



Constraint Guided Deep Learning to Add Semantics to Acoustic Signals



Peter Karsmakers

DTAI Section

Declarative Languages and Artificial Intelligence Section

13 Faculty

1 IOF Research Manager

± 10 Postdoctoral Researchers

± 60 PhD Researchers

> 80 Alumni

4 campi (Leuven, Bruges, De Nayer, Diepenbeek, Geel)



LEUVEN.AI INSTITUTE

Basic Research

Probabilistic Programming

Predictive Learning and Clustering

Automated Data Science

Exploratory Data Mining

Privacy, Non-discrimination and Ethics

Constraints

Functional Programming

Knowledge Base Systems

Applications

Sports Analytics

Anomaly Detection

Robotics

Engineering and Sensors

Health

Texts and Web

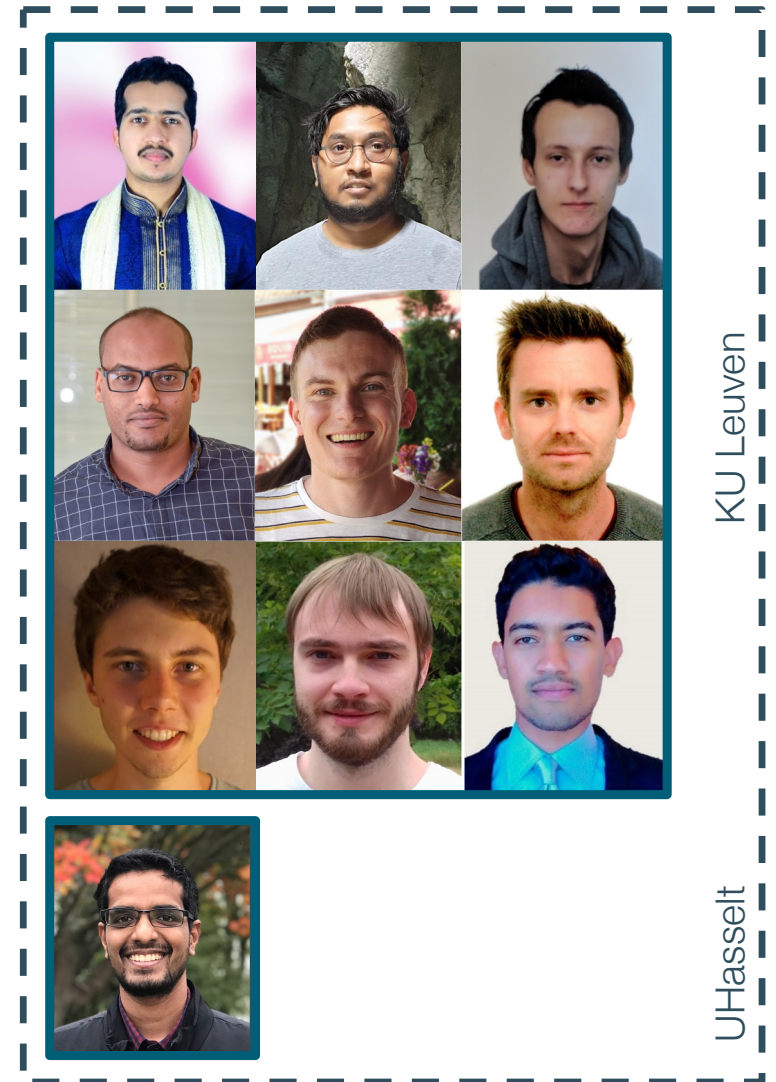
Games

<http://dtai.cs.kuleuven.be/stories>

DTAI@Geel



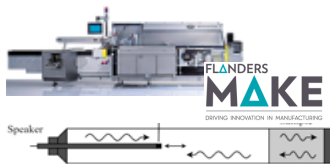
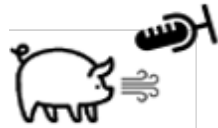
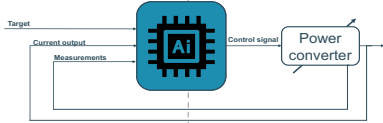
- DTAI@Geel people
 - 1 Professor
 - 1 Postdoc
 - 8 PhDs (+ 1 PhD U Hasselt)
- Member of Flanders Make Core Lab



Research focus DTAI@Geel

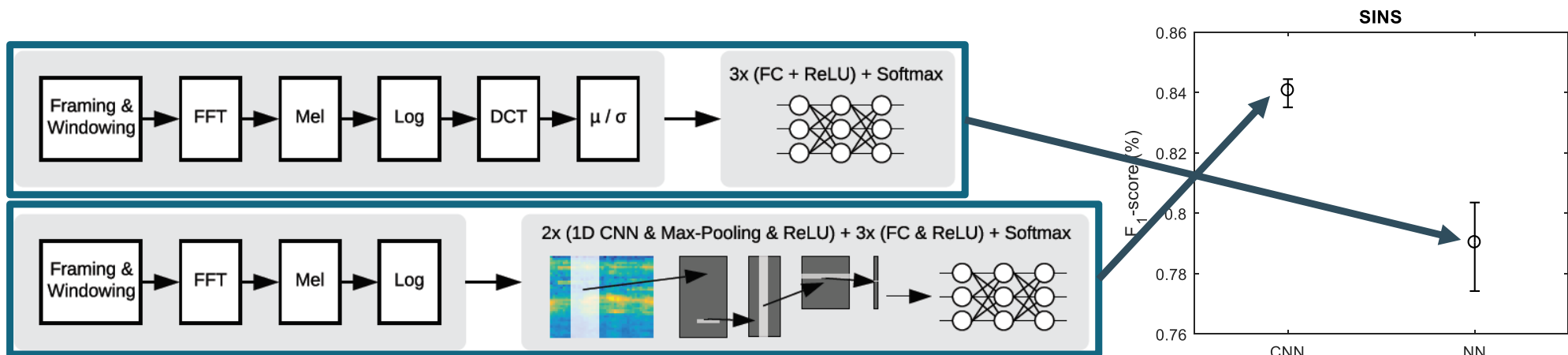
- ML Applications: automated interpretation of time-series -sensor- data
 - Focus on acoustic signals
- Constraint-guided ML
 - Computational limitations (TinyML)
 - Flexible supervision

ML applications

	Home	Industry	Bio
Audio			
Radar			
ACM			
Piezo-electric			
I/V			

Popular in acoustic event/scene classification: DL

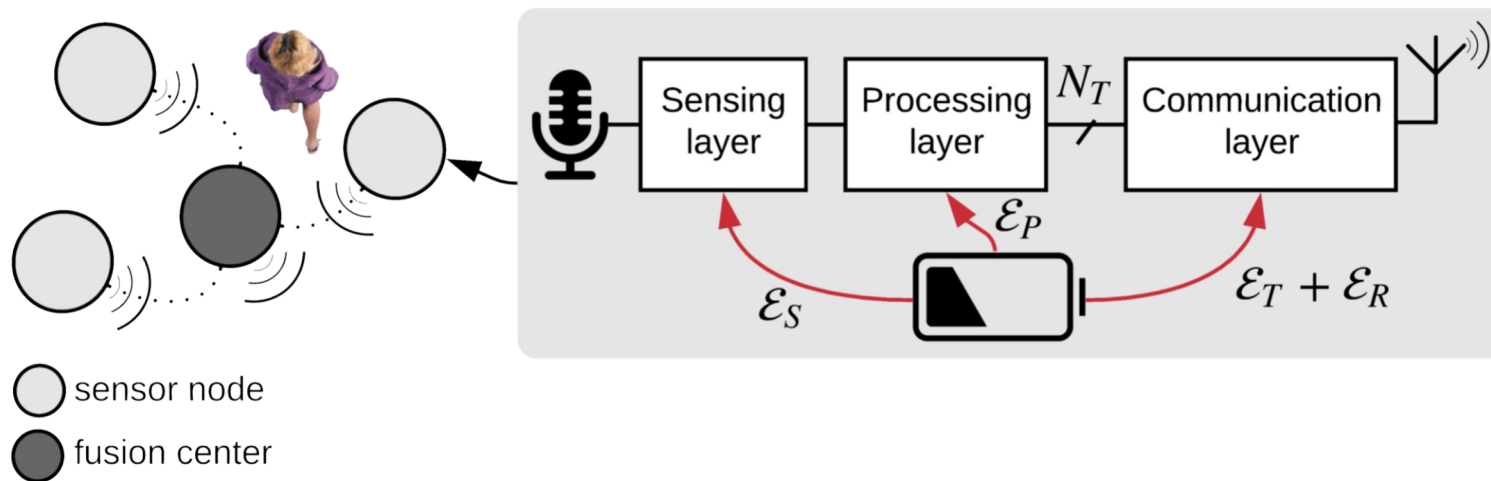
Shift from regular machine learning algorithms in 2013 into deep learning in 2018



Constraint-guided ML

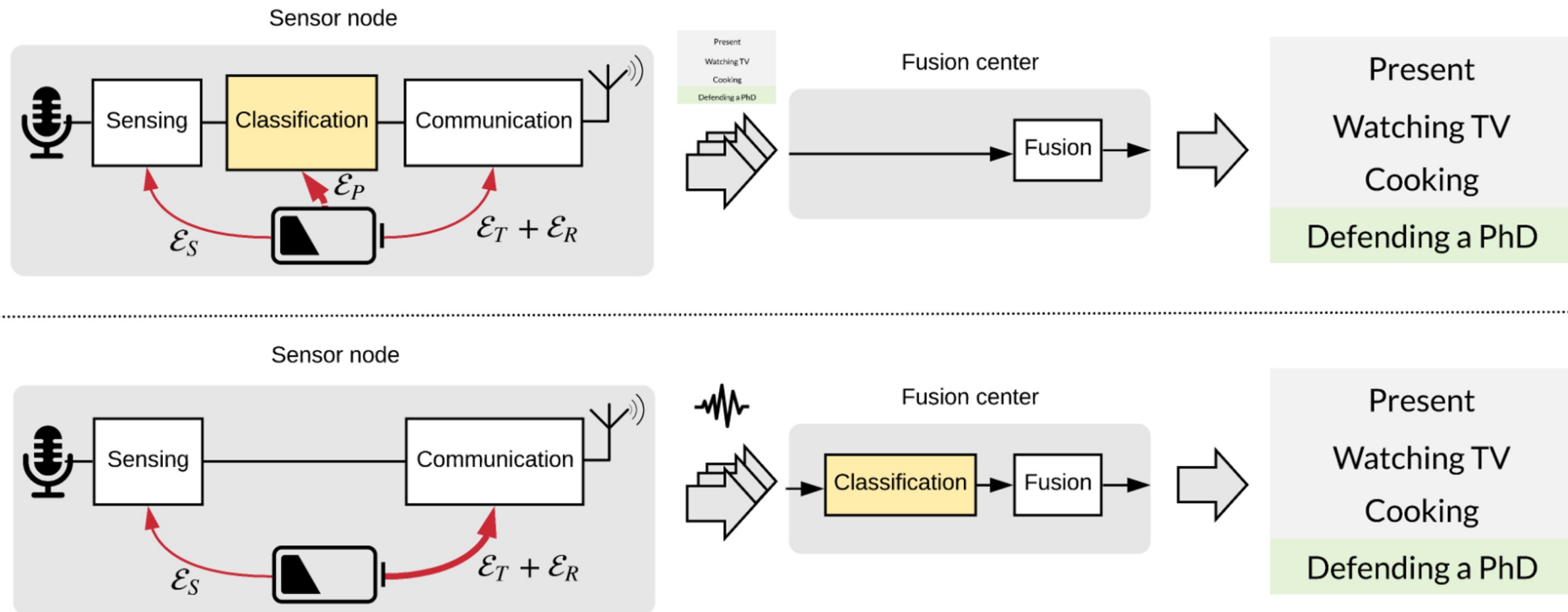
- Computational limitations (TinyML)
- Flexible supervision

Motivation: battery-powered acoustic sensing nodes



$$\bar{\mathcal{E}}_{\text{node}} = \mathcal{E}_S + \mathcal{E}_P + N_T \bar{\mathcal{E}}_T + N_R \bar{\mathcal{E}}_R$$

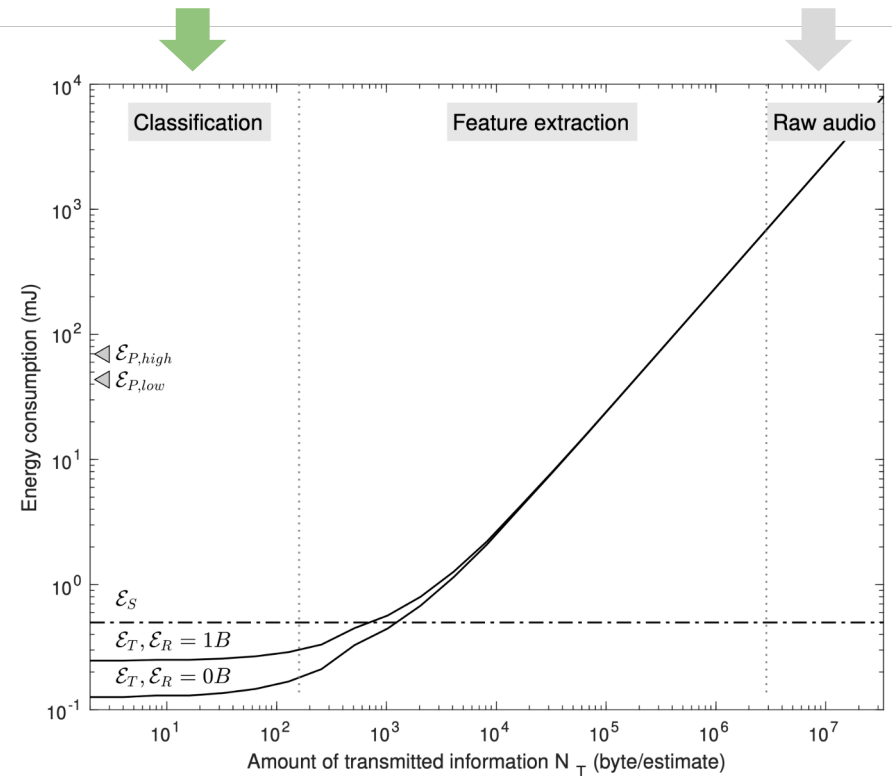
Motivation: compute and communication



Motivation: compute at the edge

- Sensing + communication:
Lifetime... $\pm$ few days
Dominated by communication
- Sensing + classification + communication:
Lifetime... $\pm$ tens of days
Dominated by processing

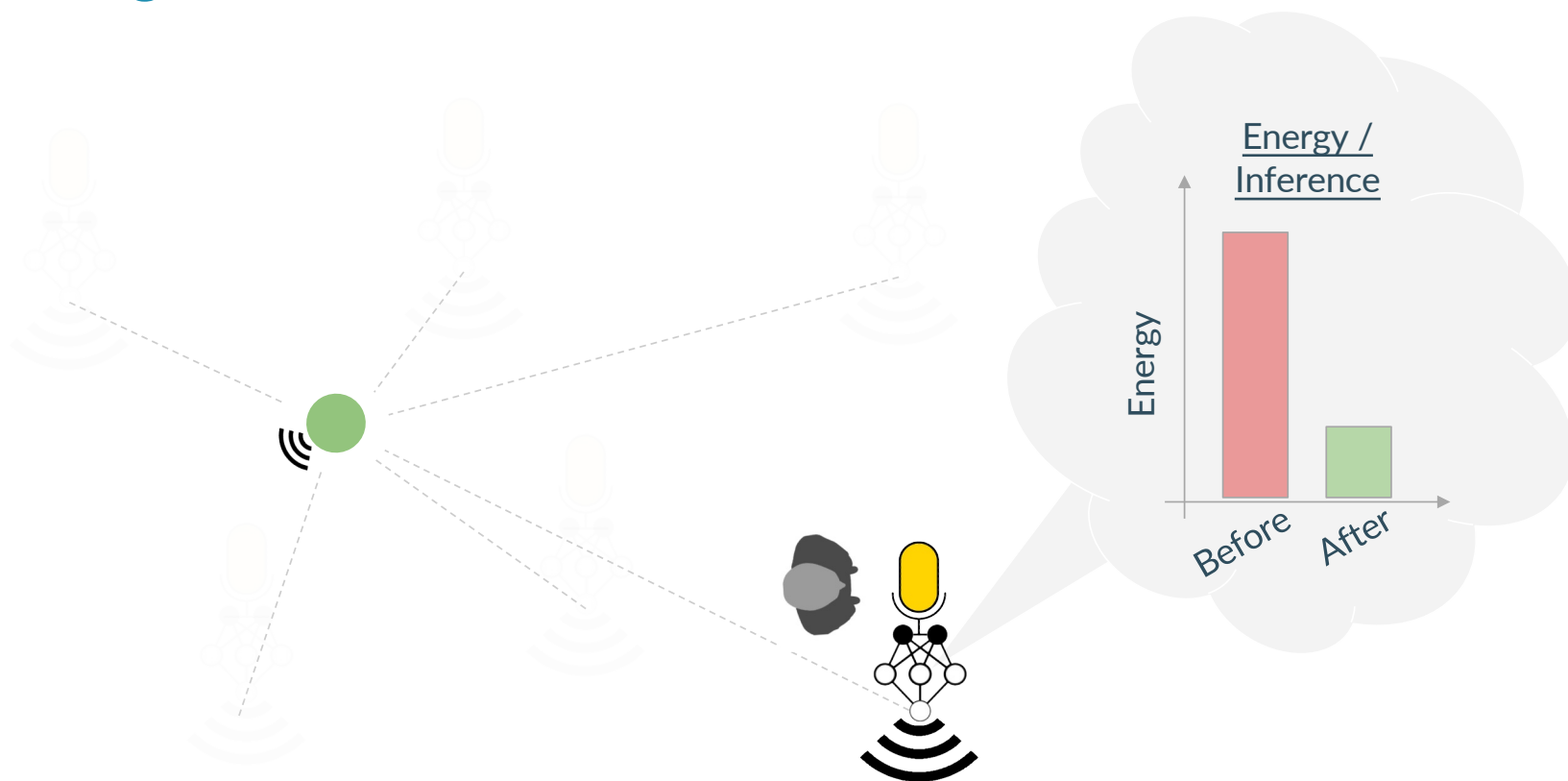
Choose: local classification



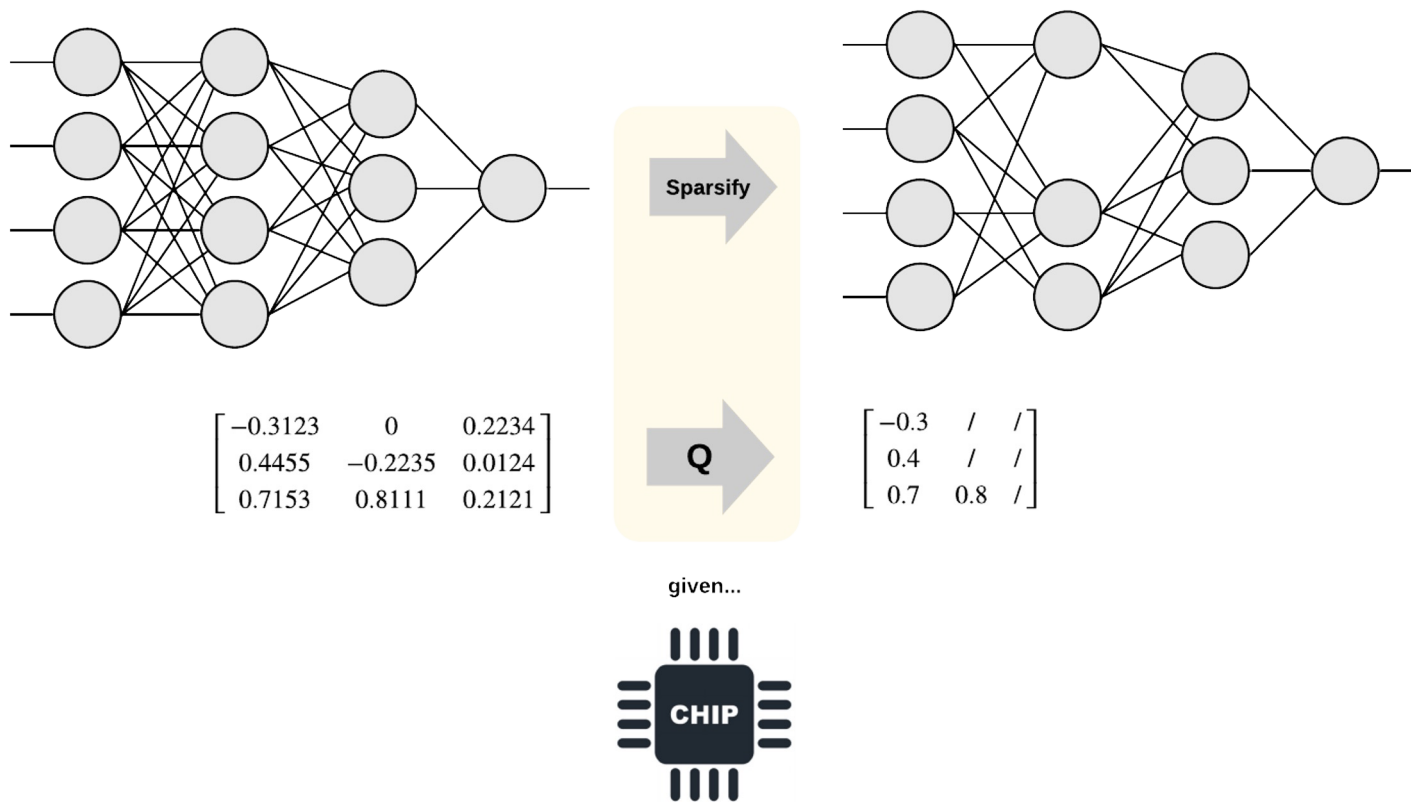
Motivation: reduce energy consumption – duty cycling



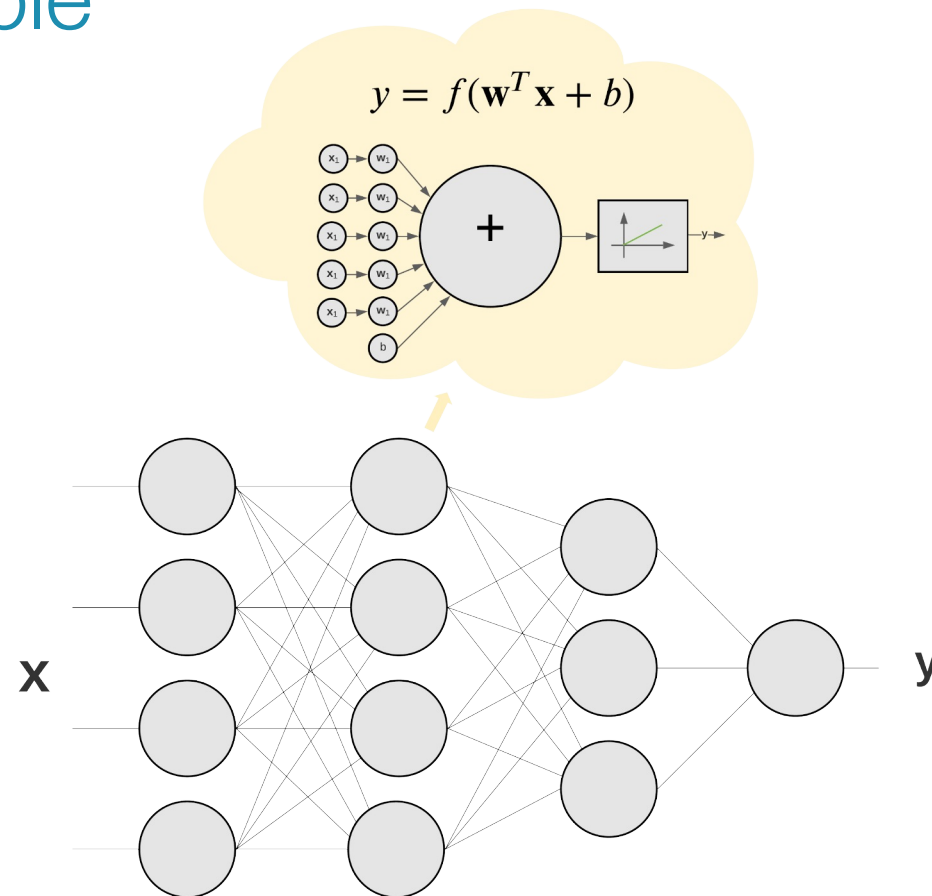
Motivation: reduce energy consumption - quantisation and pruning



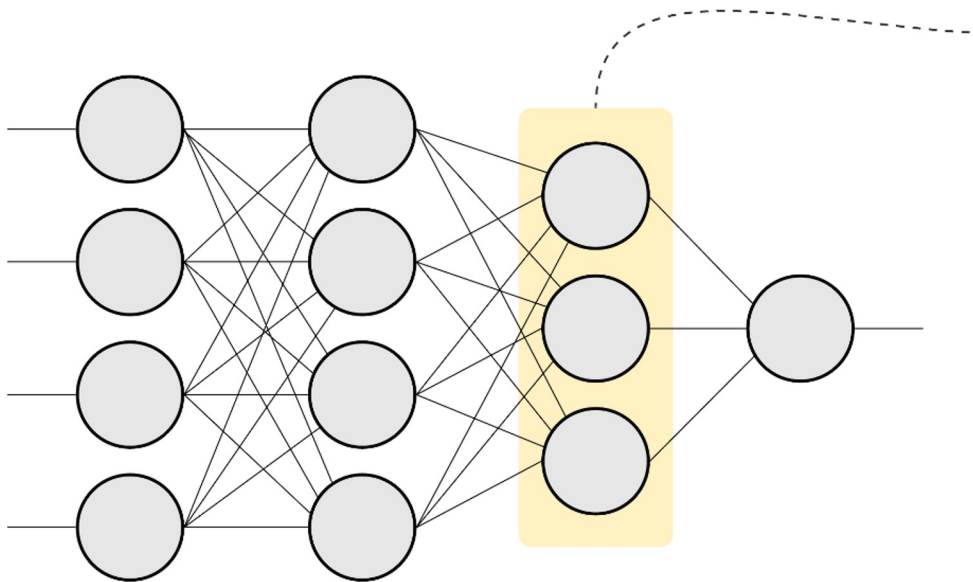
Constraint guided ML: computational limitations



Model example



Pruning



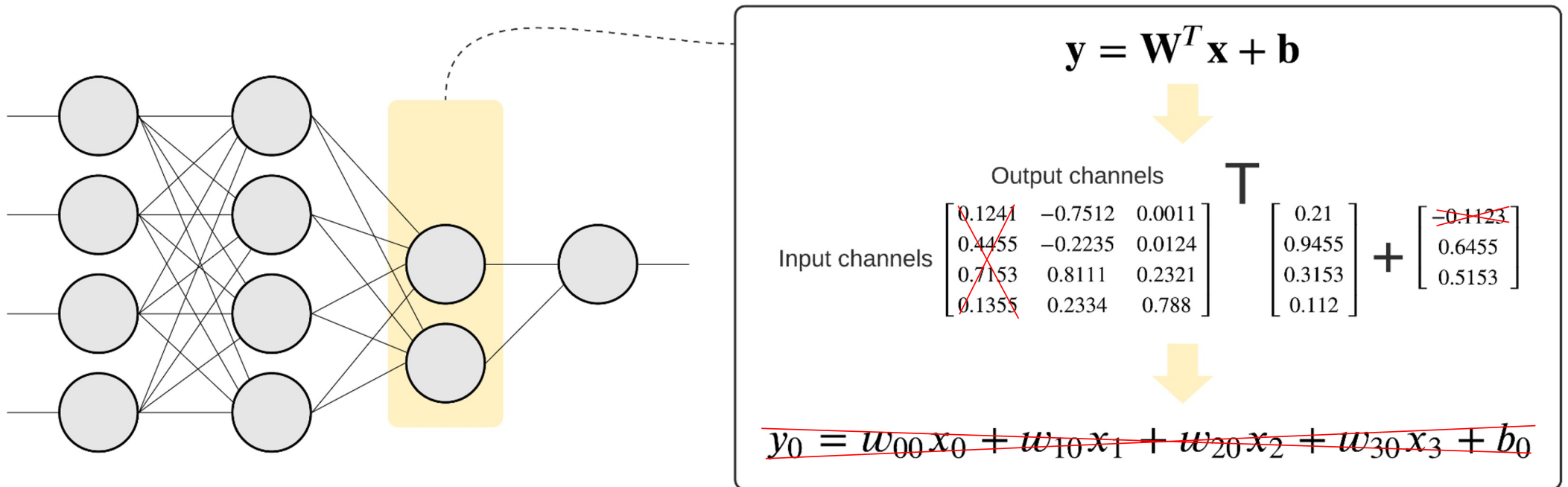
$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

Output channels

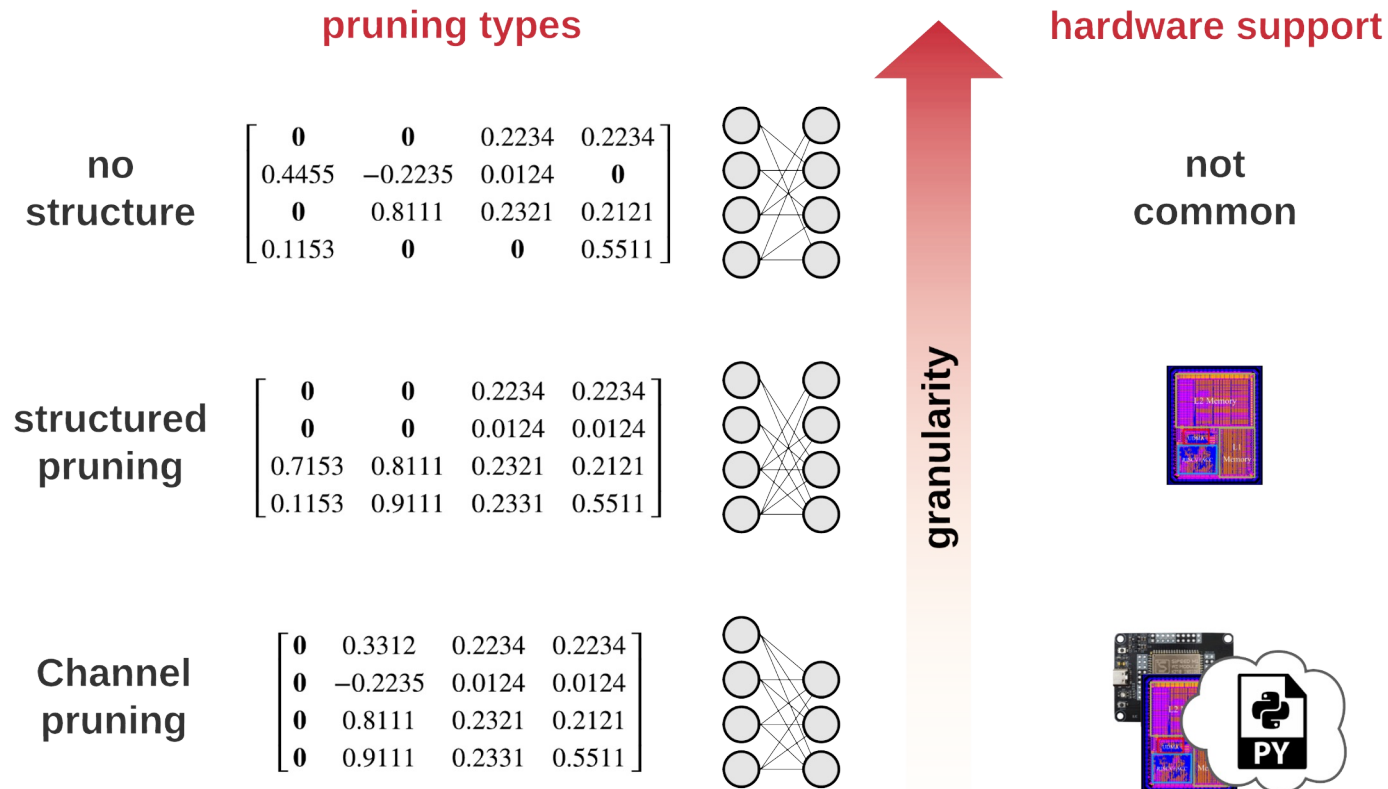
$$\text{Input channels} \begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_2 + b_0$$

Pruning



Pruning and hardware support



Pruning

$$\min_{\boldsymbol{\theta}'} \sum_{d=0}^{D-1} L\left(f(\mathbf{X}_d, \boldsymbol{\theta}), f(\mathbf{X}_d, \boldsymbol{\theta}')\right)$$

$$\text{s.t.} \quad \text{cost}(\boldsymbol{\theta}') \leq K \cdot \text{cost}(\boldsymbol{\theta}) \quad ,$$



$$\Delta L_g(\boldsymbol{\theta}, \mathcal{D}) = L(\boldsymbol{\theta}, \mathcal{D}) - L\left((\mathbf{1}^\top - \mathbf{I}_g) \odot \boldsymbol{\theta}, \mathcal{D}\right)$$

Pruning

$$\min_{\boldsymbol{\theta}'} \sum_{d=0}^{D-1} L\left(f(\mathbf{X}_d, \boldsymbol{\theta}), f(\mathbf{X}_d, \boldsymbol{\theta}')\right)$$

$$\text{s.t.} \quad \text{cost}(\boldsymbol{\theta}') \leq K \cdot \text{cost}(\boldsymbol{\theta}) \quad ,$$



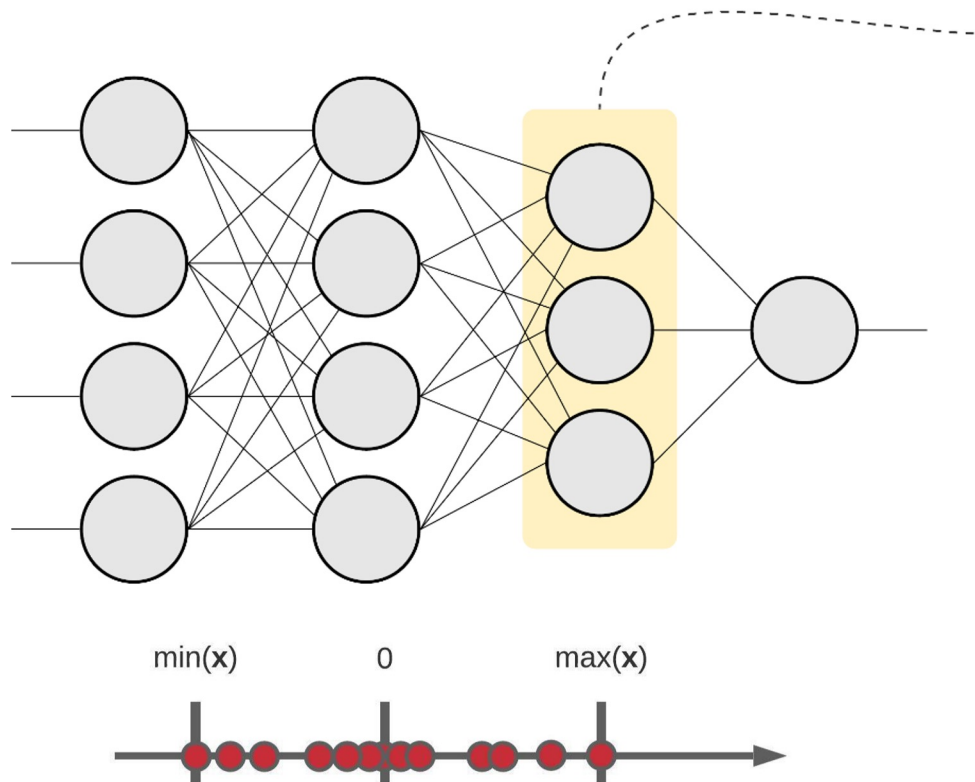
$$\Delta L_g(\boldsymbol{\theta}, \mathcal{D}) = L(\boldsymbol{\theta}, \mathcal{D}) - L\left((\mathbf{1}^\top - \mathbf{I}_g) \odot \boldsymbol{\theta}, \mathcal{D}\right)$$

exhaustive search too expensive

State-of-the-art

- Approximate saliency criteria
 - Magnitude of a weight (Han et al, 2015)
 - Magnitude of the gradient of the loss w.r.t. a connection (Lee et al, 2019 and Molchanov et al, 2017)
- Regularization
 - Approximate saliency criterion + L1/L2 regularization (Han et al, 2015 and Lebedev et al, 2016)
 - Sparse variational dropout (Molchanov et al, 2017)
 - L0-regularization (Louizos et al, 2017)
- Search-based pruning (NAS)
 - Automated Model Pruning (He et al, 2018)
- Other
 - Once-for-All (Cai et al, 2020)
 - Low-rank approximation, i.e. SVD (Denton et al, 2014)
 - Efficient network architectures, i.e. MobileNet (Howard, 2017)

Quantisation



$$\mathbf{y} = \mathbf{W}^T \mathbf{x} + \mathbf{b}$$

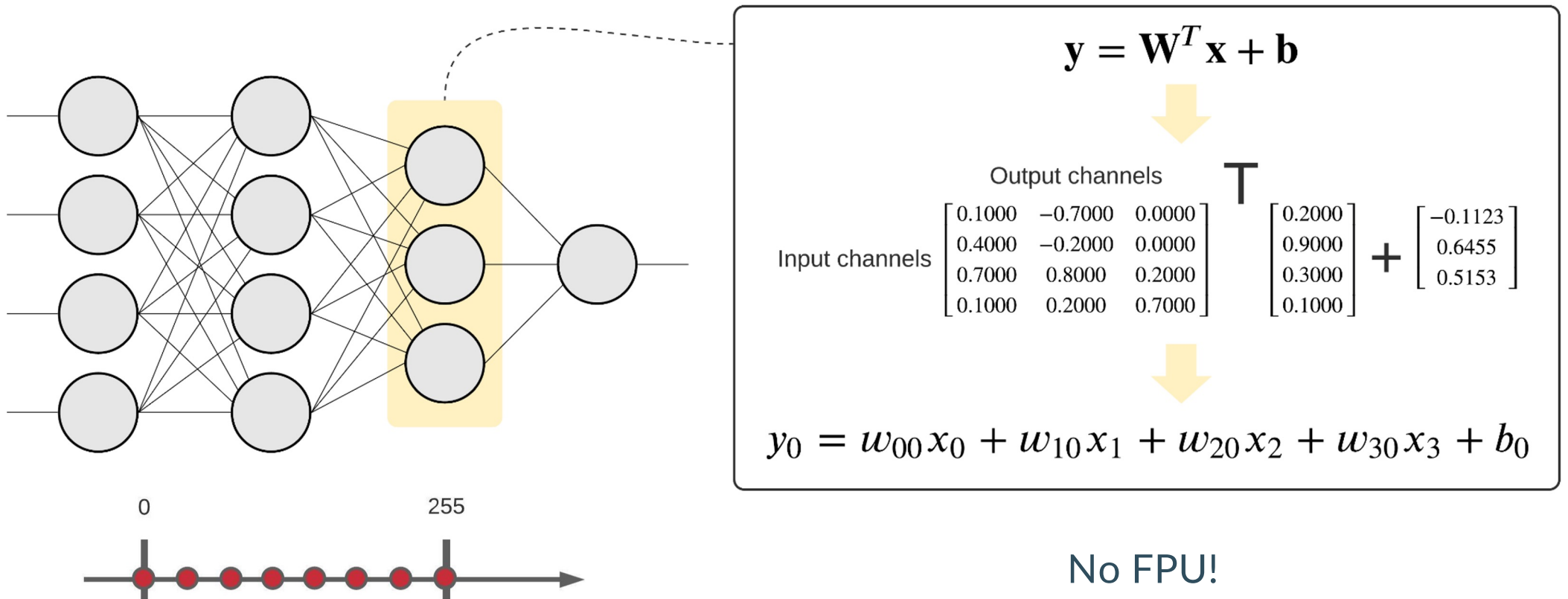
Output channels

Input channels

$$\begin{bmatrix} 0.1241 & -0.7512 & 0.0011 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2321 \\ 0.1355 & 0.2334 & 0.788 \end{bmatrix}^T \begin{bmatrix} 0.21 \\ 0.9455 \\ 0.3153 \\ 0.112 \end{bmatrix} + \begin{bmatrix} -0.1123 \\ 0.6455 \\ 0.5153 \end{bmatrix}$$

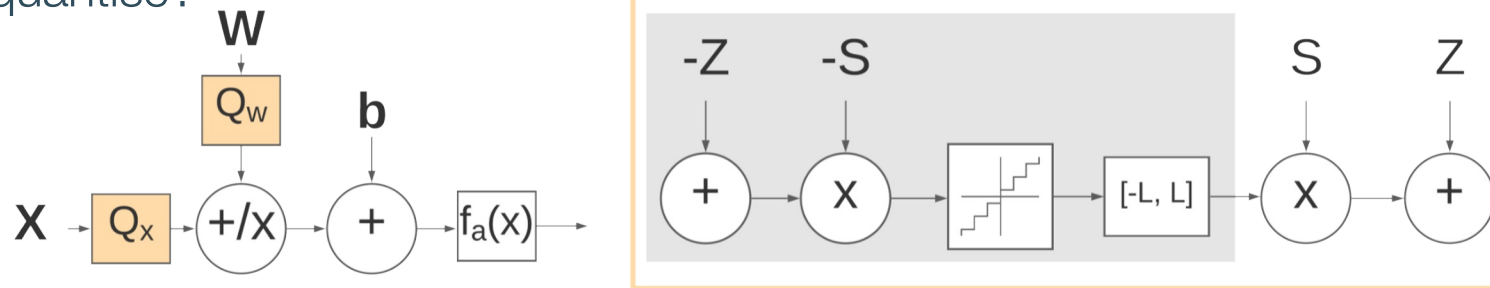
$$y_0 = w_{00}x_0 + w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + b_0$$

Quantisation



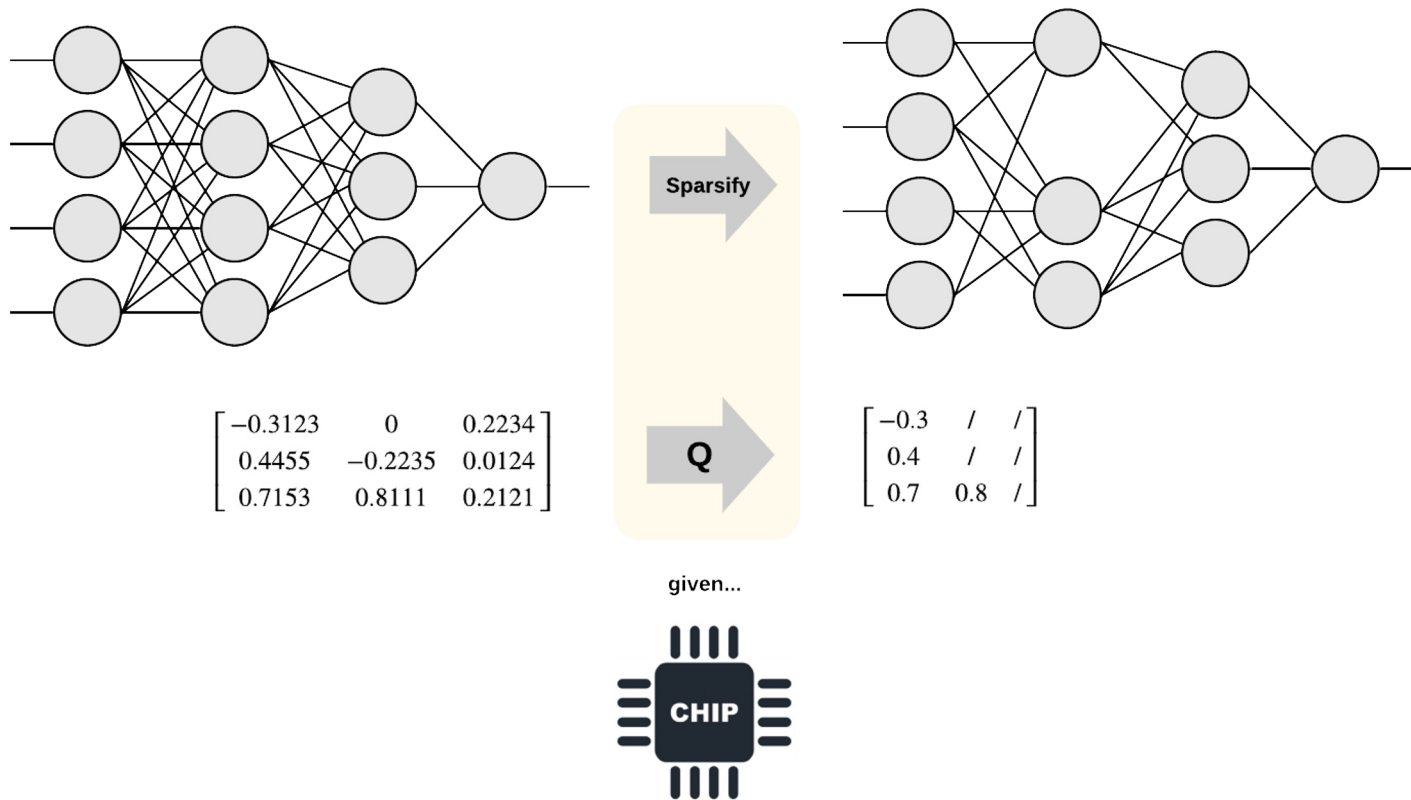
State-of-the-art

- How to quantise?



- Estimates ranges based on min-max of statistical distribution (Krishnamoorthi, 2018)
- Quantisation-aware training (Jacob, 2019)
- Data-free quantisation (Nagel, 2019)
- Hardware-aware automated quantisation with mixed-precision (Wang, 2019)
- ...
- When? After, During

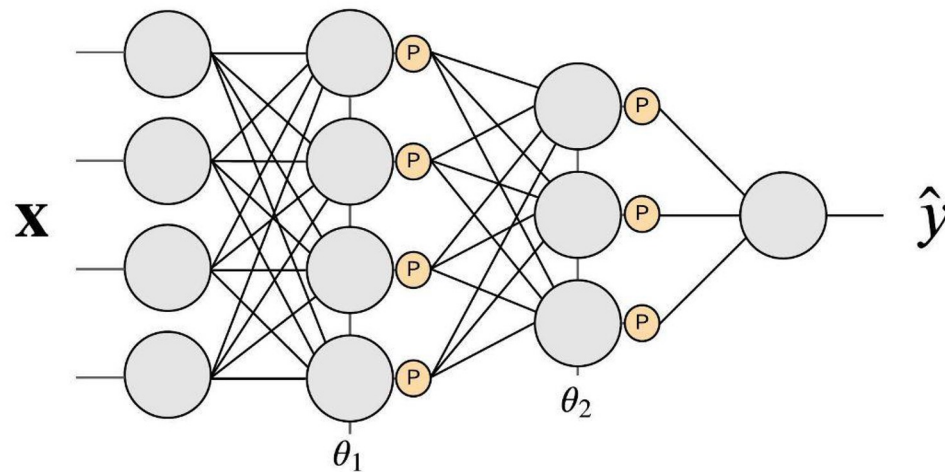
Joint pruning and quantisation



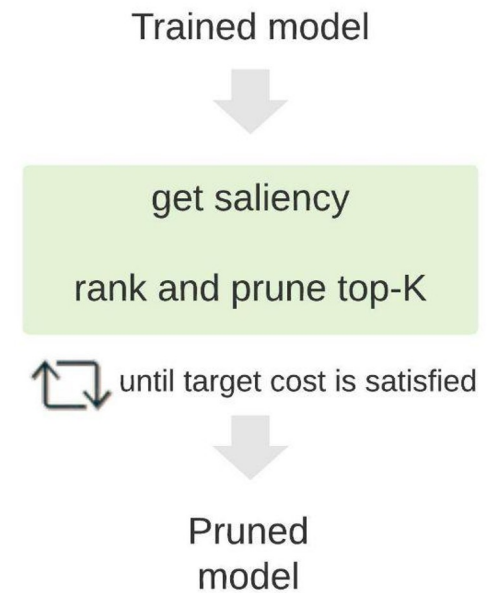
State-of-the-art

- Bayesian Bits (van Baalen et al, 2020)
 - During training
 - Can't exactly target a cost
- APQ (Wang et al, 2020)
 - During or after training
 - Multiple training cycles
- Our approach:
 - During or after training
 - “Quick”
 - Targets exactly a cost

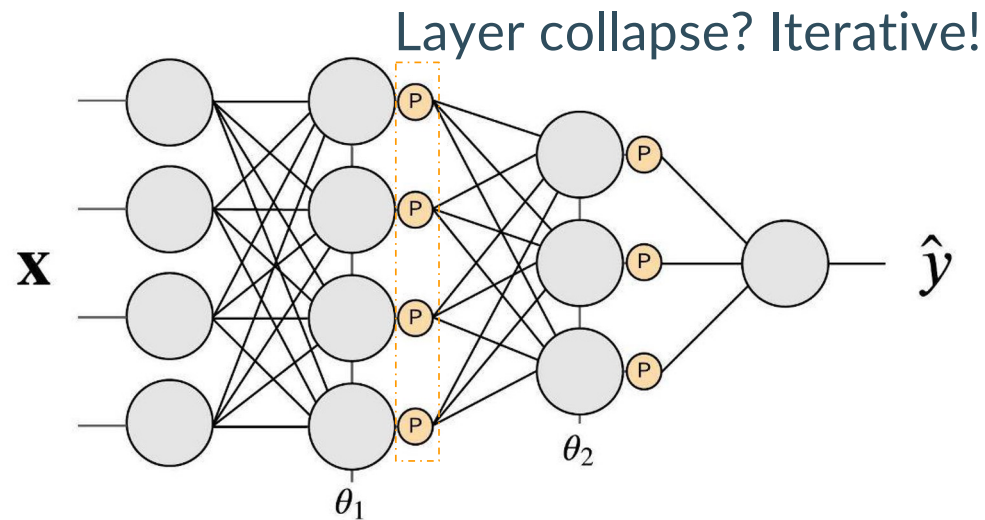
SNAP pruning*



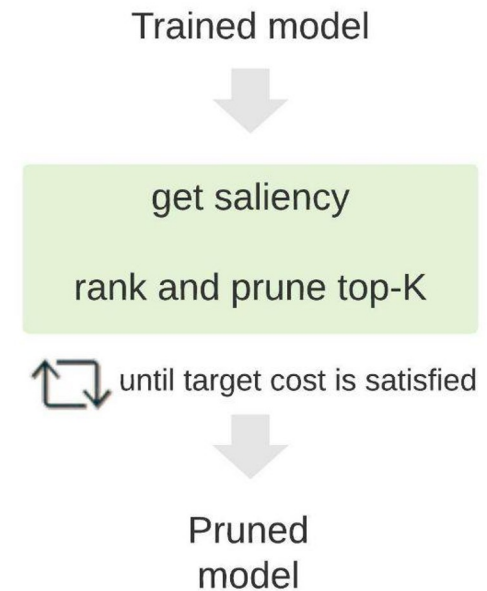
$$s(\mathbf{g}_i) = \left| \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{g}_i} \right| = \text{absolute gradient of loss w.r.t. the gate}$$



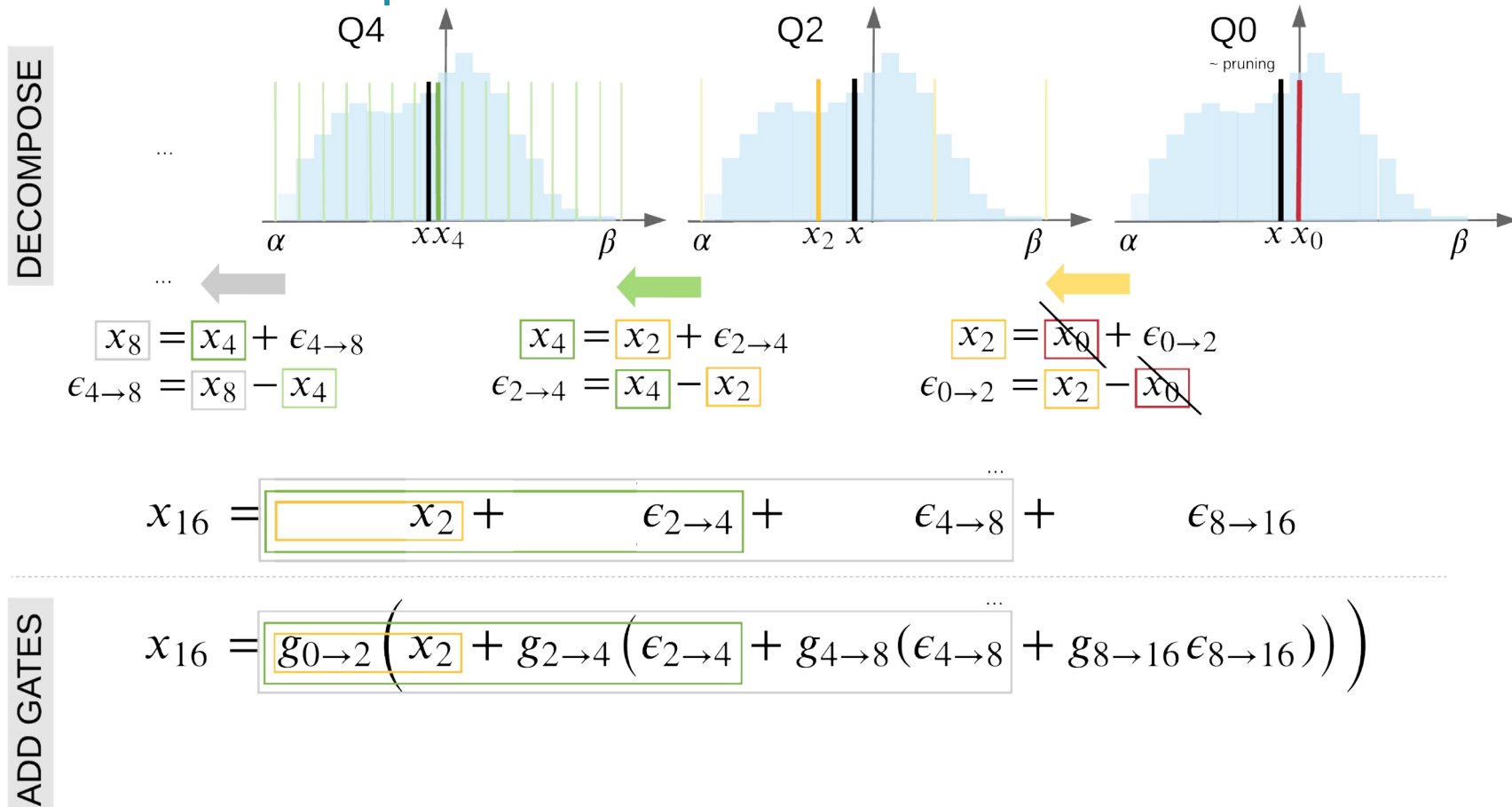
SNAP pruning*



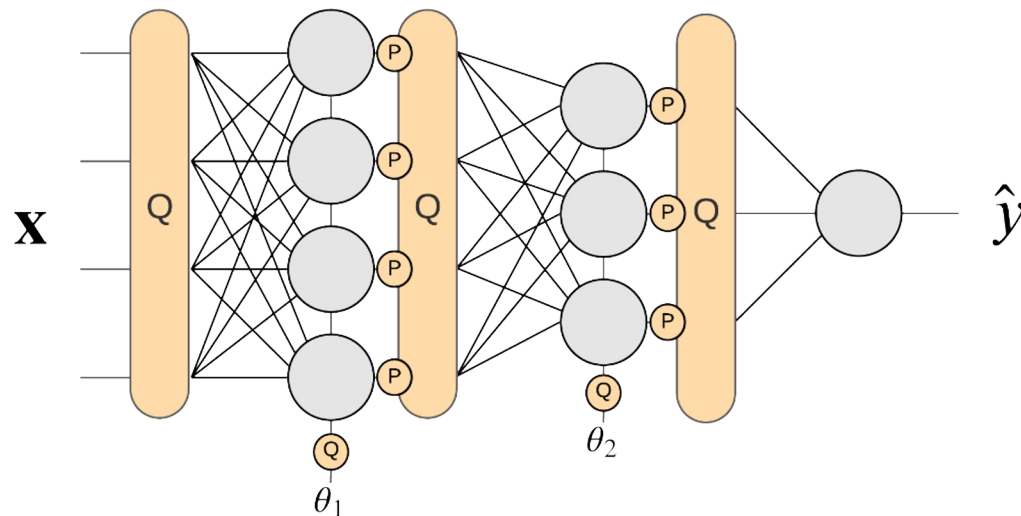
$$s(\mathbf{g}_i) = \left| \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{g}_i} \right| = \text{absolute gradient of loss w.r.t. the gate}$$



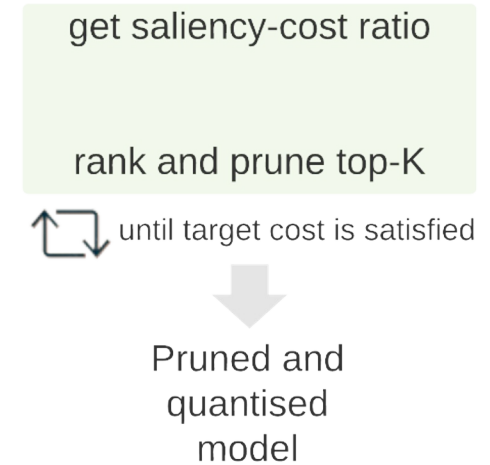
Gated decomposition*



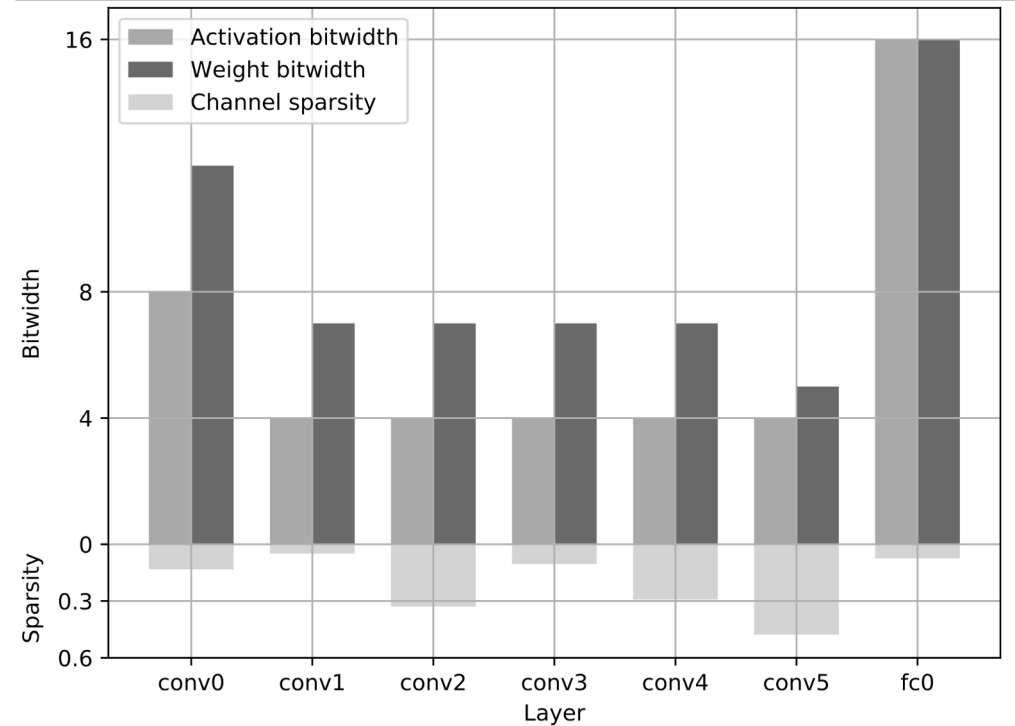
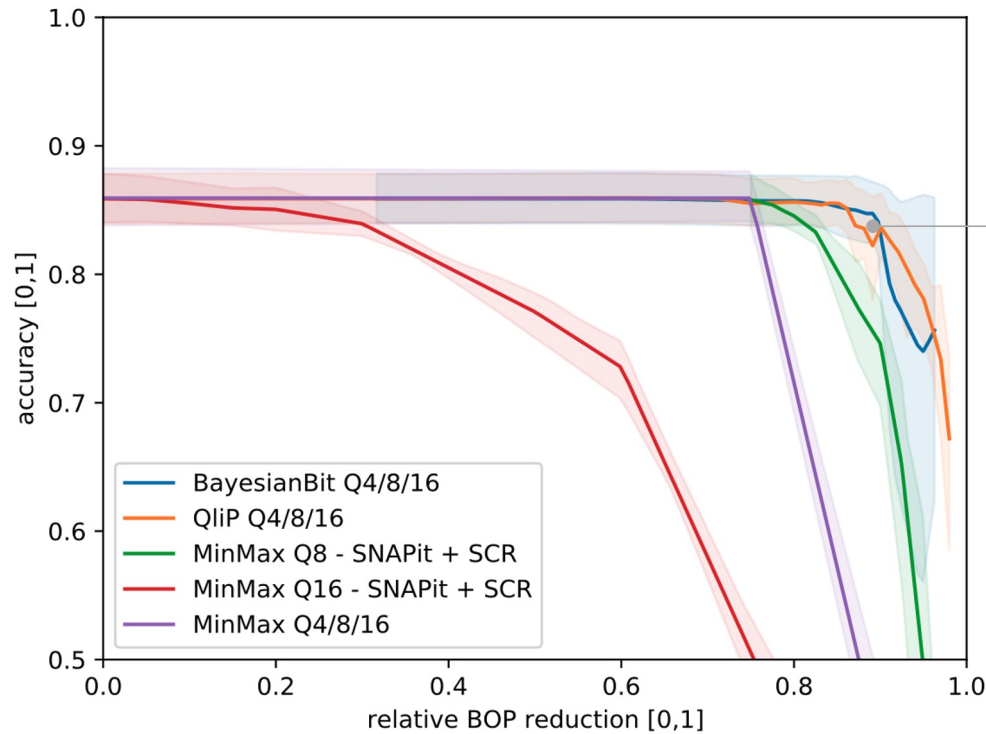
Proposed: QliP



$$s(\mathbf{g}_i) = \frac{\left| \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{g}_i} \right|}{\Delta \kappa_i} = \frac{\text{absolute gradient of loss w.r.t. the gate}}{\text{relative cost gain if gate is zero}}$$

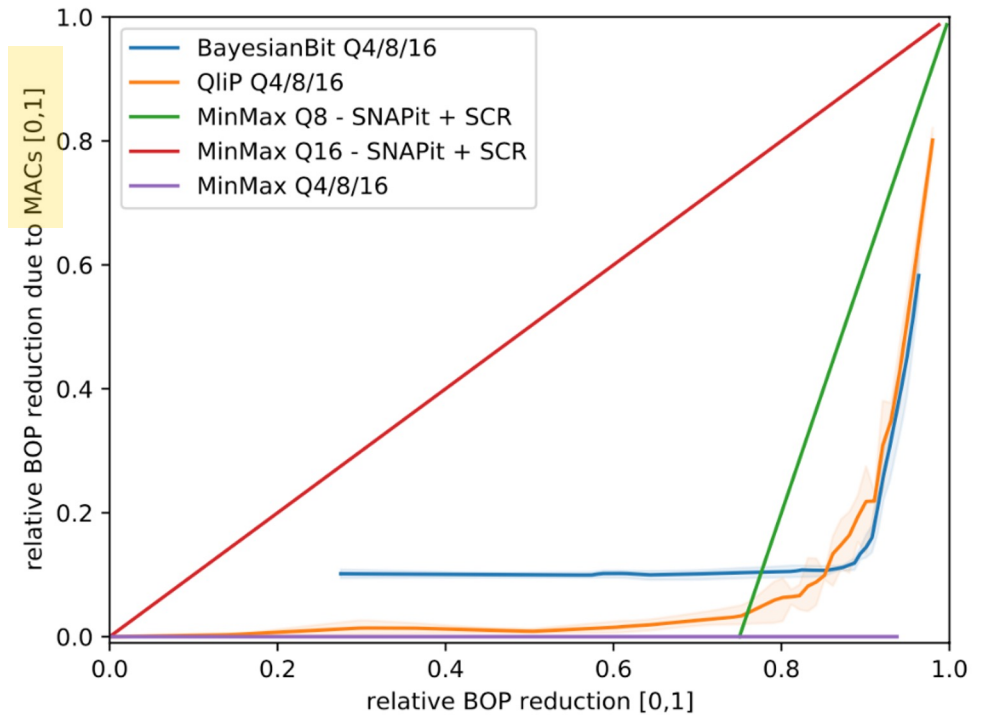
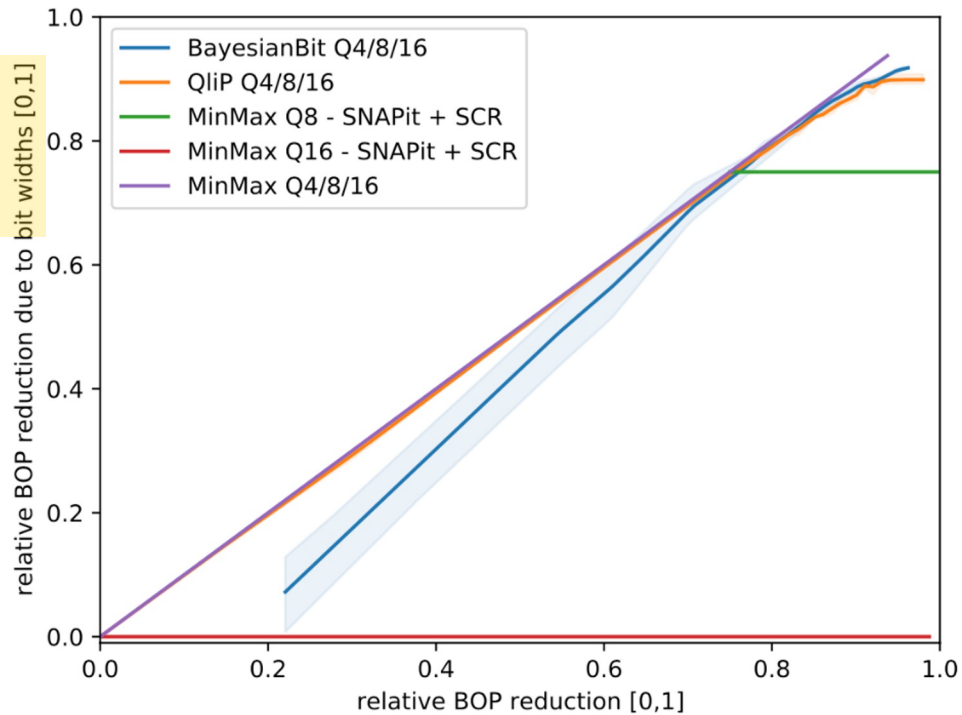


Result snapshot



$$1 - \sum_{l \in \text{remaining}} \frac{MAC(l)b_{weight}(l)b_{act}(l)}{\sum_{l \in All} MAC(l)b_{weight}(l)b_{act}(l)}$$

Result snapshot



BACKGROUND



Constraint-guided ML

- Computational limitations (TinyML)
- Flexible supervision

Motivation

- DL methods need huge amount of annotated data during supervised learning
- Learn solely from data or try also to include available background/domain knowledge?

Background knowledge = refers to information about the problem

- Forms of background knowledge: relevant features, knowledge graphs, rules-of-thumb, logical constraints, probability distributions, mathematical formula's, causal connections
- Domain knowledge can be used to customize:
 - features processing
 - model architecture
 - learning mechanism

Motivation

- DL methods need huge amount of annotated data during supervised learning
- Learn solely from data or try also to include available background/domain knowledge?

Background knowledge = refers to information about the problem

- Forms of background knowledge: relevant features, knowledge graphs, rules-of-thumb, logical constraints, probability distributions, mathematical formula's, causal connections
- Domain knowledge can be used to customize:
 - features processing
 - model architecture
 - learning mechanism ← enforce ML model to be consistent with background knowledge

Motivation: flexible supervision

- Enforce ML model to be consistent with background knowledge
→ flexible supervision (alternative to the sole use of labeled data to train models)
- Background knowledge formatted as constraints, e.g.:
 - $\text{addition}(X,Y,Z) \text{ :- digit}(X,N1), \text{digit}(Y,N2), Z \text{ is } N1+N2$
 - $(y_1 \wedge \neg y_2 \wedge \neg y_3) \vee (\neg y_1 \wedge y_2 \wedge \neg y_3) \vee (\neg y_1 \wedge \neg y_2 \wedge y_3)$
 - $\frac{\partial u(t,x)}{\partial t} + \mathcal{N}_\theta[u](t,x) = 0$
 - Sentence S with structure A but B $\rightarrow$ sentiment of B dominates
 - $T_{\min}^{(t+1)} \leq T_{\max}^{(t+1)}$

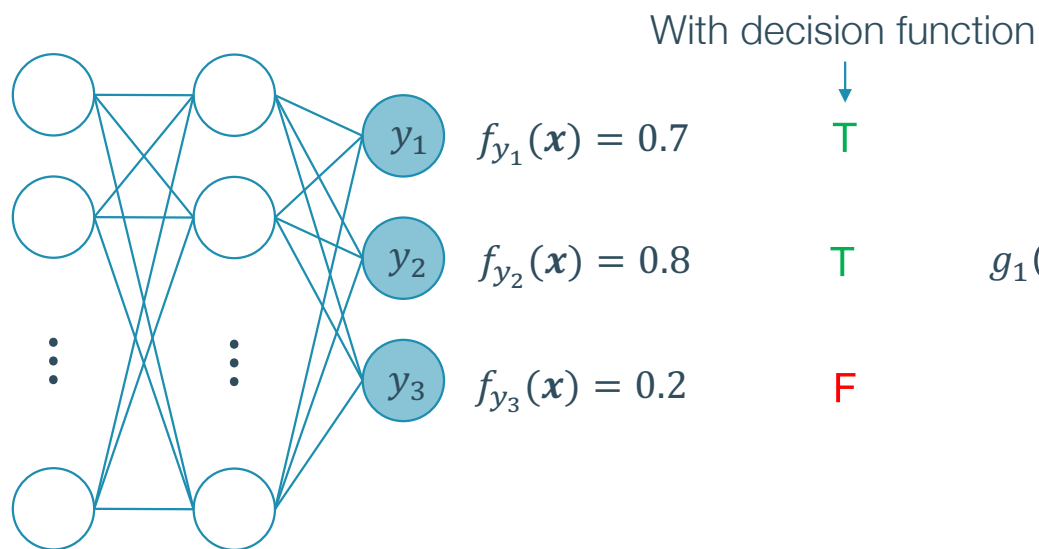
Constraint-guided ML: flexible supervision

- How to incorporate constraints during the learning cycle?
- Typical approach:
 - Transform constraints to differentiable “constraint” functions $g_i(\mathbf{x}) \in \mathbb{R}$ where the higher $g_i(\mathbf{x})$ the better $\mathbf{x}$ w.r.t. the knowledge expressed by $g_i(\mathbf{x})$
 - Add such term to the loss function used in ML

$$\theta^* = \operatorname{argmin}_{\theta} L(f_{\theta}(\mathbf{x}), y) - \sum_i \lambda_i \log(g_i(\mathbf{x}))$$

Constraint-guided ML: flexible supervision

Example:



Consider the following constraint

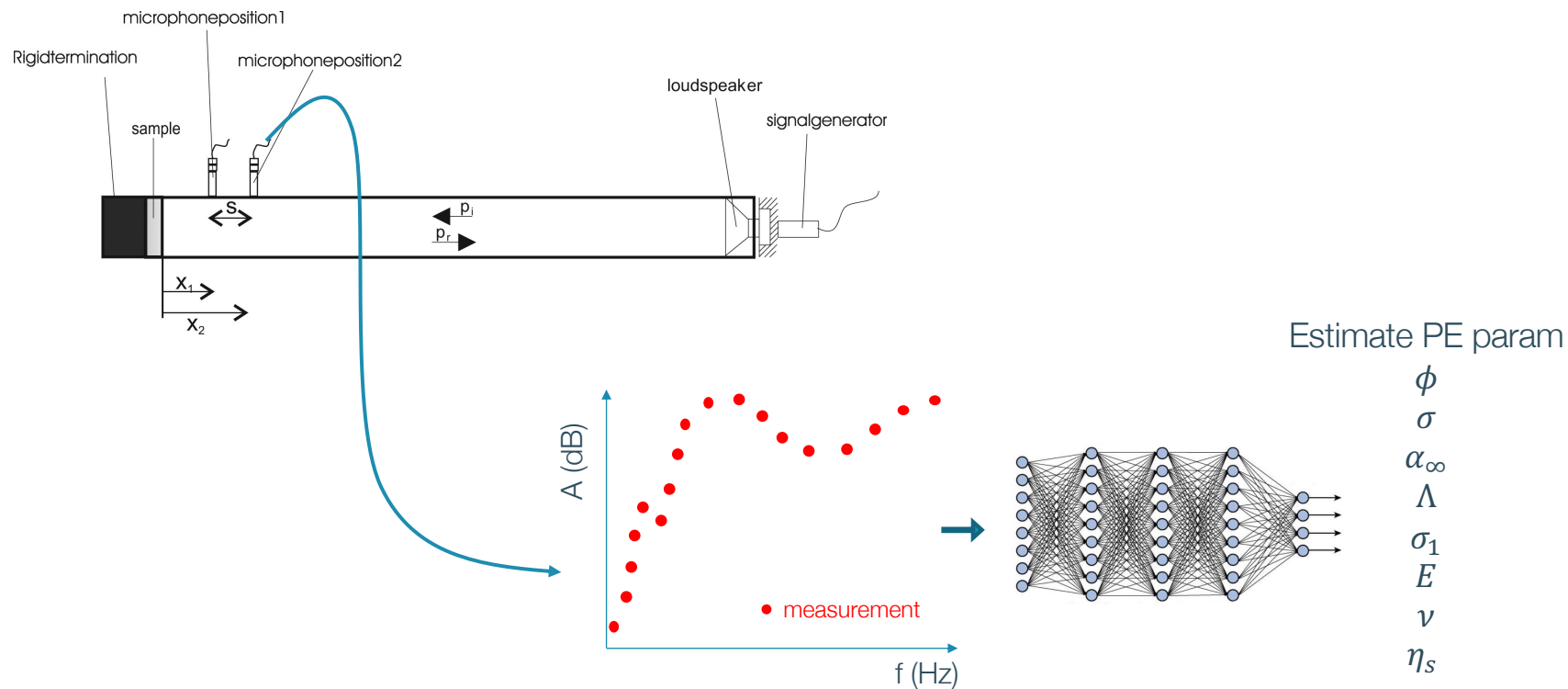
$$\text{Con}_1: (y_1 \wedge \neg y_2 \wedge \neg y_3) \vee (\neg y_1 \wedge y_2 \wedge \neg y_3) \vee (\neg y_1 \wedge \neg y_2 \wedge y_3)$$

Fuzzy logic
Gödel t-norm

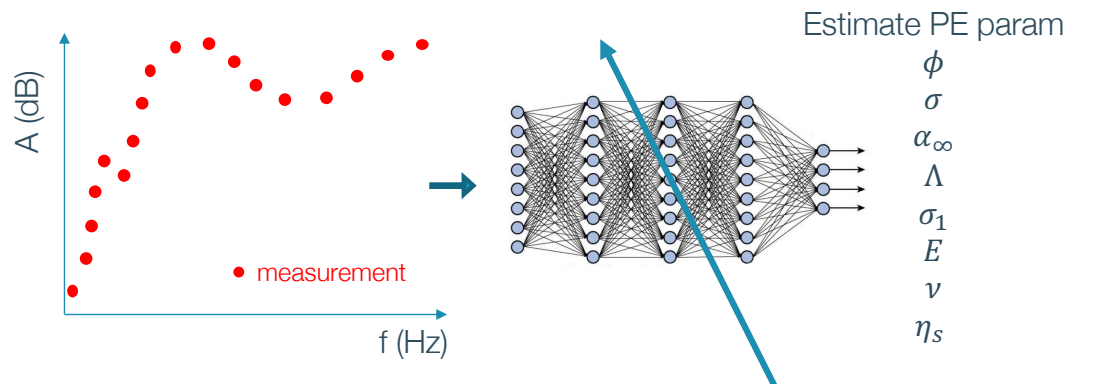
$$g_1(\mathbf{x}) = \max[\min(f_{y_1}(\mathbf{x}), (1 - f_{y_2}(\mathbf{x})), (1 - f_{y_3}(\mathbf{x}))) + \min((1 - f_{y_1}(\mathbf{x})), f_{y_2}(\mathbf{x}), (1 - f_{y_3}(\mathbf{x}))) + \min((1 - f_{y_1}(\mathbf{x})), (1 - f_{y_2}(\mathbf{x})), f_{y_3}(\mathbf{x}))]$$

$$\theta^* = \operatorname{argmin}_{\theta} L(h_{\theta}(\mathbf{x}), \mathbf{y}) - \lambda_1 \log(g_1(\mathbf{x}))$$

Use-case: vibro-acoustic modelling of poroelastics



Use-case: vibro-acoustic modelling of poroelastics



Bound constraints

Variable	Lower Bound	Upper Bound
ϕ	0.7	1
σ	10^3	10^5
α_{∞}	1	2
ρ_1	0	100
E	10^3	10^6
ν	0	0.5
η_s	0	0.5

Connection constraints

$$\frac{1}{3.3} \sqrt{\frac{8\alpha_{\infty}\eta}{\sigma\phi}} \leq \Lambda \leq \Lambda'$$

Technical problem setting

- Learn a neural network to perform a multi-output regression task by minimizing a mean squared error loss while satisfying inequality constraints
- Restrictions:
 - No transformations of the constraints that adjust the crispness or hardness of the statement (rules out fuzzy logic approach)
 - Framework should hold for other tasks and other kind of constraints

Technical problem setting

- Assumptions:
 1. Neural network is trained using gradient descent
 2. For each constraint, there is a method available to determine which direction to move towards in order to find a point that satisfies the constraint
 - E.g., consider the constraint $T_{min} \leq T_{max}$:
if the T_{min} is strictly larger than the T_{max} , then the T_{min} prediction should be made smaller and/or the T_{max} prediction should be made larger

Constraint Guided Gradient Descent

- Constrained optimization problem

$$\operatorname{argmin}_{\theta} L(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{y}; \theta)$$

$$s. t. \forall (\mathbf{x}, \hat{\mathbf{y}}) \in (\mathbf{x}, \hat{\mathbf{y}}): C_i(\mathbf{x}, \hat{\mathbf{y}}; \theta) \leq 0, i = 1, \dots, N$$

- Modify gradient step

$$\theta^{(k+1)} := \theta^k - \eta_i \left(\nabla L(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{y}; \theta^{(k)}) + \gamma \sum_i \overrightarrow{dir} \left(C_i(\mathbf{x}, \hat{\mathbf{y}}; \theta^{(k)}) \right) \right)$$

Constraint Guided Gradient Descent

$\overrightarrow{dir} \left(C(x, \hat{y}; \theta^{(k)}) \right)$:

E.g., consider the constraint $T_{min} \leq T_{max}$:

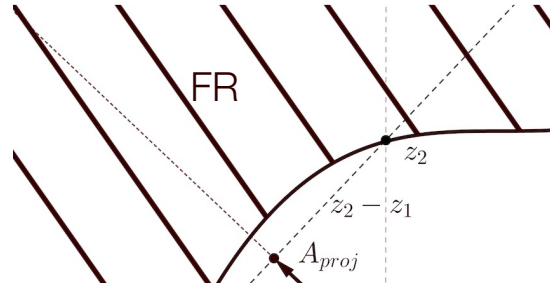
if the T_{min} is strictly larger than the T_{max} , then

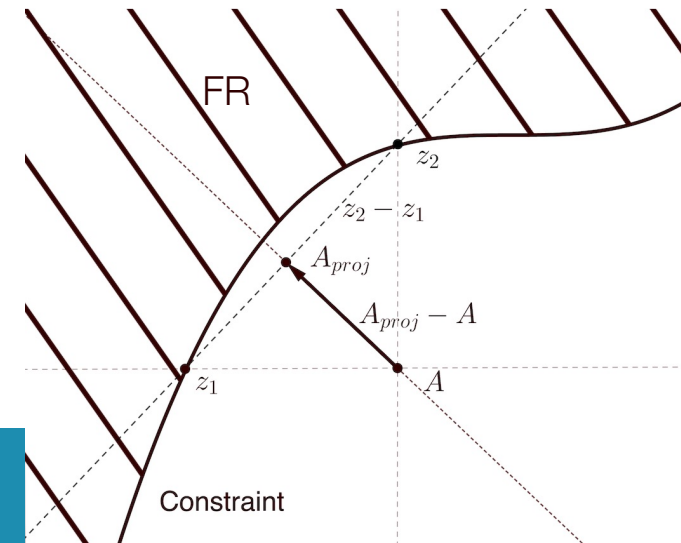
the T_{min} prediction should be made smaller and/or

the T_{max} prediction should be made larger

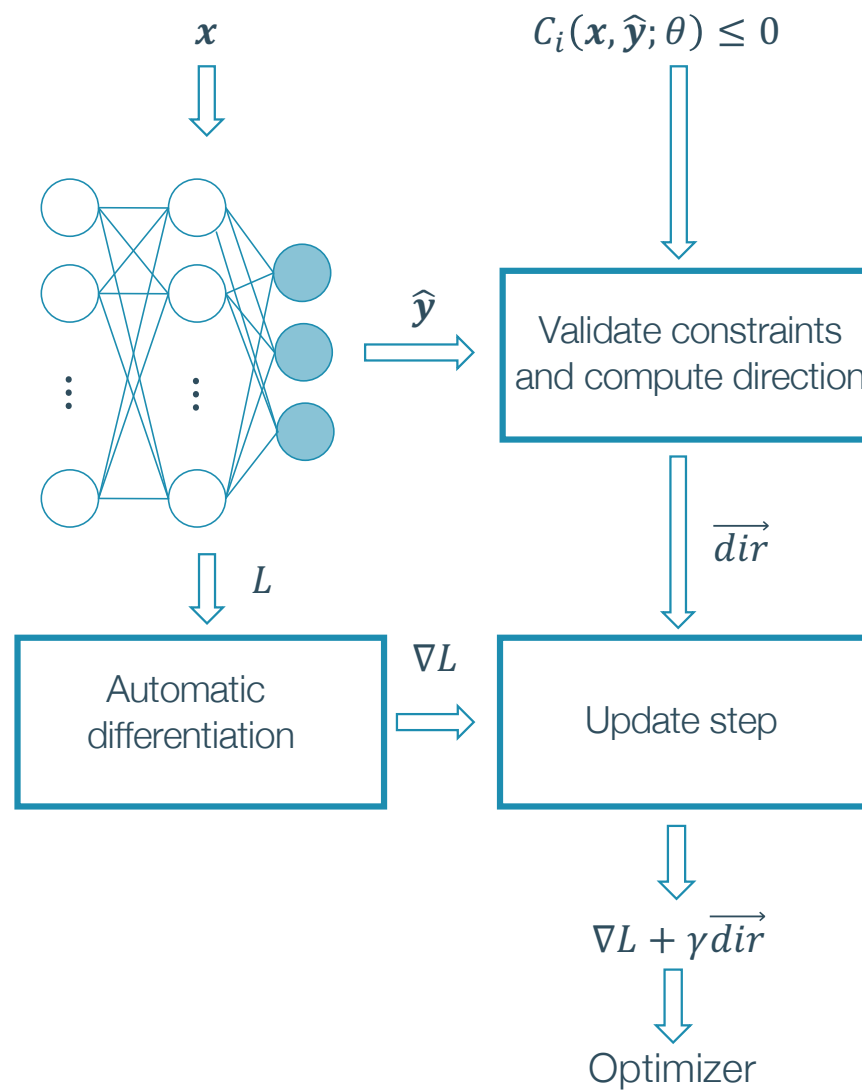
Constraint Guided Gradient Descent

$$\theta^{(k+1)} := \theta^k - \eta_i \left(\nabla L(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{y}; \theta^{(k)}) + \gamma \sum_i \overrightarrow{dir} \left(C_i(\mathbf{x}, \hat{\mathbf{y}}; \theta^{(k)}) \right) \right)$$

- $\overrightarrow{dir}\left(C(\mathbf{x}, \hat{\mathbf{y}}; \theta^{(k)})\right)$: denotes the direction corresponding to the closest path w.r.t. the Euclidean distance from FR to $\theta^{(k)}$
 - γ : is chosen such that it will **dominate** the update step when at least one constraint is not satisfied
 - Remark: it is not required to compute this direction by taking a gradient
- 



CGGD



Results

Data set: *poro-elastic materials*

- Metric: mean squared error

Variable	Lower Bound	Upper Bound
ϕ	0.7	1
σ	10^3	10^5
α_∞	1	2
ρ_1	0	100
E	10^3	10^6
ν	0	0.5
η_s	0	0.5

$$\frac{1}{3.3} \sqrt{\frac{8\alpha_\infty \eta}{\sigma \phi}} \leq \Lambda \leq \Lambda'$$

Method	Size used for training relative to full training set (%)				
	100	75	50	25	10
Baseline	0.1391 ± 0.0076	0.1405 ± 0.0043	0.1508 ± 0.0026	0.1518 ± 0.0037	0.1788 ± 0.0041
Fuzzy	0.1700 ± 0.0104	0.2040 ± 0.0573	0.3351 ± 0.1547	0.1802 ± 0.0180	0.2488 ± 0.1384
CGGD	0.1660 ± 0.0019	0.1701 ± 0.0046	0.1713 ± 0.0018	0.1706 ± 0.0025	0.1721 ± 0.0047

- Metric: satisfaction ratio = #satisfied / #constraints

Method	Size used for training relative to full training set (%)				
	100	75	50	25	10
Baseline	99.11 ± 0.10	99.34 ± 0.18	99.19 ± 0.29	99.19 ± 0.36	99.52 ± 0.23
Fuzzy	99.96 ± 0.03	99.95 ± 0.02	99.90 ± 0.07	99.89 ± 0.05	99.86 ± 0.11
CGGD	99.97 ± 0.02	99.98 ± 0.02	99.94 ± 0.06	99.96 ± 0.04	99.96 ± 0.04

Results

Data set: *bias correction**

- Metric: mean squared error

Constraints:

- $T_{\min}^{(t+1)} \leq T_{\max}^{(t+1)}$
- $T_{\min}^{(t+1)} \in [a, b]$
- $T_{\max}^{(t+1)} \in [a, b]$

Method	Size used for training relative to full training set (%)				
	100	75	50	25	10
Baseline	0.5998 ± 0.5856	0.6687 ± 0.4934	0.6930 ± 0.5330	0.7267 ± 0.5437	0.7441 ± 0.5593
Fuzzy	$0.0020 \pm 9.8e-5$	0.0023 ± 0.0002	0.0026 ± 0.0003	0.0062 ± 0.0042	0.0129 ± 0.0049
CGGD	$0.0020 \pm 7.4e-5$	$0.0022 \pm 6.9e-5$	0.0027 ± 0.0002	0.0044 ± 0.0026	0.0079 ± 0.0084

- Metric: satisfaction ratio = #satisfied / #constraints

Method	Size used for training relative to full training set (%)				
	100	75	50	25	10
Baseline	80.64 ± 16.24	75.96 ± 10.27	75.90 ± 10.71	75.20 ± 10.17	74.90 ± 10.21
Fuzzy	100.00 ± 0.00	100.00 ± 0.00	99.96 ± 0.08	99.82 ± 0.16	99.52 ± 0.48
CGGD	100.00 ± 0.00	100.00 ± 0.00	99.94 ± 0.12	99.96 ± 0.05	99.96 ± 0.05

*Dongjin Cho et al. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science*, 7, 4 2020 .

Quinten Van Baelen et al., "Constraint Guided Gradient Descent: Guided Training with Inequality Constraints", internal report, 2022.

Conclusion

Computational limitations (TinyML)

- QLIP: a method to jointly do pruning and quantisation
Reduction of BOPs up to 85% without loss in classification performance
- Outperforms traditional methods, similar performance to the state-of-the-art.
However, proposed method targets a **predefined cost**.
- Quantisation is favoured over pruning up to 70% relative BOP reduction
- First and last layer are the least affected*
- Activations are quantised at lower precision than the weights*

Conclusion

Flexible supervision

- Proposed CGGD: supervision during learning (for regression task) is guided by constraints
- Enables incorporating non-linear inequality constraints without losing crispness
- Does not require careful balancing a loss with a regularization term
- Offers perspective to reduce the need for annotated data
- Perspective to reduce the impact of weight initialization on performance

Future

Generalize and improve CGGD procedure

- Enable knowledge integration for different types of background knowledge (inequalities, PDE's, probabilistic rules)
- Validate for classification and regression tasks