AI & Energy:
Scenario Based Analysis of Energy Demand of AI
&
SOSX Based Training of ANN

Metin Türkay

Department of Industrial Engineering

Koç University, Istanbul, Türkiye

CAPD, Carnegie Mellon University – September 17, 2025

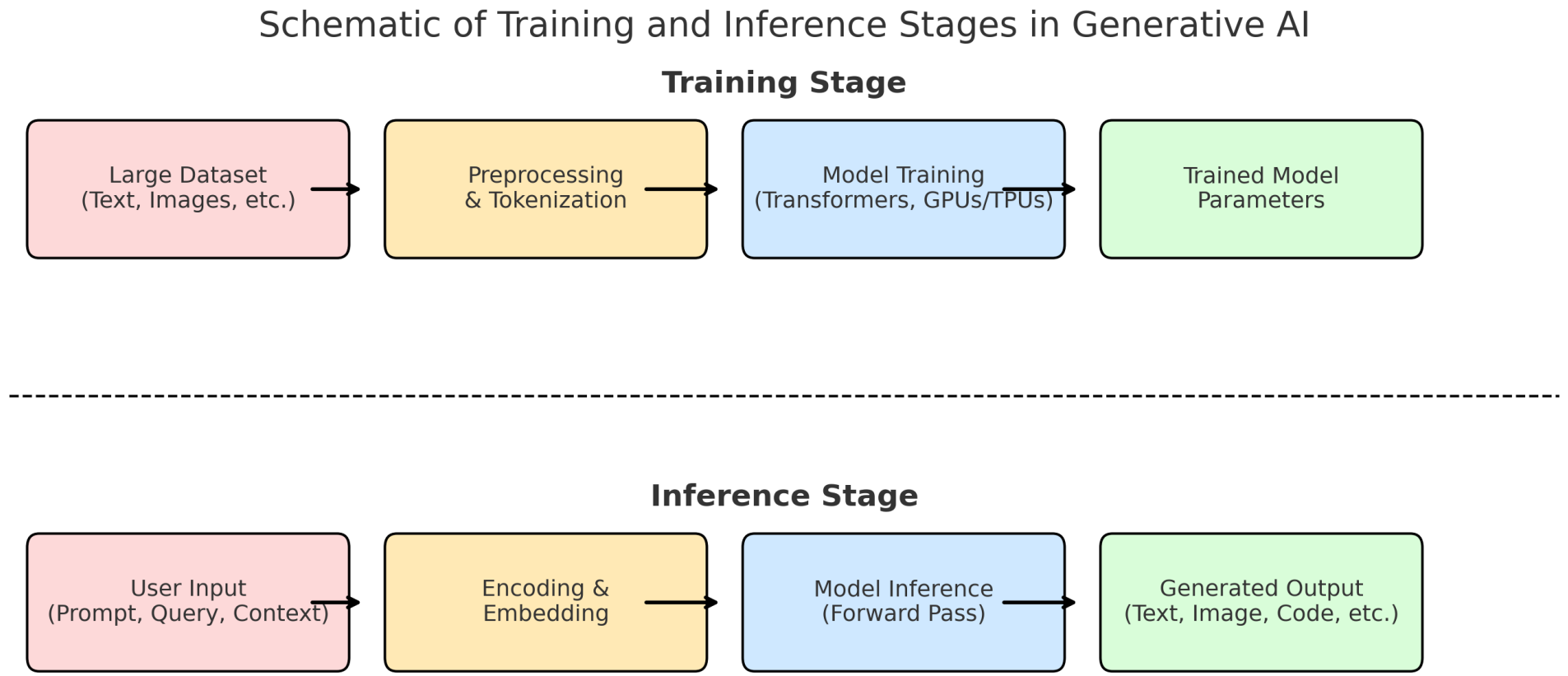
OUTLINE

- Motivation
  - ✓AI: How does it work?
  - ✓AI & Energy
  - ✓Why so much energy is needed for AI?
- AI: Training Problem
  - ✓The structured approach for using AI: **Training** (Training, Validation, Testing) and **Inference**
  - ✓Training problem: activation function in ANNs
  - ✓SOSX approach for instant and accurate training of ANNs
  - ✓Results on Benchmark Problems
- Discussion & Conclusions
  - ✓Energy conundrum of AI

MOTIVATION

- What is the interplay between AI and Energy
  - ✓Scenario based projections of
    - AI's Adoption
    - AI Strategy (Baseline, Fewer-Larger, More-Smaller)
    - Computing Power
    - Algorithmic Efficiency
  - ✓Carbon emissions considering
    - Business As Usual
    - Emission Targets
- Why does AI need so much energy?
  - ✓Training and Inference stages
  - ✓Deeper analysis of algorithms

HOW DOES AI (GenAI) WORK?



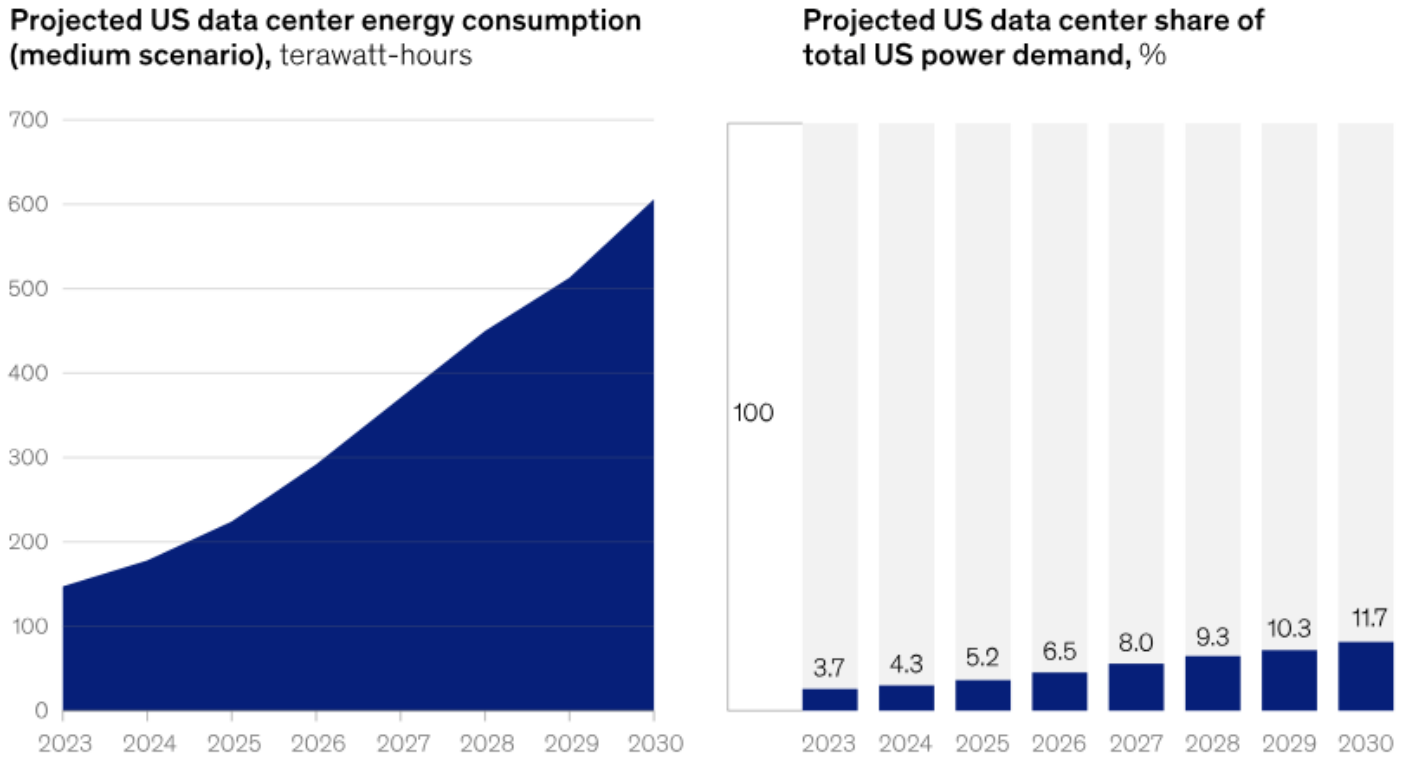
Source: ChatGPT 5

AI & ENERGY

AI & Energy

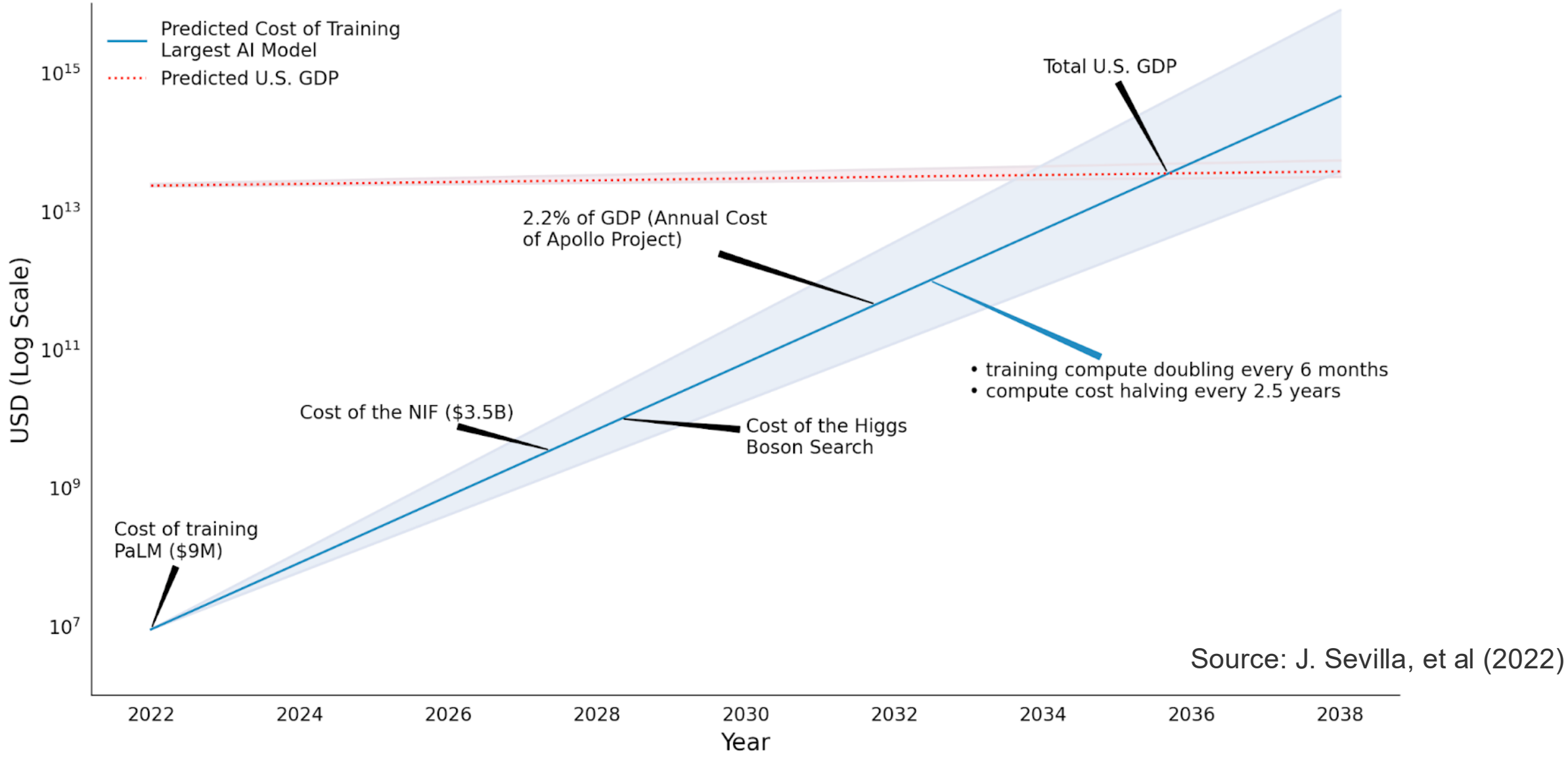
- ✓ AI needs significant amounts of energy
- ✓ Demand for AI is increasing resulting in
- ✓ Higher energy consumption share
  - 4.3% of electricity is used for AI in US (2024)
  - 1% of global energy consumption is for AI (2024)
- ✓ How AI models are
  - Trained
  - Inferred/Served
  - Used
- ✓ Need to consider energy demand during
  - Hardware
  - Training Models
  - Inference/Service Models
  - Adaptation & Usage Profile
- ✓ Efficiency gains
  - Hardware
  - Training Models
  - Inference/Queries/Requests
- ✓ AI & Energy is a complex issue

Demand for power for data centers is expected to rise significantly in the United States.



https://www.mckinsey.com/featured-insights/sustainable-inclusive-growth/charts/ais-power-binge

AI ADOPTION



Source: J. Sevilla, et al (2022)

METHODICAL APPROACH

Energy Cost of AI
Power required per year

1) Energy Efficiency
Power required per FLOP

A) Hardware Efficiency
Improvements in energy requirements of GPUs

B) Algorithm Efficiency
Improvements in optimizing energy use of algorithms

C) Power Usage Effectiveness (PUE)
Improvements in operational efficiency of data centers (DCs)

2) Computational Demand
FLOPs required per year

A) Training Demand
Computational requirements of training new AI models

I) Model Complexity
Computational requirements per new AI model

II) Model Demand
New AI models and versions developed per year

B) Usage Demand
Computational requirements of training new AI models

I) AI Users
Projected AI users based on population growth and adoption rate

II) Query Complexity
New AI models and versions developed per year

1) Electricity Demand
Electricity required to meet energy cost, in Terawatt hours (TwH), computational demand divided by energy efficiency

2) GPU Demand
GPUs required to meet energy demand in DCs

GPU Efficiency
Energy cost of GPUs, modeled using utilization rate, power, and average lifetime

Carbon Footprint of AI
CO2 Emissions per year

1) Electricity Carbon Footprint
Emissions from supplying power to DCs

A) Carbon Cost of Electricity
Emissions by power source

I) Energy Pricing
Cost per TwH by energy source e.g. coal, solar

II) Emission Factor
Emissions per TwH per unit price for each energy source

B) Energy Demand
Electricity demand by country for each power source

I) Global Share
Each country's share of global DCs count

II) Allocation
Distribution of demand by energy source per country

C) Total Carbon Footprint
Comprehensive footprint by country

I) Integration
National policy targets for energy mix

II) Estimation
Total carbon cost, split for training and usage

2) Supply Chain Carbon Footprint
Emissions from supplying GPUs to DCs

A) Manufacturing
Emissions from producing GPUs

B) Transportation
Emissions from transporting GPUs to DCs

AI ADOPTION SCENARIOS

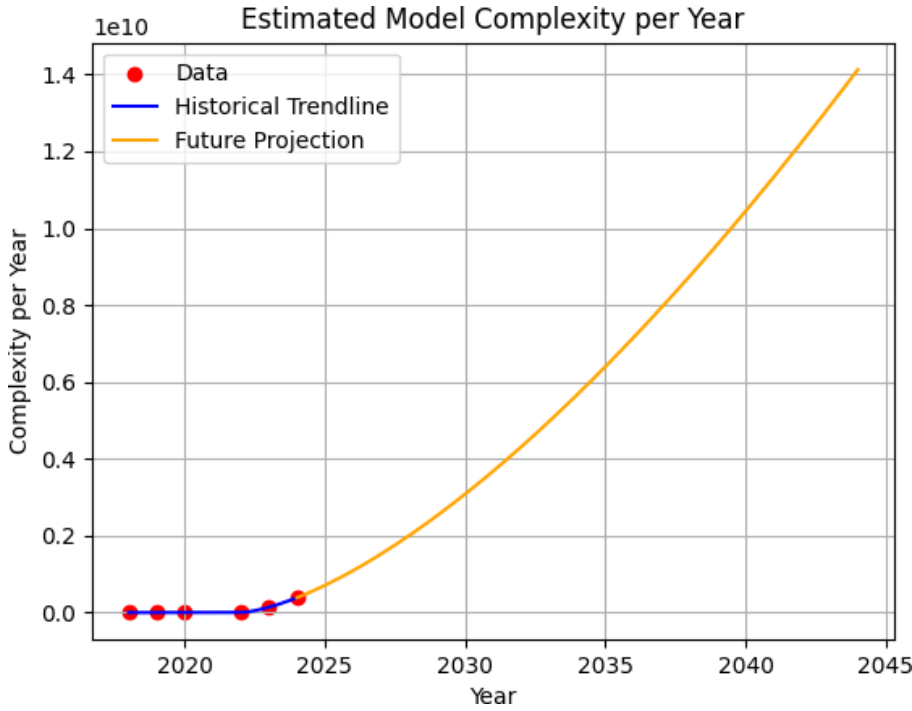
➢3 Scenarios for AI Adoption

- ✓**Baseline:** Continues the current trend of AI development without any significant change in the number of platforms used for AI training and inference
- ✓**Fewer-Larger:** A small number of platforms remain to serve different geographies and alliances across the globe – Artificial General Intelligence
- ✓**More-Smaller:** A very large number of platforms serve to different geographies and sectors across the globe – micro & specialized platforms

AI ADOPTION SCENARIOS

➢Complexity of AI Models

- ✓The use of AI for tasks will become more complex requiring training more complex models
- ✓Developed world is expected to lead the new uses of AI for different tasks
  - US already accounts for 9% of active AI users globally only with 4.22% of global population but responsible from 66.8% AI-related emissions (2024)
- ✓It is important to distinguish the energy demand during
  - Hardware manufacturing (GPU) and installation
  - Data/Computing Center locations for Training AI models
  - Data/Service Center locations for AI Inference hosting
- ✓Hardware Energy Efficiency Gains
- ✓AI model complexity increase vs algorithmic efficiency
  - ✓ Training
  - ✓ Inference
- ✓Energy sector efficiency gains
- ✓Carbon mitigation targets



HARDWARE EFFICIENCY

➢Hardware Efficiency

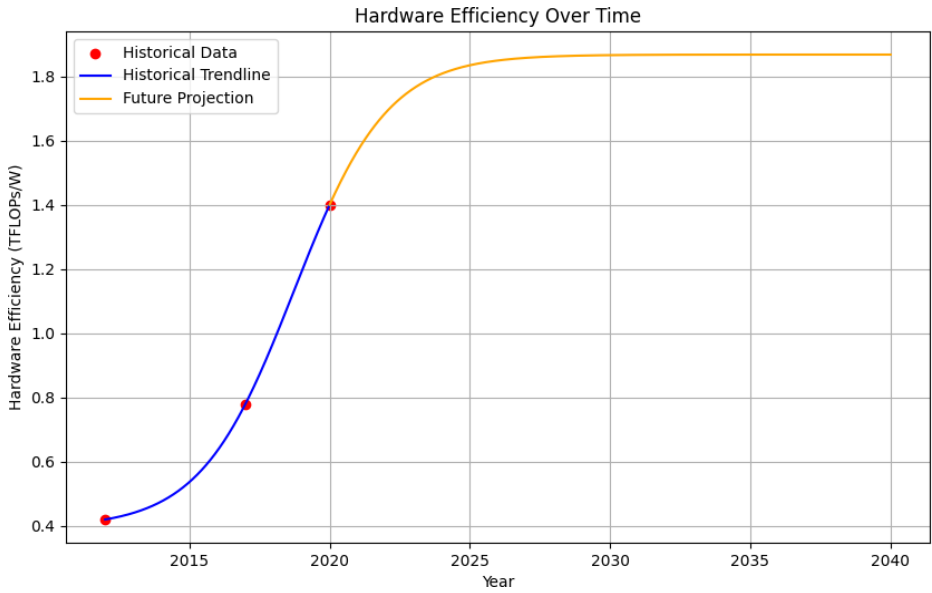
- ✓Hardware efficiency captures improvements in the energy consumption of data center IT equipment over time
- ✓The two major components are central processing units (CPUs) and graphics processing units (GPUs), with GPUs playing a central role in AI workloads.
- ✓Hardware efficiency captures improvements in the energy consumption of data center IT equipment over time.

$$E_h(t) = E_{h0} exp\left(\frac{\alpha_h}{\beta_h}\left(1 - exp(-\beta_h t)\right)\right)$$

$\boldsymbol{E_{h0}}$ represents the baseline hardware efficiency at $t = 0$

$\boldsymbol{\alpha_h}$ denotes the initial rate of improvement in hardware efficiency

$\boldsymbol{\beta_h}$ is the deceleration factor, capturing the slowdown in efficiency gains over time.



AI ALGORITHMS & THEIR EFFICIENCY

Algoritmic Efficiency

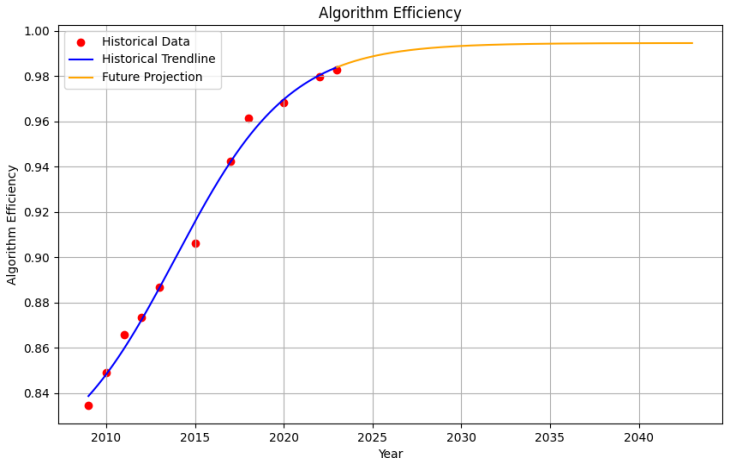
- ✓ As AI workloads grow, efficiency gains from algorithmic improvements become critical to mitigating overall energy consumption
- ✓ Algorithmic efficiency reflects the energy savings achieved through advances in computation, such as reduced redundancy and optimization of processing steps.

$$E_a(t) = E_{a0} exp\left(\frac{\alpha_a}{\beta_a}\left(1 - exp(-\beta_a t)\right)\right)$$

$\boldsymbol{E_{a0}}$ represents the baseline algorithmic efficiency at $t = 0$

$\boldsymbol{\alpha_a}$ denotes the initial rate of improvement in algorithmic efficiency

$\boldsymbol{\beta_a}$ accounts for diminishing returns as optimization approaches fundamental limits



Year	Unsolved Problems
2009	1655
2010	1511
2011	1343
2012	1265
2013	1134
2015	939
2017	575
2018	385
2020	319
2022	202
2023	173

Gurobi Benchmark dataset

EFFECTIVE USE OF POWER

➢Power Usage Effectiveness

- ✓ Power Usage Effectiveness (PUE) measures the ratio of total facility energy consumption to IT equipment energy consumption.
- ✓ A PUE of 1.0 represents perfect efficiency, while real-world data centers typically operate at values closer to 1.5, with the remainder of energy use attributed to cooling and infrastructure.

$$E_p(t) = E_{p0} exp\left(\frac{\alpha_p}{\beta_p}\left(1 - exp\left(-\beta_p t\right)\right)\right)$$

$\boldsymbol{E_{p0}}$ is the initial PUE at $t = 0$

$\boldsymbol{\alpha_p}$ is the rate of improvement in PUE, reflecting efficiency enhancements in cooling and facility operations

$\boldsymbol{\beta_p}$ accounts for diminishing returns as data centers approach optimal efficiency

ENERGY EFFICIENCY

➢ Total Energy Efficiency

✓ All improvements in affect the Total Energy Efficiency

- Hardware Efficiency
- Algorithmic Efficiency
- Power Usage Effectiveness

$$E(t) = \frac{E_h(t)}{E_a(t)E_p(t)}$$

$E(t)$ values indicate higher efficiency in energy use per FLOP.

AI ADAPTATON SCENARIOS

➢ Demand for AI

✓ AI Computation Demand has 2 components: Training and Inference (Usage):

$$\boldsymbol{C(t) = C_t(t) + C_u(t)}$$

$C(t)$ is the total demand for AI
$C_t(t)$ accounts for model complexity and frequency of model training
$C_u(t)$ is driven by AI adoption and query intensity

$$\boldsymbol{C_t(t) = C_c(t) + C_m(t)}$$

$C_c(t)$: model complexity (FLOPs per model), evolves as $C_c(t) = C_{c0}(t + \lambda_{c1})^{\eta_{c1}}$

$C_m(t)$: model demand (FLOPs per model), evolves as $C_m(t) = C_{m0}(t + \lambda_{m1})^{\eta_{m1}}$

$$C_u(t) = \sum_{i=1}^{I} (P_{i0} e^{r_i t})\, A_i(t) \left(Q_0 e^{\chi^t}\right)$$

P_{i0} is the initial population of user group i.

r_i is the population growth rate in group i.

$A_i(t)$, the adoption rate, follows a logistic function: $A_i(t) = \frac{1}{1+\exp(-\gamma_i(t-t_{0i}))}$

Q_0 represents initial query complexity, and χ is the growth rate of query complexity.

DATA CENTER LOCATIONS

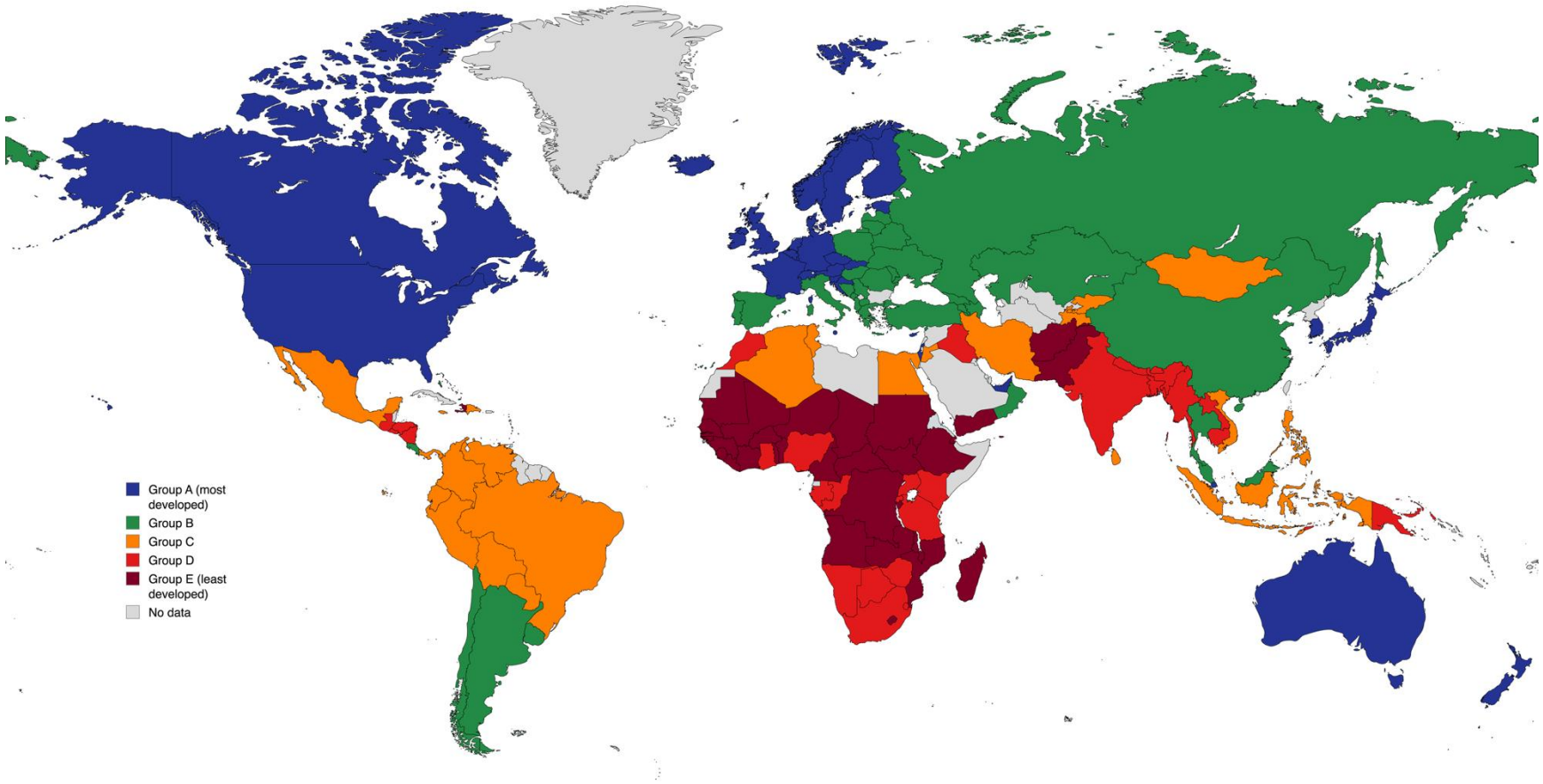
➢ENERGY MIX (Energy Production Targets by Country for 2030)

Table 12: Energy demand allocation by country based on global data center distribution for 2030 (TWh).

Country Name	Data Centers	% Share	Baseline	Fewer-Larger	More-Smaller
United States	1958	49.0%	448	308	594
China	375	9.4%	86	59	114
United Kingdom	243	6.1%	56	38	74
Germany	198	5.0%	45	31	60
Canada	168	4.2%	38	26	51
France	143	3.6%	33	22	43
India	141	3.5%	32	22	43
Netherlands	138	3.5%	32	22	42
Australia	132	3.3%	30	21	40
Japan	121	3.0%	28	19	37
Brazil	118	3.0%	27	19	36
Singapore	96	2.4%	22	15	29
Hong Kong	59	1.5%	13	9	18
Spain	57	1.4%	13	9	17
Indonesia	49	1.2%	11	8	15

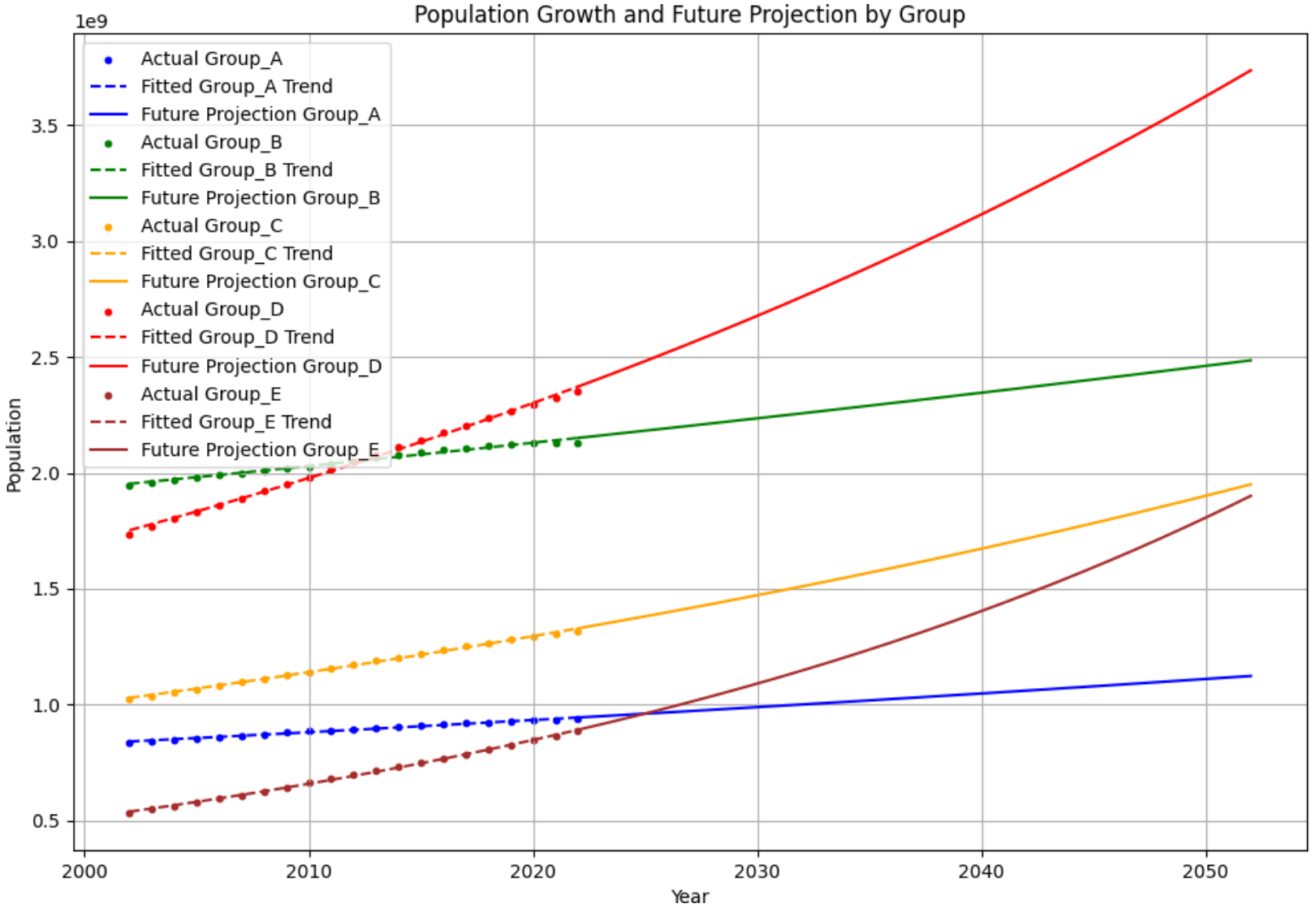
GLOBAL DISTRIBUTION OF ENERGY DEMAND

- **Group A (Most Developed):** Includes the United States, Canada, Western European nations, Japan, Australia, and South Korea.
- **Group B:** Consists of emerging economies with strong technological bases, such as China, Eastern European nations, and select Latin American countries like Brazil.
- **Group C:** Includes middle-income countries with moderate AI adoption, such as Mexico, Egypt, Philippines, and Indonesia.
- **Group D:** Consists of lower-middle-income nations in Southern Africa and South Asia.
- **Group E (Least Developed):** Comprises countries with minimal AI presence, including regions in Sub-Saharan Africa, war-affected nations, and those with severe energy access issues.

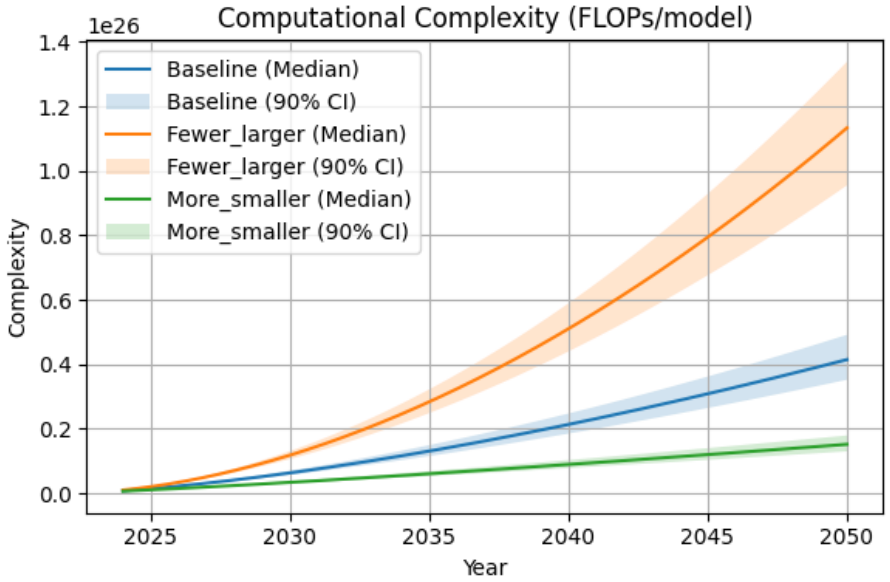


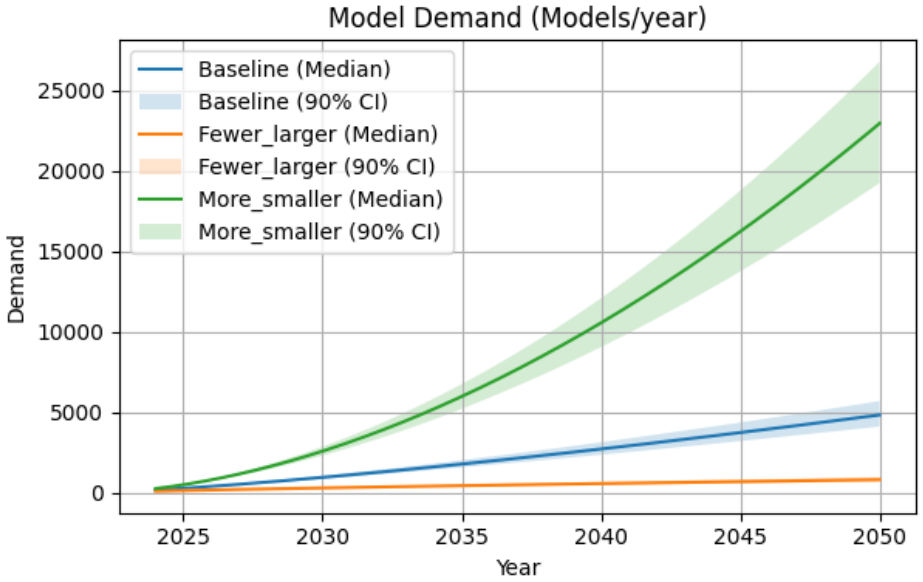
GLOBAL DISTRIBUTION OF ENERGY DEMAND

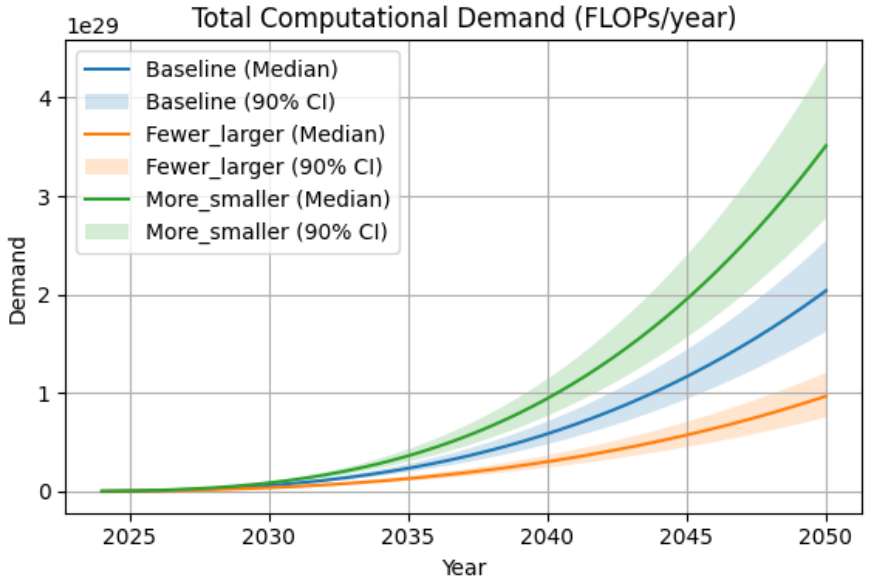
- **Group A (Most Developed):** Includes the United States, Canada, Western Euro- pean nations, Japan, Australia, and South Korea.
- **Group B:** Consists of emerging economies with strong technological bases, such as China, Eastern European nations, and select Latin American countries like Brazil.
- **Group C:** Includes middle-income countries with moderate AI adoption, such as Mexico, Egypt, Philippines, and Indonesia.
- **Group D:** Consists of lower-middle-income nations in Southern Africa and South Asia.
- **Group E (Least Developed):** Comprises countries with minimal AI presence, including regions in Sub-Saharan Africa, war-affected nations, and those with severe energy access issues.

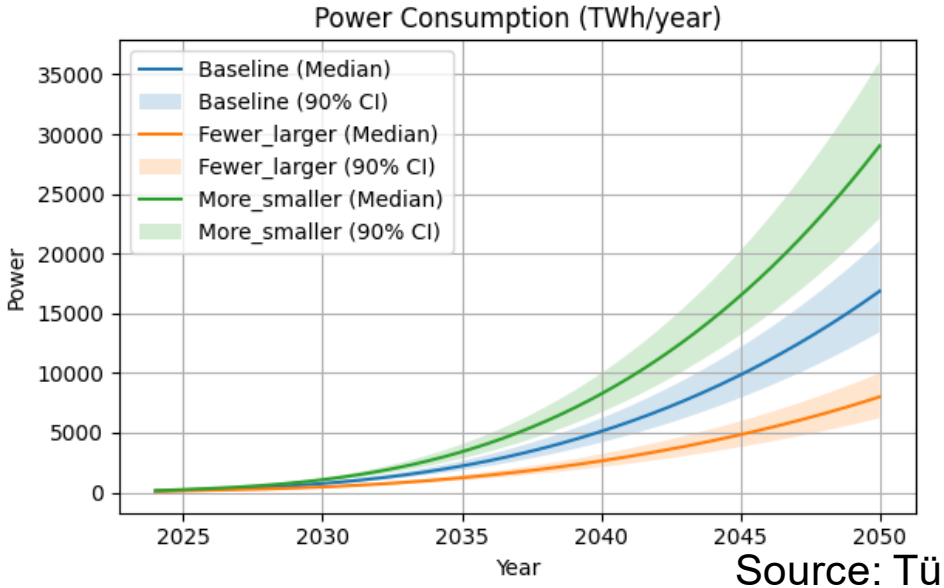


RESULTS: AI ADOPTION









Source: Türkay et al (2025)

AI: GOLD RUSH

Pressure to deliver fast

- openAI, Google (Gemini), Microsoft (pilot), X (Grok) and Meta
- Apple and amazon are getting ready to join
- Look beyond concrete fortresses and 36-month timelines. Their new weapon in the $400 Billion Ai compute race
- Billion-dollar GPU clusters running under hurricane-proof canvas, not steel, in just 3 weeks

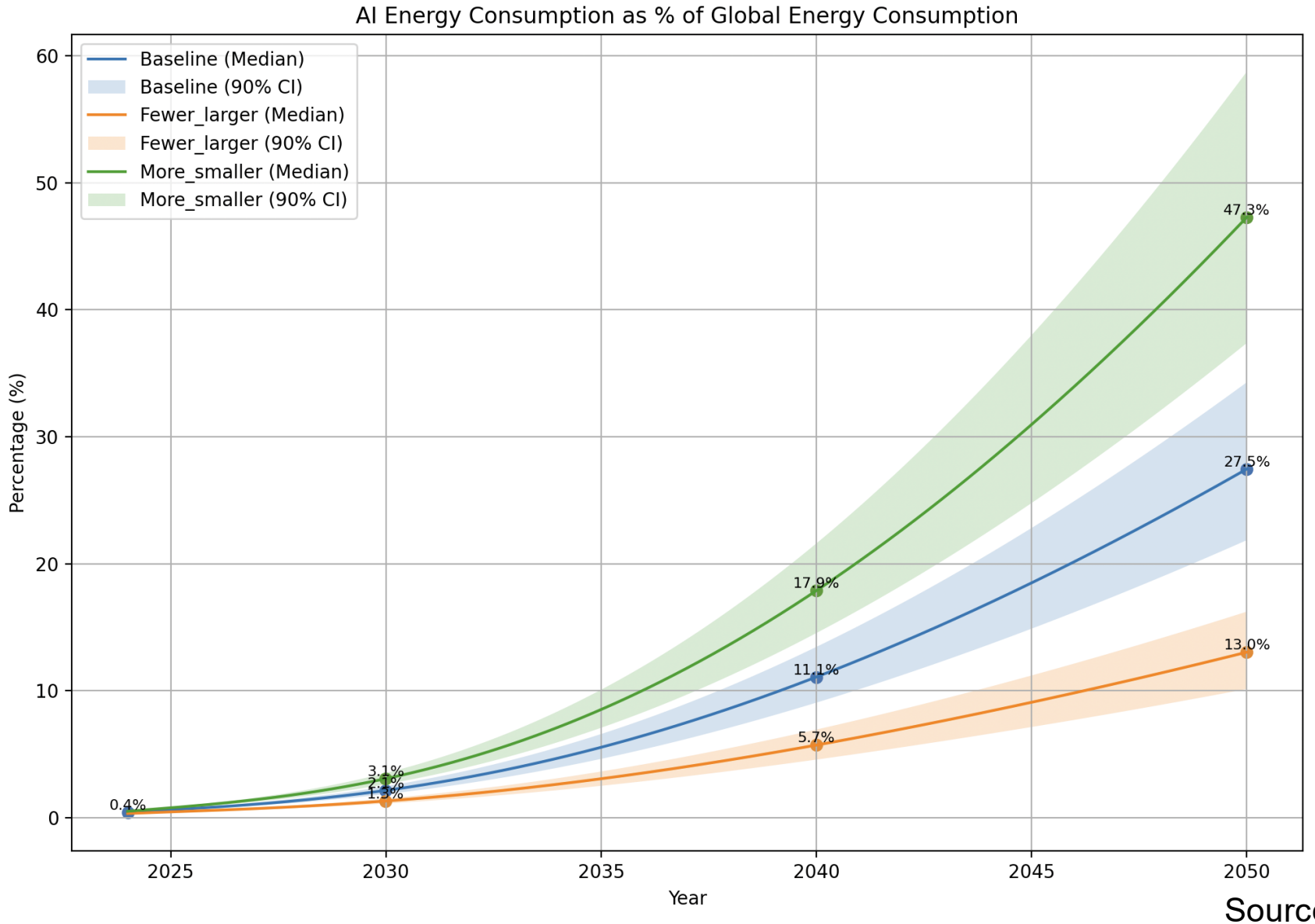
Materials

- Aerospace-grade aluminum frames
- 400 MW on-site power substations
- $2-3B worth of compute under each tent
- Zero backup generators, no fortress walls
- Next-gen liquid cooling pushing 95% efficiency

AI Training

- Meta is running some of the world's most powerful Ai:
- 20,000+ GPUs per site
- 24/7 orchestrated AI training
- Elite engineers running everything like an F1 pit crew
- 95% uptime

RESULTS: AI's ENERGY DEMAND PROJECTIONS



Source: Türkay et al (2025)

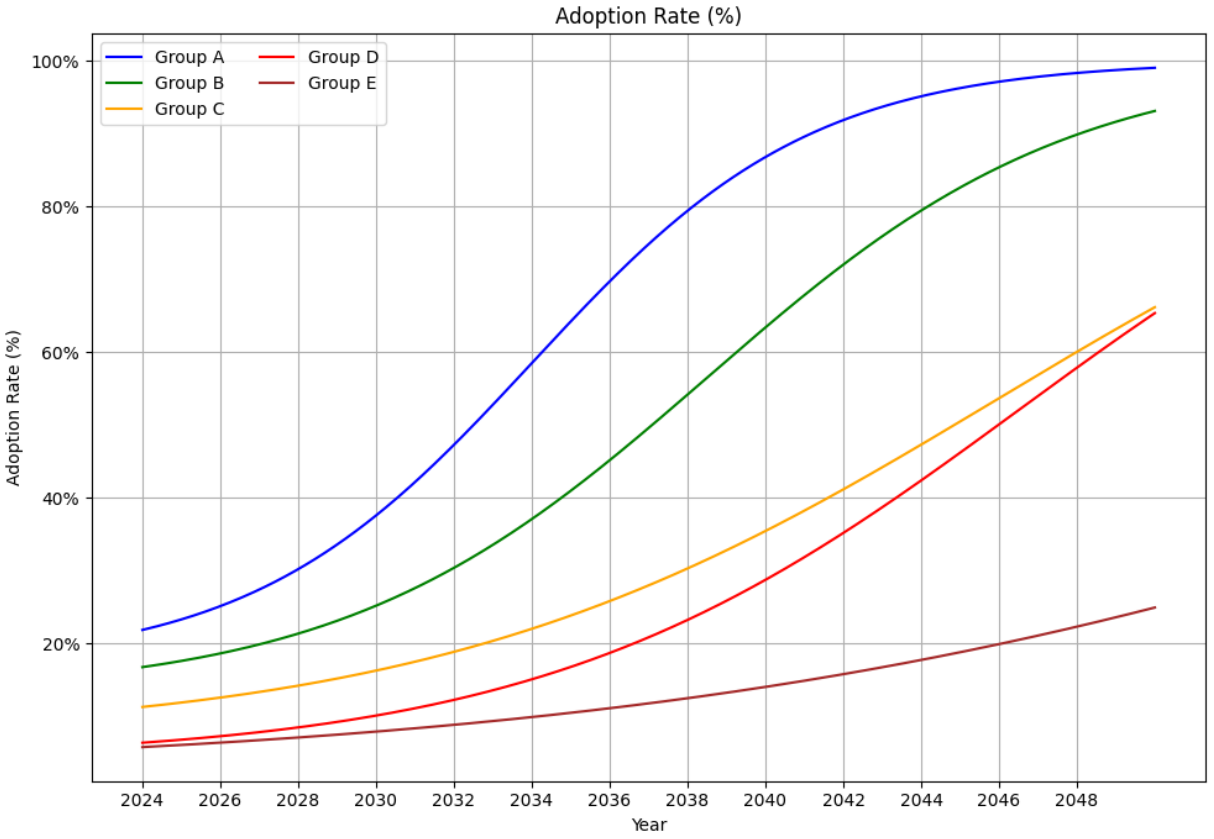
AI CENTERS & ENERGY SOURCE/EMISSIONS

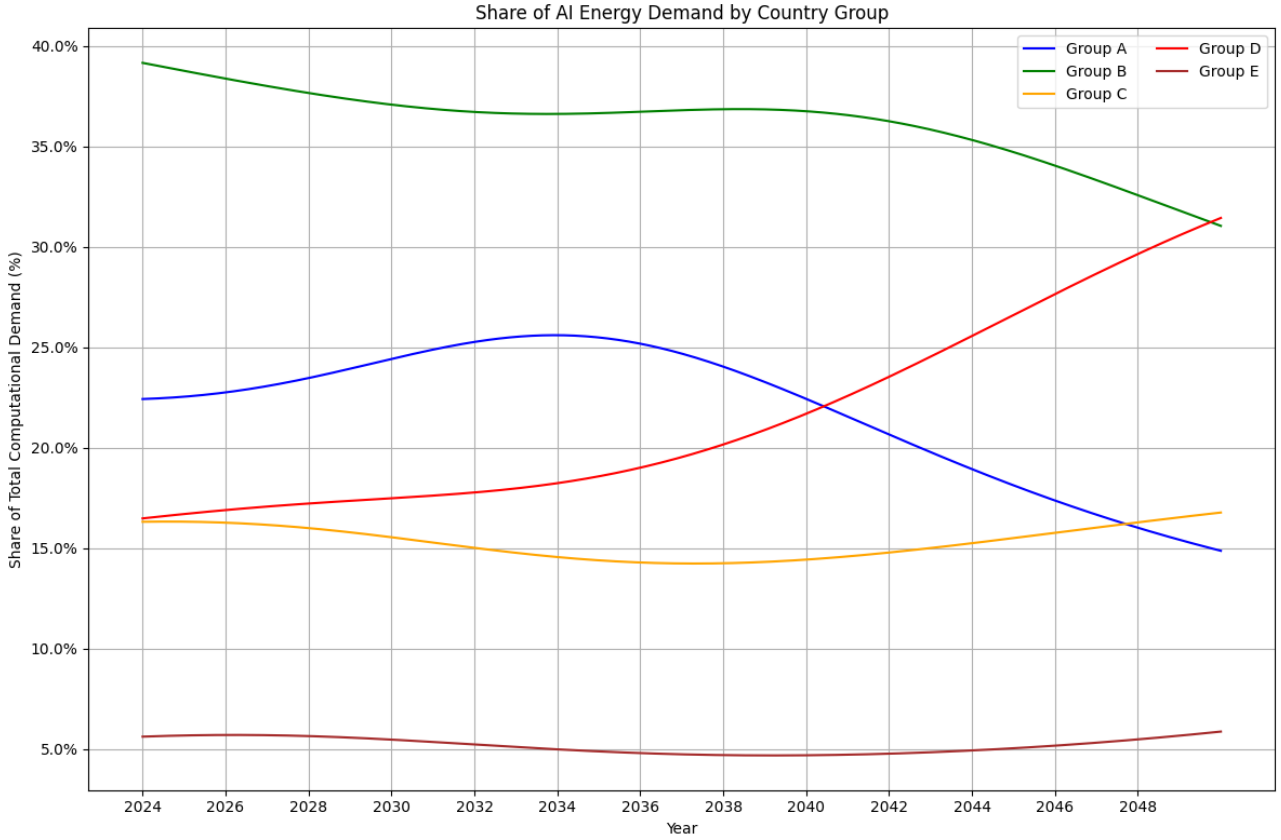
➢Data Centers & Their Energy Source

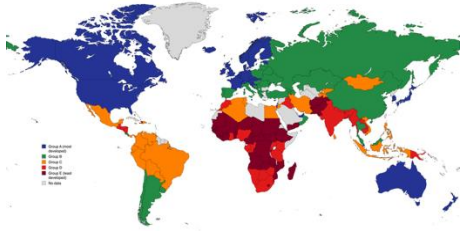
Table 9: Emission Factor GWP*Price: kgCO_2/TwH by country and energy type

Country	Other ren	Bioenergy	Solar	Wind	Hydro	Nuclear	Oil	Gas	Coal
Australia	0	385625542	95259.4572	2945726.89	18048819.8	0	372482830	830201501	1596456432
Brazil	0	852146.118	0	288380.742	834790.903	2656462.72	952398330	620232941	406512518
Canada	0	14571851.4	143146600	16265675.7	1745287.96	1102165.33	1884575052	234064211	1534625910
China	0	6627593.23	401496.57	757681.115	15448123.1	5445121.81	57211678.9	321583318	744582542
France	38217159.9	52364092.2	264330.61	6049937.13	11557847.9	18788607	159455983	337748925	2093224609
Germany	0	116496762	36256672.6	3079718.29	7882972.74	17189057.7	57216492.9	77642257.2	1110729875
India	0	4388737.65	3389060.39	9520558.31	9306567.15	5598860.21	1526785464	494387466	906494468
Indonesia	101358201	161434.384	0	0	4590581.35	0	1061547536	23035881.1	101290176
Japan	0	370097986	71285.6374	2035459.63	1937046.87	280229807	1426096103	629249883	801304835
Netherlands	0	2105035.82	418854.48	13875516.3	10605436.4	14949638.4	11042205.6	43899024.4	984150282
Spain	85199000000	58636529.2	547370.944	34261979.7	31469223.7	31634391.8	820286082	305143368	12076000000
United Kingdom	0	31762320.7	28041592.7	25046164.1	11316999.1	3314333.44	74594931.2	302233442	10135000000
United States	673856.67	277461887	2542832.33	315496.971	4633629.73	1270051.35	400854666	443602027	1535621222

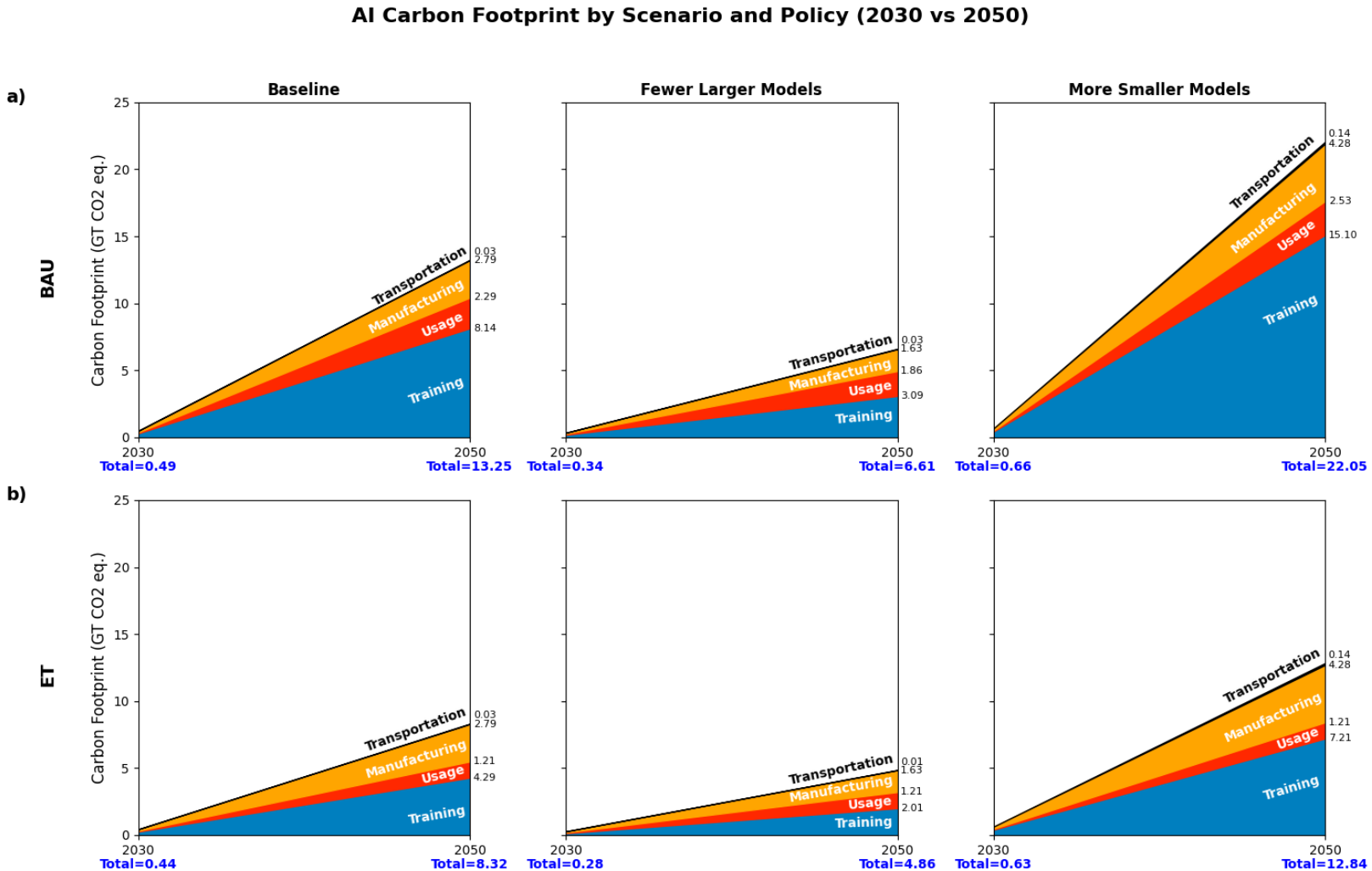
RESULTS: GLOBAL DISTRIBUTION OF ENERGY DEMAND







ENERGY NEEDED: TRAINING & INFERENCE

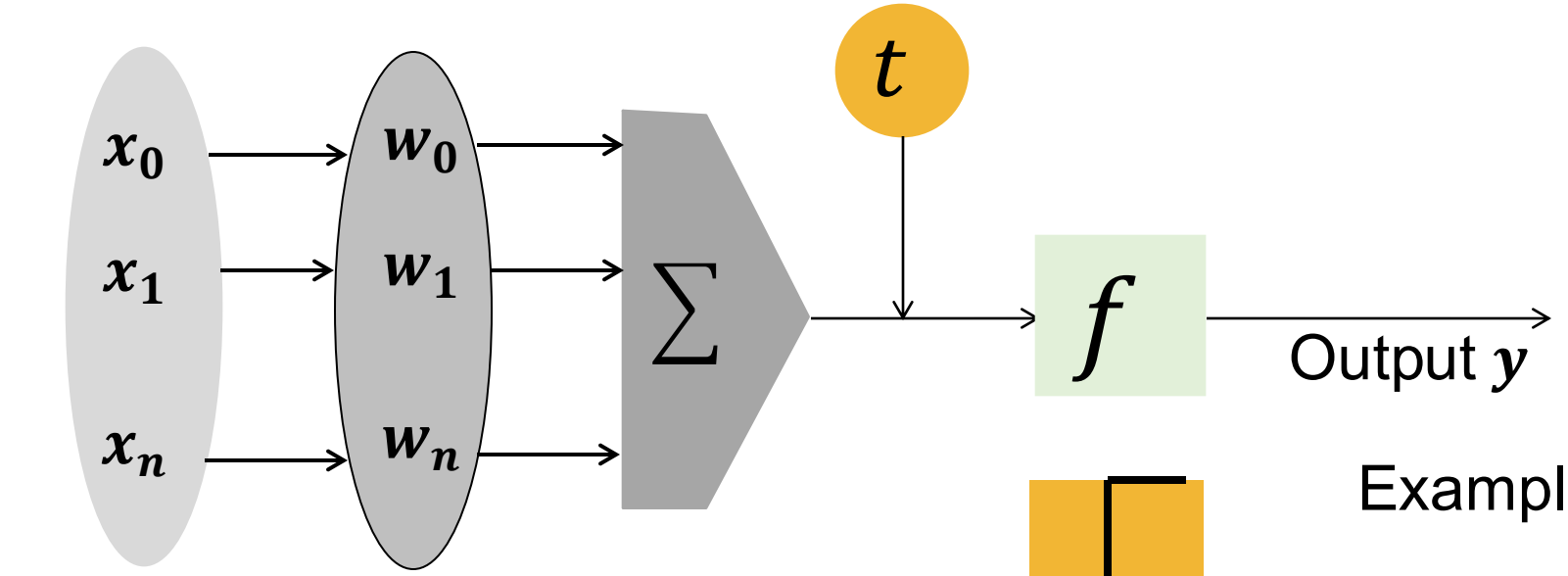


Source: Türkay et al (2025)

SUMMARY OF RESULTS

- AI's impact on energy is significant
  - ✓13-47% of global energy consumption by 20250!
- AI scaling strategies significantly impact carbon output
  - ✓More-Smaller Models (BAU) scenario producing the highest emissions at 22.05 Gt CO2, followed by the Baseline (13.25 Gt CO2) and Fewer-Larger Models (6.59 Gt CO2).
- Implementing Emissions-Targeted (ET) strategies leads to substantial reductions across all cases, with the most pronounced decrease in the More-Smaller Models scenario, where emissions drop by 41.8% to 12.83 Gt CO2
- The primary contributor to emissions remains the Training phase, underscoring the urgent need for low-carbon energy solutions, efficiency improvements in AI infrastructure and better training algorithms.

THE TRAINING PROBLEM



Example:

$$y = sign\left(\sum_{i=0}^{n} w_i x_i - t\right)$$

✓n-dimensional input vector x, via scaler product and mapping a nonlinear function are used to determine the output variable y

ANNs

- A feed-forward, fully connected artificial neural network with a single layer is defined as:

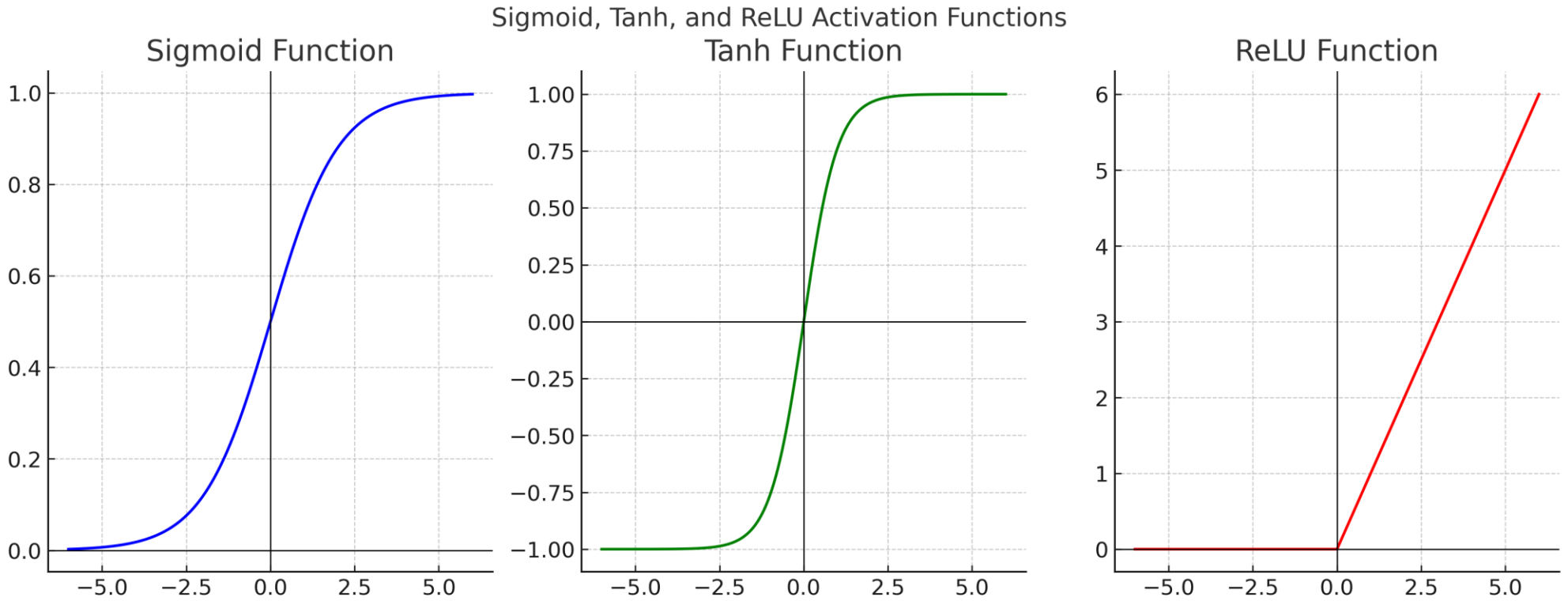
$$\hat{y} = f_1(w_1 f_2(w_2 x + b_2) + b_1)$$

x input matrix
$\hat{y}$ output matrix
f_1, f_2 activation functions at output and hidden layers
w_1, w_2 weight matrices
b_1, b_2 bias matrices

- Typically, the training problem with N training data points is defined as minimizing mean absolute error (MAE) which is defined as:

$$\min_{w_1, w_2, b_1, b_2} \frac{1}{N} \sum_{i=1}^{N} |f_1(w_1 f_2(w_2 x + b_2) + b_1) - y_i|$$

ACTIVATION FUNCTIONS



- Nonlinear, moreover a nonconvex function
- State-of-the-art AI platforms (chatGPT, Grok, Gemini) uses:
  - ✓ **Gradient Descent Algorithm** (**Cauchy, A.-L. (1847).** *Méthode générale pour la résolution des systèmes d'équations simultanées.* Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, 25, 536–538.)
  - ✓ **Second Order / Quasi-Newton Method** **Fletcher, R., & Powell, M. J. D. (1963).** *A rapidly convergent descent method for minimization.* Computer Journal, 6(2), 163–168.

HARDWARE NEEDED FOR TRAINING

Family / Version	Parameters	Training Compute / Time	GPUs / TPUs Used
OpenAI – GPT-3 (2020)	175B	~3.1×10^23 FLOP	~1024 A100 (est.)
OpenAI – GPT-4 family (2023–2025)	Undisclosed	>10^25 FLOP (est.)	10k–25k GPUs (est.)
OpenAI – GPT-5 (2025, projected)	Undisclosed	Undisclosed (likely >10^26 FLOP)	Undisclosed (multi-10k GPUs est.)
Google – Gemini (1.5 / 2.0 / 2.5)	Undisclosed	Undisclosed	Thousands of TPUs (est.)
xAI – Grok-2 (MoE)	~270B total / ~115B active	Undisclosed	Undisclosed
Anthropic – Claude 3 / 3.5	Undisclosed	Undisclosed	Undisclosed
Meta – Llama-3 (2024)	8B, 70B	Undisclosed	2k–8k GPUs (est.)
Mistral – Mixtral 8×7B (2023–2024)	~47B total / ~13–14B active	Undisclosed	512–1024 GPUs (est.)

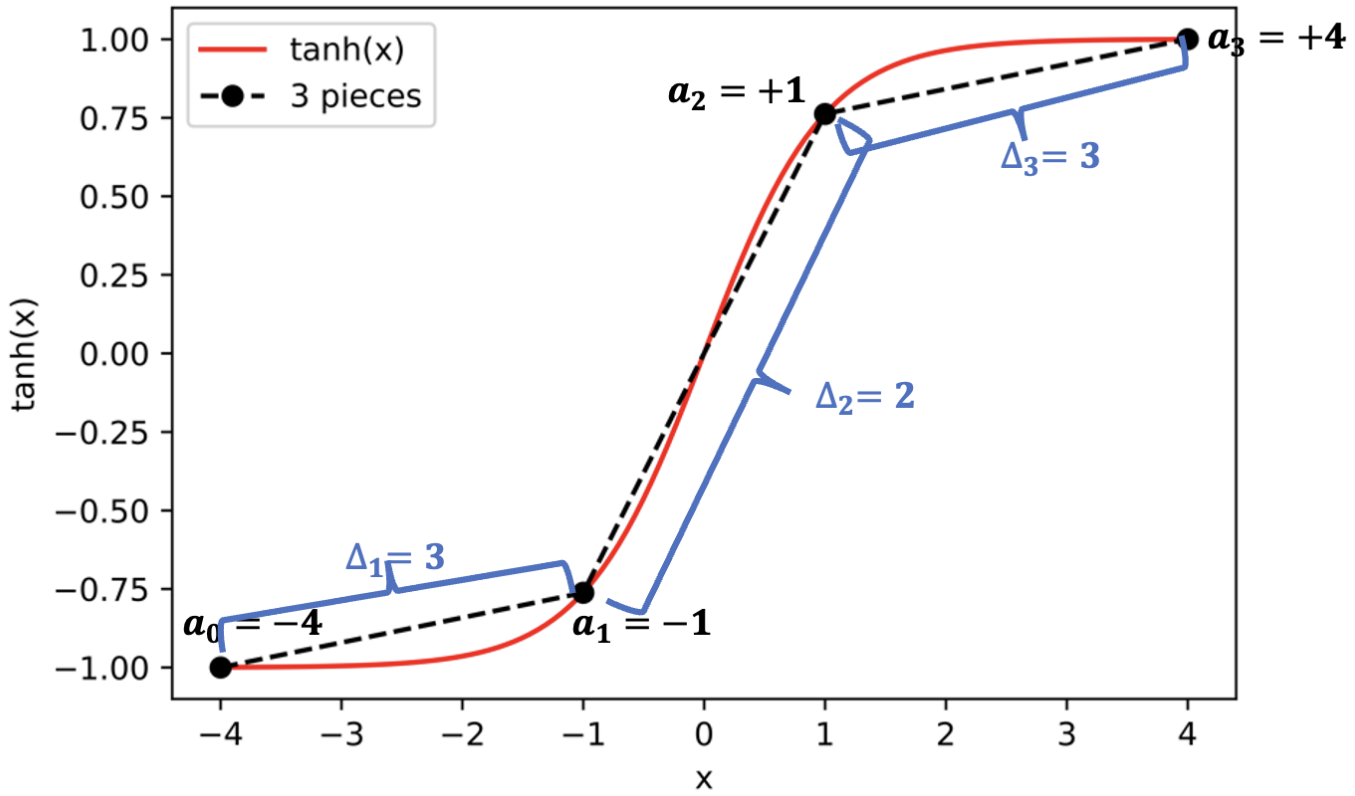
Source: ChatGPT 5

- Despite all of this effort:
  - ✓ Global optimality cannot be guaranteed
    - Stochastic behavior of the tweaked gradient descent and quasi-Newton algorithms
  - ✓ Need to retrain the entire system with new data
    - No warm start
  - ✓ Every version needs
    - More memory
    - Faster GPUs/TPUs
    - More GPUs/TPUs
    - More Energy
    - More Water for Cooling

CONVEX TRAINING

$$min_{w_1,w_2,b_1,b_2} \frac{1}{N}\sum_{i=1}^{N}\left|\left(f_1(w_1 f_2(w_2 x_i + b_2) + b_1)\right) - y_i\right| \longrightarrow min_{w_1,w_2,b_1,b_2}|\nu'| = \frac{1}{N}\sum_{i=1}^{N}|(w_1 f_2(w_2 x_i + b_2) + b_1) - y_i|$$

non-smooth, nondifferentiable



$$\nu' \leq |\nu'|$$

$$-|\nu'| \leq \nu'$$

$$a_0 \leq w_2, b_1, b_2 \leq a_P$$

$$\Delta_k = a_k - a_{k-1}$$

$$g_k = \sigma'(a_k) - \sigma'(a_{k-1}) \;\; \forall\, k \in \{1, \ldots, P\}$$

$$a_0 \leq w_2 x_i + b_2 \leq a_P$$

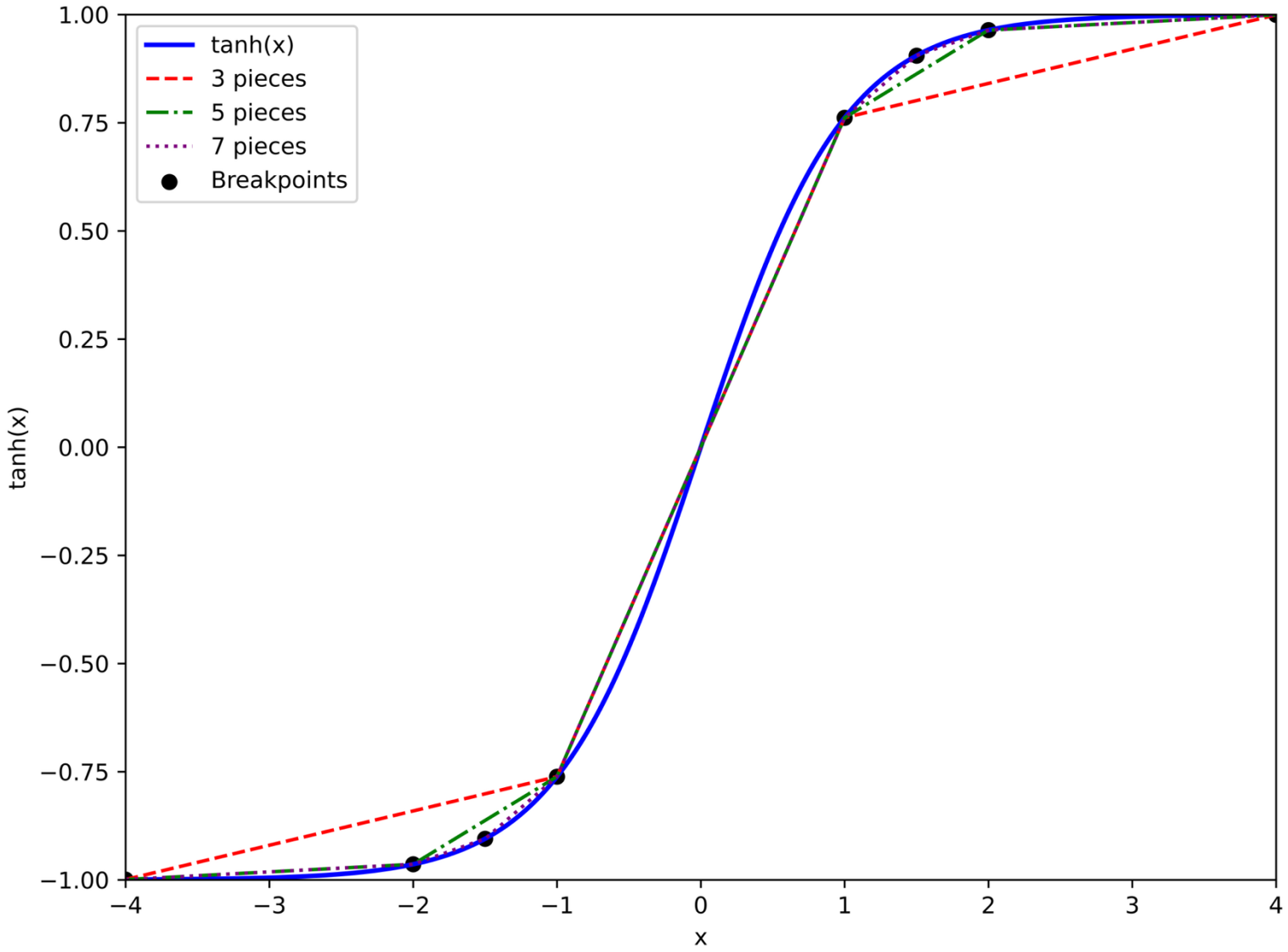
$$w_2 x_i + b_2 = a_0 + \sum_{k=1}^{P} y_k$$

$$\sigma'(w_2 x_i + b_2) = \sigma(a_0) + \sum_{k=1}^{P} \frac{g_k}{\Delta_k} y_k$$

D'Ambrosio, C., Lodi, A., & Martello, S. (2010). Piecewise linear approximation of functions of two variables in MILP models. *Operations Research Letters*, *38*(1), 39–46. https://doi.org/10.1016/j.orl.2009.09.005

Keha, A. B., De Farias, I. R., & Nemhauser, G. L. (2004). Models for representing piecewise linear cost functions. *Operations Research Letters*, *32*(1), 44–48. https://doi.org/10.1016/S0167-6377(03)00059-2

tanh: PIECEWISE LINEAR APPROXIMATION



Possible Formulations:

Unary (Jeroslow 1984)

- ✓ convex combination / incremental / SOS2
- ✓ introduce **binary variables** that indicate which interval is active
- ✓ each breakpoint is assigned its own binary
- ✓ compact, tight LP relaxation
- ✓ needs many binaries if the function has many segments

Binary (Vielma & Nemhauser, 2011)

- ✓ known as logarithmic formulation
- ✓ encode the active segment using a binary variable
- ✓ introduce binary variables that indicate which interval is active
- ✓ each breakpoint is assigned its own binary
- ✓ fewer binaries compared to unary
- ✓ the LP relaxation is usually weaker than unary

SOS2 and SOSX

SOS2

$$w_2 x_i + b_2 = \sum_{m=0}^{P} a_m \lambda_m$$

$$\sigma'(w_2 x_i + b_2) = \sum_{k=0}^{P} \sigma(a_m) \lambda_m$$

$$\sum_{m=0}^{P} \lambda_m = 1$$

$$\lambda_m \geq 0, \forall\, m \in \{0, \dots, P\}\ and\ SOS2$$

Explicit

$$\lambda_0 \leq z_0$$

$$\lambda_m \leq z_{m-1} + z_m\ , \forall m \in \{1, \dots, P-1\}$$

$$\lambda_m \leq z_{m-1}$$

$$\sum_{m=0}^{P-1} z_m = 1$$

$$z_m \in \{0,1\}$$

SOSX

$$\Delta_m Y_{m+1} \leq \Delta_{m+1} Y_m \quad \forall m \in \{1, \dots, P-1\}$$

$$Y_1 \leq \Delta_1$$

$$Y_P \geq 0$$

if $\tilde{Y}$ is the current solution (LP relaxation) with $\tilde{Y}_i < \Delta_i$ and $\tilde{Y}_{i+k} > 0$ for some $k \in \{1, \dots, t-i\}$, the are two branches:

$$Branch\ 1\colon\ Y_k = \Delta_k$$

$$Branch\ 2\colon\ Y_{k+1} = \cdots = Y_P = 0$$

Keha, A. B., De Farias, I. R., & Nemhauser, G. L. (2004). Models for representing piecewise linear cost functions. *Operations Research Letters*, *32*(1), 44–48. https://doi.org/10.1016/S0167-6377(03)00059-2

Vielma, J. P. (2015). Mixed integer linear programming formulation techniques. In *SIAM Review* (Vol. 57, Issue 1, pp. 3–57). Society for Industrial and Applied Mathematics Publications. https://doi.org/10.1137/130915303

TRAINING WITH SOSX

Algorithm 1 Training with SOSX variables

Input: N (number of data points), M (number of neurons), P (number of pieces), breakpoints $[\, a_0, \ldots, a_P \,]$, w_1^*

Output: Optimized parameters w_2, b_1, b_2 and minimized objective $|\nu'|$

Initialize parameters w_2, b_1, b_2.

for $i = 1$ to N **do**

  for $j = 1$ to M **do**

    for $k = 1$ to P **do**

      Solve the model:

      Minimize:

$$|\nu'| = \frac{1}{N} \sum_{i=1}^{N} \left| \left(w_1^* f'(w_2 x + b_2) + b_1 \right) - y_i \right|$$

      Subject to:

$$\nu' \leq |\nu'|, \quad |\nu'| \leq -\nu',$$

$$a_0 \leq w_2, b_1, b_2 \leq a_P, \quad w_2 x + b_2 \in [a_0, a_P],$$

$$u_k = a_k - a_{k-1}, g_k = f'(a_k) - f'(a_{k-1}), \quad f'(w_2 x + b_2) = f(a_0) + \sum_{k=1}^{P} \frac{g_k}{u_k} Y_k$$

$$u_m Y_{m+1} \leq u_{m+1} Y_m, \quad \forall m \in \{1, \ldots, P-1\}, \quad Y_1 \leq u_1, \quad Y_P \geq 0$$

      Branching conditions:

      if $\hat{Y}_{i,j,k} < u_k$ and $\hat{Y}_{k+1} > 0$ **then**

        Branch 1: Set $Y_{i,j,k} = u_k$

        Branch 2: Set $Y_{i,j,k+1} = \cdots = Y_{i,j,P} = 0$

      end if

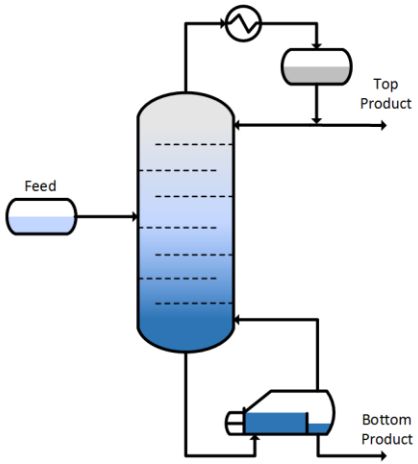
    end for

  end for

end for

Return: Optimized parameters w_2, b_1, b_2 and minimized $|\nu'|$

CASE STUDY

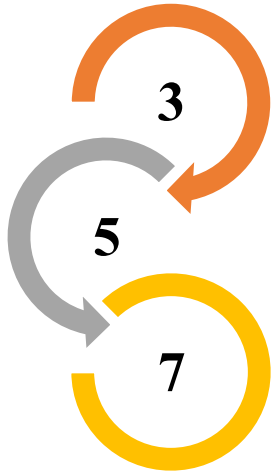


Case study:

- Distillation column with 27 features
- Temperature, pressure,…
- Vapor pressure as output

Number of neurons used:

Number of piecewise linear segments used:

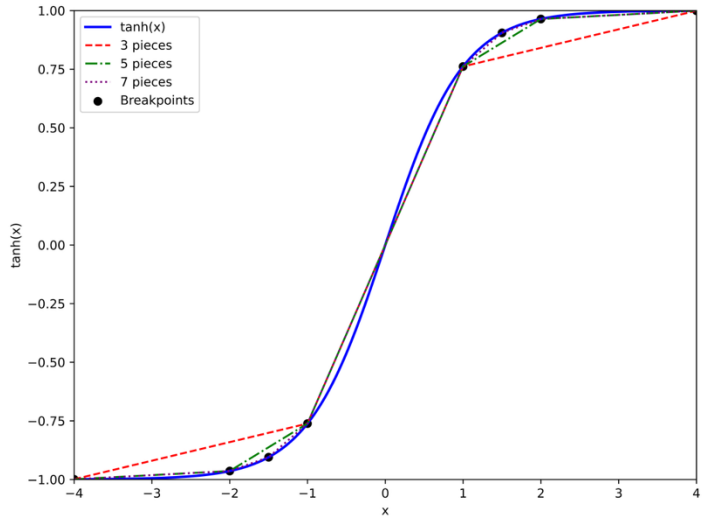


NLP, SOSX, MIP trained FFNs

SOSX CPU times are recorded

MIP does not converge a solution

MIP is stopped at SOSX CPU time



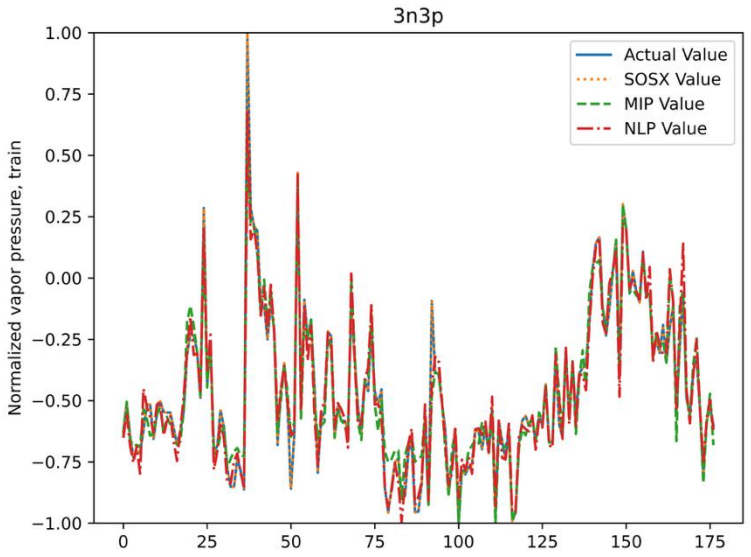
Koksal, E.S., Turkay, M., Aydin, E. (2025). An Efficient Convex Training Algorithm for Feedforward Neural Networks by Utilizing Semi-Continuous Piecewise Linear Formulations.

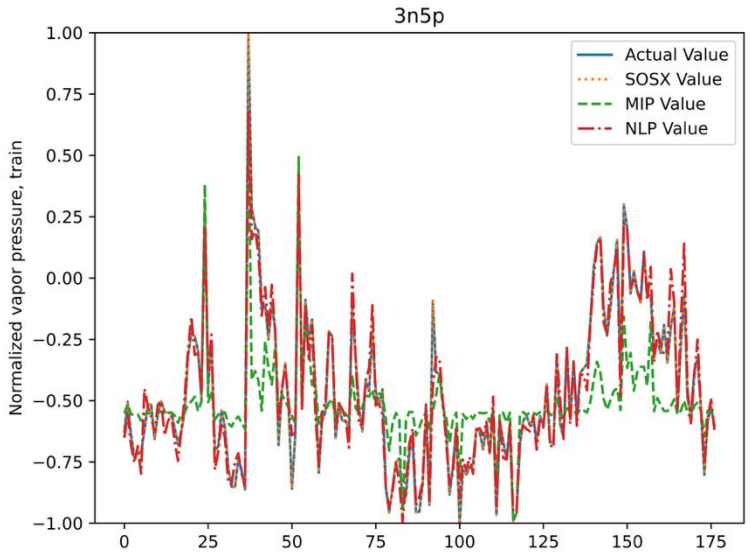
RUN SETTINGS

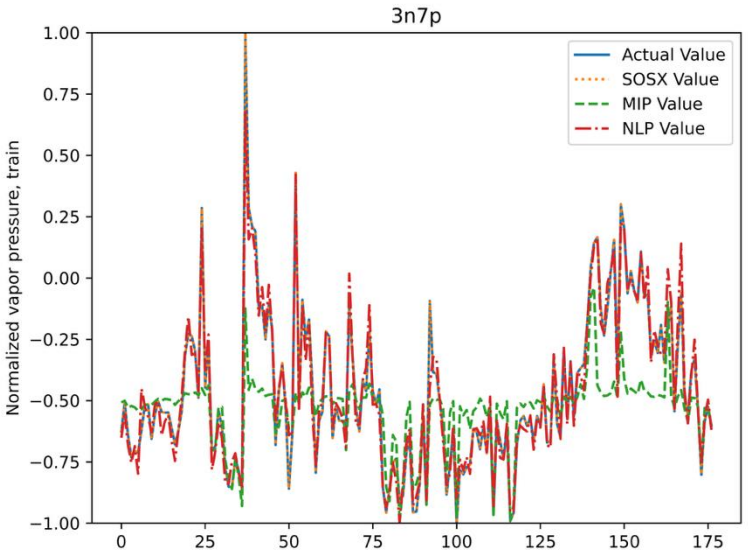
```
solver.options['Cuts'] = 0                  # Disable cuts
solver.options['Heuristics'] = 0             # Disable heuristics
solver.options['Presolve'] = 0               # Disable presolve
solver.options['MIPFocus'] = 1               # Focus on optimality
solver.options['VarBranch'] = 0              # Use classical branching
result=solver.solve(model)
```

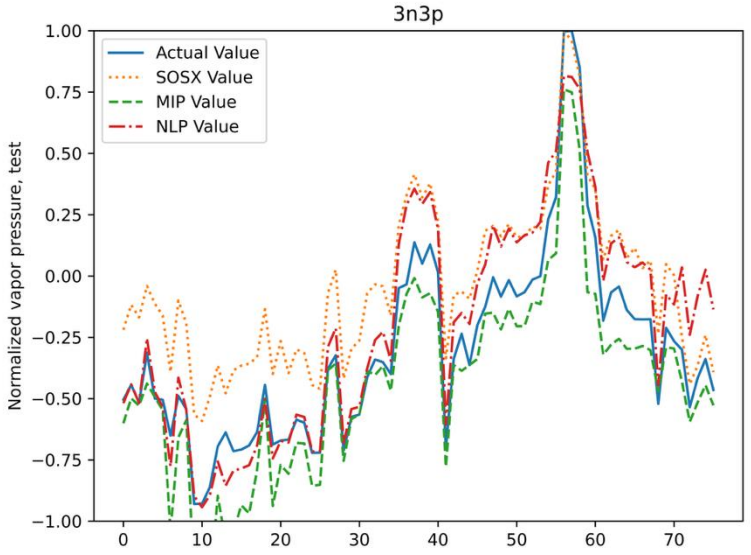
- SOSX does not use heuristics
- Changing MIP focus disturbs test performance
- "NLP training," uses the Adam optimizer (Kingma & Ba, 2014) and original hyperbolic tangent activation function in TensorFlow's Keras (Chollet et al, 2015) framework with 100 epochs
- "MIP training" incorporates binary variables into the problem structure in a standard fashion and uses standard B&B algorithm
- $\boldsymbol{n}$: number of neurons at the hidden layer, $\boldsymbol{p}$: number of piecewise linear segments
- 12000 seconds of CPU time limit

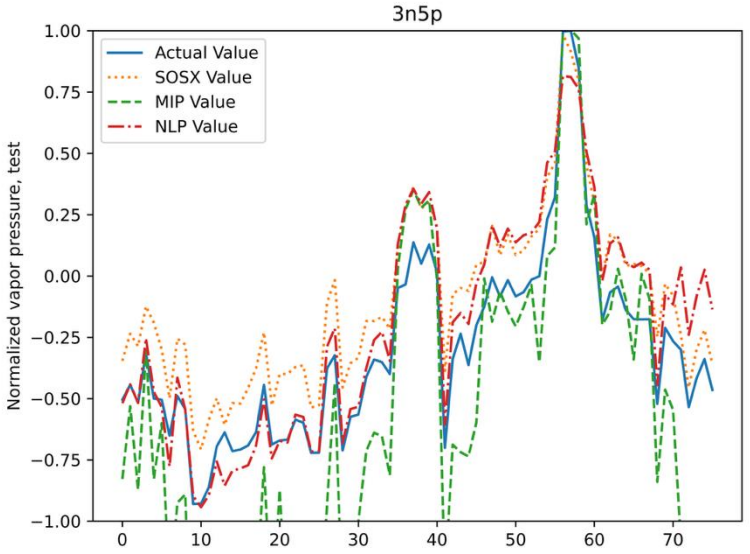
RESULTS: 3 neurons

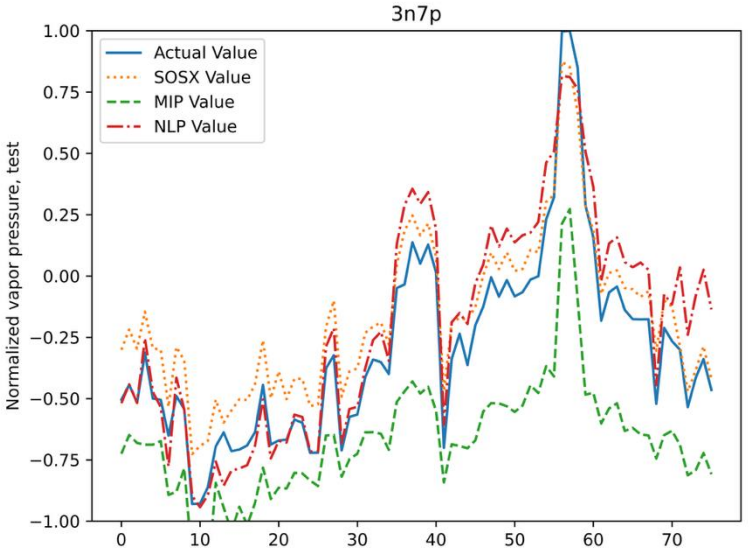




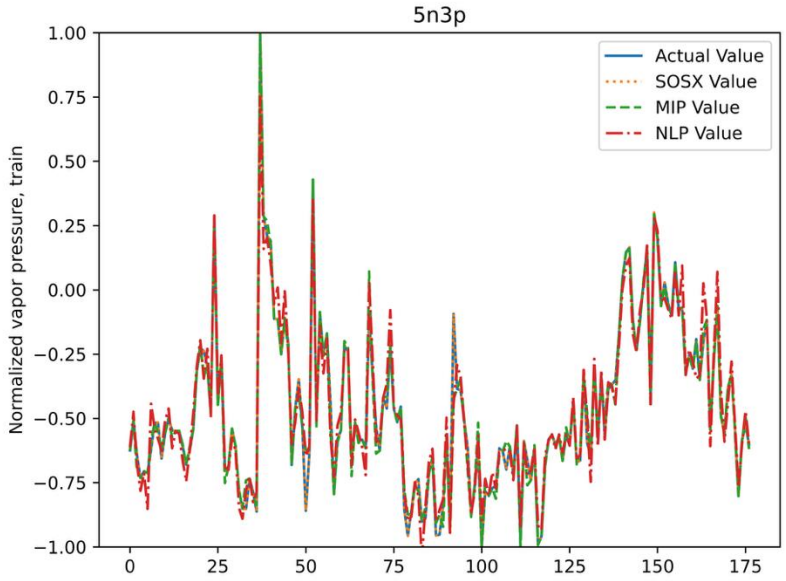


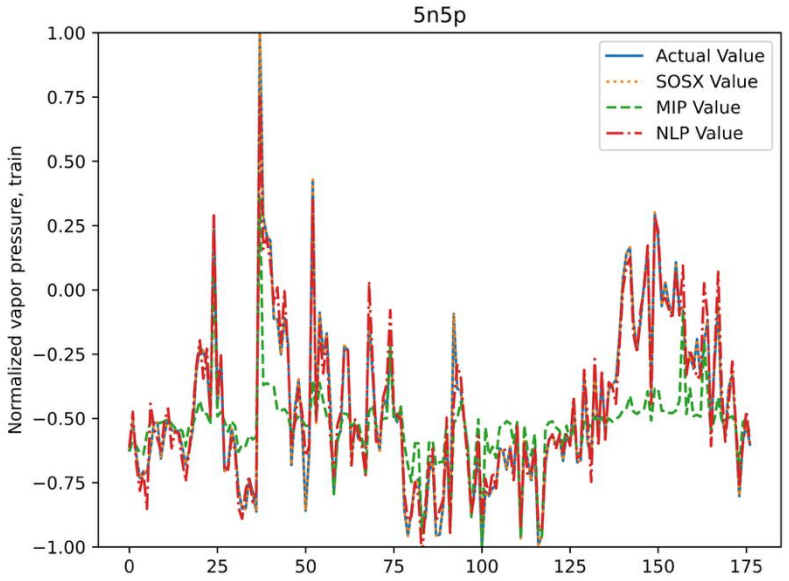


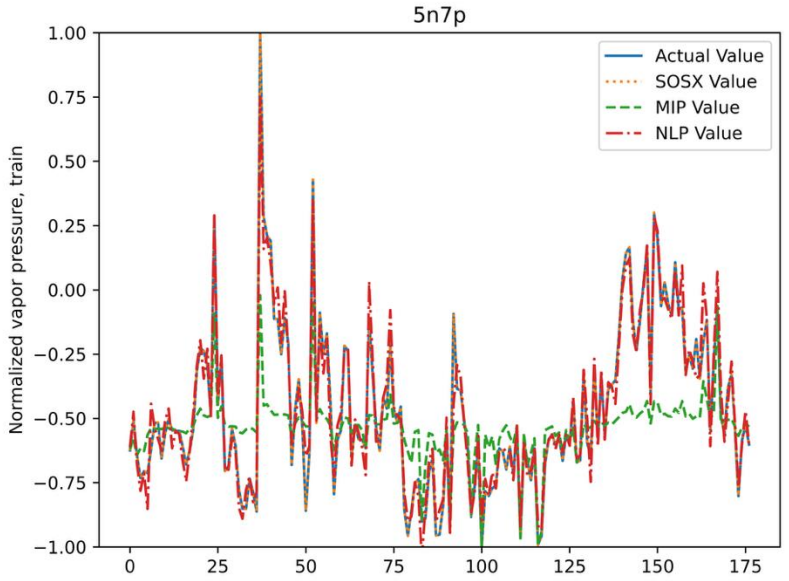


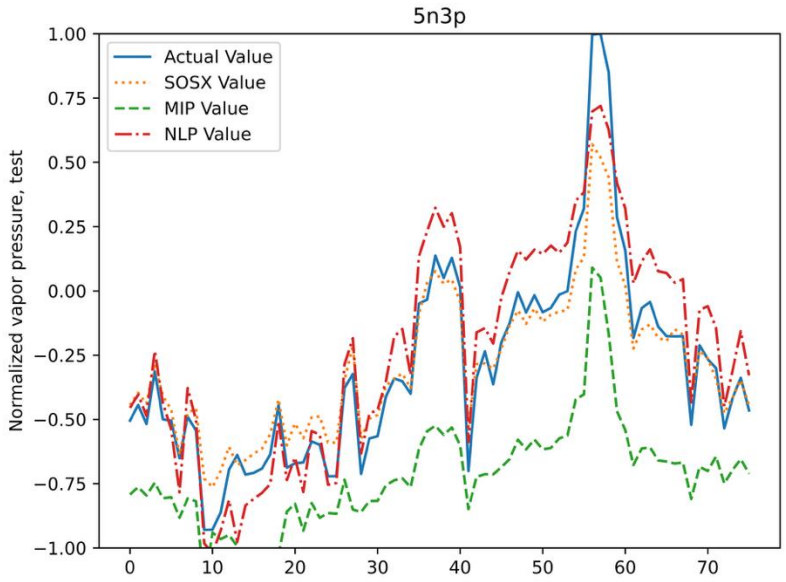


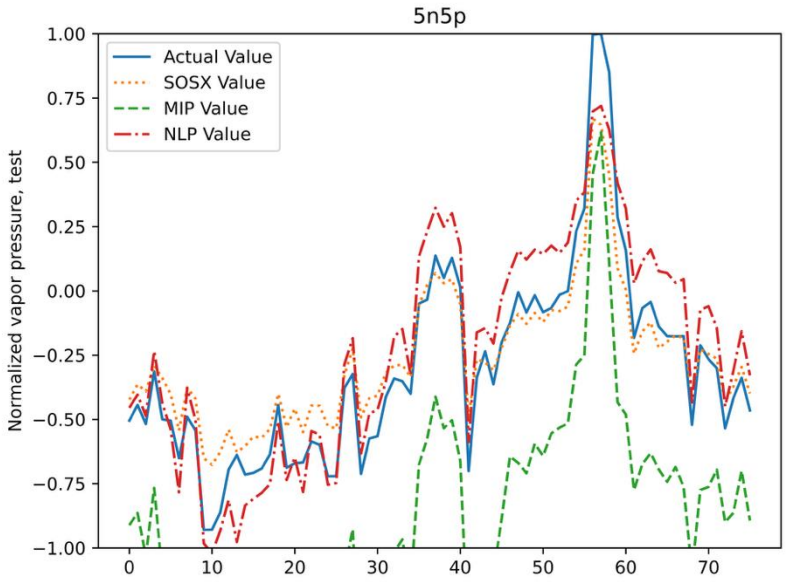
RESULTS: 5 neurons

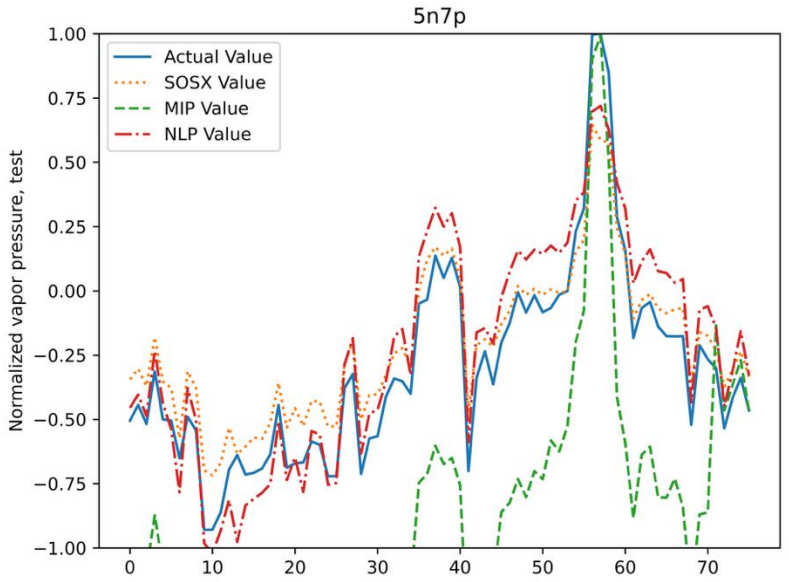




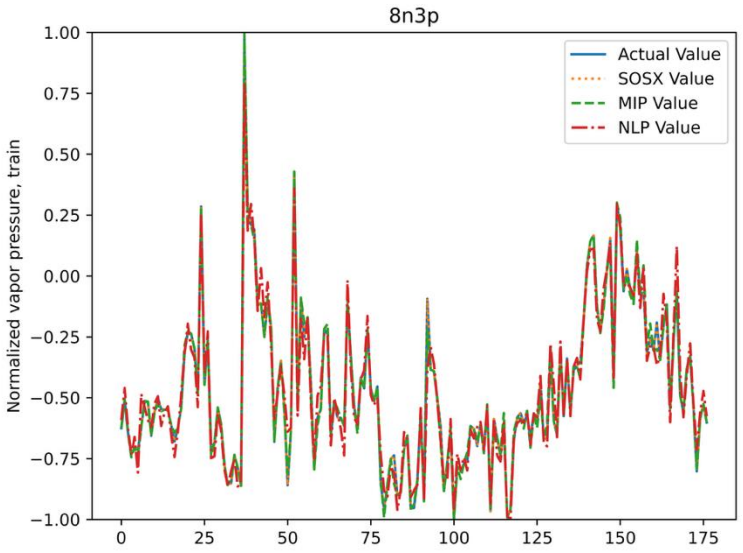


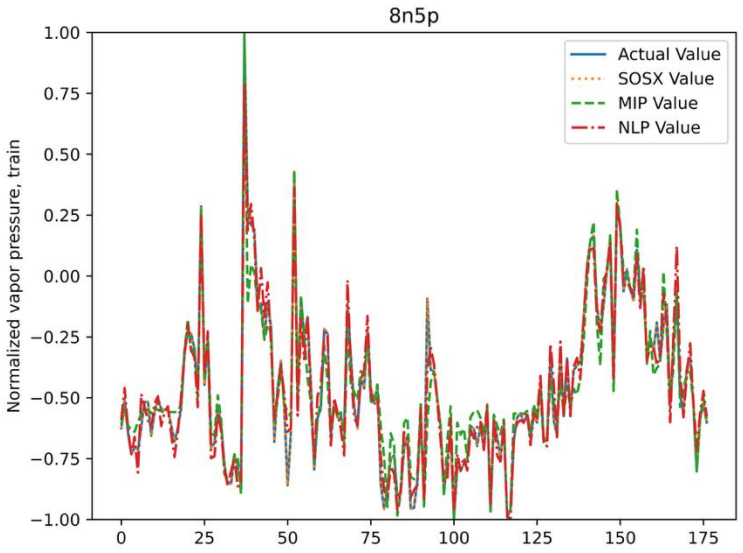


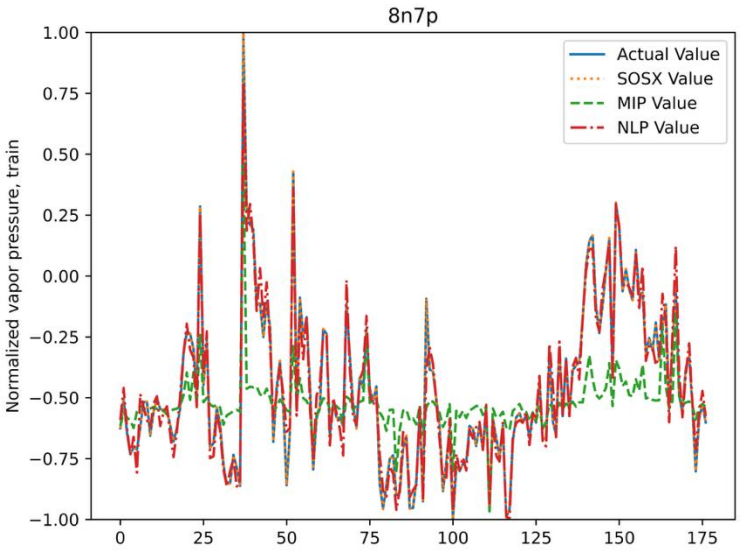


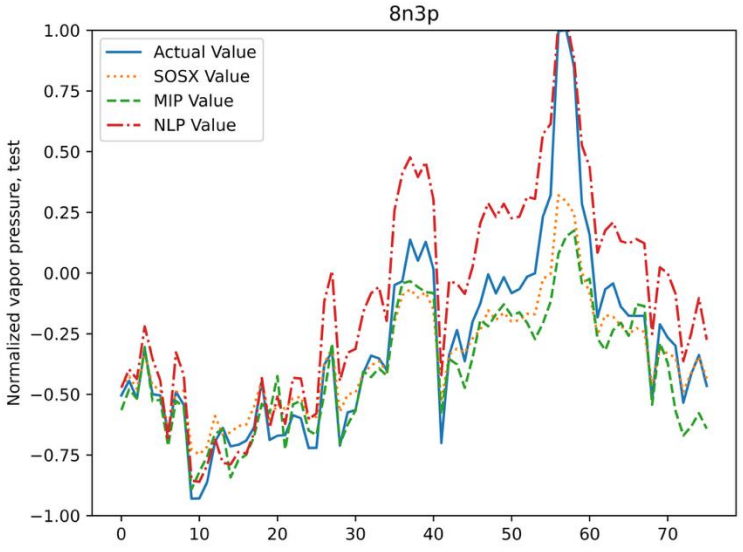


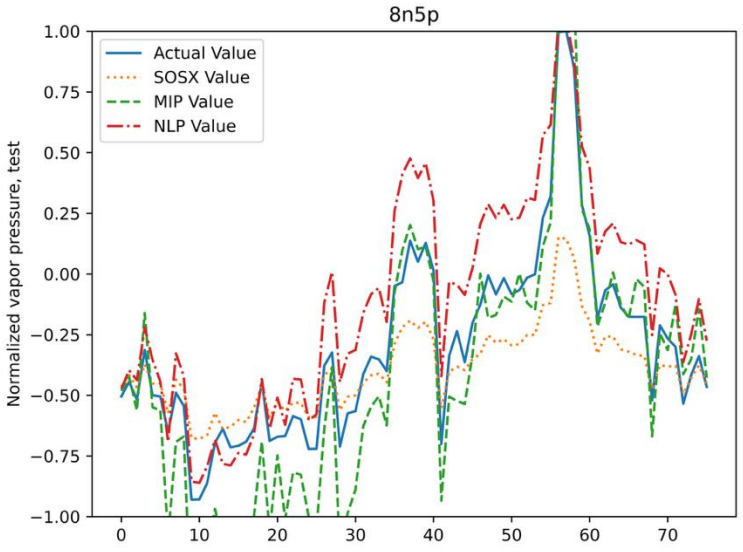
RESULTS: 8 neurons

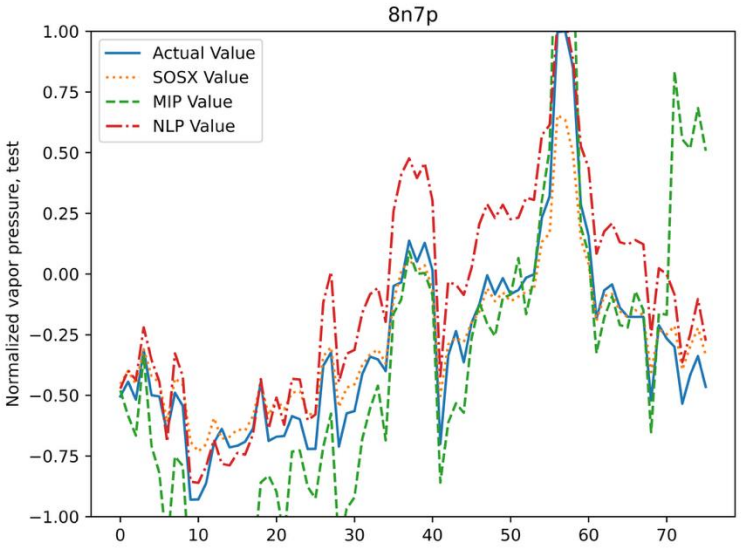




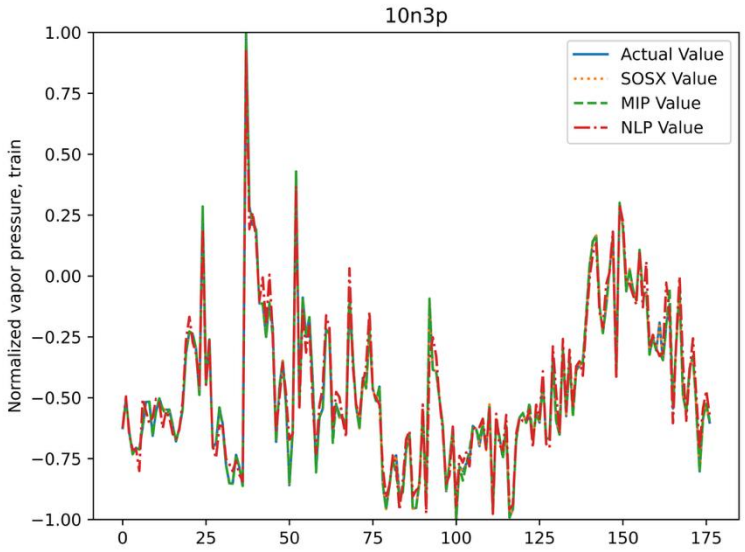


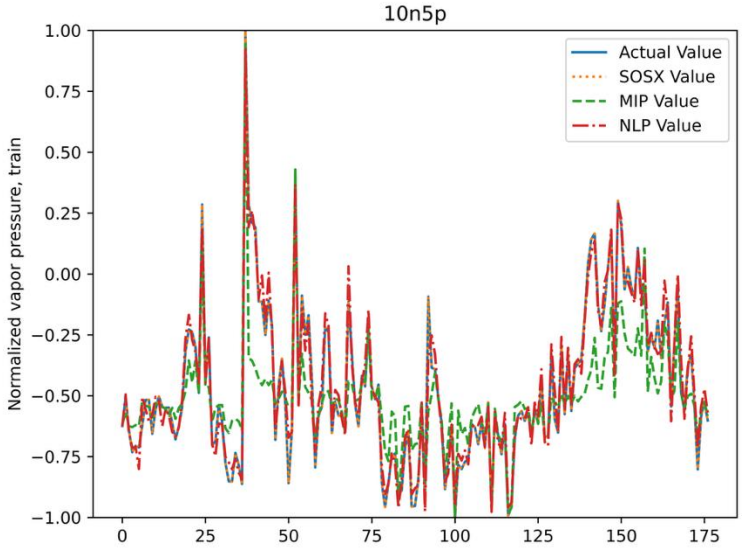


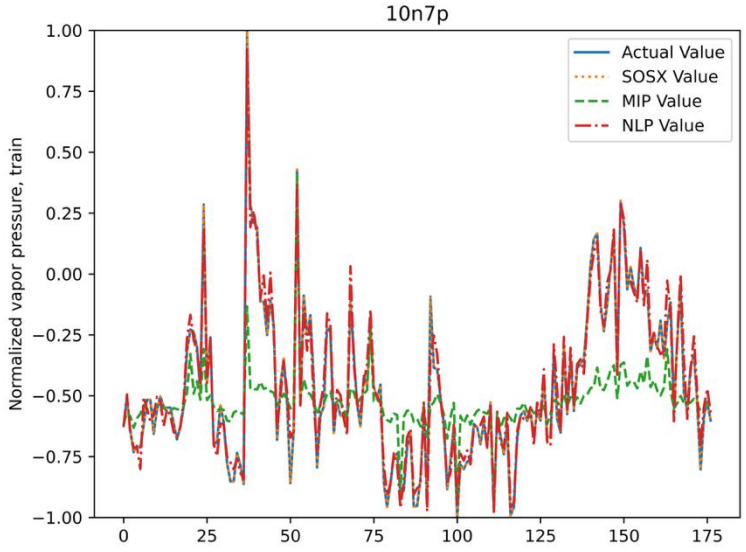


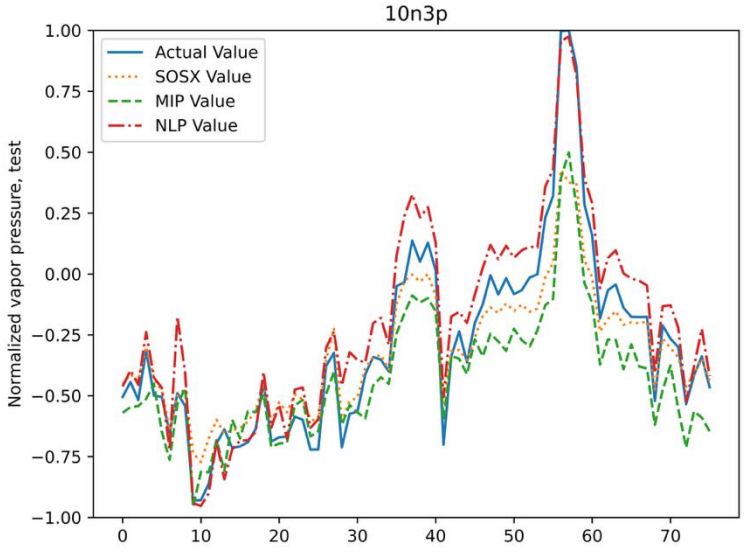


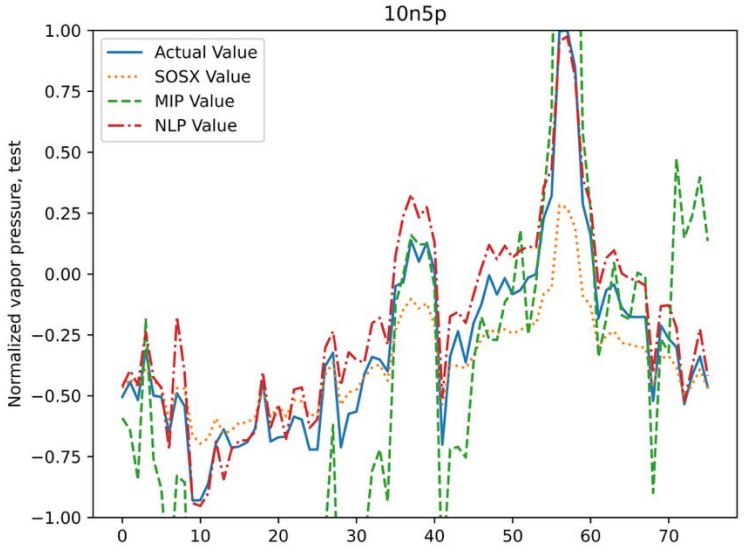
RESULTS: 10 neurons

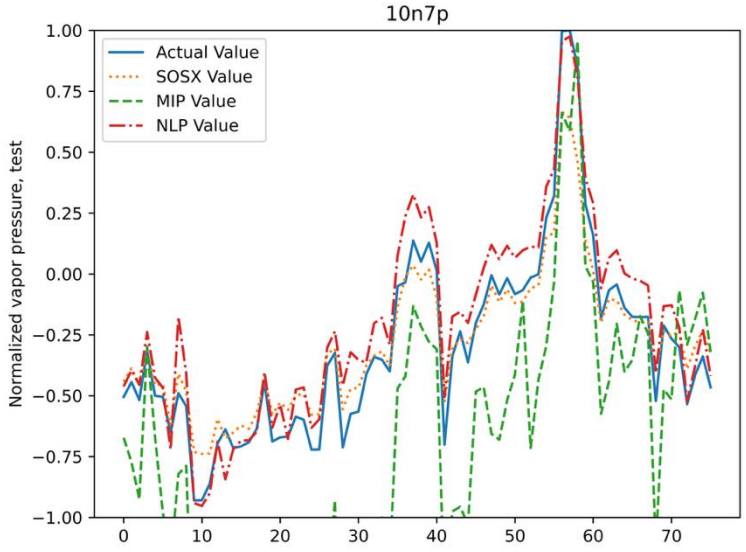




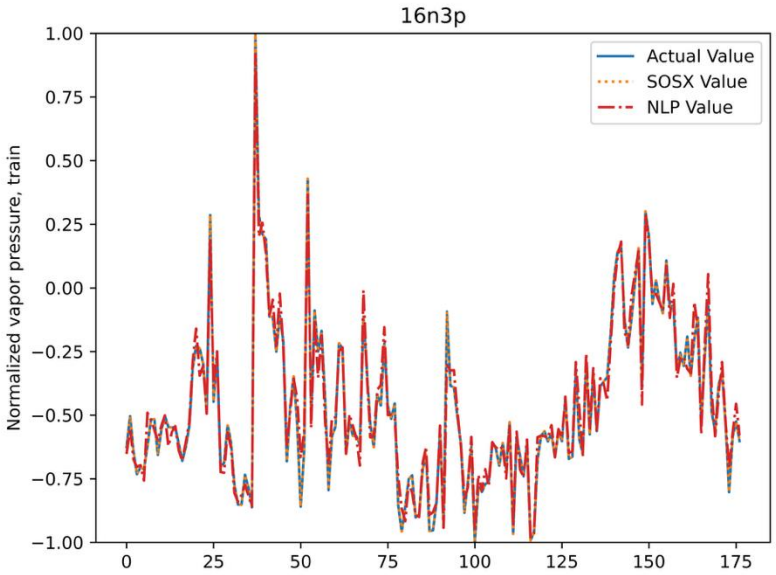


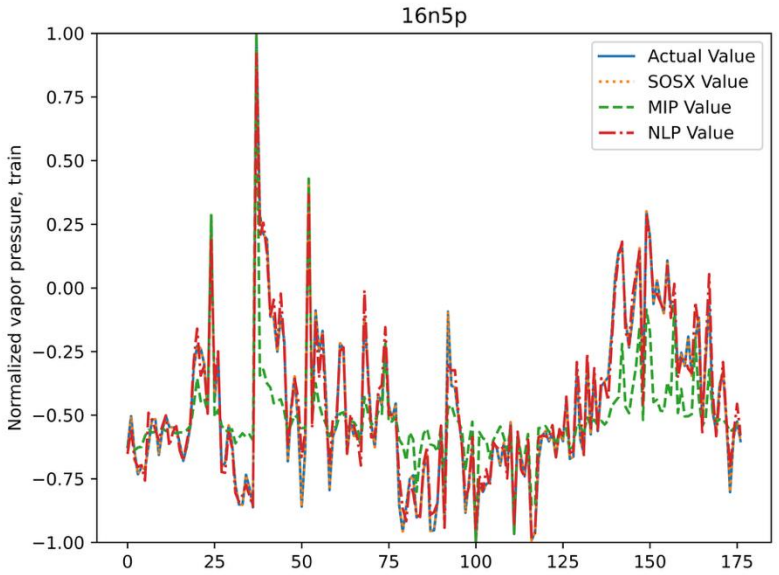


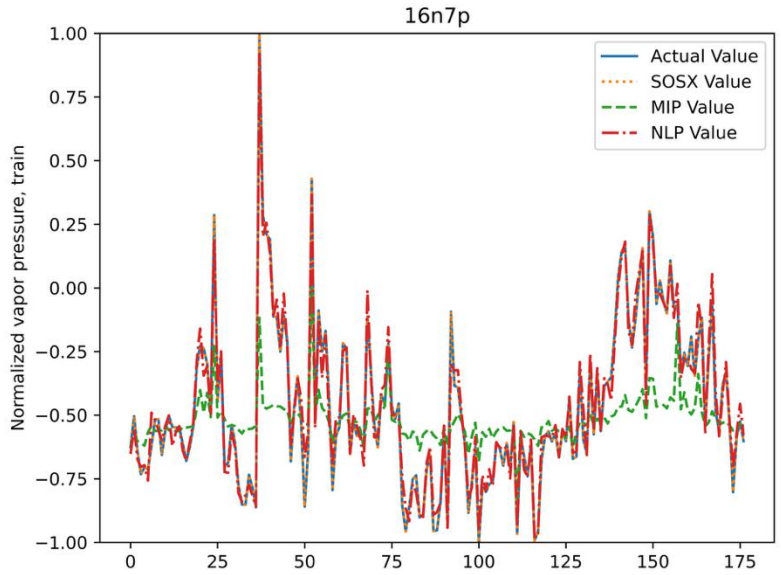


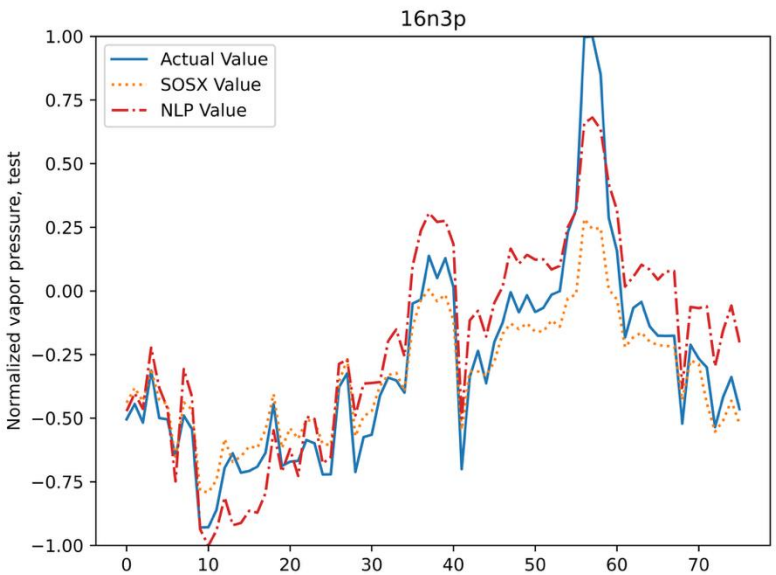


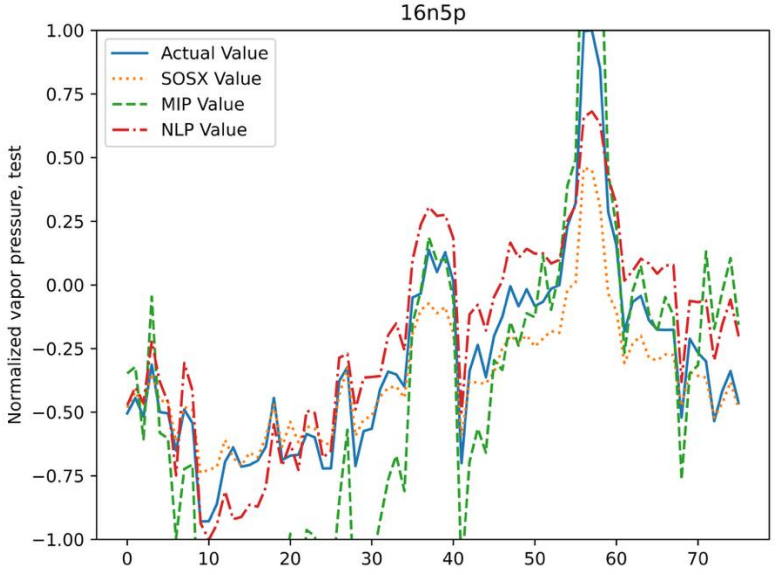
RESULTS: 16 neurons

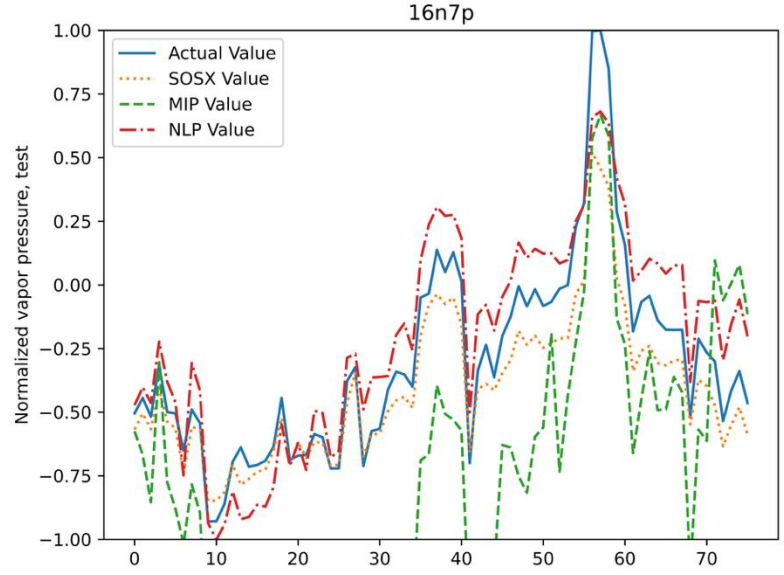




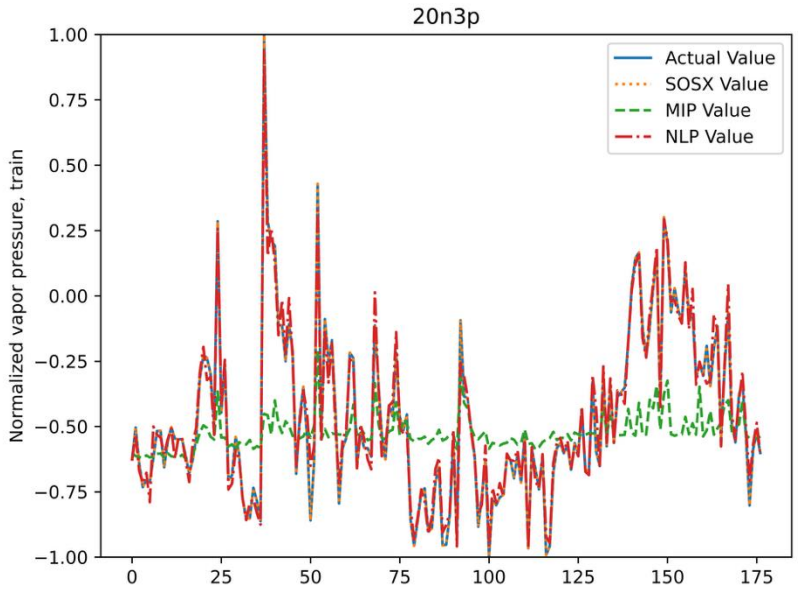


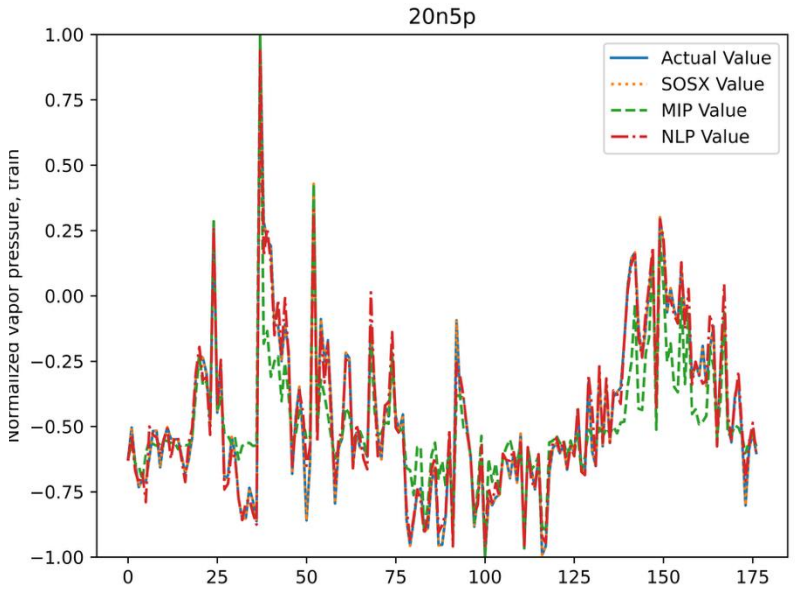


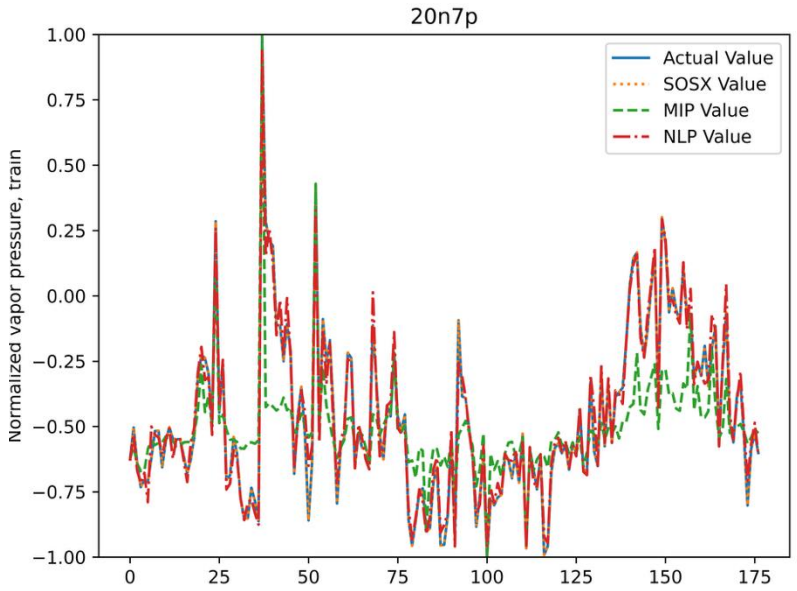


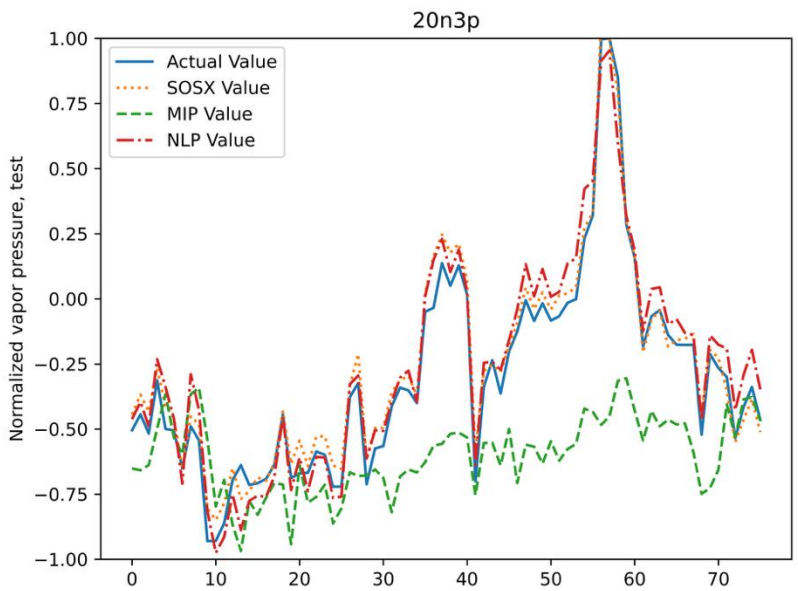


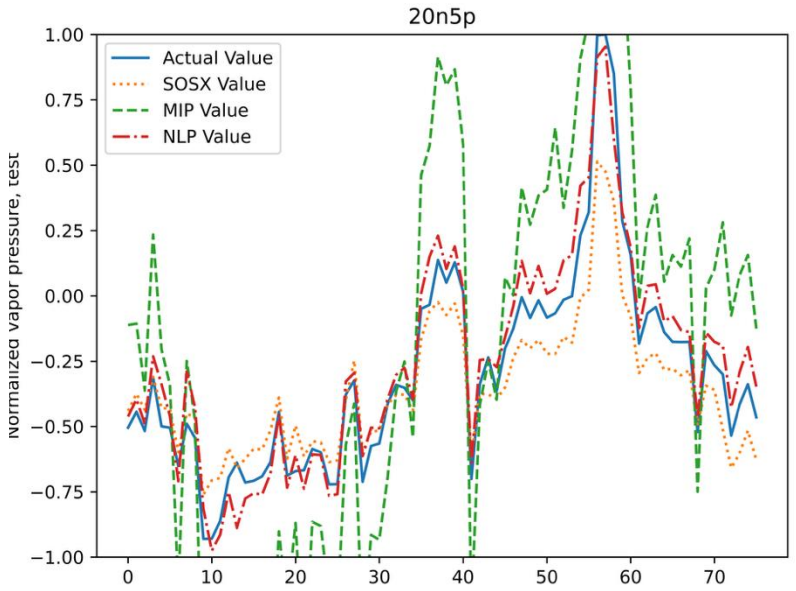
RESULTS: 20 neurons

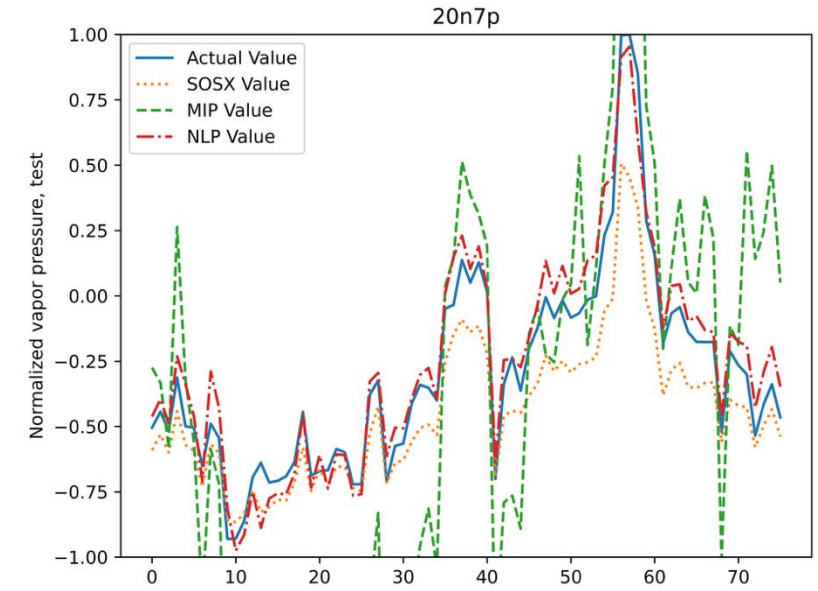




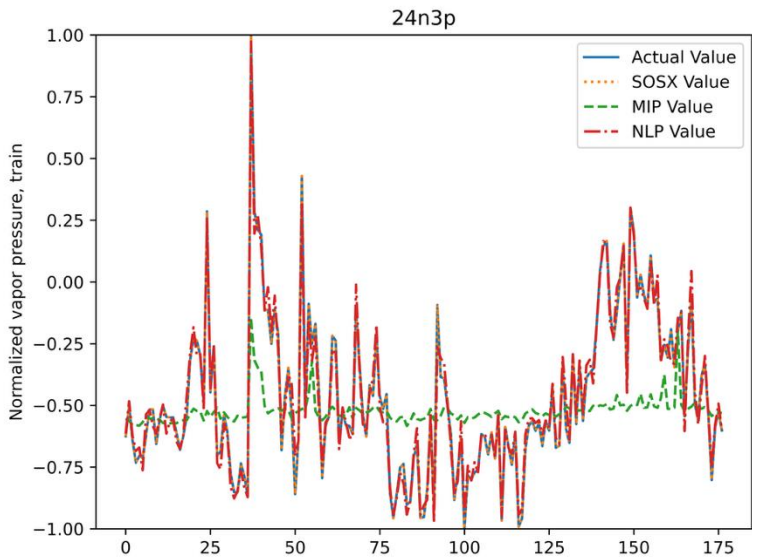


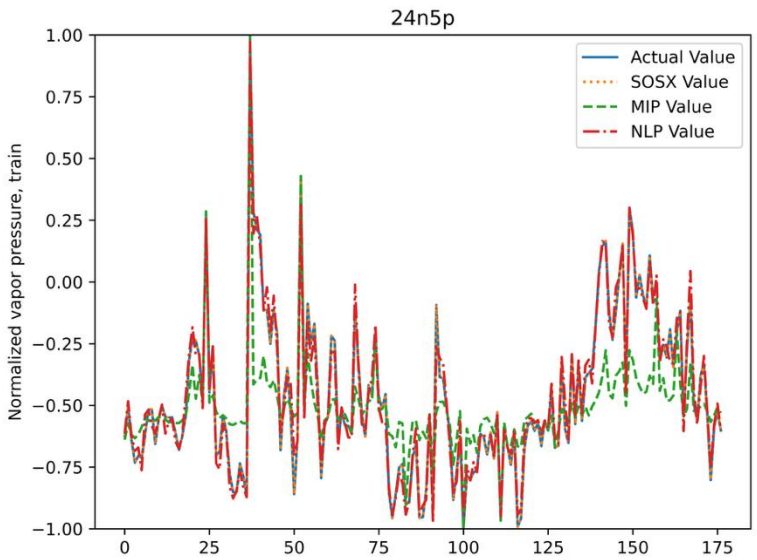


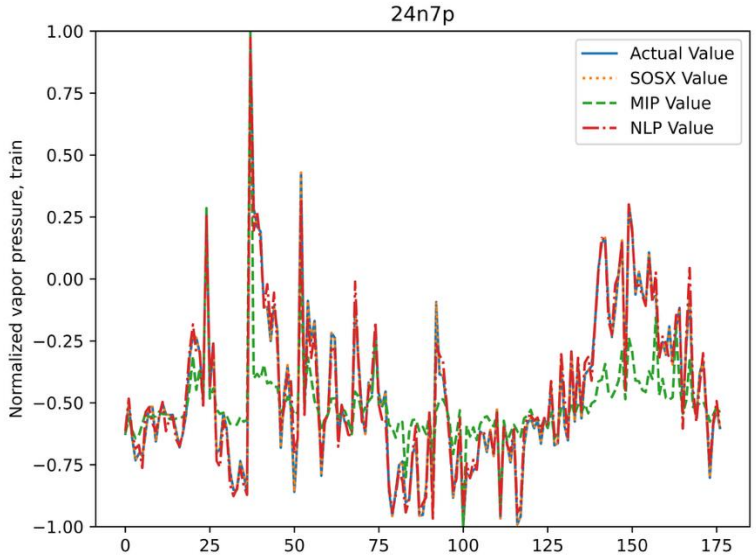


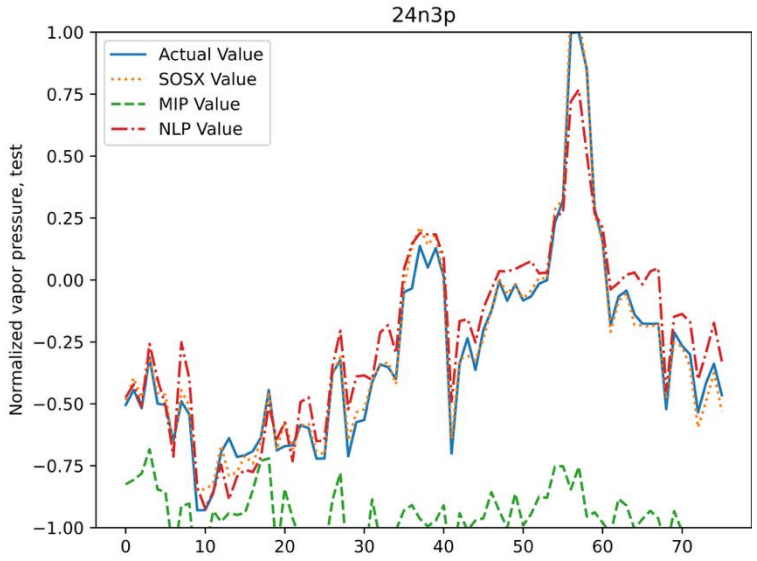


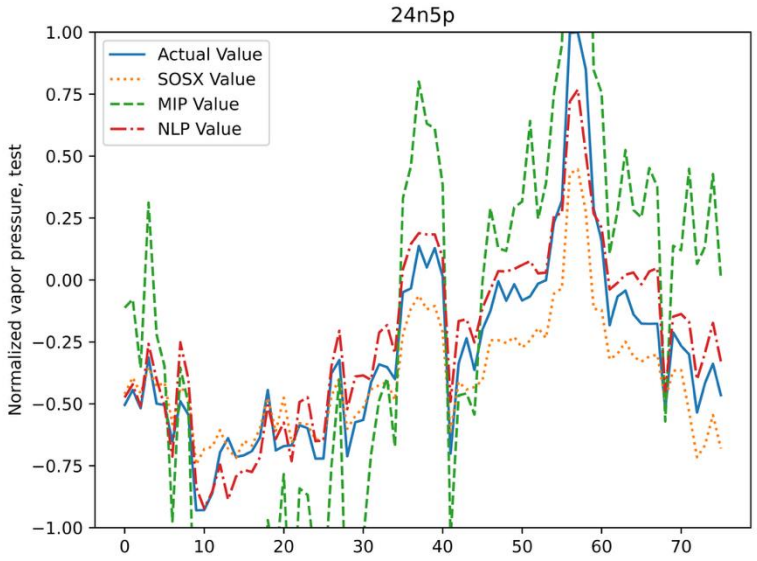
RESULTS: 24 neurons

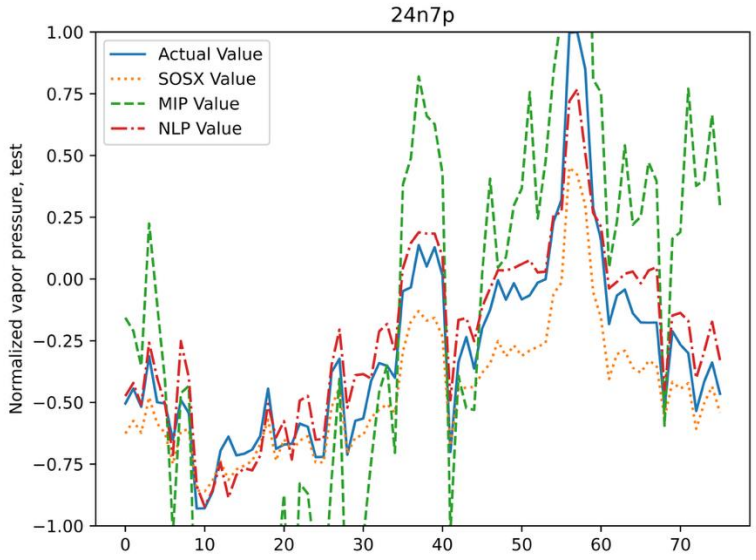




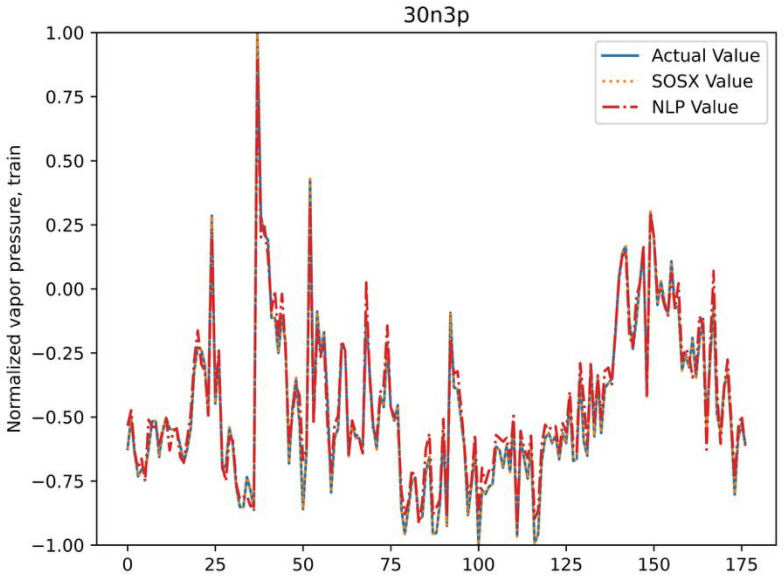


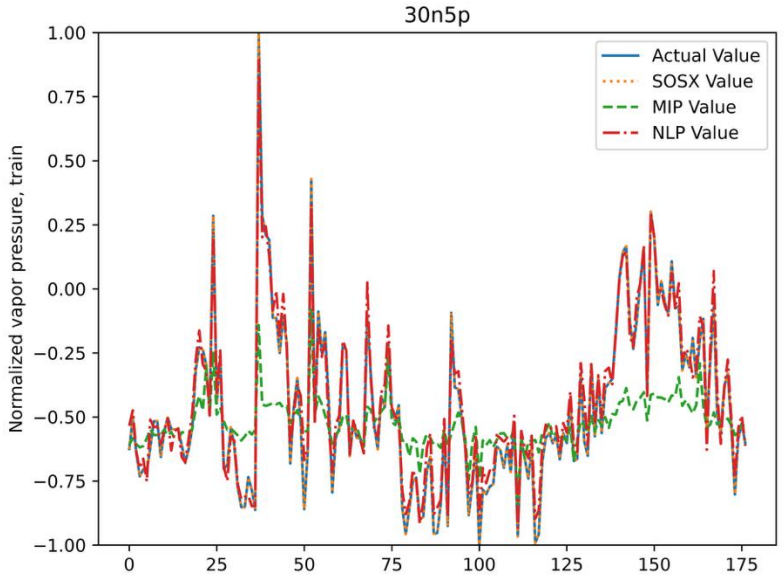


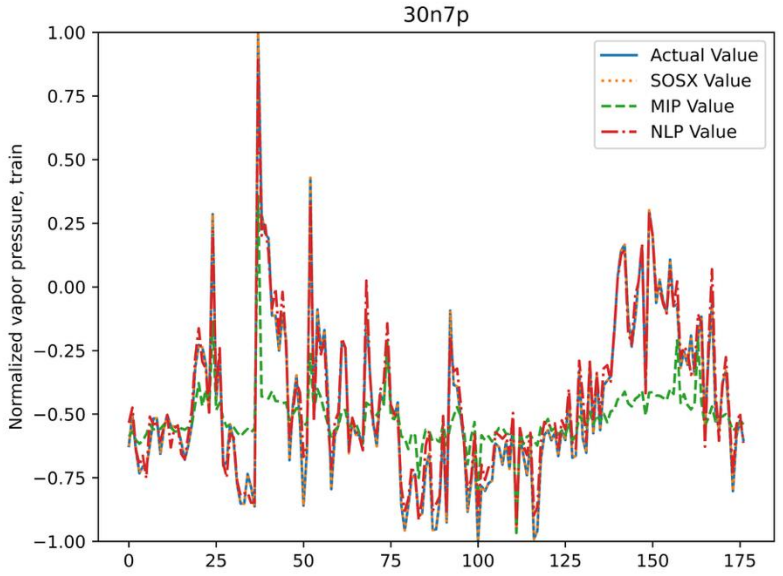


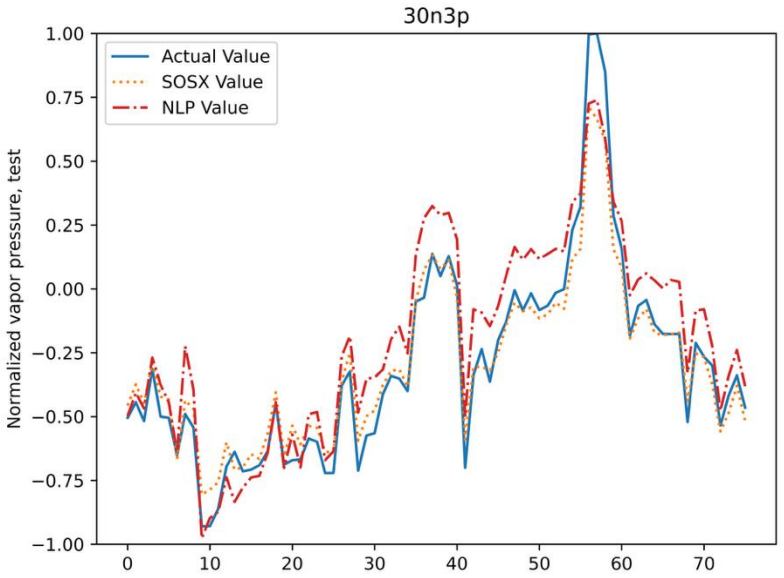


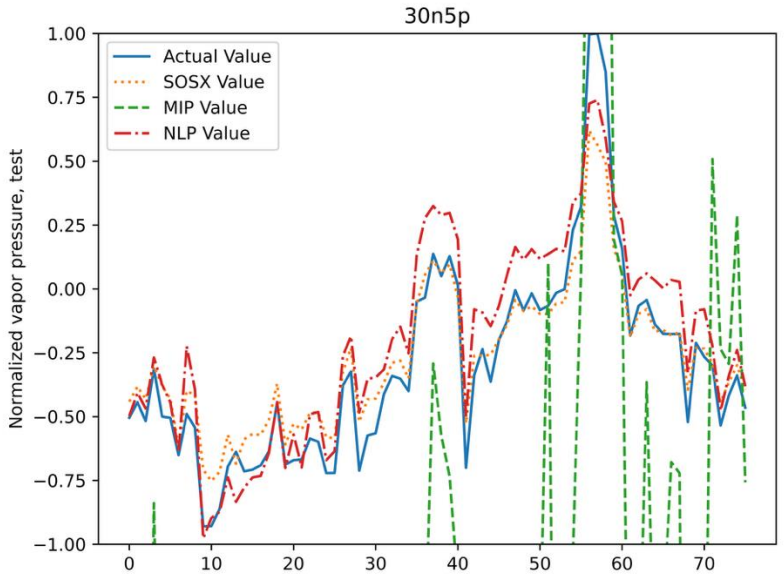
RESULTS: 30 neurons

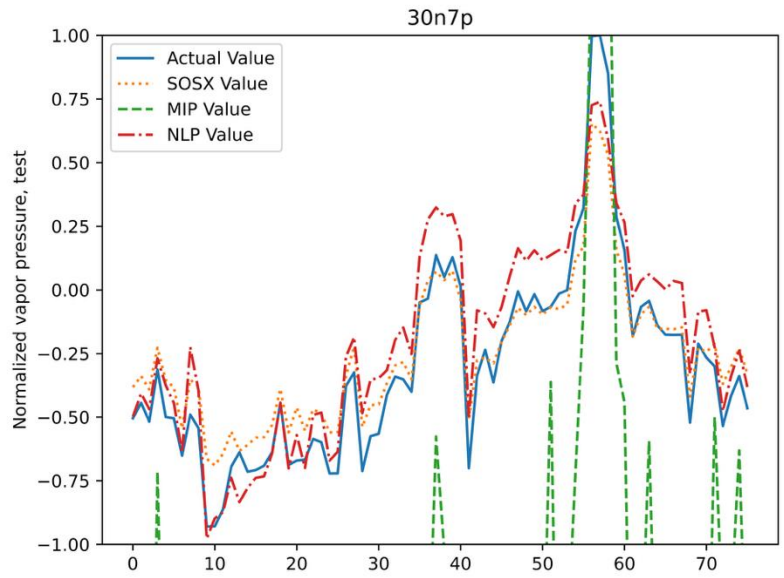




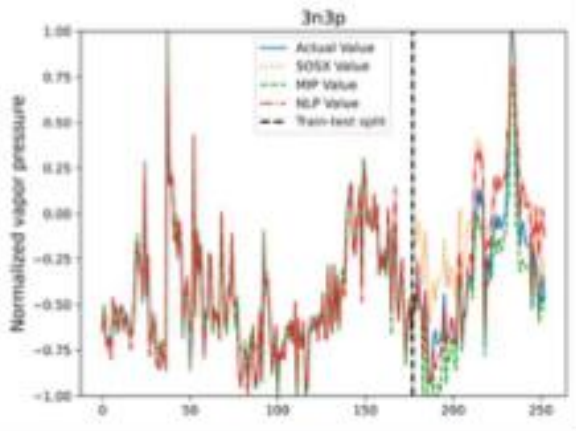




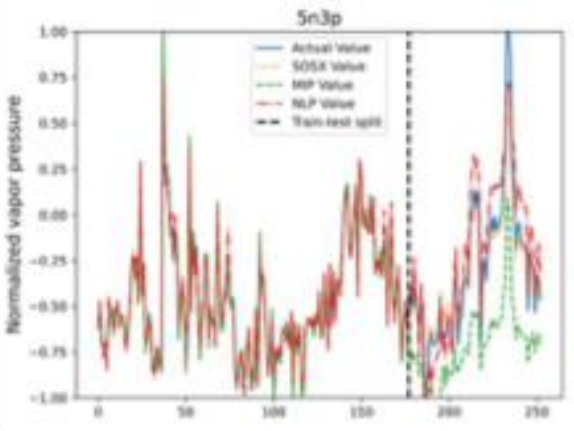




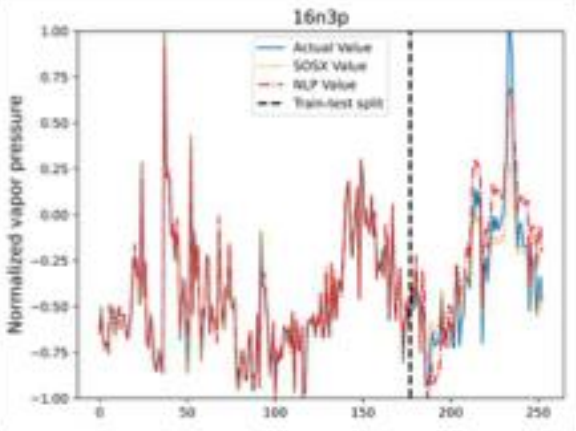
RESULTS: Training & Testing (p=3)



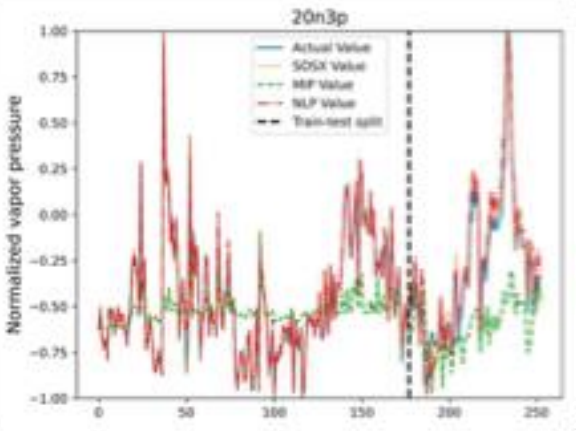
(a)



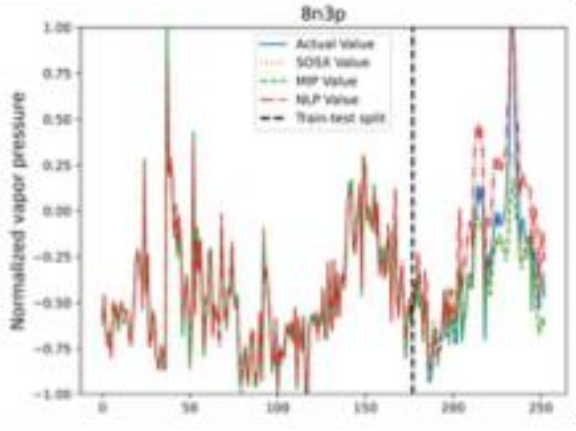
(b)



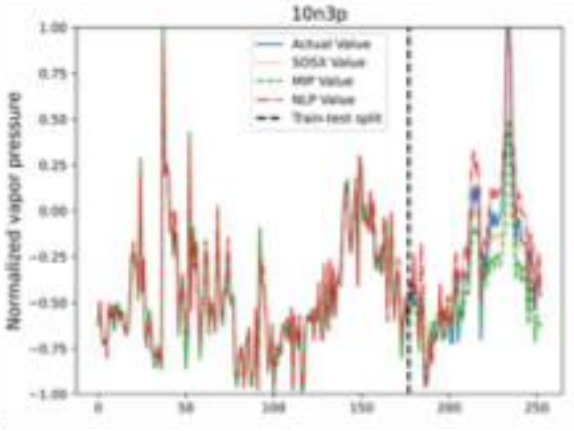
(e)



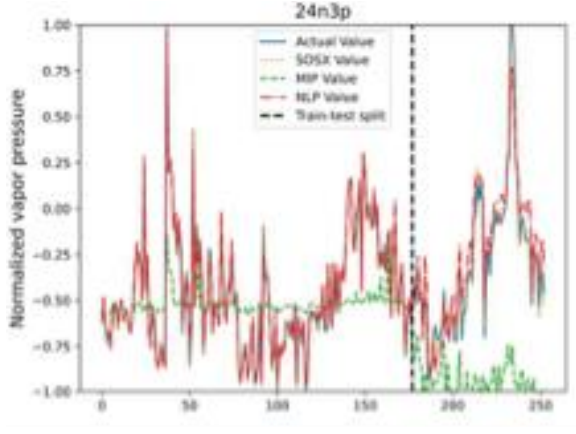
(f)



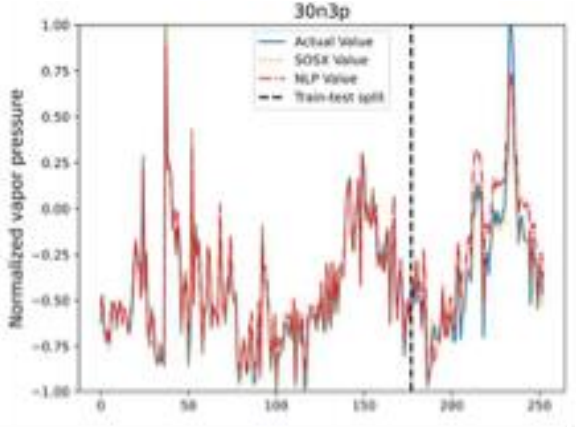
(c)



(d)

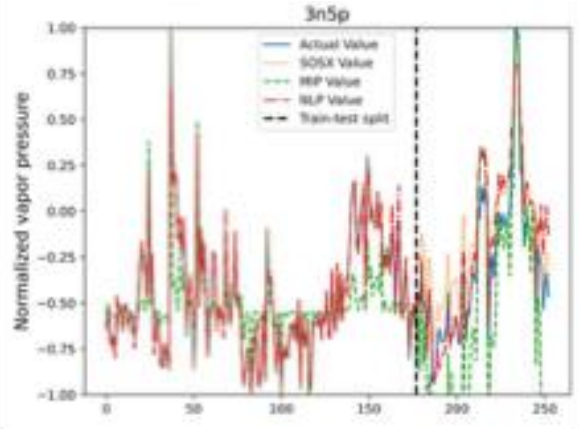


(g)

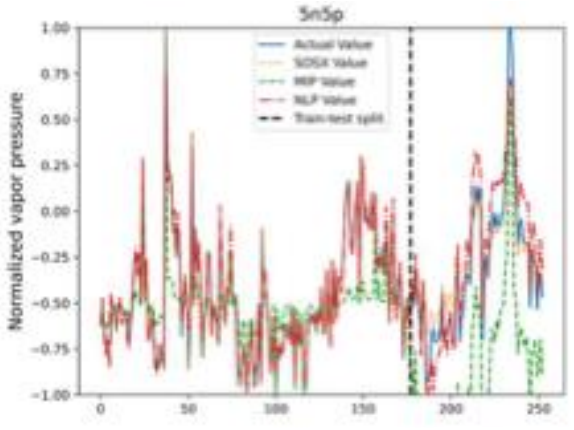


(h)

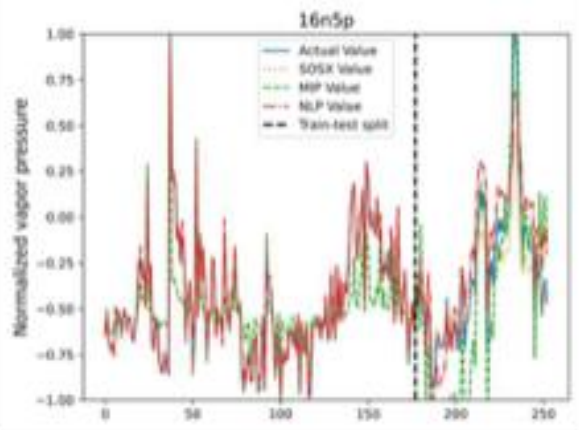
RESULTS: Training & Testing (p=5)



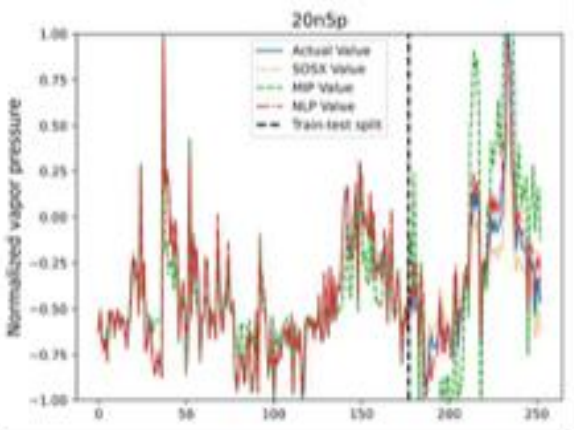
(a)



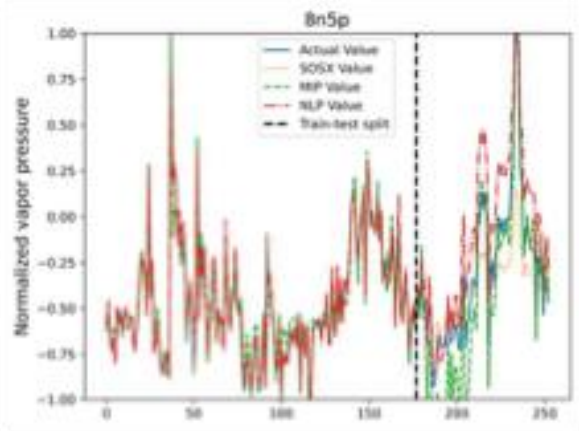
(b)

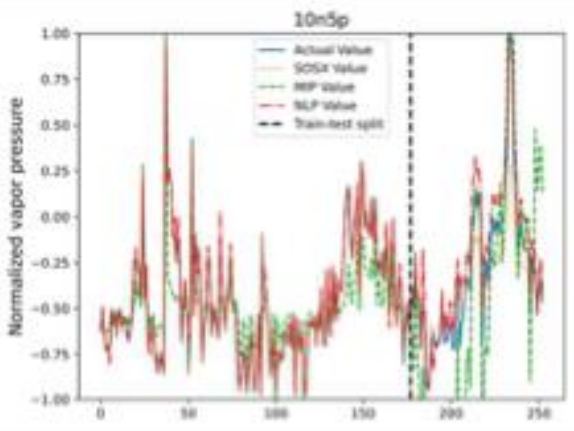


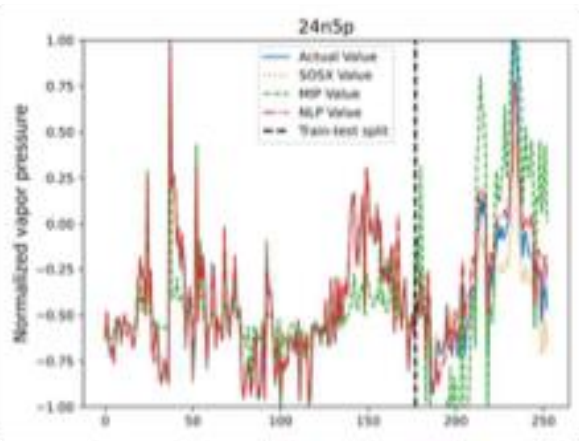
(c)



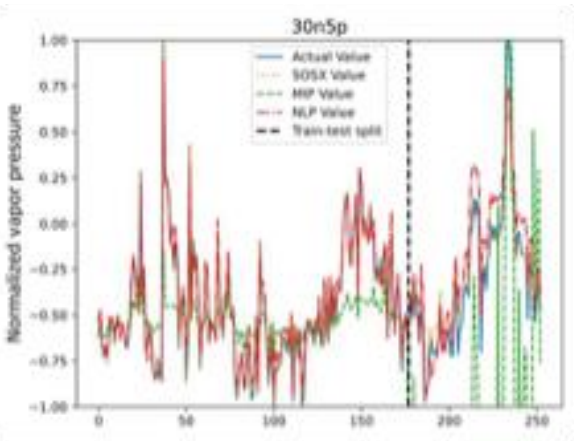
(f)





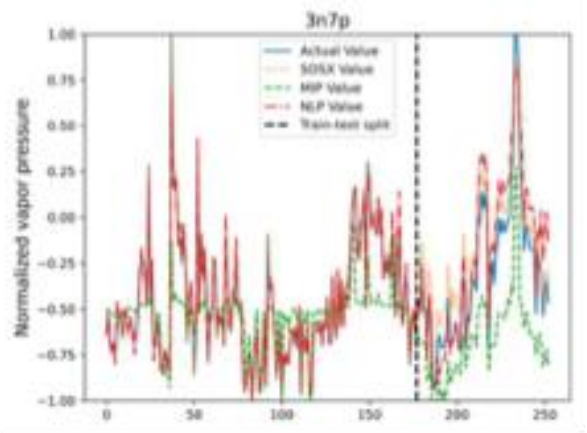


(g)

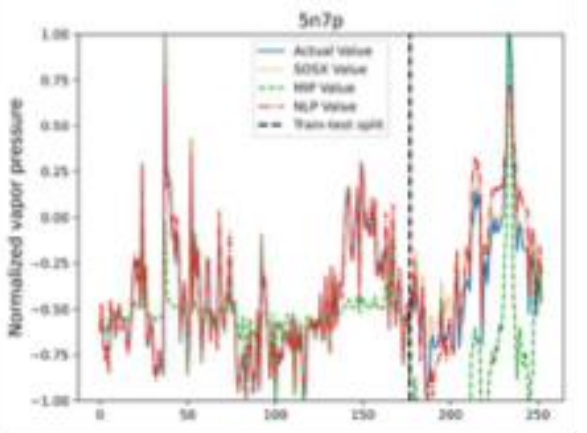


(h)

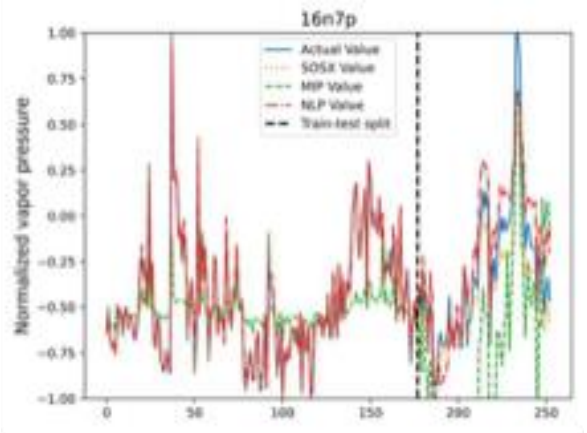
RESULTS: Training & Testing (p=7)



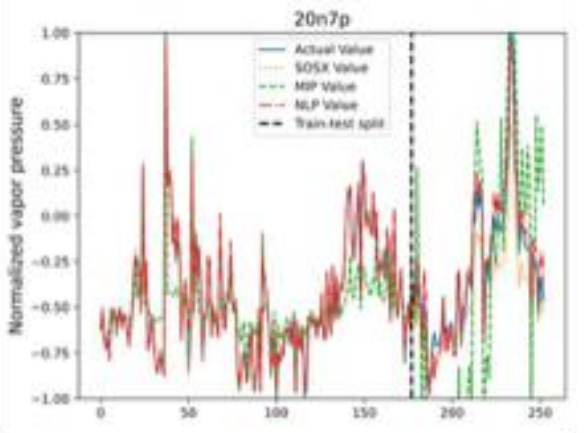
(a)



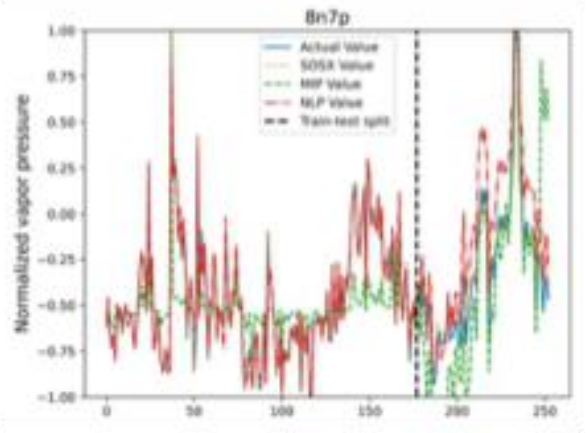
(b)



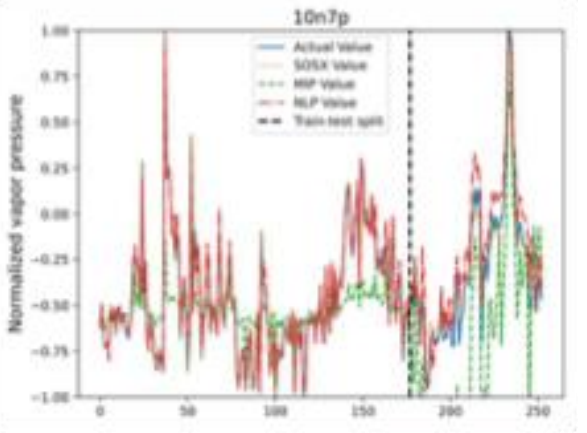
(c)



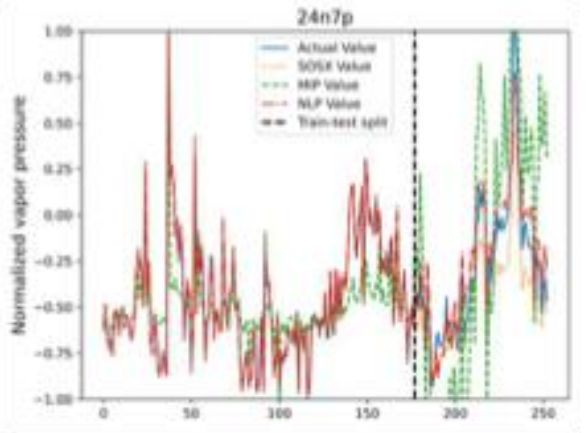
(f)



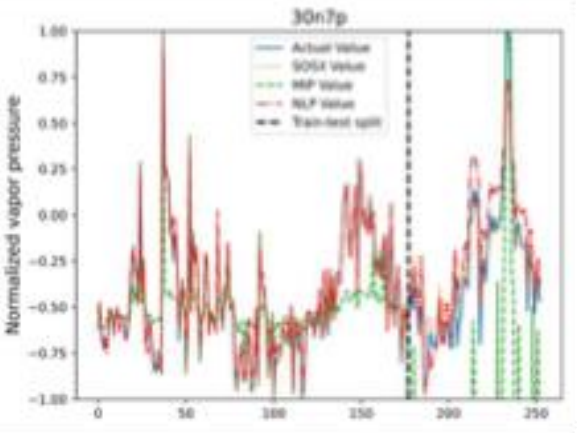
(c)



(d)



(g)



(h)

RESULTS: MAE

			3 pcs (p=3)		5 pcs (p=5)		7 pcs (p=7)	
	MAE	NLP	MIP	SOSX	MIP	SOSX	MIP	SOSX
n=3	MAE train	0.0454	0.0502	0	0.1665	0	0.1673	0
	MAE test	0.1337	0.1390	0.2527	0.3824	0.1970	0.3723	0.1402
n=5	MAE train	0.0488	0.0211	0	0.1616	0	0.1770	0
	MAE test	0.1367	0.4300	0.0837	0.6425	0.1980	0.6726	0.1194
n=8	MAE train	0.0459	0.0136	0	0.0596	0	0.1844	0
	MAE test	0.2016	0.1340	0.1144	0.1772	0.1679	0.2755	0.0841
n=10	MAE train	0.0436	0.0049	0	0.1296	0	0.1804	0
	MAE test	0.1087	0.1560	0.0974	0.4689	0.1439	0.5367	0.0851
n=16	MAE train	0.0406	-	0	0.1484	0	0.1895	0
	MAE test	0.1522	-	0.1112	0.3220	0.1178	0.6054	0.1206
n=20	MAE train	0.0375	0.2088	0	0.1044	0	0.1457	0
	MAE test	0.0791	0.3382	0.0539	0.5243	0.1267	0.5625	0.1491
n=24	MAE train	0.0371	0.2171	0	0.1477	0	0.1547	0
	MAE test	0.1052	0.6602	0.0360	0.4758	0.1490	0.5211	0.1609
n=30	MAE train	0.0466	-	0	0.1799	0	0.1760	0
	MAE test	0.1320	-	0.0649	3.0313	0.0883	2.8112	0.0997

- Main Observations:

Training:

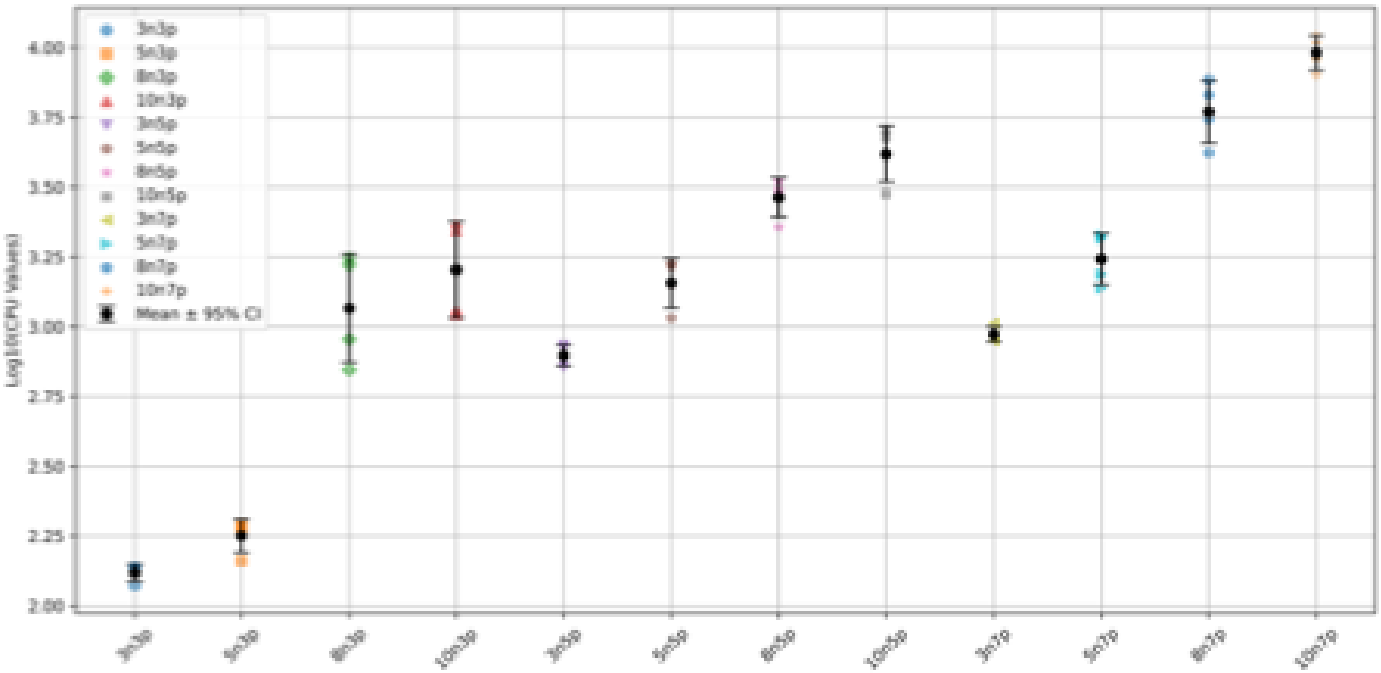
- ✓ SOSX achieves 0 error at the training stage → they are solved to optimality
- ✓ MIP could not solve any of the training problems to optimality
- ✓ NLP was stuck with locally optimal solutions in all cases

Testing:

- ✓ SOSX is superior to MIP in all cases
- ✓ SOSX is better than NLP in 7 out of 8 cases
- ✓ NLP is better only when $n = 3$

RESULTS: CPU Time on Benchmark Problems

# of segments	Set	n=3	n=5	n=8	n=10
3 pcs	Set 1 (Distillation tower)	118	145	903	1102
	Set 2 (Wastewater treatment)	135	187	699	1151
	Set 3 (US cancer data)	138	193	1666	2346
	Set 4 (Boston house prices)	135	193	1763	2208
5 pcs	Set 1 (Distillation tower)	857	1073	2281	4127
	Set 2 (Wastewater treatment)	754	1421	3316	5050
	Set 3 (US cancer data)	840	1679	3036	2994
	Set 4 (Boston house prices)	713	1656	3128	4779
7 pcs	Set 1 (Distillation tower)	888	2074	4191	11103
	Set 2 (Wastewater treatment)	924	1372	5554	7993
	Set 3 (US cancer data)	1030	2124	6780	9080
	Set 4 (Boston house prices)	926	1544	7683	10439

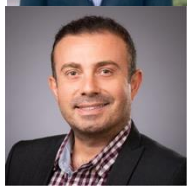


CONCLUSIONS

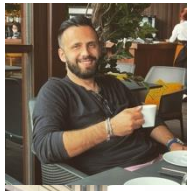
- Training for AI is a major resource (GPU time, energy and water) consumer
- Training models and algorithms are not very effective in most state-of-the-art systems
- Algorithmic efficiency needs to be addressed
- Proposing a SOSX based algorithm for training ANNs
  - ✓ SOSX approach for quasi-convex training neural networks using piecewise linear approximations of activation functions
  - ✓ consistently achieves almost zero training error while avoiding overfitting, showcasing a striking balance between computational efficiency and model accuracy
  - ✓ Regular NLP-based training, while relying on standard optimization frameworks and providing reasonable test performance, is generally outperformed by SOSX, particularly when higher piecewise approximations are employed to better capture activation function behavior
  - ✓ Unlike MIP-based training, which exhibits exponential increase in CPU time as problem size increases, SOSX demonstrates polynomial increase

ACKNOWLEDGEMENTS

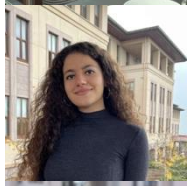
Erdal Aydın
Asst.Prof., Department of Chemical & Biological Engineering
Koc University, Istanbul, Türkiye



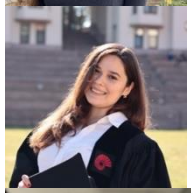
Dr. Murat Küçükvar
Professor of Sustainable Business, Daniels College of Business
University of Denver, USA



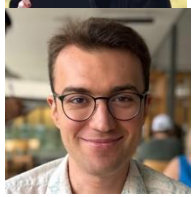
Dr. Nuri C. Onat
Assoc.Prof., Qatar University, Doha, Qatar



Ece Koksal
Research Associate, TUPRAS Research Center, Kocaeli, Türkiye



İpek Pehlivan
Research Asst., Koc University, İstanbul, Türkiye



Berke M. Türkay
PhD Candidate, Purdue University, IN