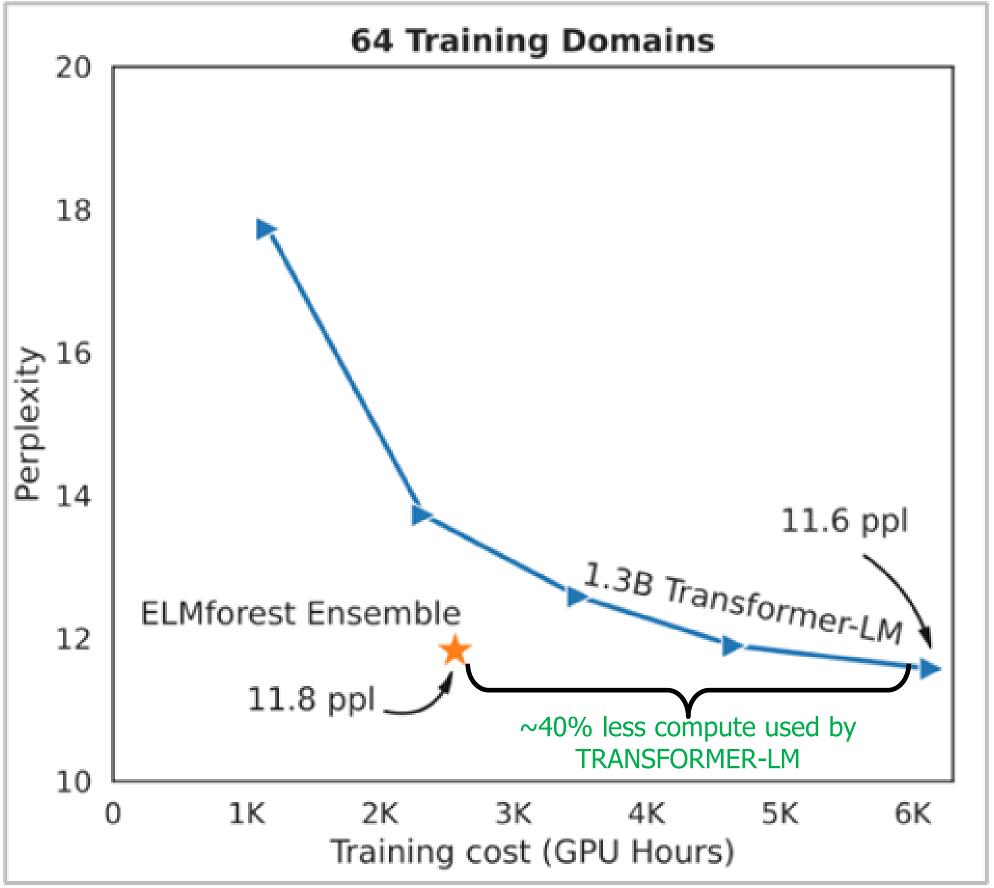
H3: Empirical evidence demonstrates ML efficiency improves via tuning objective function and optimizing model selection

Multi-objective hyperparameter tuning demonstrates performance gain. [1] Candelieri et al. (2024)



FanG-HPO – Fair and Green Hyperparameter Optimization
Hyper Volume - in multi-objective optimization, quality in balancing multiple objectives

Empirical data demonstrates selection of dataset and algorithms can increase performance. [2] Smith et al. (2022)

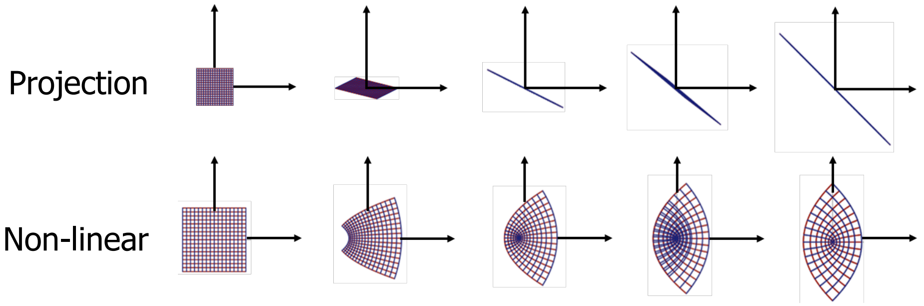


Ppl – Perplexity: degree to which a language model predicts the next word correctly

SVM – Support Vector Machine
ADULT - Binary classification dataset from UCI Repository

H8: Lower bound of joules of energy used for ML model lifecycle, computed using MLB-Linpack metric

LB ML Training



A single matrix operation to transform a pre-trained matrix in into a trained matrix is the lower bound training a model

LB ML Inference

$$\begin{pmatrix} a_1 & a_2 \\ \vdots & \vdots \\ a_{n-1} & a_n \end{pmatrix} \boxed{\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}} = \boxed{(i)}$$

A single row computation for a matrix of size n x 2 for inference is the lower bound model inference

MLB-Linpack metric

$$\frac{\|Ax - b\|_\infty}{(\|A\|_\infty \|x\|_\infty + \|b\|_\infty) n\epsilon} \leq O(1)$$

[Eq. 1]

Let A be a matrix, b be a vector with equal cardinality to A, ϵ is the HW's precision, n is the size of the problem*, $\|\cdot\|_\infty$ is a matrix norm, and $O(1)$ corresponds to Big-O notation.

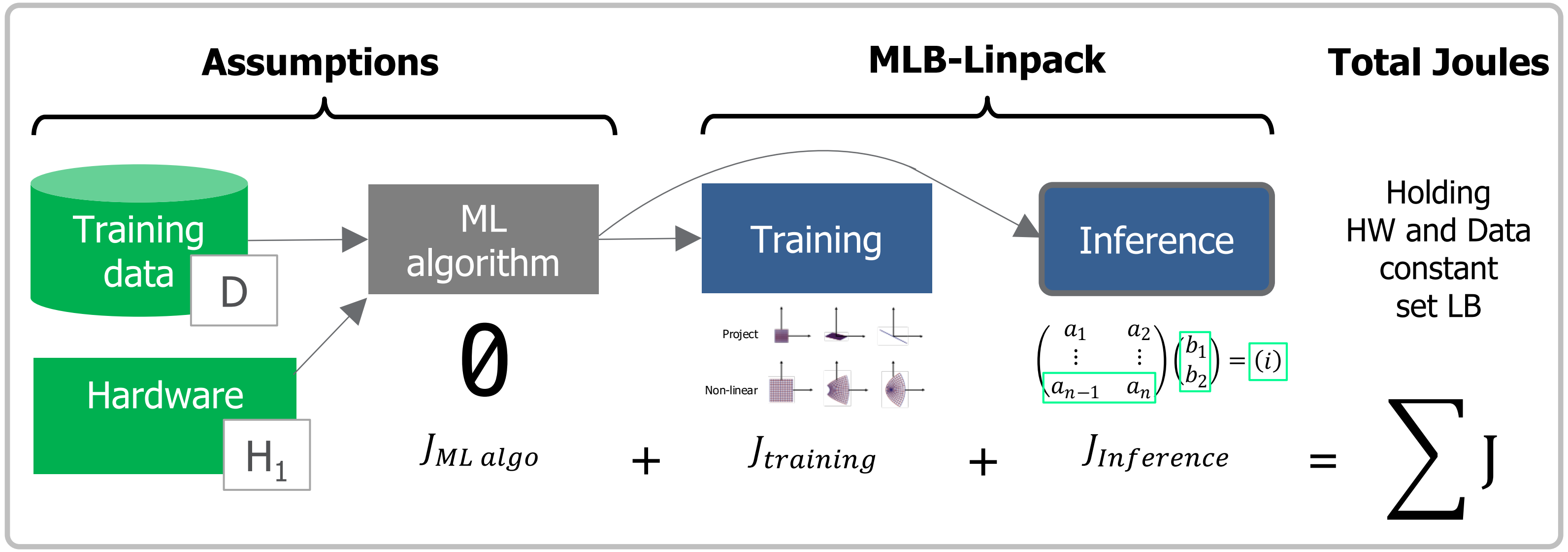
Test Procedure

- Assumptions; performer defines set of HW and dataset properties (e.g., matrix size)
- Compute MLB-Linpack for LB for training and inference with the constraints from the performer, where the size of A fits the HW memory.

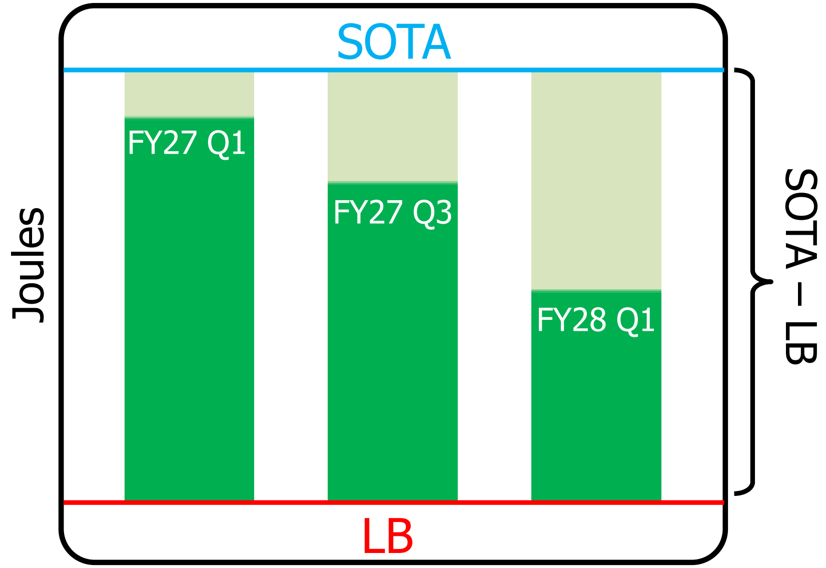
Define the lower bound to baseline the power budget

ML – Machine Learning
HW – Hardware
LB –Lower Bound
*n size must be large enough to fill available memory, but not induce swapping

H8: Metrics



SOTA vs. ML2P vs. LB



The possible improvement power budget is the SOTA - LB

Metrics	FY27 Q1	FY27 Q3	FY28 Q1
Accuracy of predicted energy usage for ML2P selected datasets	60%	80%	99%
Delta Joules holding scores (A/P/R/F1) constant SOTA vs. ML2P vs. LB for training and inference	< 90% (SOTA – LB)	< 75% (SOTA - LB)	< 50% (SOTA - LB)

A – Accuracy / P – Precision / R – Recall / F1 score – harmonic mean of precision and recall
ML – Machine Learning, J - Joule
LB – Lower Bound, SOTA – State of the Art

H7: Schedule

Phase 1 (12 months)				Phase 2 (12 months)			
FY26			FY27				FY28
Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1
Experiment Setup		Experimentation					

Experiment Setup

- Develop Forensic hardware instrumentation (direct power measurements)
- Select objective functions
- Create machine readable ES-ML
- Develop algorithms to convert trained models to different hardware types

Experimentation

- Explore multi-objective optimization (Joules, A/P/R/F1)
- Explore interactions of optimization via energy semantics of ML
- Discover algorithms for constructing energy-aware ML for any point in a model's life cycle
- Evaluation Team: Design energy usage experiments

Kickoff

< 90% (SOTA – LB), 60%

< 75% (SOTA – LB), 80%

< 50% (SOTA – LB), 99%

Go/No go   Code drop

A – Accuracy / P – Precision / R – Recall / F1 score – harmonic mean of precision and recall
ML – Machine Learning, ES-ML - Energy Semantics of Machine Learning
LB – Lower Bound, SOTA – State of the Art

Selection gates and reminders

PS announced	Written abstracts due	Invitation to oral and written proposal

- Please note open-source license <u>CLEARLY</u> in your proposals!
  - (e.g., MIT License)
- Submit On Time: Please do not wait until the end to submit!
  - Late is late, even by a few seconds
- Proposals are defining experiments and giving evidence as to why their approach is valid and novel
- In the technical section – please be technical!
  - It is encouraged to jump technical details levels between the executive and the technical sections

Vision and transition

Vision: ML2P software is the gold standard for ML construction and simulation of power usage

Impact Objective: Establish presence on machine learning development sites such as Scikit-learn, enabling broad adoption by public/private AI communities of practice.

Strategy:

- **Mature technology:** Publish documentation, algorithms, code, and tutorials licensed as open-source via existing ML repository sites (e.g., scikit-learn) and the DARPA GitHub page code enable community developers
- **Advance scientific research:** Publish in conferences (e.g., NeurIPS) and peer reviewed journals (e.g., IEEE)
- **Attract investment:** Seek strategic partnerships with standards and requirements consortium (e.g., Sensor Open Systems Architecture) to influence adoption and engage industry focused on low powered ML (e.g., EDGE AI Foundation)

ML – Machine Learning, AI – Artificial Intelligence

DARPA

www.darpa.mil