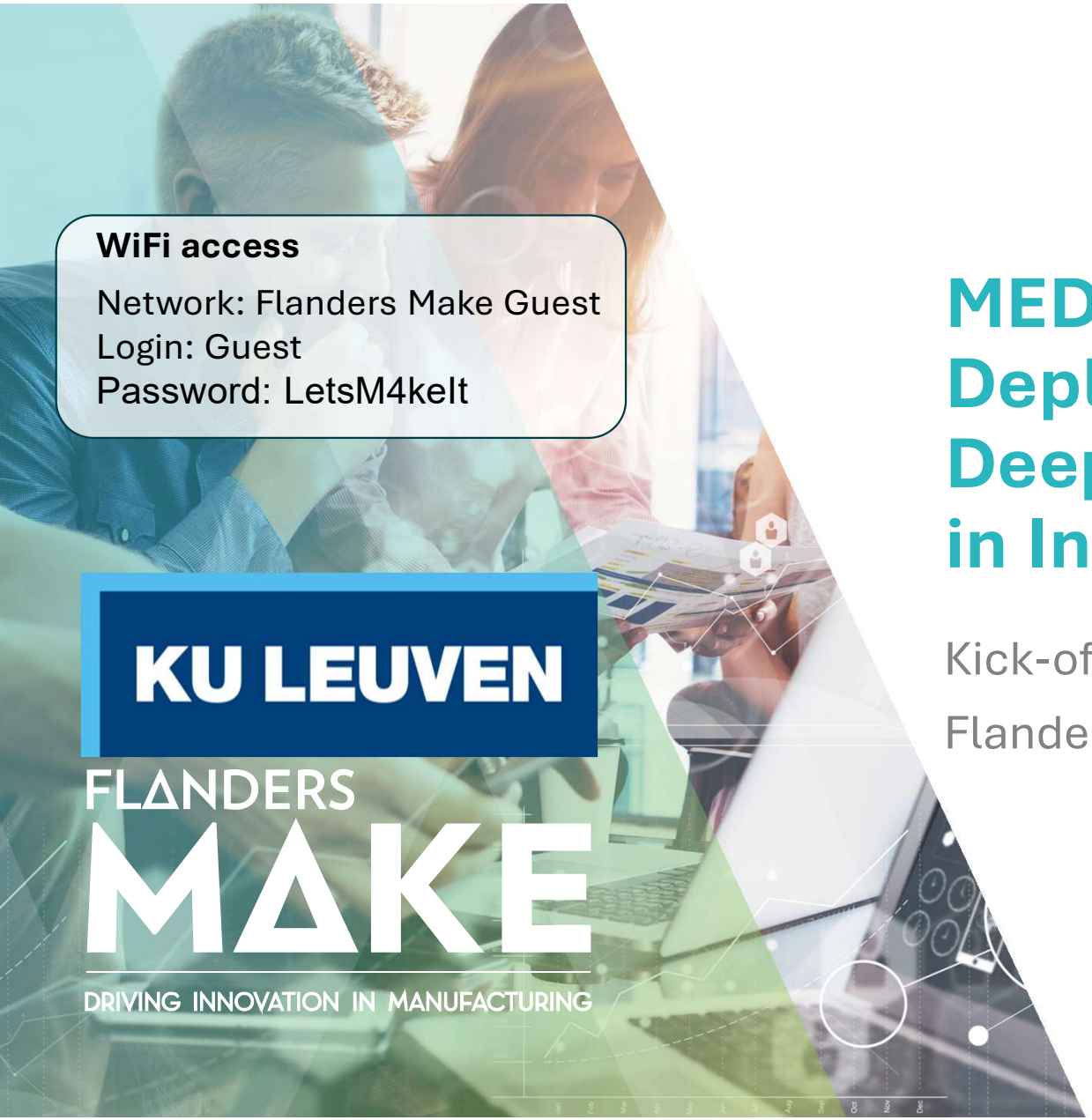


WiFi access

Network: Flanders Make Guest
Login: Guest
Password: LetsM4kelt

KU LEUVEN

FLANDERS MAKE

DRIVING INNOVATION IN MANUFACTURING

VLAIO

MEDLI: Managing Edge Deployment of Large Deep Learning Models in Industry

Kick-off meeting 18/03/2025

Flanders Make Leuven

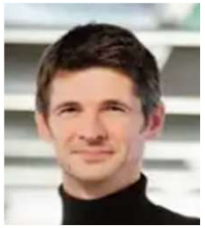
DTAI

Declarative Languages & Artificial Intelligence

Welcome to the MEDLI kick off meeting

Start: 1 March 2025 - Duration: 2 years

Peter Karsmakers



DTAI FLANDERS MAKE@ KU LEUVEN

Daan Luyckx

Lode Vuegen

Marjolein Deryck

Abdel Bey Temsamani

FLANDERS MAKE

<u>ProductionS (Vision AI)</u>

Cyril Blanc

Patrizio Perugini

Emilio Gamba

Ted Ooijevaar

FLANDERS MAKE

<u>MotionS</u>

Steven Michiels

Kerem Eryilmaz

User Group

- Act as an advisory and as a **sounding board** to explore the possibilities of economic implementation of the reusable results
- **User group meetings, 2x/year**: to provide information about project status, and to collect feedback from the companies
- Follow up activities

User group members

Sound Talks

Wim Buyens
Sreenivasa Upadhyaya

ADDAX

Koen Verhellen

RideOnTrack

Daan Lochs
Ferre Sneyers

DAIKIN

Albert Rosich

CybridThrust

Matteo Baronio
Thibaut Gerard
Sebastien Depret

MAGICS

Gert Dekkers
Toon Vinck

ZF

Florin Tatar

BMSvision
Empowering your productivity

Pierre van den Bulck
Sam Decoster
Jan Bogaert

vntecc

Glen Aesaert

CTRL engineering

Mathieu Dutré

Pickit
robot vision made easy

Timothée Habra

Sentigrate

Thomas De Moor

REYNAERS aluminium

Svetlana Ebzeeva
David Deschutter
Damien Hoyen

CONTEC
Industrial automation solutions

Peter Van Mieghem
Nick Leus

televic healthcare

Pieter Crombez
Kristof Tjonck

ikologik

Tom Neels

Dockflow

Michiel Valee

GOODiX

Xueru Zhang

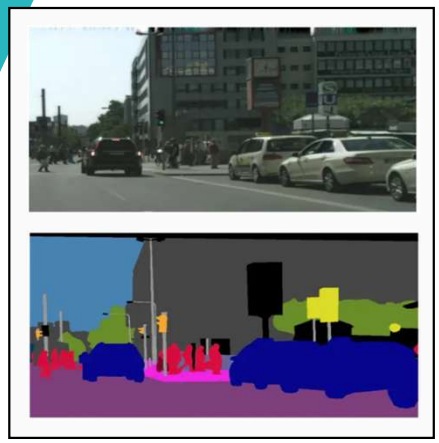
Agenda

13u30	Welcome & meeting objectives
13u40	Overall project goals
14u25	Tutorial : From pretrained model to edge deployment
14u55	Coffee break
15u05	Demo's: Bearing failure monitoring & edge tower
16u05	Knowledge transfer & implementation
16u35	Planning & next steps
16u50	Closing
17u	Reception

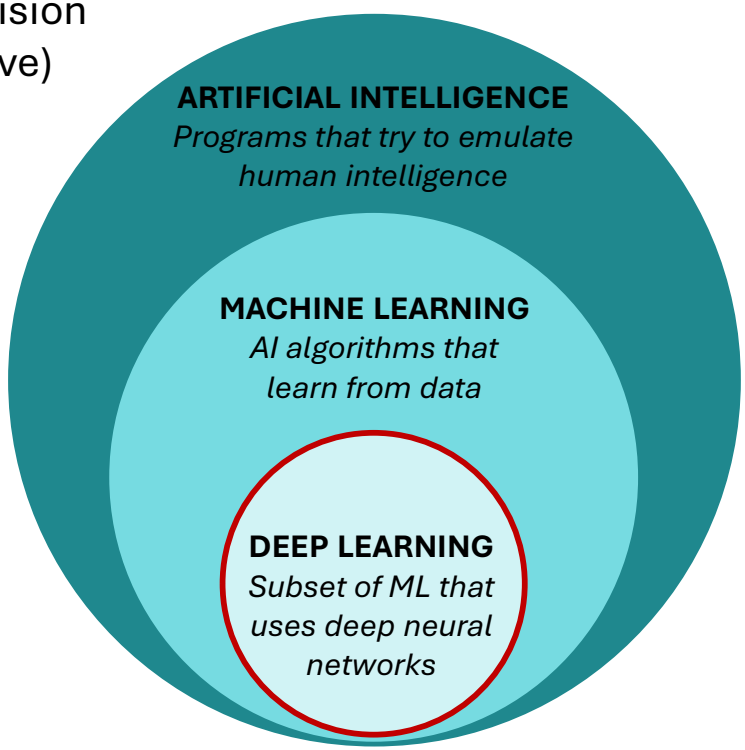
Overview

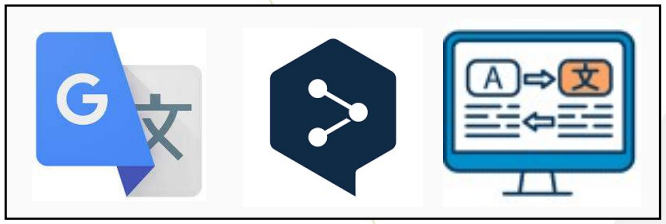
- Briefly revisit deep learning
- Observed trends
- MEDLI challenges and goals

Artificial Intelligence

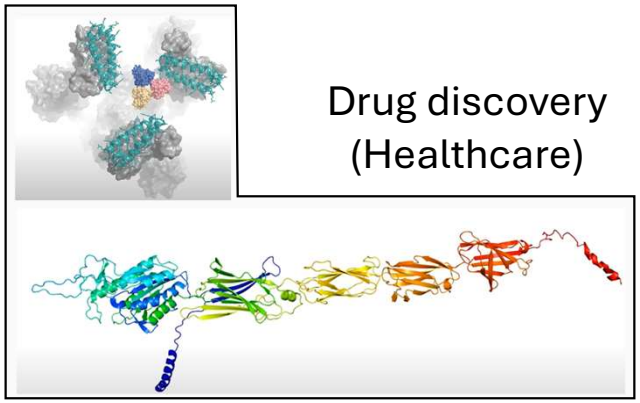


Computer vision
(Automotive)



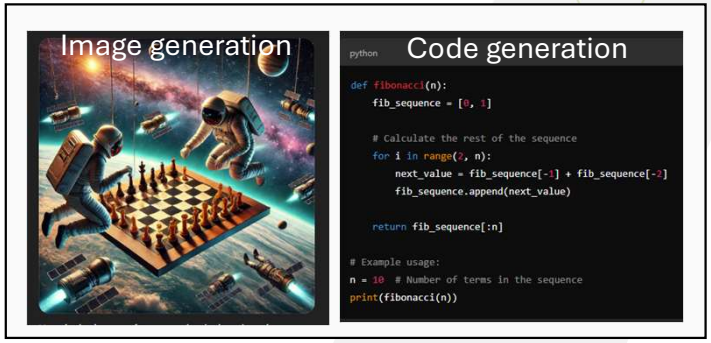


Natural Language Processing



Drug discovery
(Healthcare)

Chatbots and Virtual Assistants



What Makes Deep Learning Succesful?

learned using data

handcrafted

traditional machine learning

Handcrafted feature extraction

Classifier

Bottle

Not a bottle

deep learning

Feature extractor + classifier

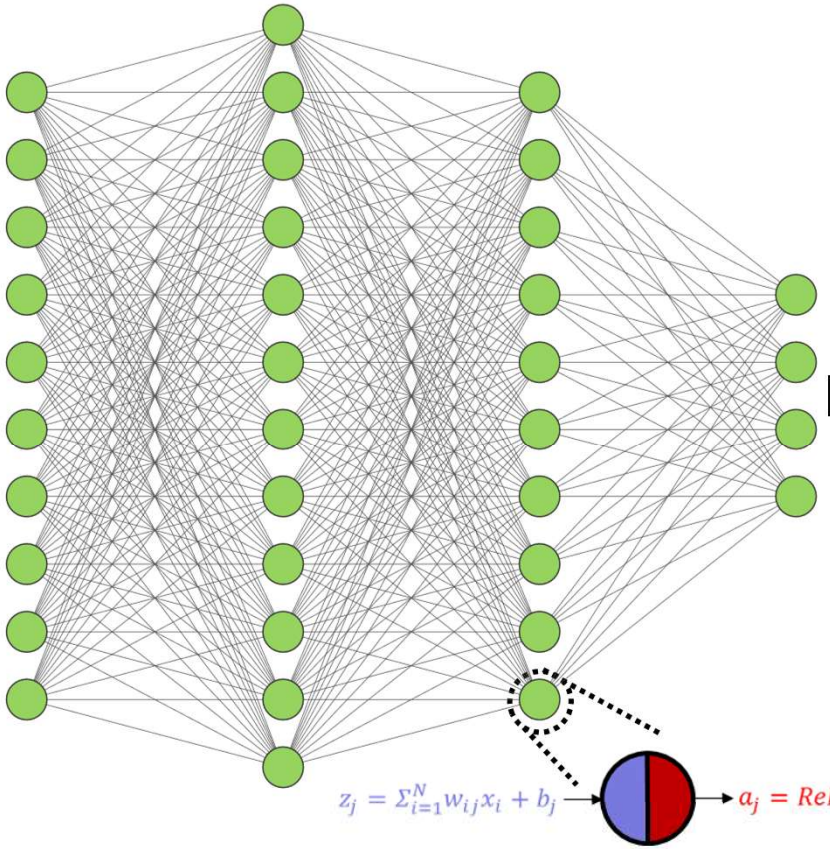
Bottle

Not a bottle

Deep Learning

Inputs

Deep Neural Network (DNN)



Outputs

0, 1, 2, …9

- Inputs
  - Images
  - Audio
  - Text (language)
  - ..

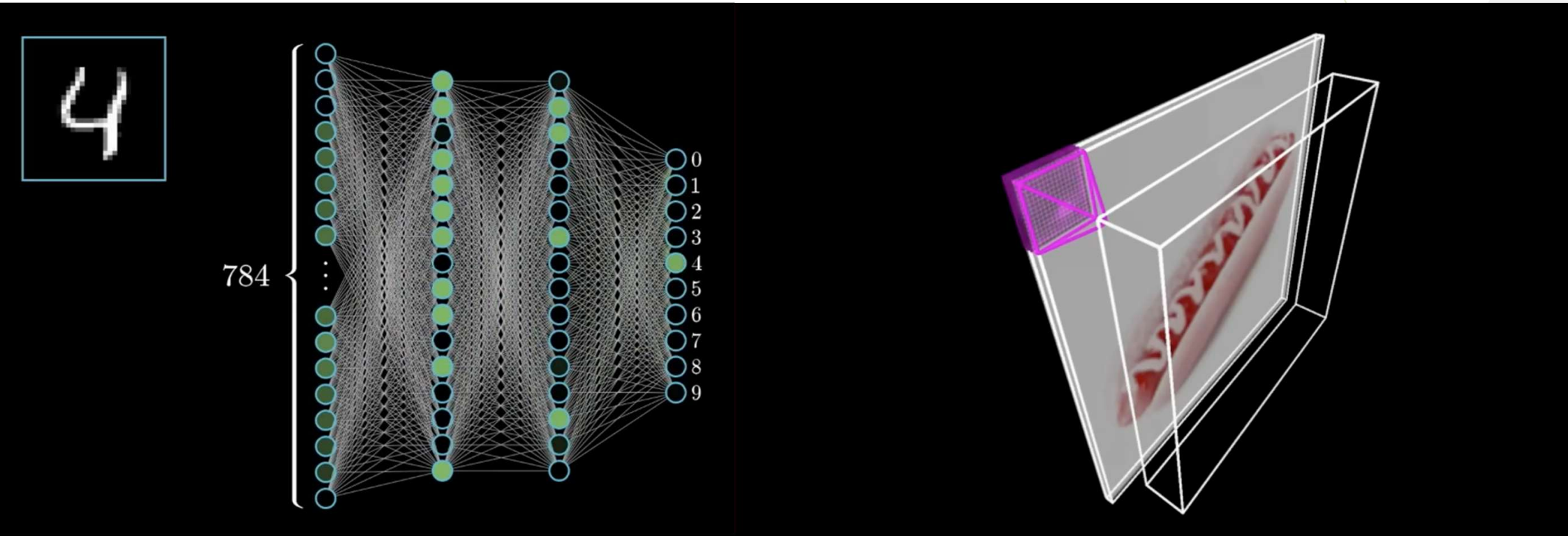
- Outputs
  - Classification probabilities
  - Text (language)
  - ..

Neuron

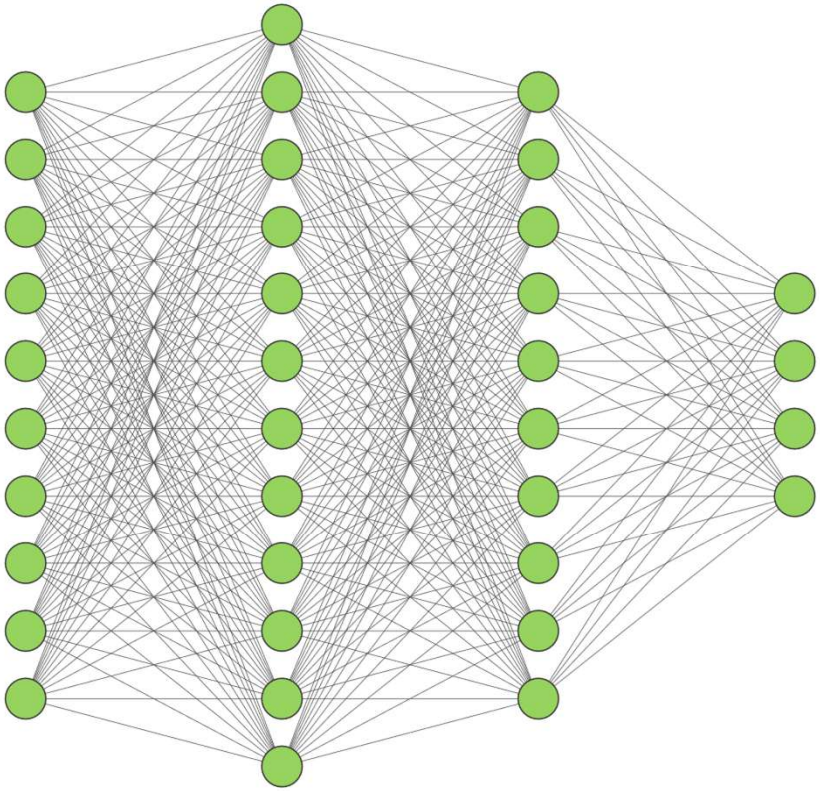
- Weighted sum of its inputs
- Non-linear activation

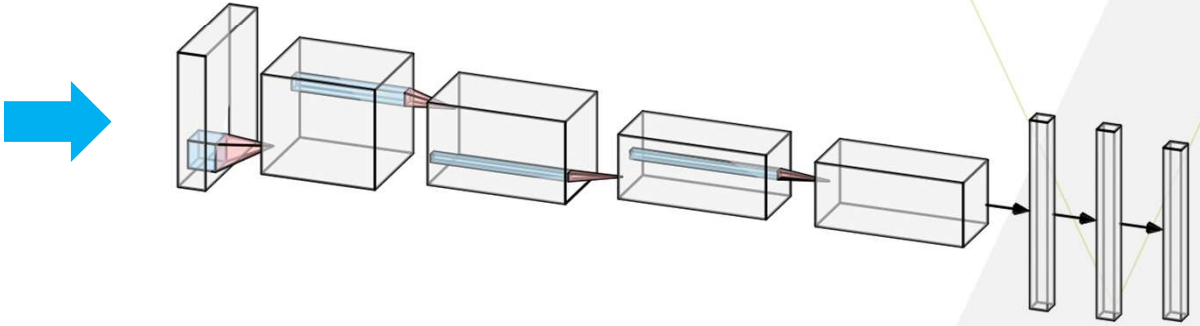
Deep Learning

- DNN inference
  - Forward propagate inputs through the DNN and predict outputs

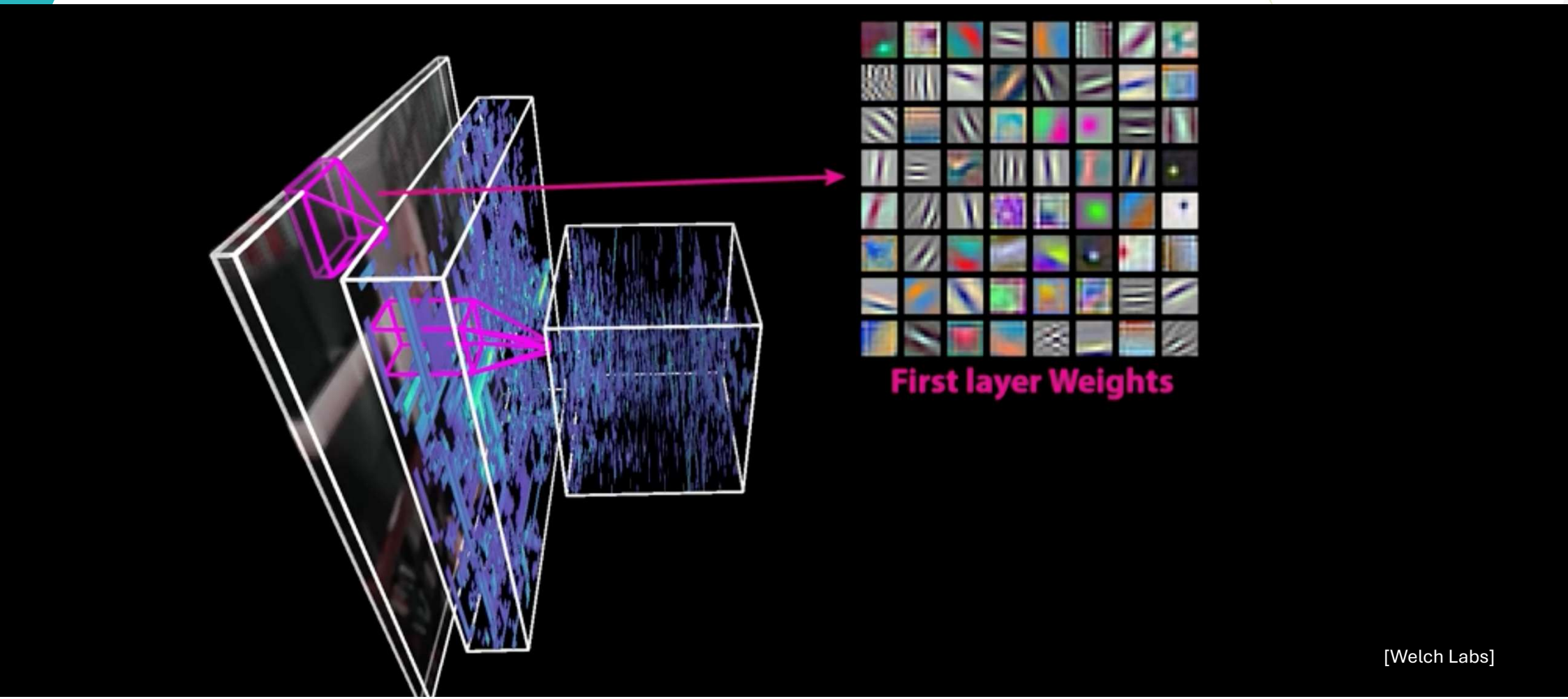


Deep Learning

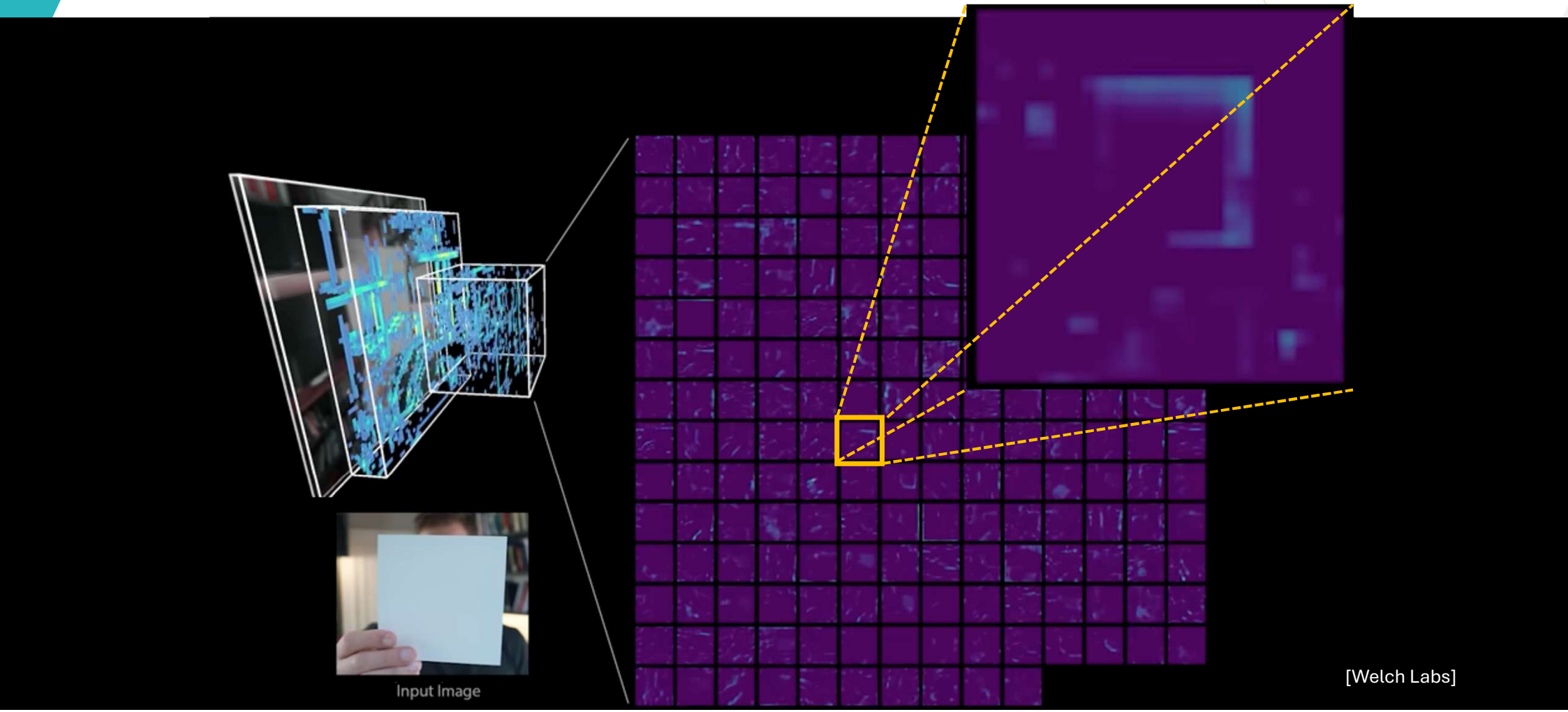




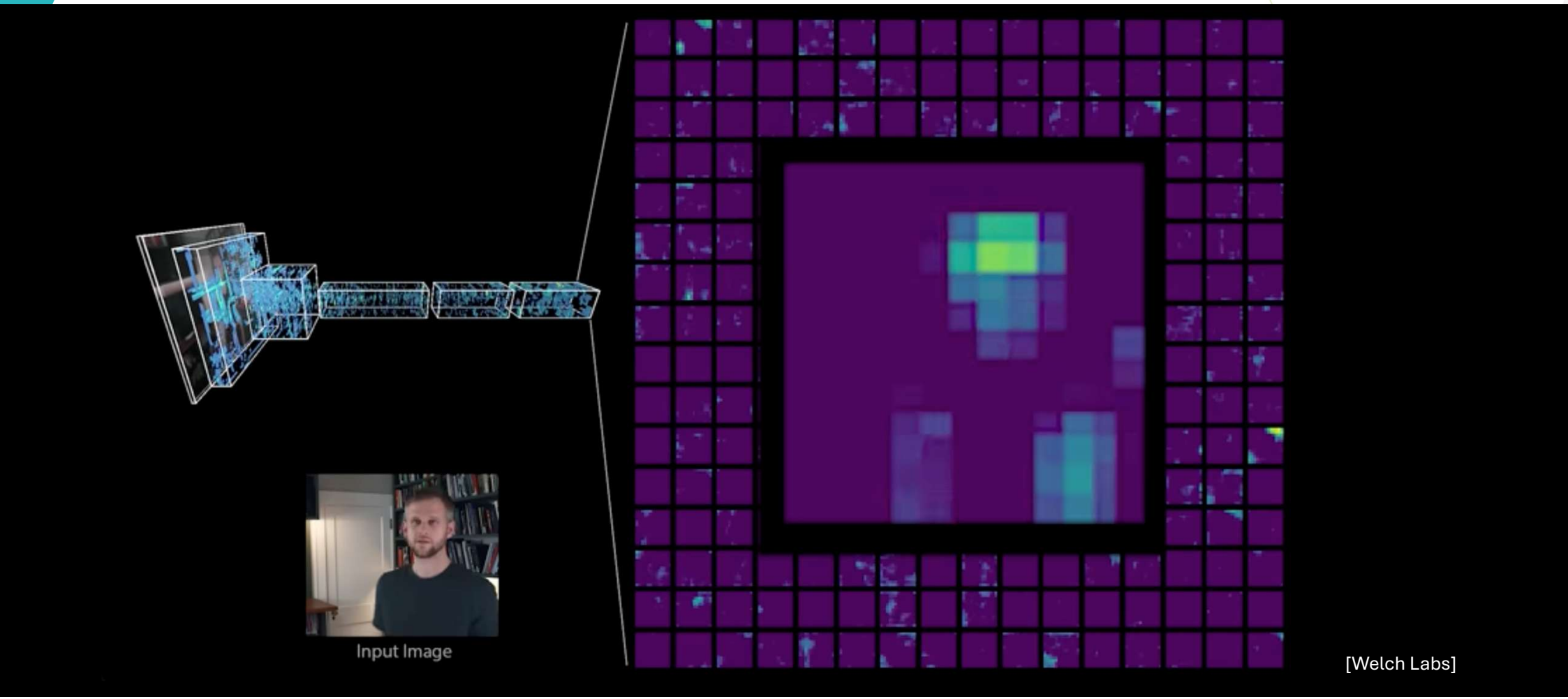
Deep learning: Interpretation Early Layer Processing



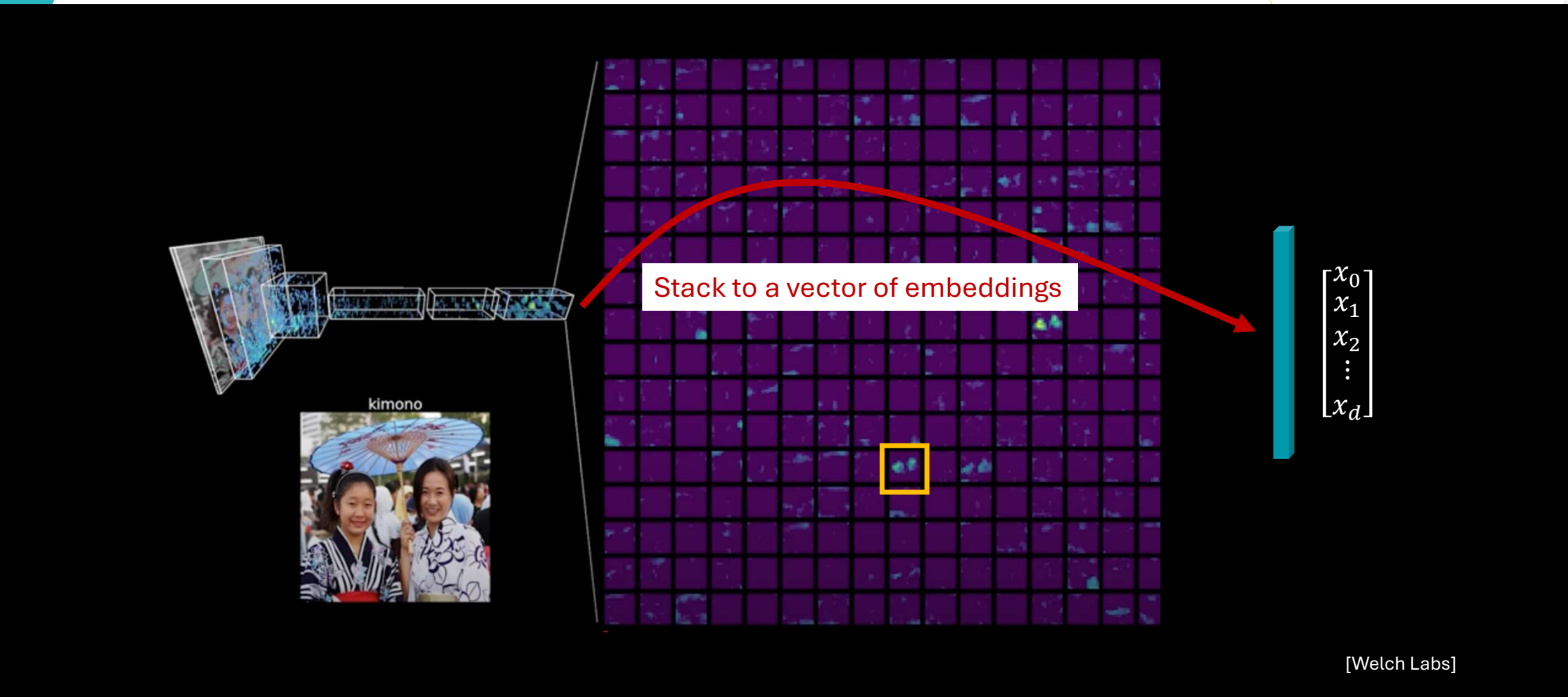
Deep learning: Interpretation **Early** Layer Processing



Deep learning: Interpretation Higher Layer Processing

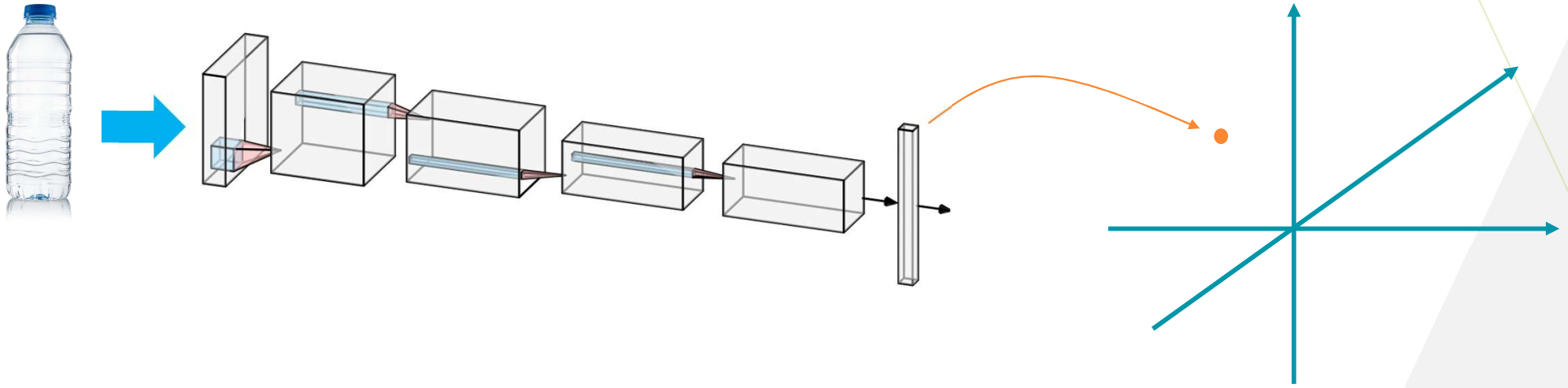


Deep learning: Interpretation **Higher** Layer Processing

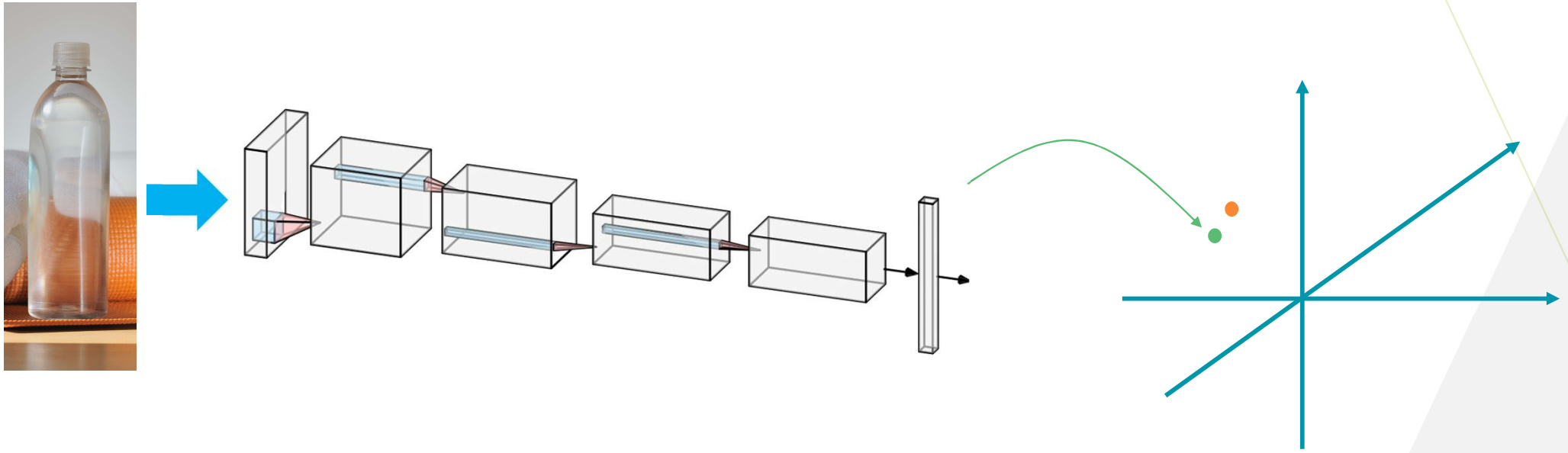


[Welch Labs]

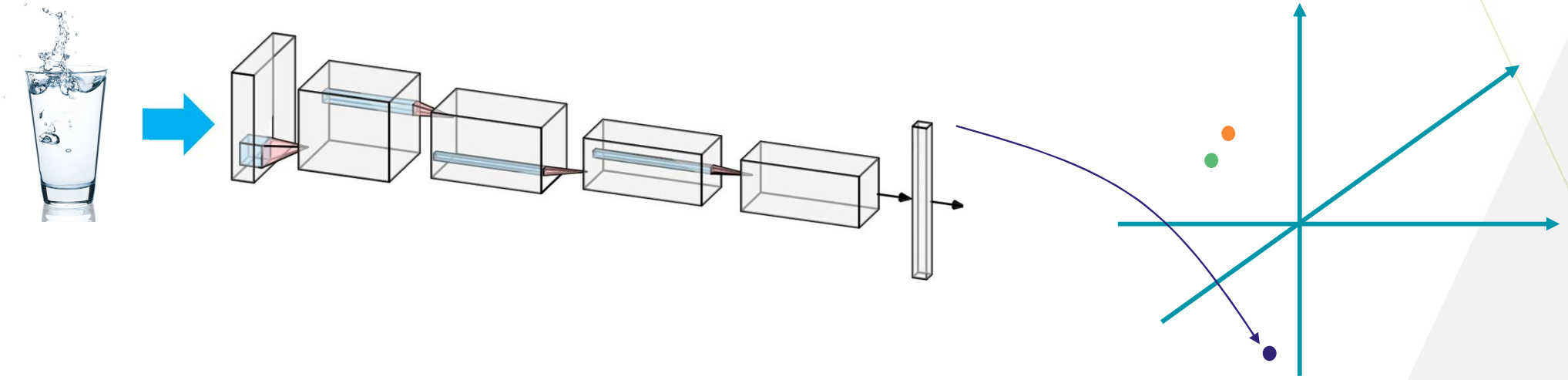
Deep learning: Interpretation Higher Layer Processing



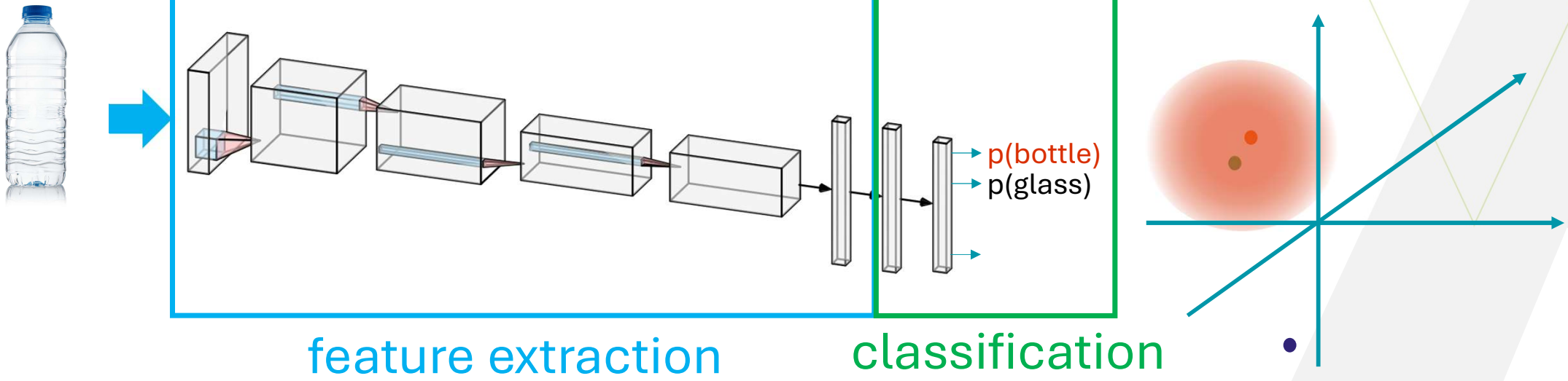
Deep learning: Interpretation Higher Layer Processing



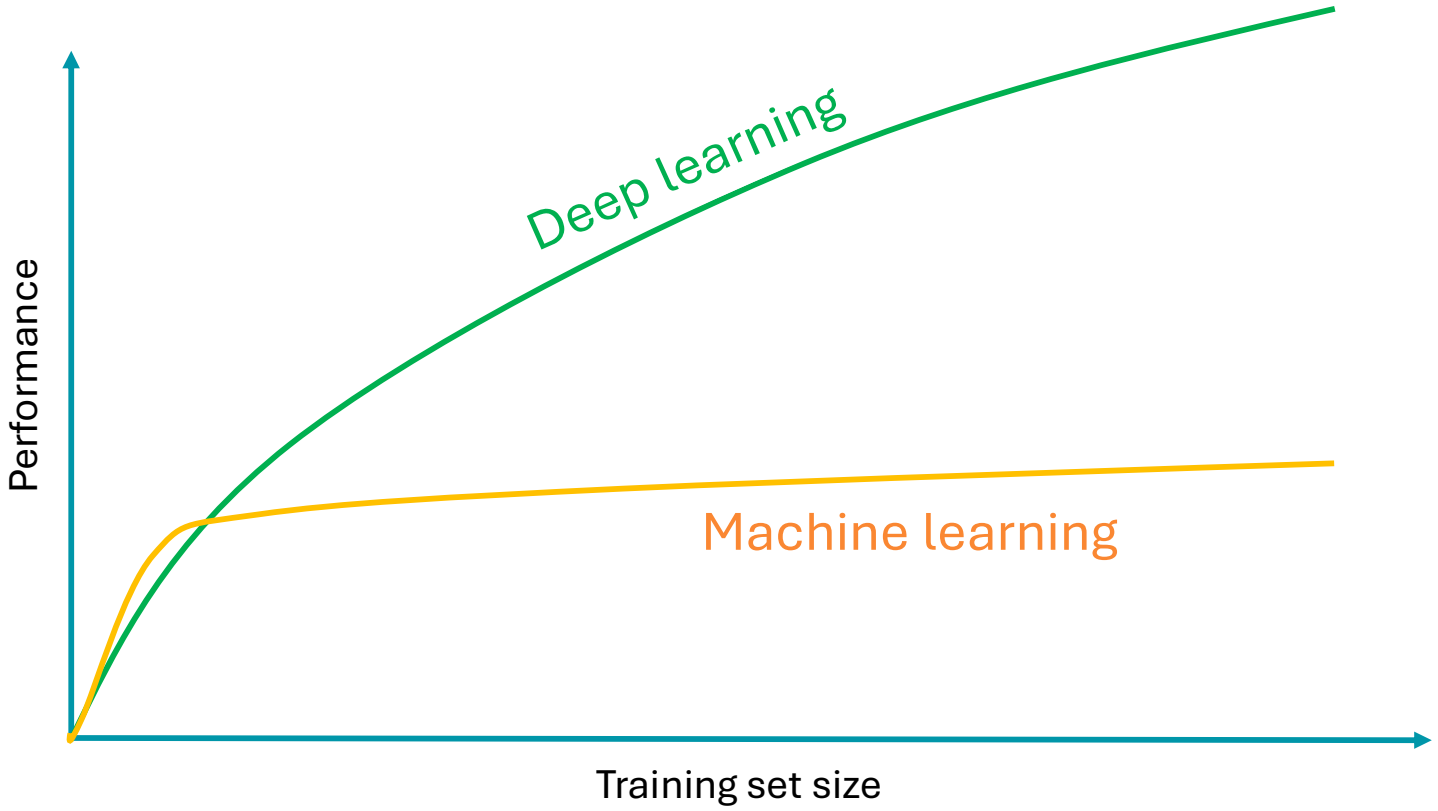
Deep learning: Interpretation Higher Layer Processing



Deep learning: Interpretation **Higher** Layer Processing

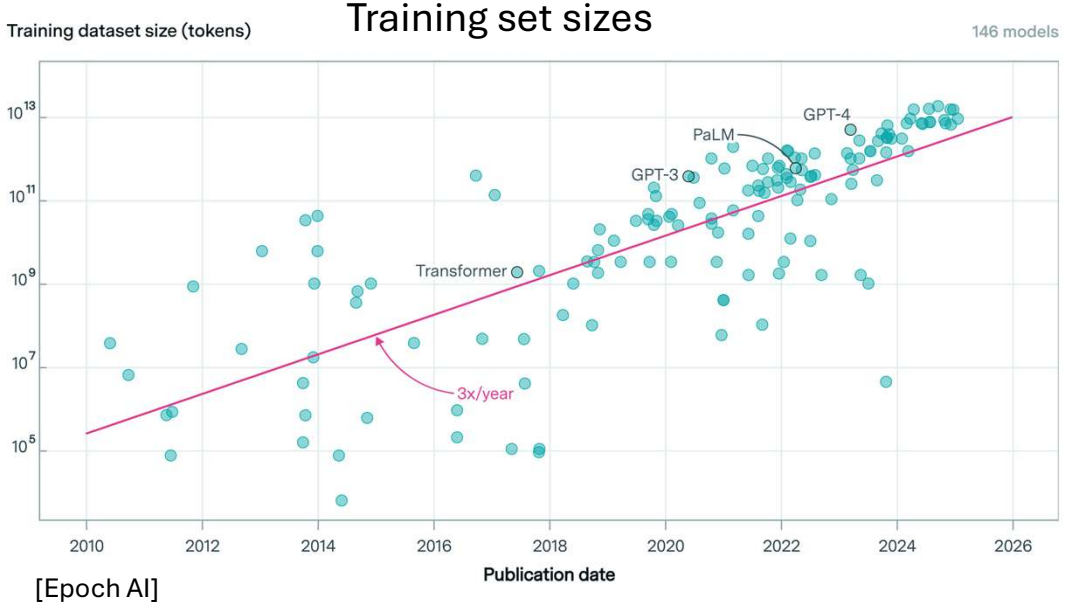


Deep Learning: Training Set Size & Model Size



Deep Learning: Training Set Size & Model Size

Training set sizes

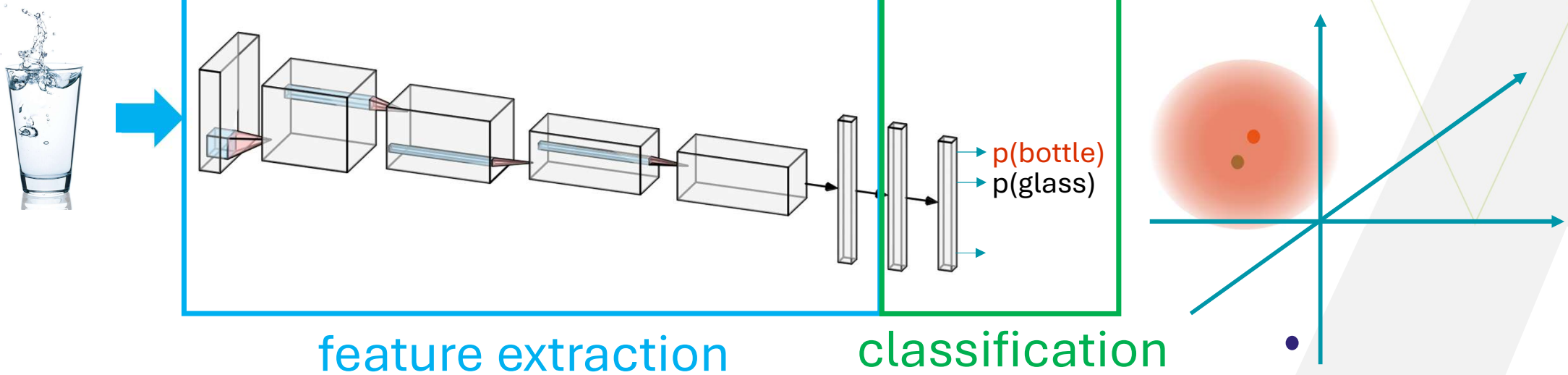


[Epoch AI]

For many applications there is no large training set size!

Deep Learning: Training Set Size & Model Size

Learn this on large data set and reuse it for similar applications for which there is limited data

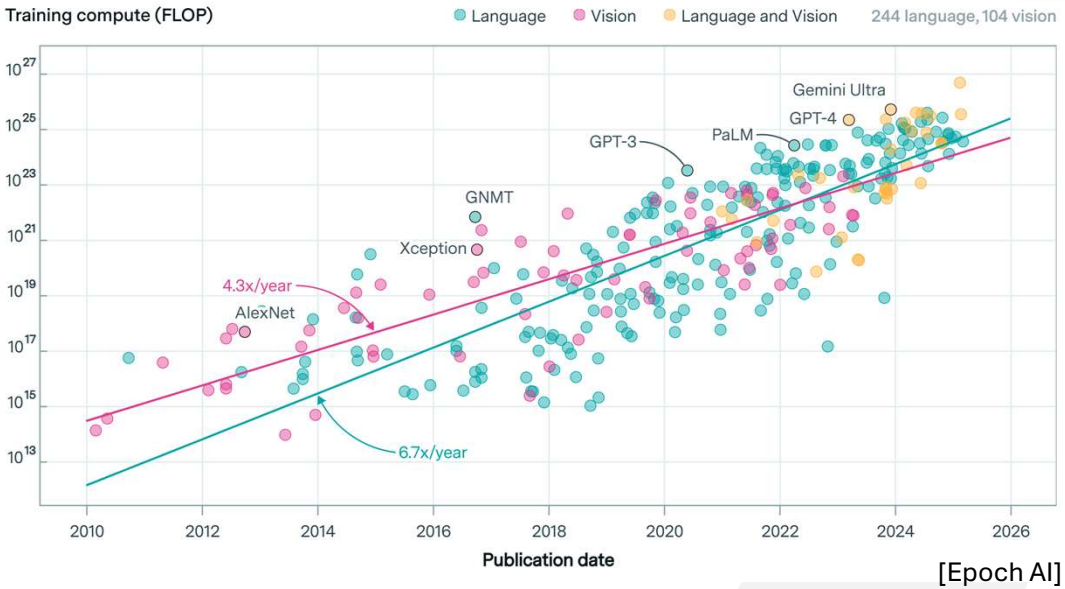


TRANSFER LEARNING: Design of DL models using modern large pre-trained models that reduce the need for training data

Deep Learning: Training Set Size & Model Size

Training set sizes

Model sizes

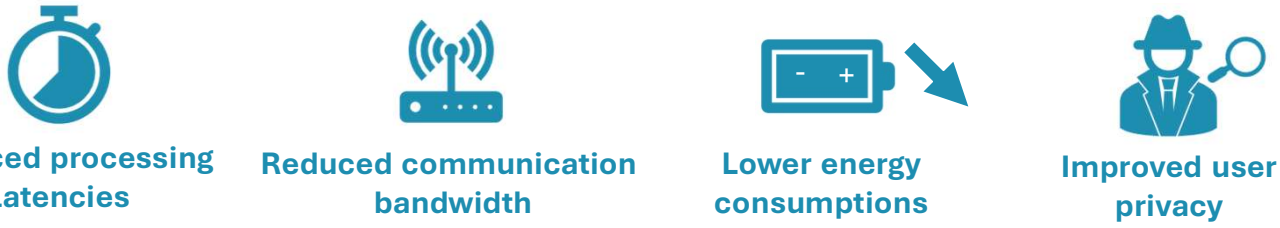


[Epoch AI]

Larger model sizes require more computing resources

Deep Learning: Cloud to Edge

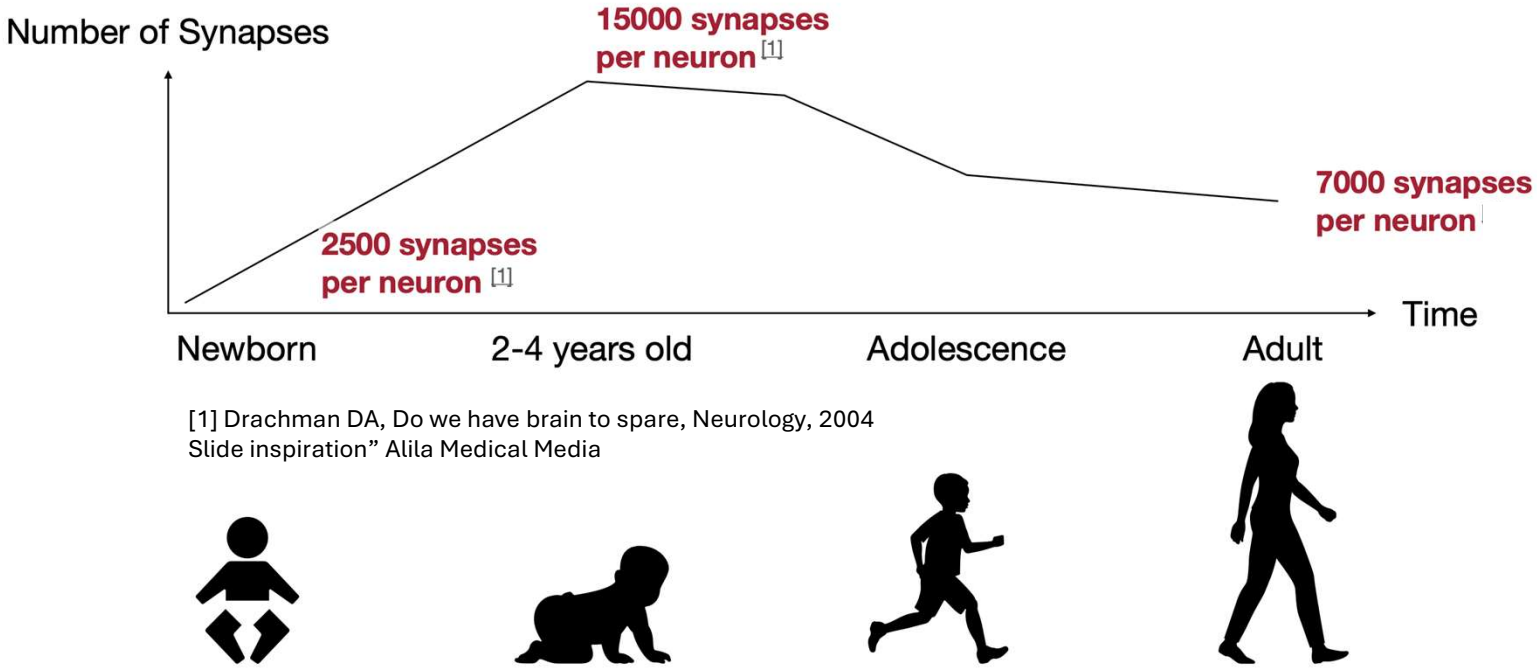
- Shifting from cloud to edge



- Limited computing resources available at the edge

Deep Learning: Learning vs Inference

- Large models are required for learning but not necessarily for inference

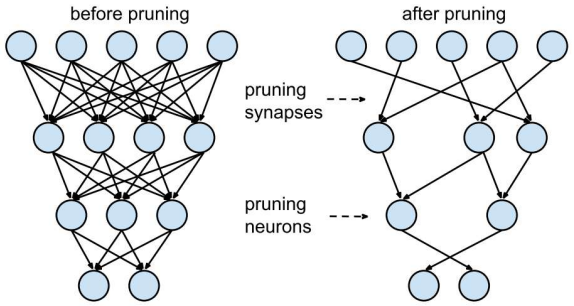


[1] Drachman DA, Do we have brain to spare, Neurology, 2004
Slide inspiration" Alila Medical Media

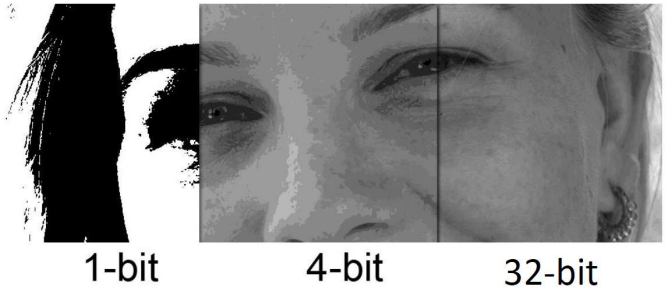
→ potential to do inference with less complex models

Deep Learning: Model Compression

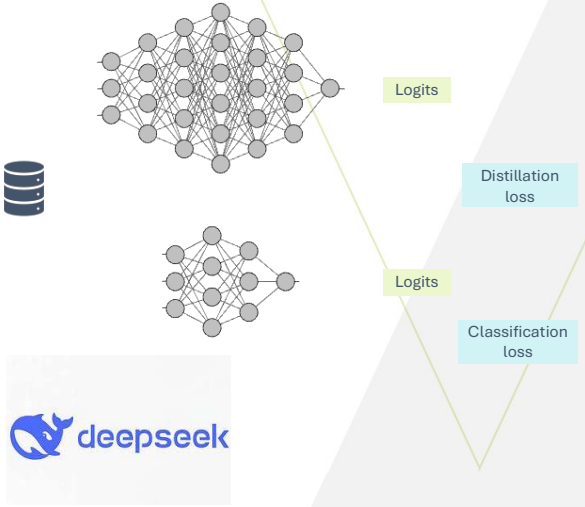
- Compress model by
  - Pruning:



  - Quantisation:

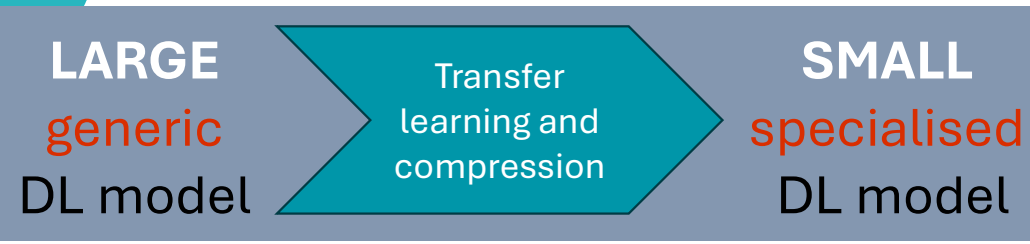


Knowledge Distillation:



MODEL COMPRESSION: Compress large DL models to (without or with limited accuracy drop) fit edge HW

MEDLI Challenge 1

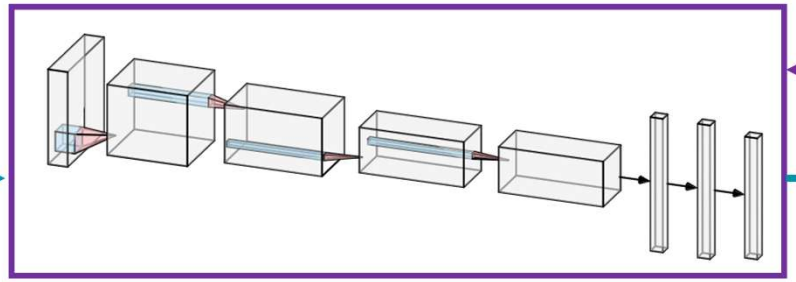


- **Challenge 1**: How to "easily" jointly transfer and compress the large pretrained model to a smaller alternative that can be deployed on the edge, with limited loss of model accuracy?

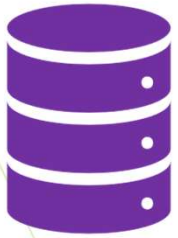
MEDLI Approach 1: Joint Specialisation and Compression

Custom small training data set

Assume this is available



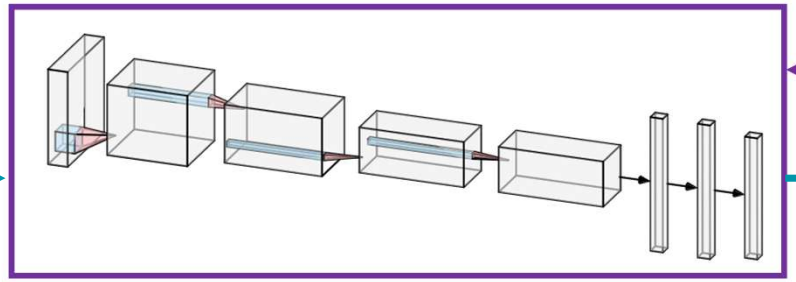
Cap
No cap



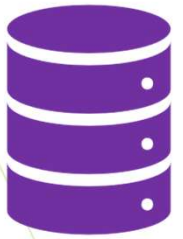
DB of pretrained models

MEDLI Approach 1: Joint Specialisation and Compression

Custom small training data set

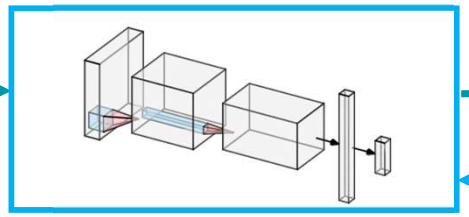


Cap
No cap

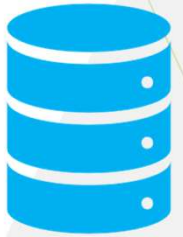


DB of pretrained models

Joint specialisation + knowledge distillation



Cap
No cap



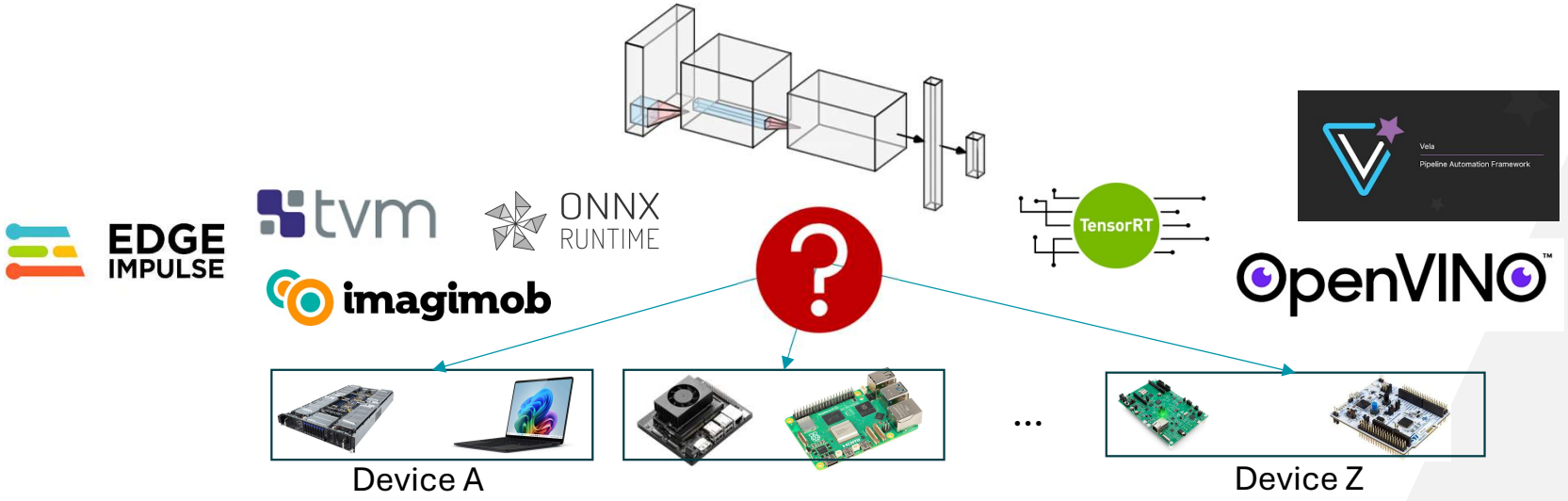
DB of small model architectures

MEDLI Goal 1: *Reduce the design time for edge AI models*

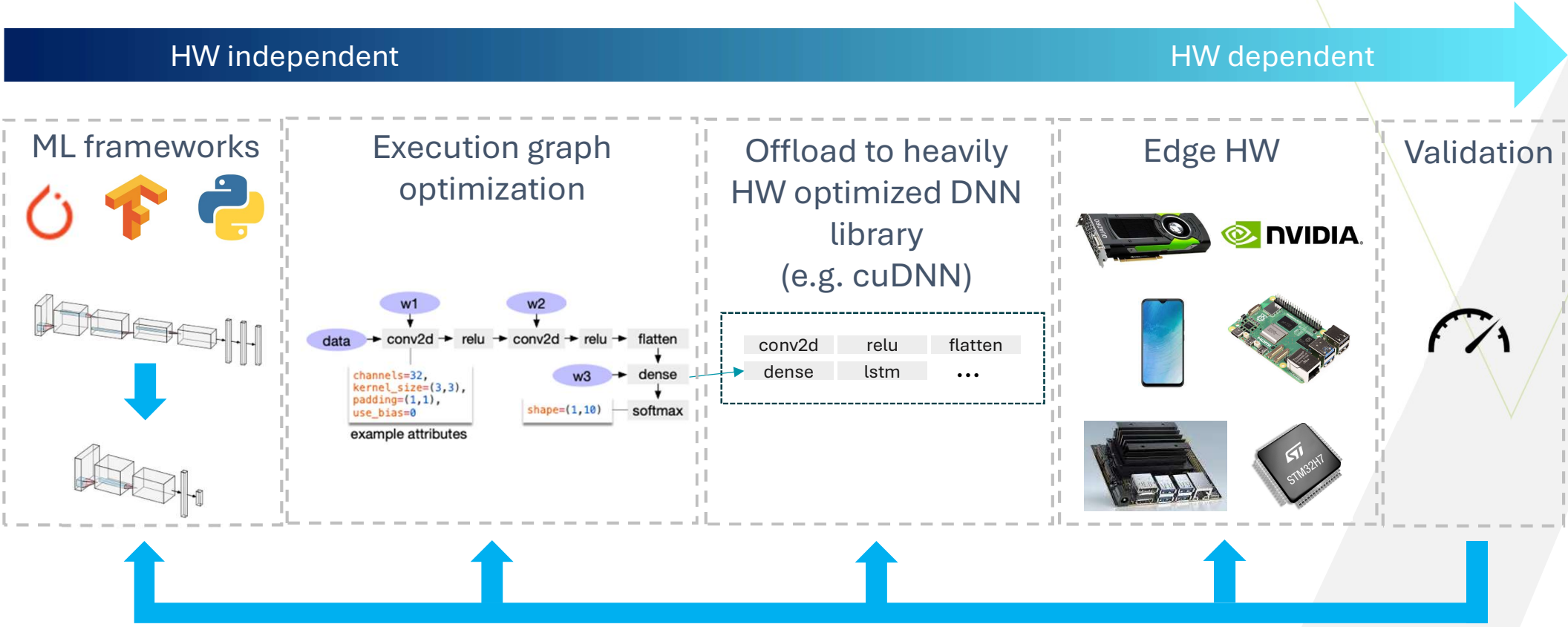
MEDLI Challenge 2

LARGE generic DL model → Transfer learning and knowledge distillation → **SMALL** specialised DL model → Optimization & edge HW deployment

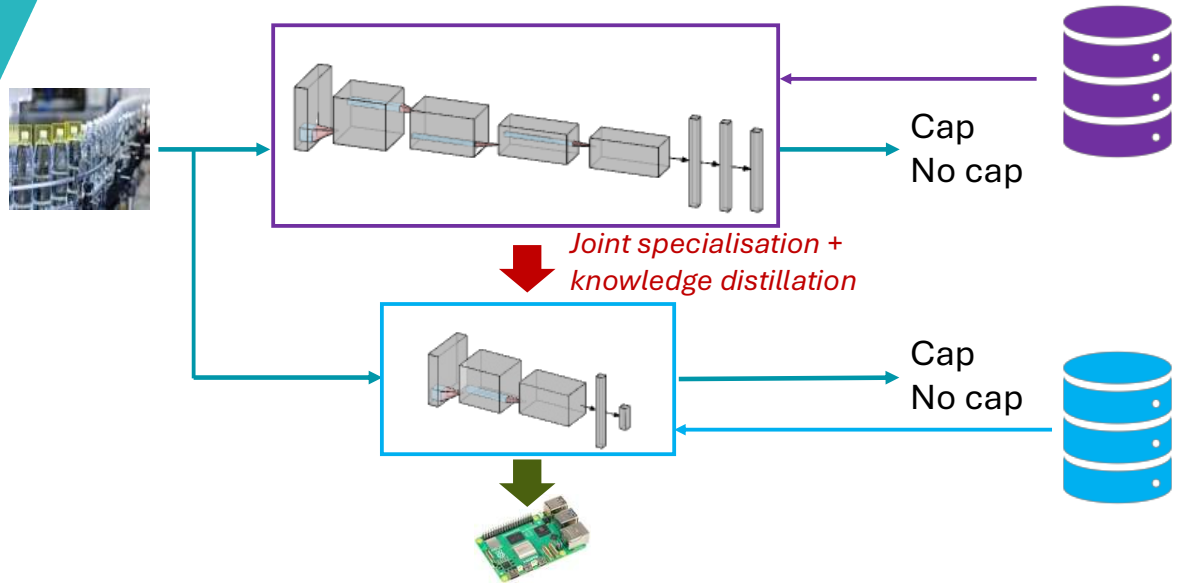
- **Challenge 2**: Which HW and/or SW ecosystem to select to deploy small model on the edge?



MEDLI Challenge 2



Approach 2: Compression to Model Architecture with Known Computational Specifications



Lottery Ticket Model Database

Parameters:				
Use case:	Images	Sound	...	Text
Latency on device:				
A:	100ms	60ms	...	200ms
B:	150ms	80ms	...	220ms
C:	250ms	120ms	...	260ms

MEDLI Goal 2:

- *Provide tools that facilitate the selection of a suitable combination of edge-HW and deployment tools for the intended application*
- *Make (and explain how to build a) DB of small model architectures a) that work well on certain tasks, b) that satisfy certain computational performance metrics on selected edge HW*

MEDLI Challenge 3

- **RQ3**: Out of the diverse set of choices, which edge model monitoring SW to select?

MEDLI Approach 3: Edge Model Monitoring

- **Monitoring edge models is hard**: With cloud inference, metrics are collected from a single endpoint, but with edge ML, each device should have own metrics (without access to targets) which need to be reported back
- Example tools:

nannyML

EVIDENTLY AI

ANOMALiB

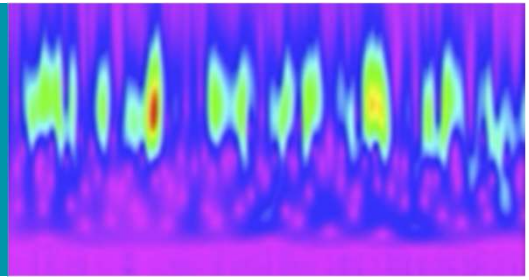
neptune.ai

Wallaroo

Azure Machine Learning

MEDLI Goal 3:
Provide overview of tools (and incl. a selection in demonstrators) for companies to monitor edge AI models when they are in use on an edge platform

Generic Use-cases

vision based object detection	
complex time-series based anomaly detection	

- **MEDLI Goal 4:** *At least two template flows that demonstrate a) how to specialize and compress a large pretrained model for edge deployment, b) how to monitor the edge model*

Summary

MEDLI Goals:

G1. *Giving companies the tools to reduce their design time for an edge AI model by up to 80%*

G2. *Provide tools that facilitate the selection of a suitable combination of edge-HW and deployment tools for the intended application*

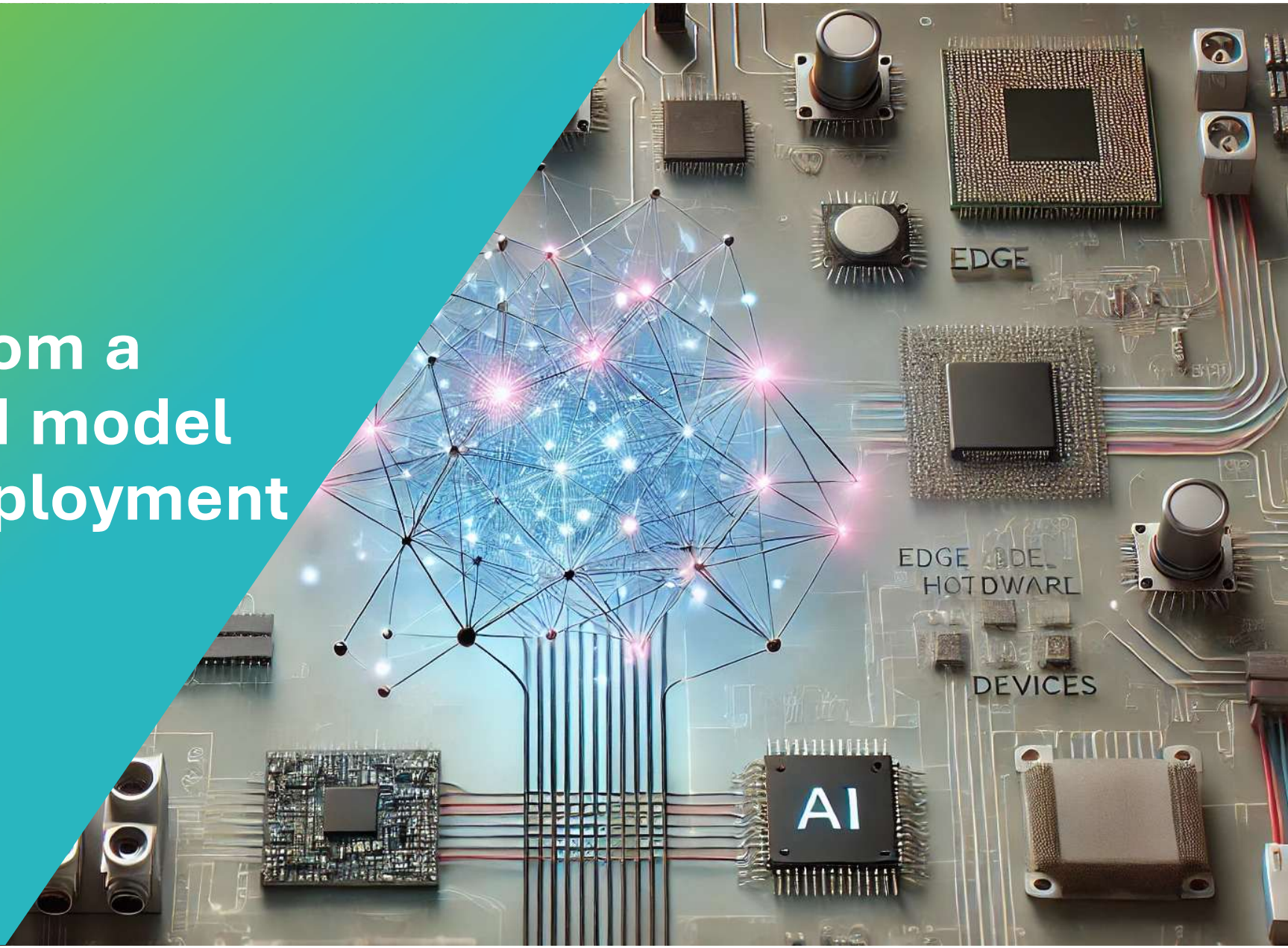
G3. *Provide tools for companies to observe edge AI models when they are in use on an edge platform*

G4. *Develop two generic case studies, linked to the manufacturing industry, with the suggested design approach and tools*

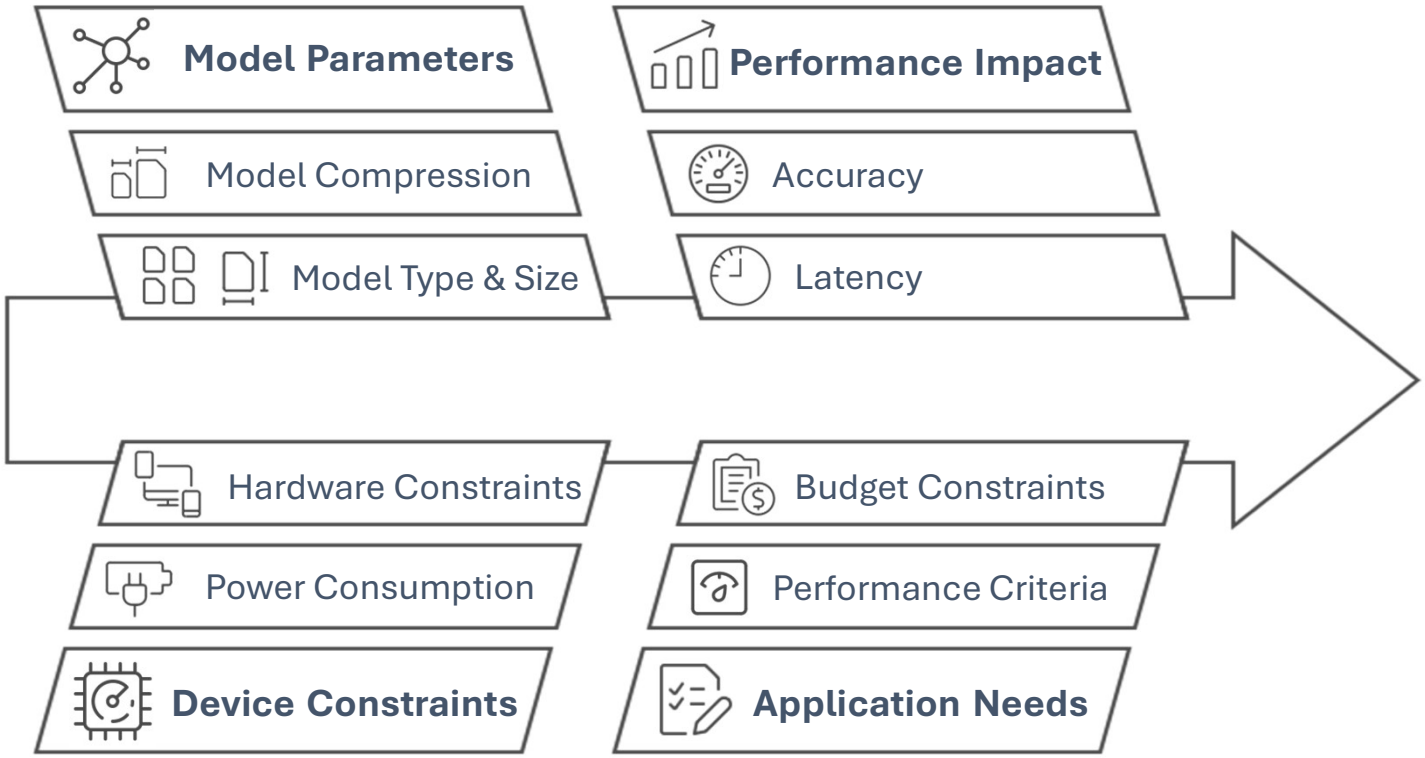
Agenda

13u30	Welcome & meeting objectives
13u40	Overall project goals
14u25	**Tutorial : From pretrained model to edge deployment**
14u55	Coffee break
15u05	Demo's: Bearing failure monitoring & edge tower
16u05	Knowledge transfer & implementation
16u35	Planning & next steps
16u50	Closing
17u	Reception

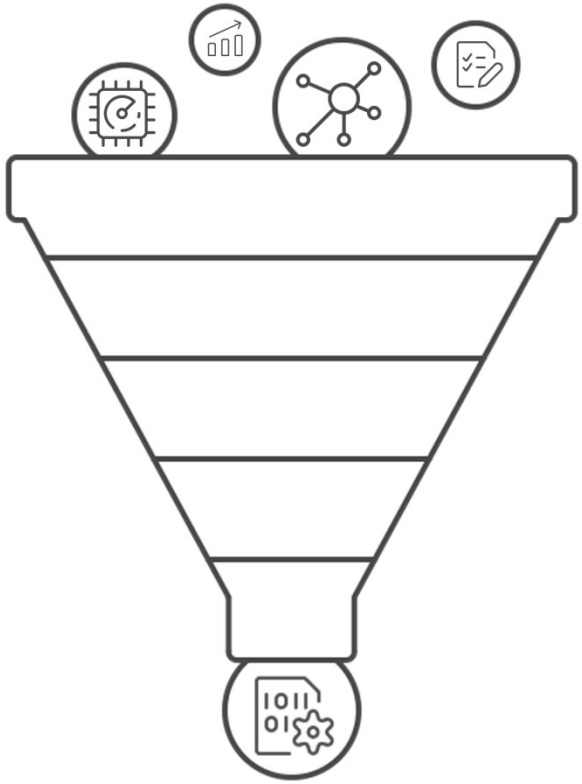
Tutorial: from a pre-trained model to edge deployment



Finding the right balance



How can you deploy AI models on the edge?



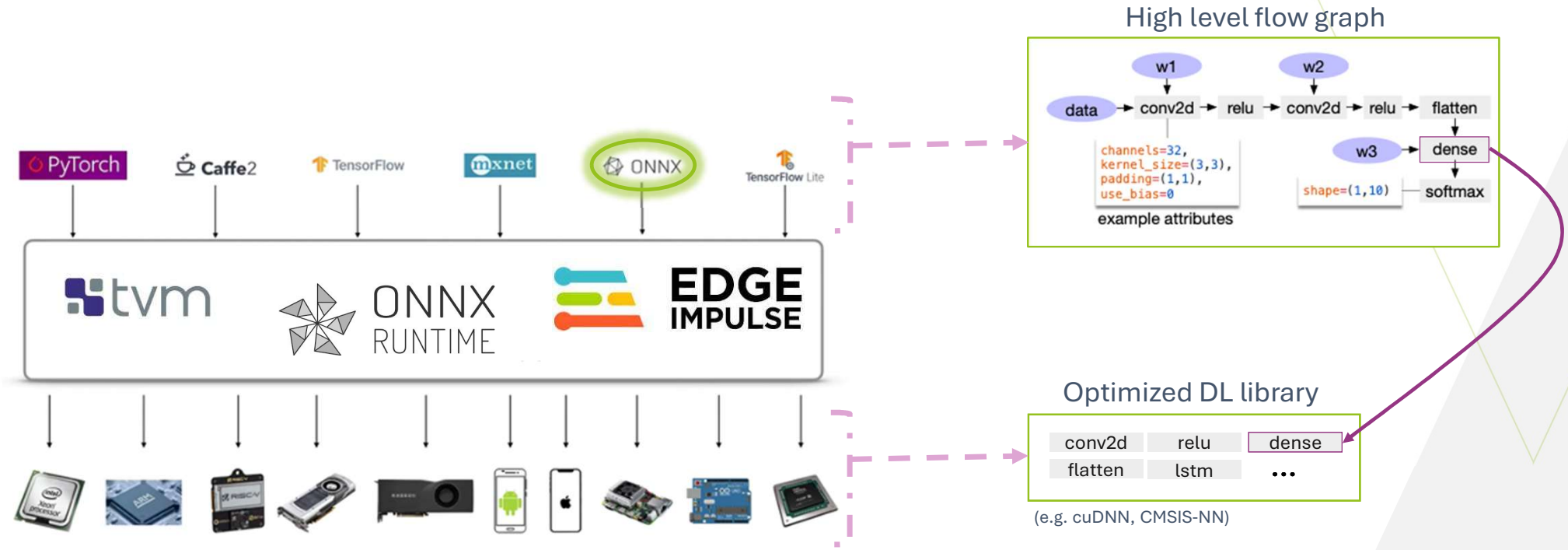
ML Compiler Framework

Optimizes and compiles DL models for edge deployment

How to choose the right framework?

How does the choice impact performance and/or other factors?

Why is an ML compiler framework needed?



Open Neural Network Exchange (ONNX)

ONNX

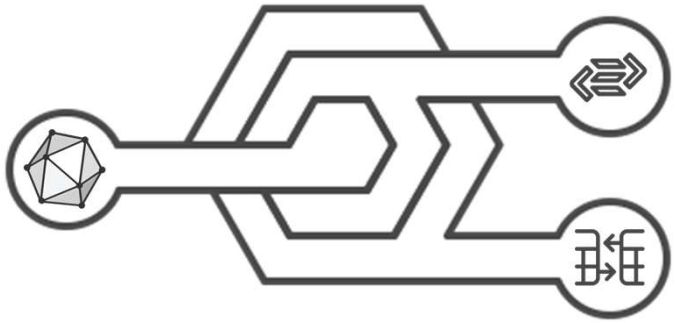
Open-source

Open standard

Founded by Facebook (PyTorch)

Supported by numerous companies:

- Facebook, Microsoft, IBM, Huawei
- Intel, AMD, Arm, Qualcomm



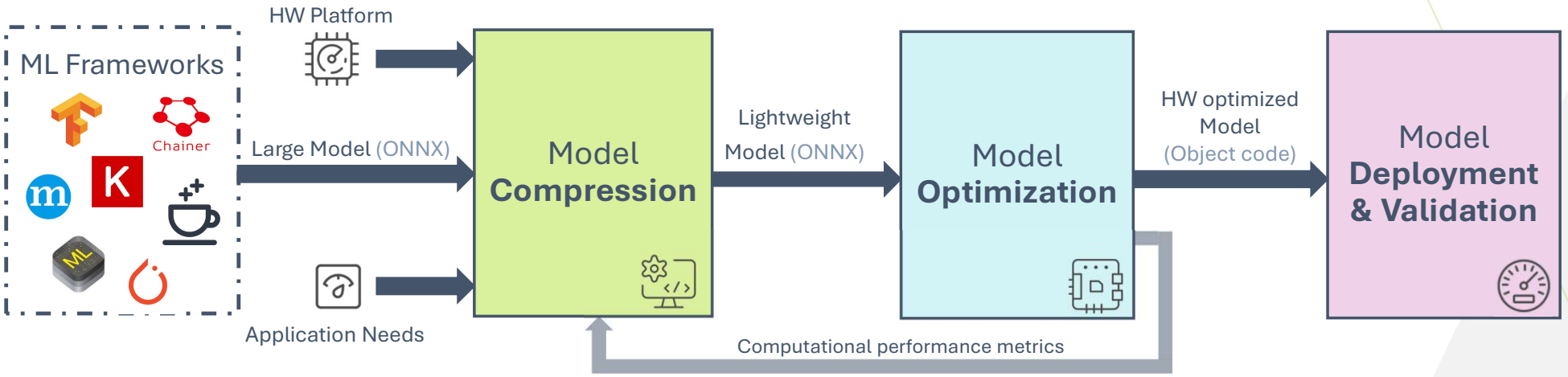
Framework Interoperability

Enables easy transitions between frameworks

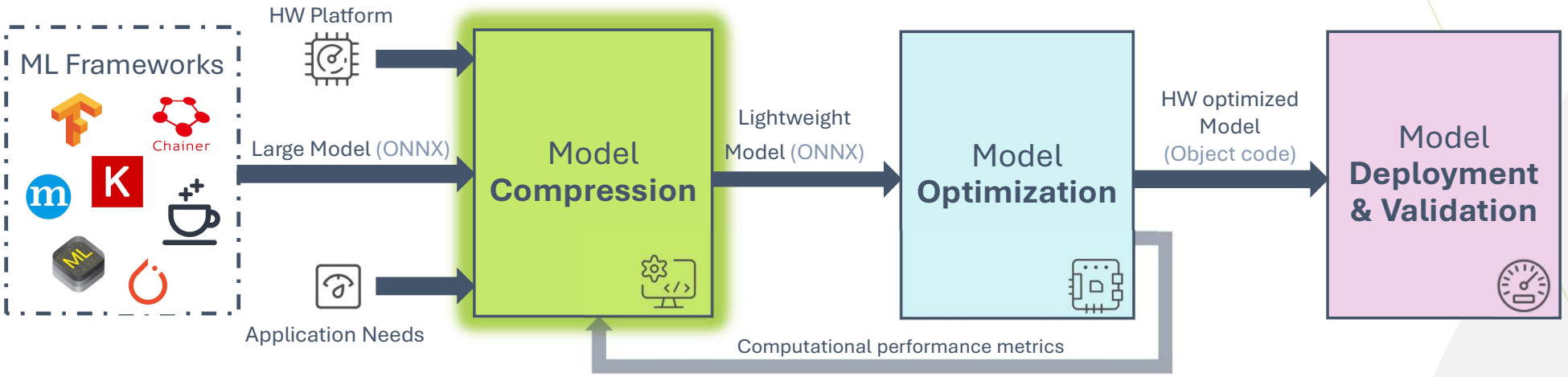
Shared Optimization

Allows hardware vendors to boost performance across multiple ML frameworks using ONNX

How does an ML compiler framework operate?



How does an ML compiler framework operate?

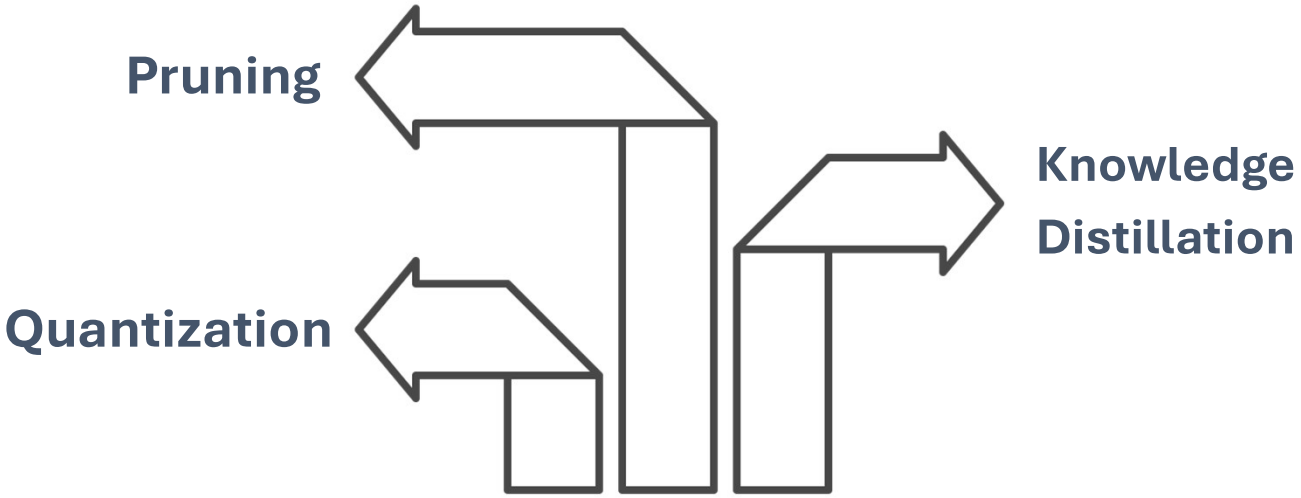


How does an ML compiler framework operate?

Aim for a more compact model without compromising performance

Concentrate on three key technique families:

Model
Compression

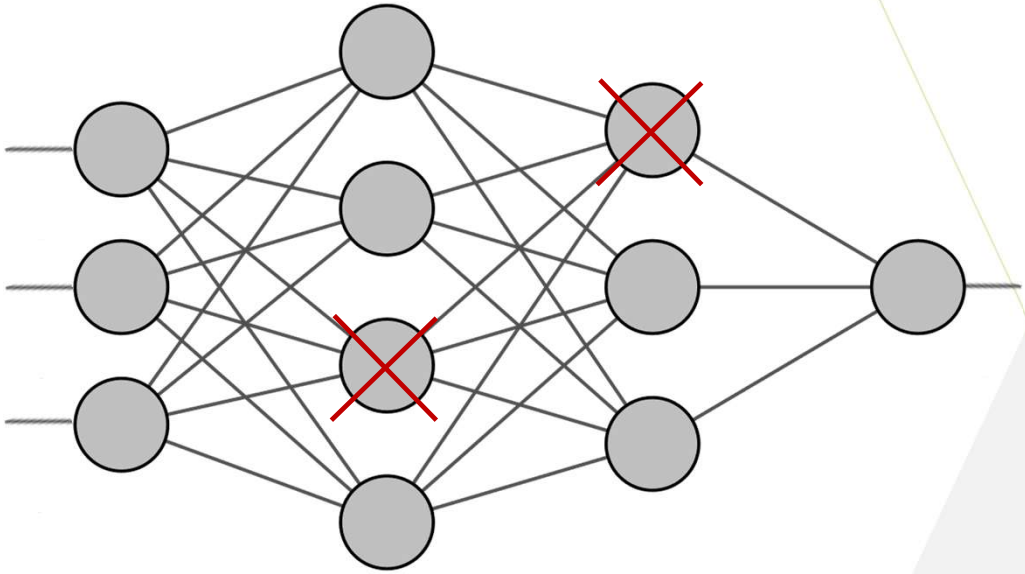


How does an ML compiler framework operate?

Model **Compression**

Pruning

Zero out weights while maintaining performance above a threshold, achieving model **sparsity**

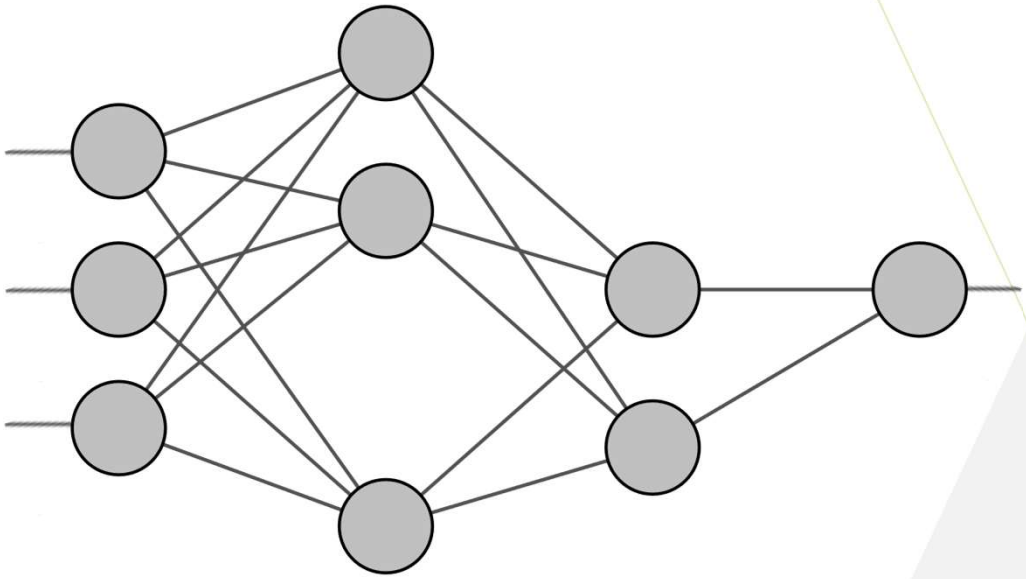


How does an ML compiler framework operate?

Model **Compression**

Pruning

Zero out weights while maintaining performance above a threshold, achieving model **sparsity**



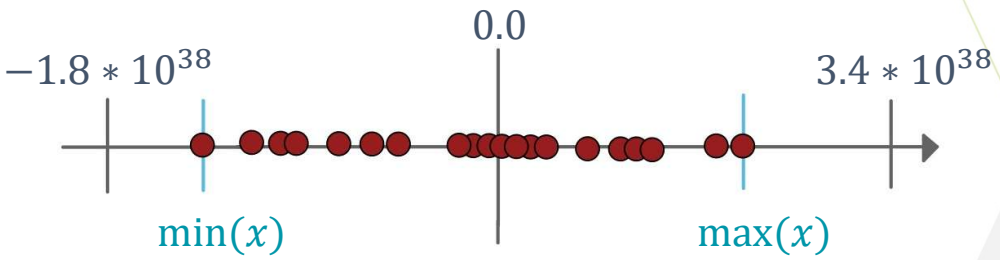
How does an ML compiler framework operate?

Model **Compression**

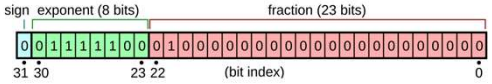
Quantization

Reduce the physical representation of weights and/or activations by using fewer bit

FP32



Note: FP32 as defined by the IEEE 754 standard



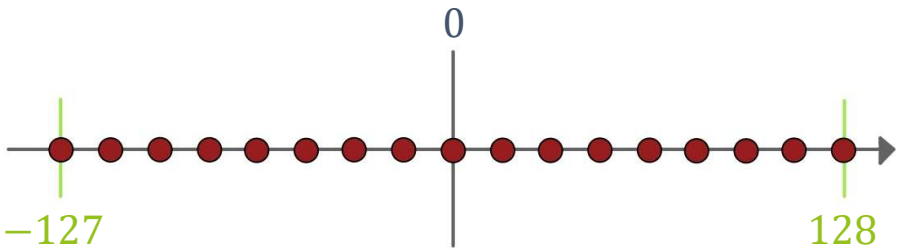
How does an ML compiler framework operate?

Model **Compression**

Quantization

Reduce the physical representation of weights and/or activations by using fewer bit

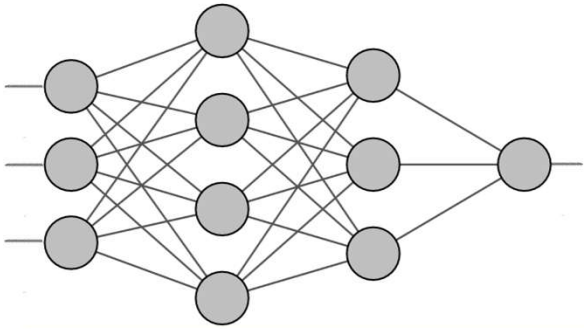
INT8



How does an ML compiler framework operate?

Pruning & Quantization

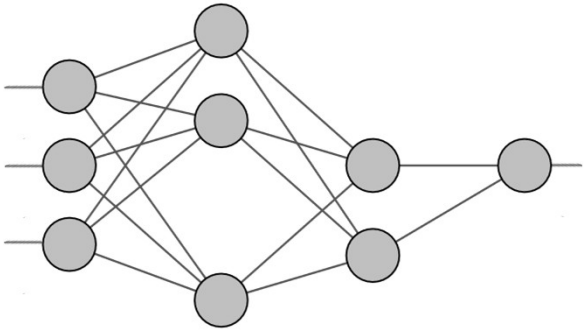
Model **Compression**



$$\begin{bmatrix} -0.3123 & 0 & 0.2234 \\ 0.4455 & -0.2235 & 0.0124 \\ 0.7153 & 0.8111 & 0.2121 \end{bmatrix}$$

Sparsify

Quant.



$$\begin{bmatrix} -0.3 & / & / \\ 0.4 & / & / \\ 0.7 & 0.8 & / \end{bmatrix}$$

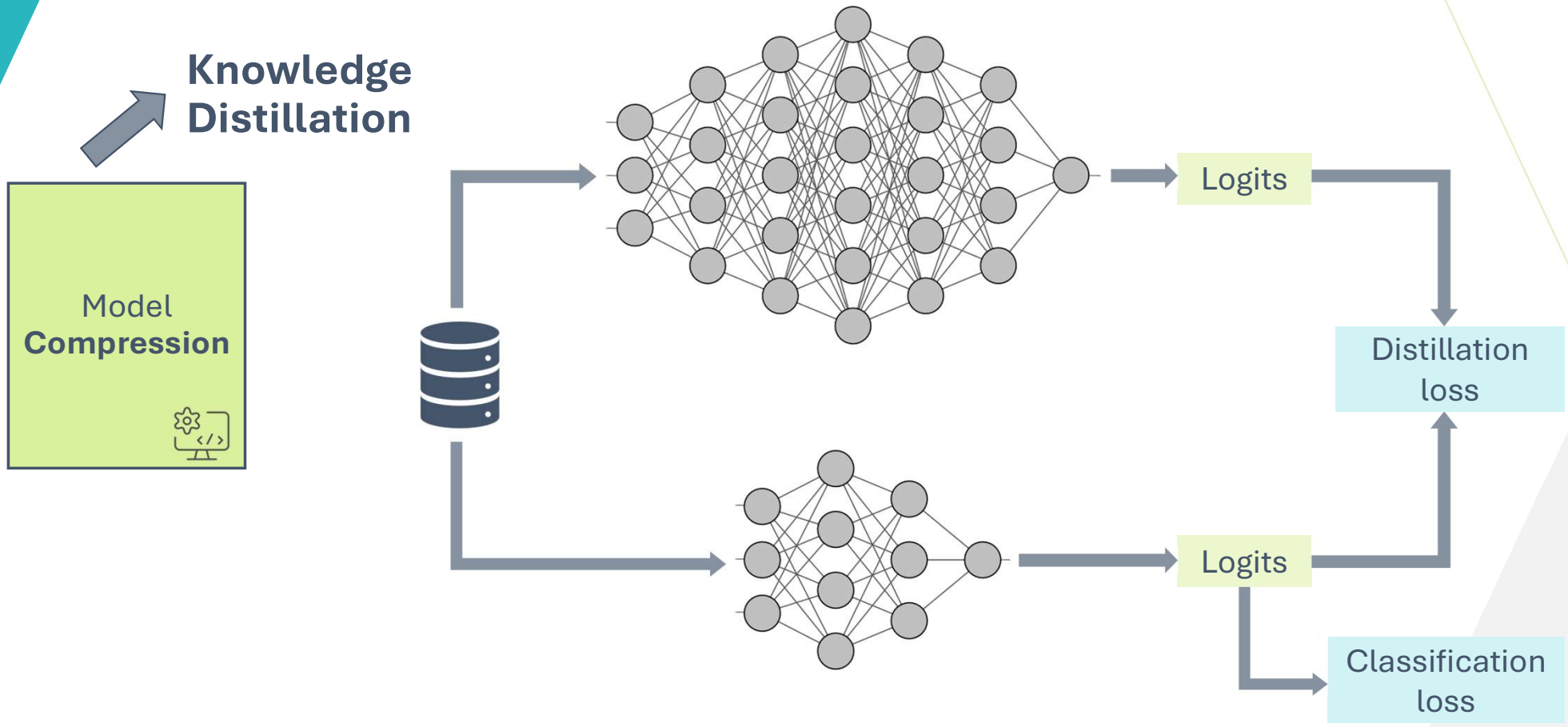
How does an ML compiler framework operate?

Knowledge Distillation

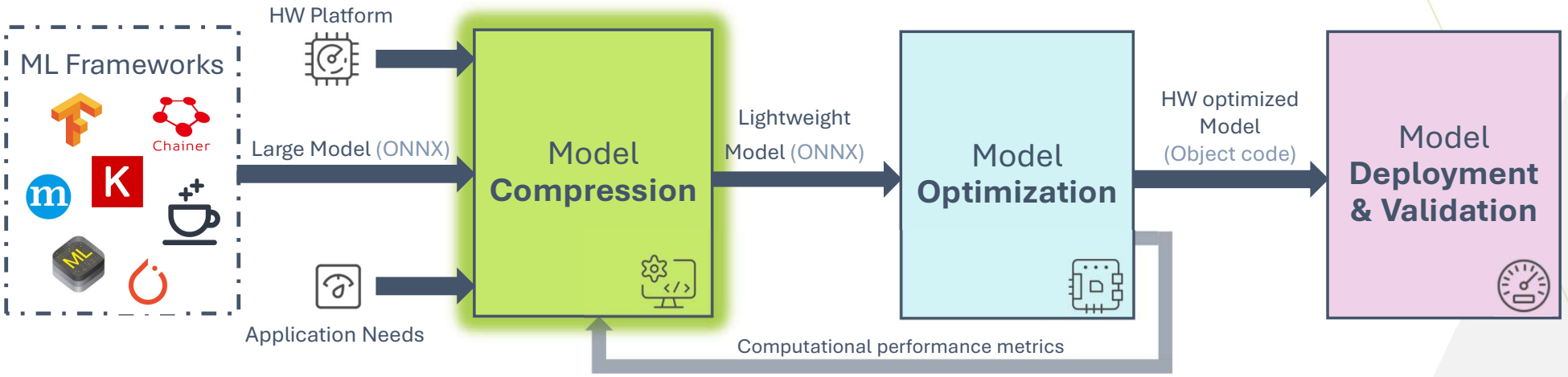
Model **Compression**

Knowledge is distilled from large, complex *“teacher”* model to a smaller *“student”* model

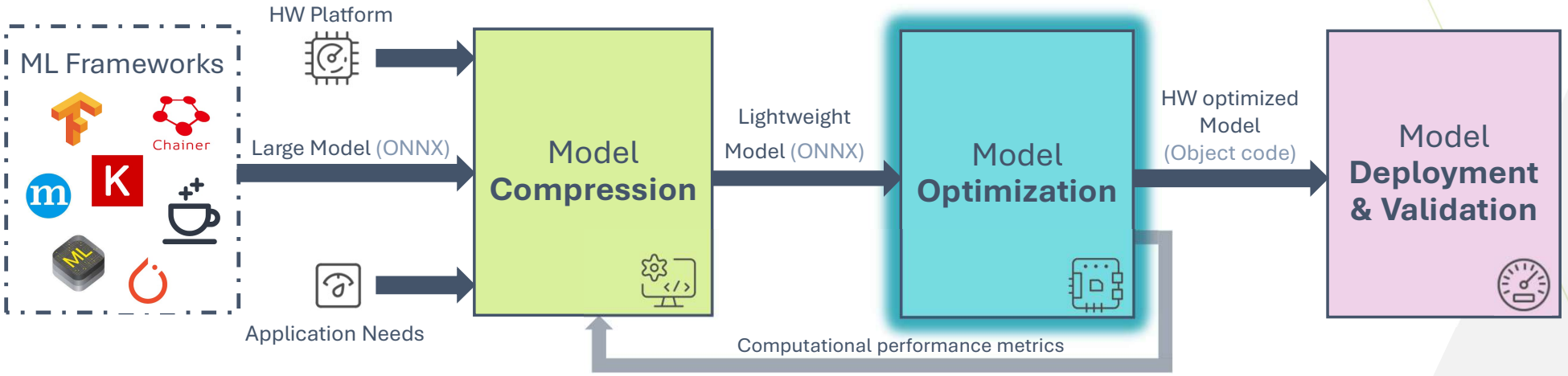
How does an ML compiler framework operate?



How does an ML compiler framework operate?



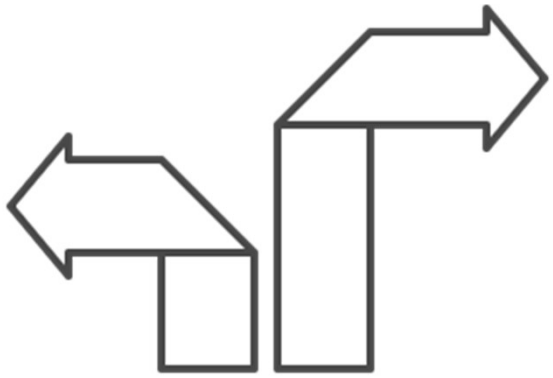
How does an ML compiler framework operate?



How does an ML compiler framework operate?

Enhance computational efficiency, reduce latency, and minimize resource consumption during inference

Model
Optimization



Operator Level
optimization

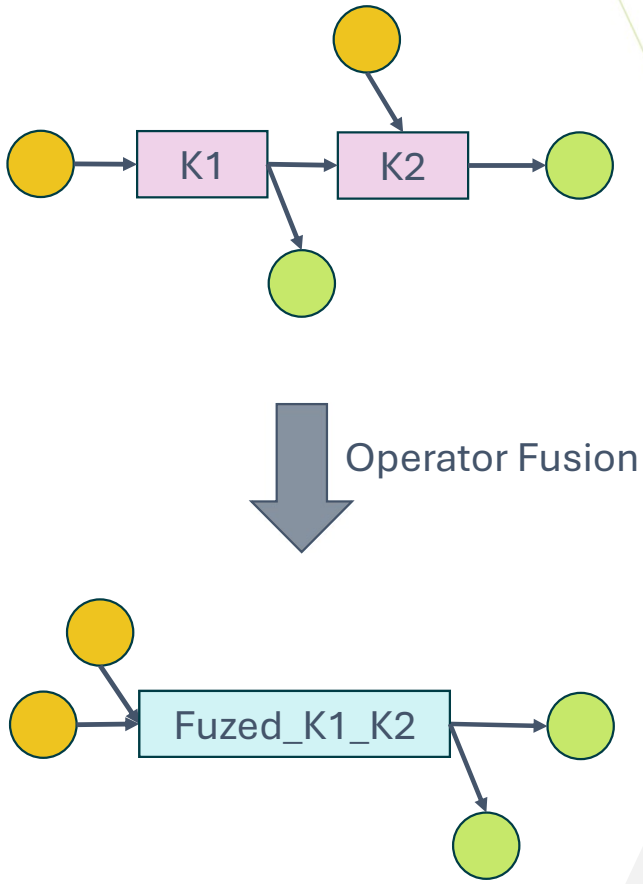
Graph Flow
optimization

How does an ML compiler framework operate?

Graph flow optimization

Model **Optimization**

Rewrite dataflow graphs to simplify graph and reduce device peak memory usage



How does an ML compiler framework operate?

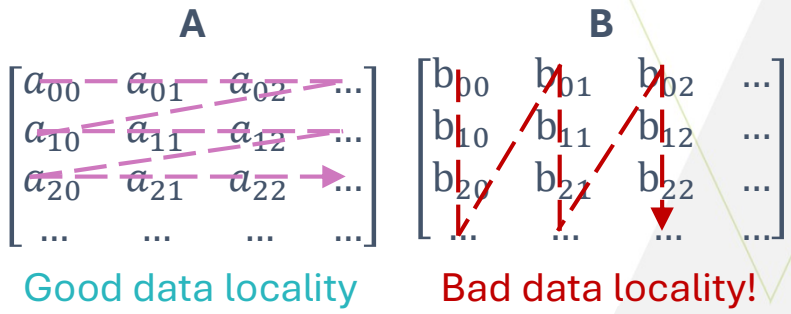
Operator level optimization

Model **Optimization**

Assigns the best schedule to an operator for fast inference and low memory use on specific hardware.

Loop optimization

$$for\ i\ in\ range(0, N):$$
$$\quad for\ j\ in\ range(0, N):$$
$$\quad\quad for\ k\ in\ range(0, N):$$
$$\quad\quad\quad C[i][j] \mathrel{+}= A[i][k] * B[k][j]$$



*Assume stored in row-major order

How does an ML compiler framework operate?

Operator level optimization

Model **Optimization**

Assigns the best schedule to an operator for fast inference and low memory use on specific hardware.

Loop optimization

SIMD (Single Instruction, Multiple Data)

$$for\ i\ in\ range(0, N):$$
$$for\ \mathbf{k}\ in\ range(0, N):$$
$$for\ \mathbf{j}\ in\ range(0, N):$$
$$C[i][j] \mathrel{+}= A[i][k] * B[k][j]$$

A

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \dots \\ a_{10} & a_{11} & a_{12} & \dots \\ a_{20} & a_{21} & a_{22} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

Good data locality

B

$$\begin{bmatrix} b_{00} & b_{01} & b_{02} & \dots \\ b_{10} & b_{11} & b_{12} & \dots \\ b_{20} & b_{21} & b_{22} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

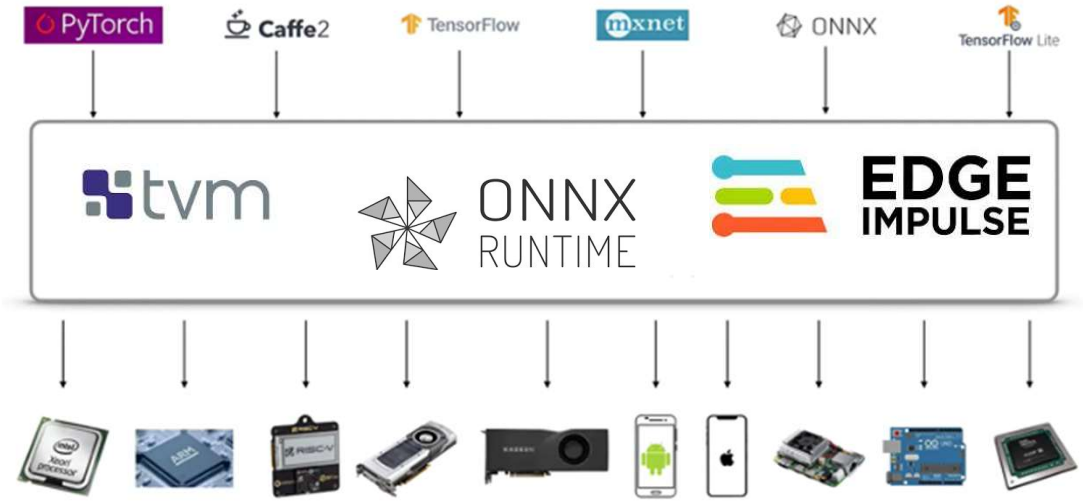
Good data locality

*Assume stored in row-major order

Choosing an Edge ML compiler framework

The Ideal framework:

- Free
- Fast
- Easy to use
- Open-source
- Well-documented
- Works across most devices
- Wide community support



ApacheTVM vs. ONNX Runtime

tvm

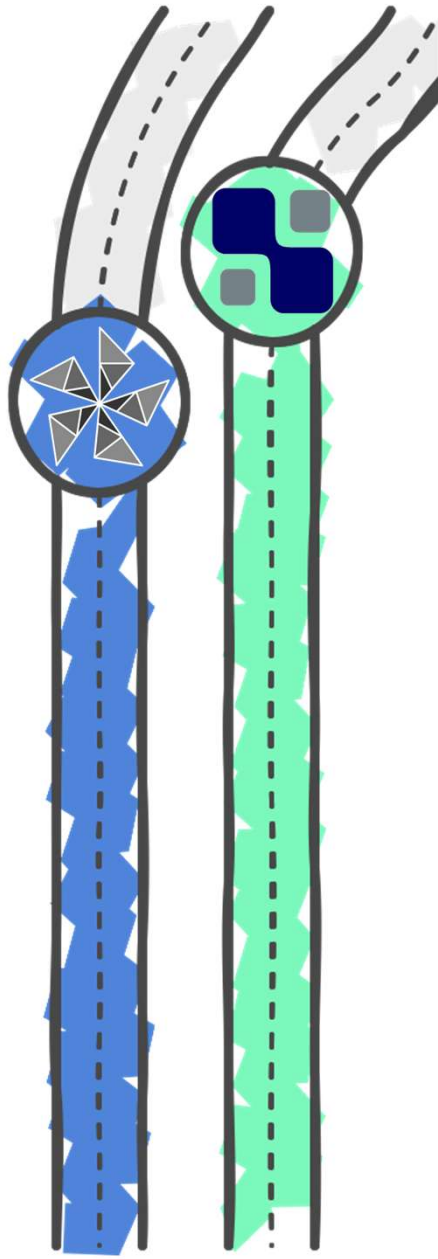
- Free, open-source, Well-documented
- Supports both OS & non-OS devices

- Hardware specific models
- More complex to setup and use

ONNX RUNTIME

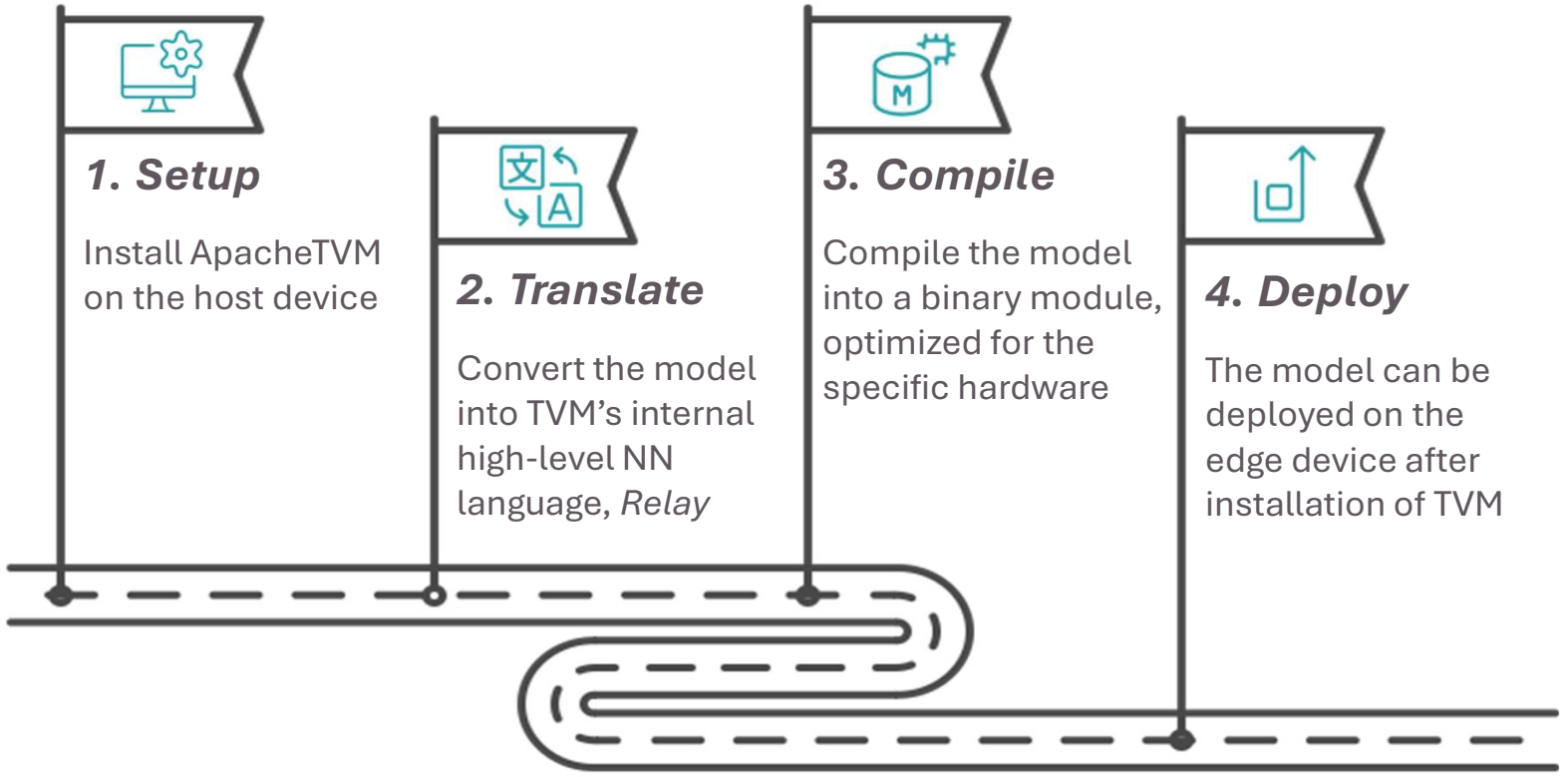
- Free, open-source, Well-documented
- Easy to setup and use

- Does not support non-OS devices



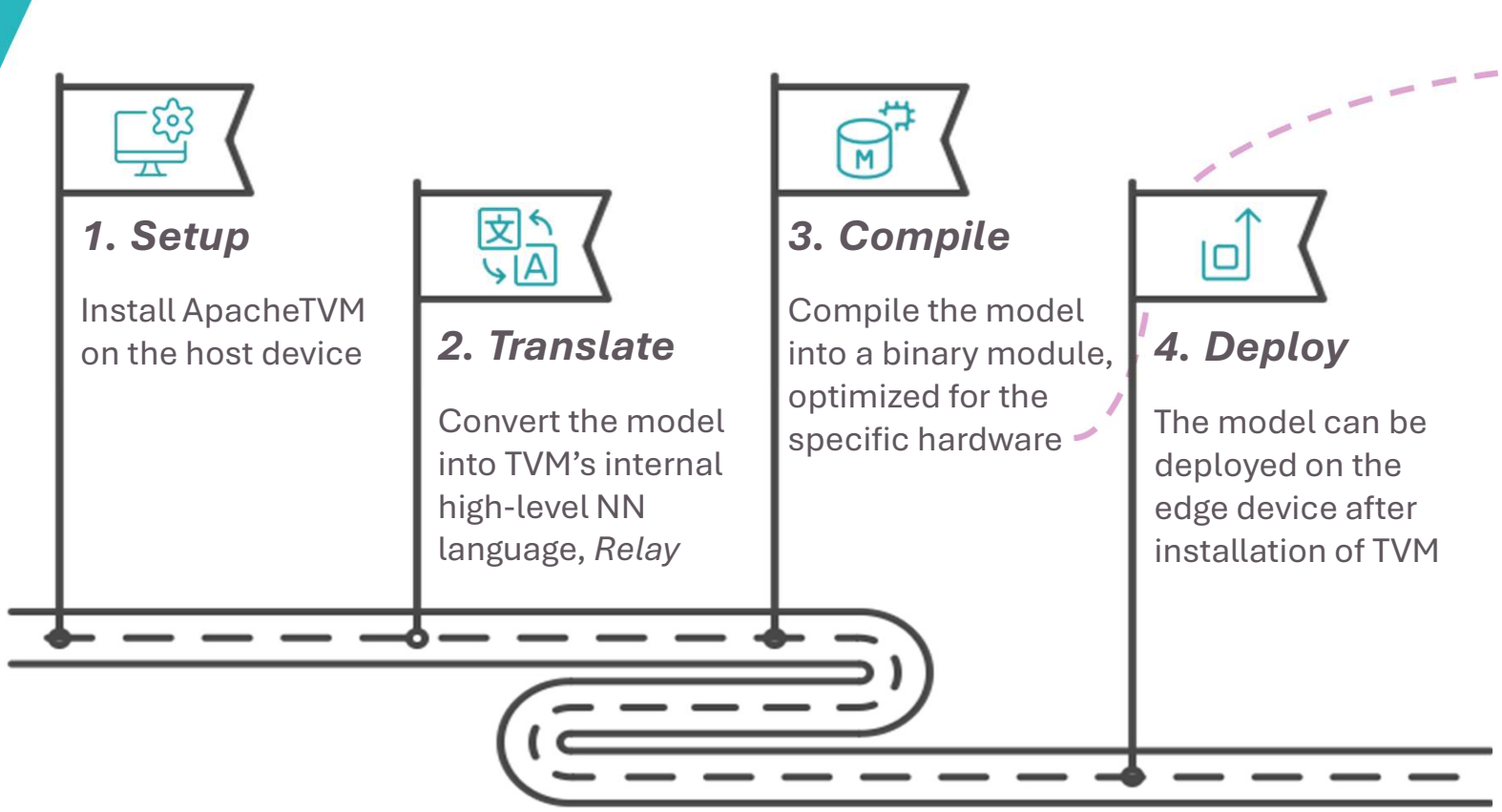
Model compilation workflow with ApacheTVM

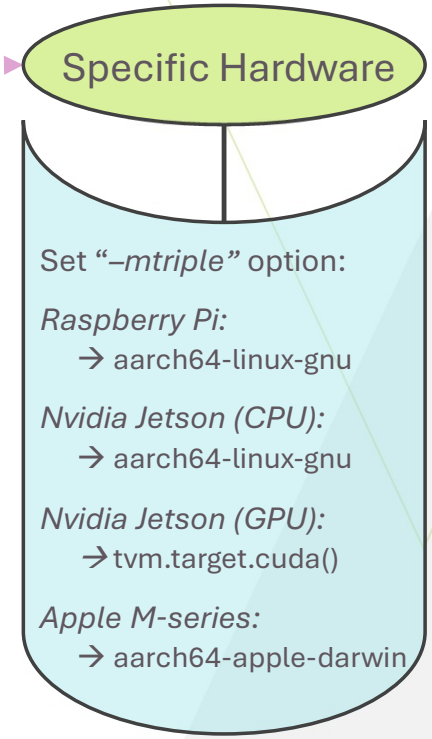
- Device with OS



Model compilation workflow with ApacheTVM

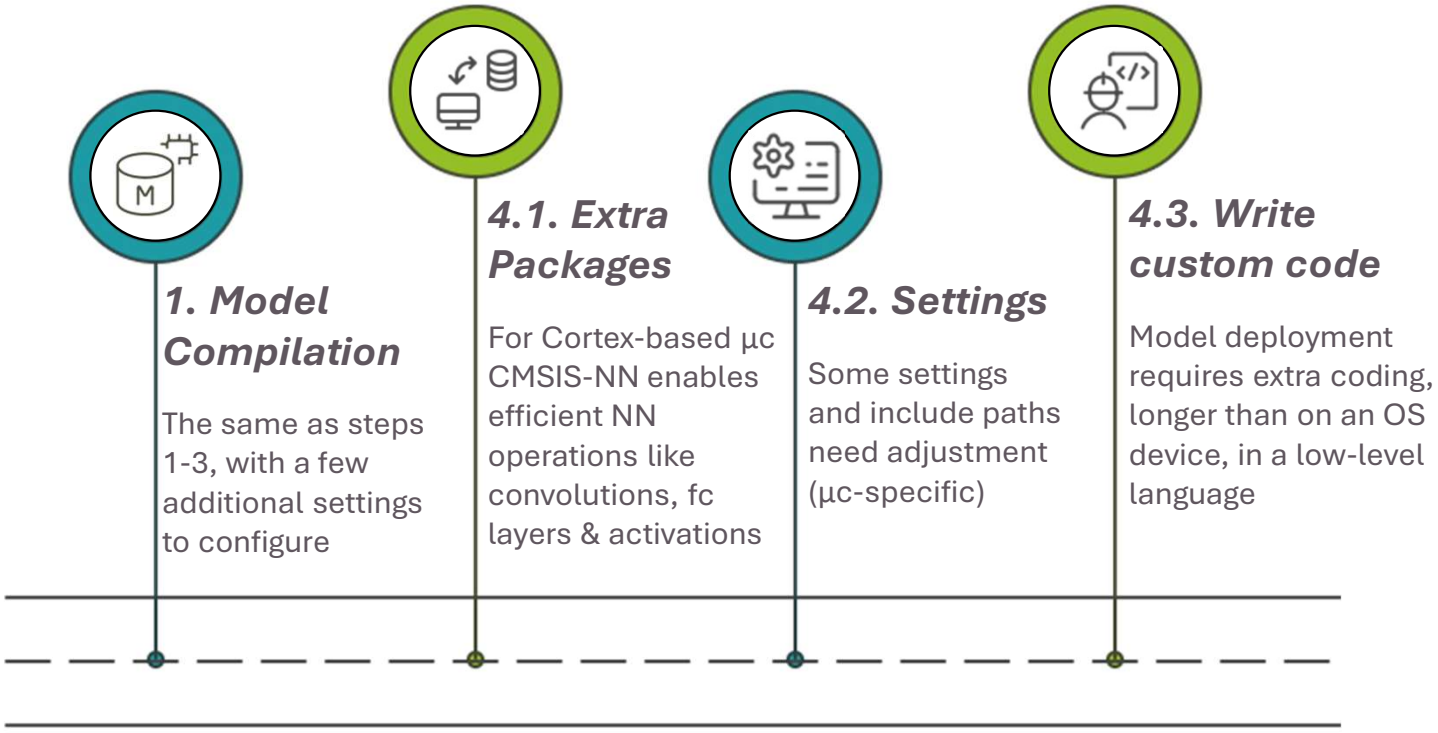
- Device with OS





Model compilation workflow with ApacheTVM

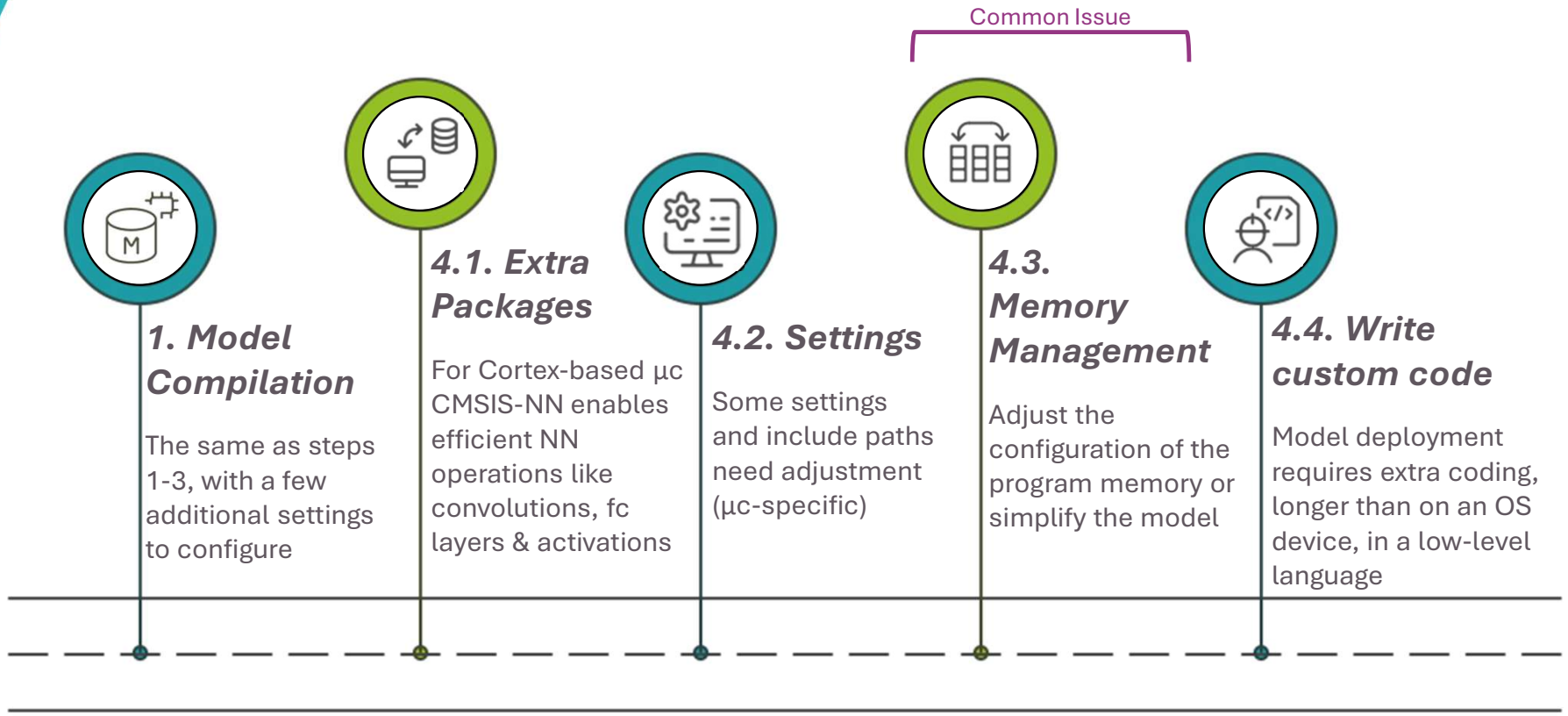
- Device without OS



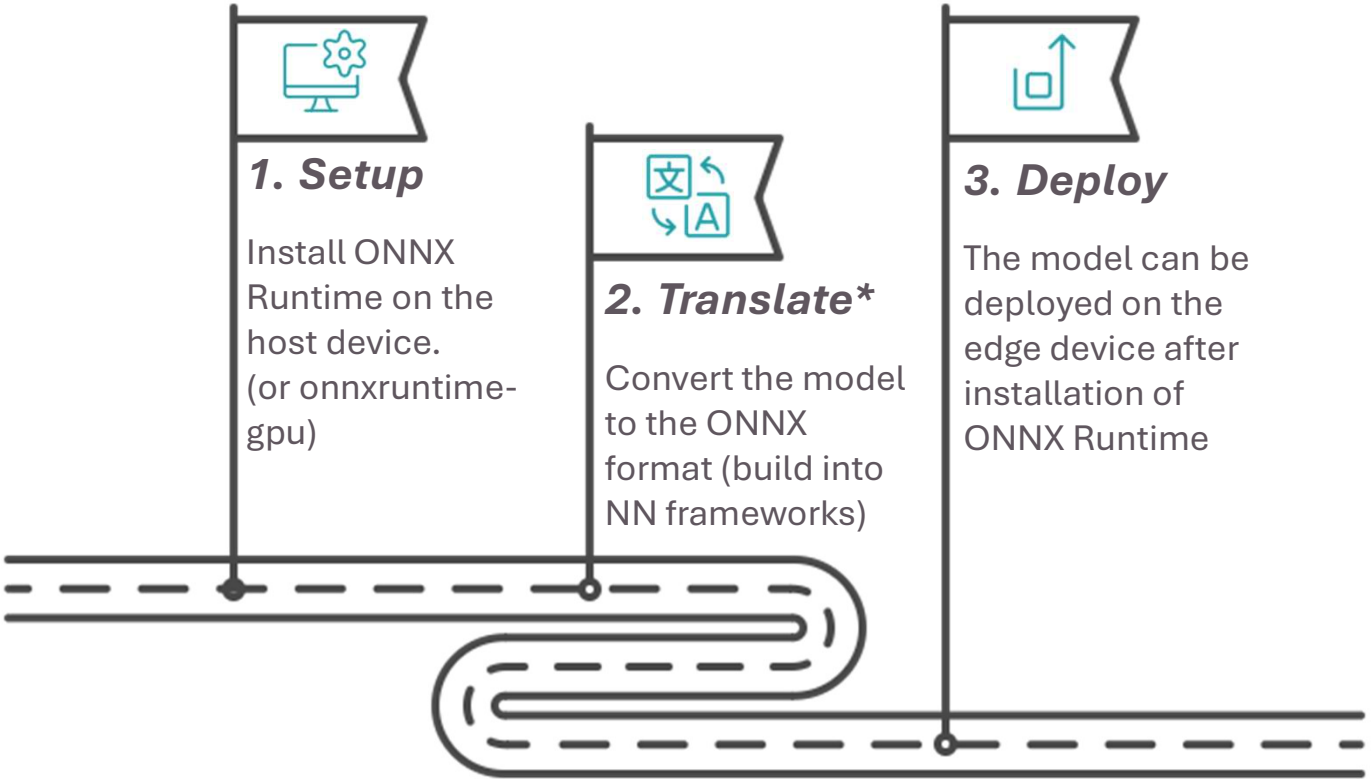
*Microcontroller = µc

Model compilation workflow with ApacheTVM

- Device without OS



Model compilation workflow with ONNX Runtime



*If necessary

Experiment: ApacheTVM vs. ONNX Runtime

What is the effect on inference time when utilizing different compilers and/or hardware platforms?

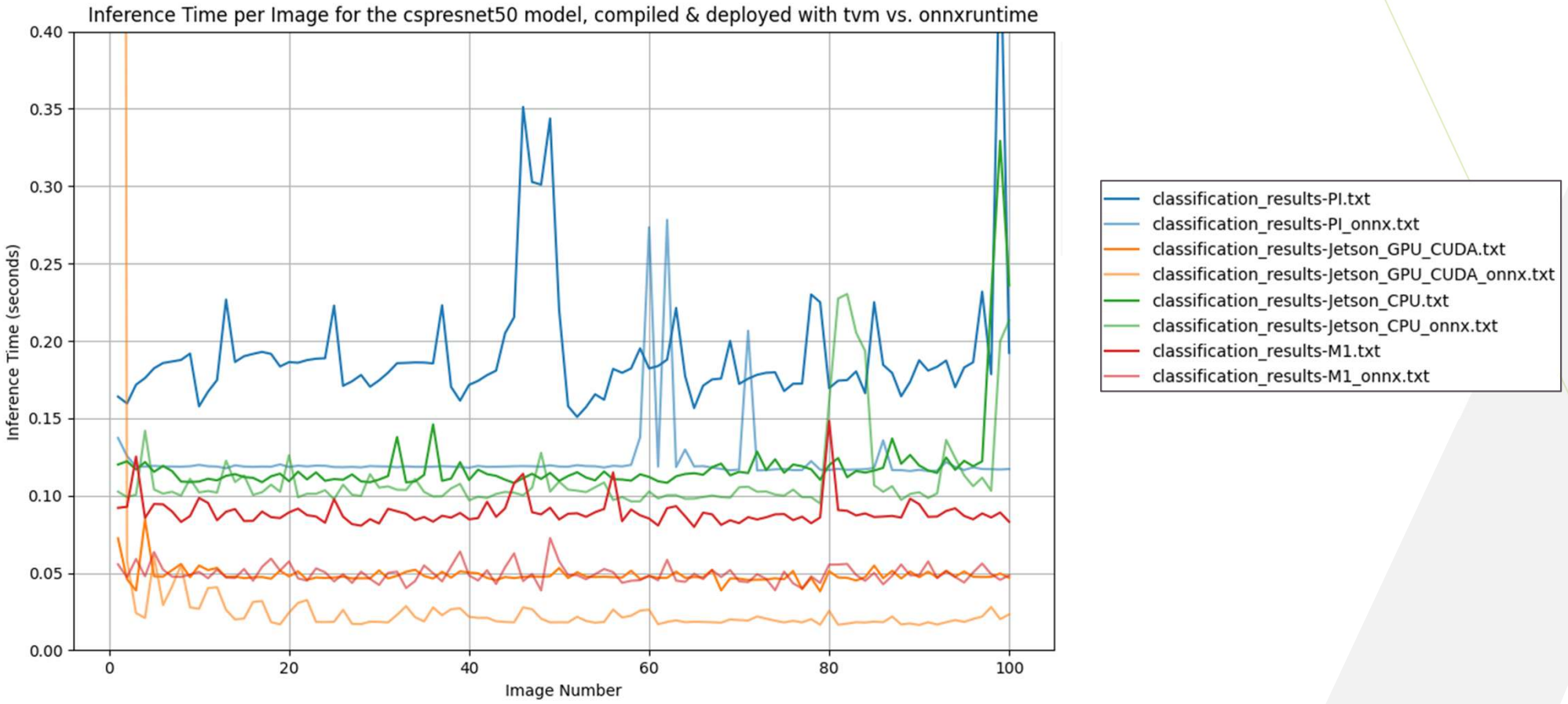
Model : Resnet50 (cspresnet50)
- → pre-trained on ImageNet
- → from ONNX model zoo
- → 22 million parameters
- → 82.66 MB

Compilers:
- → ApacheTVM
- → ONNX Runtime

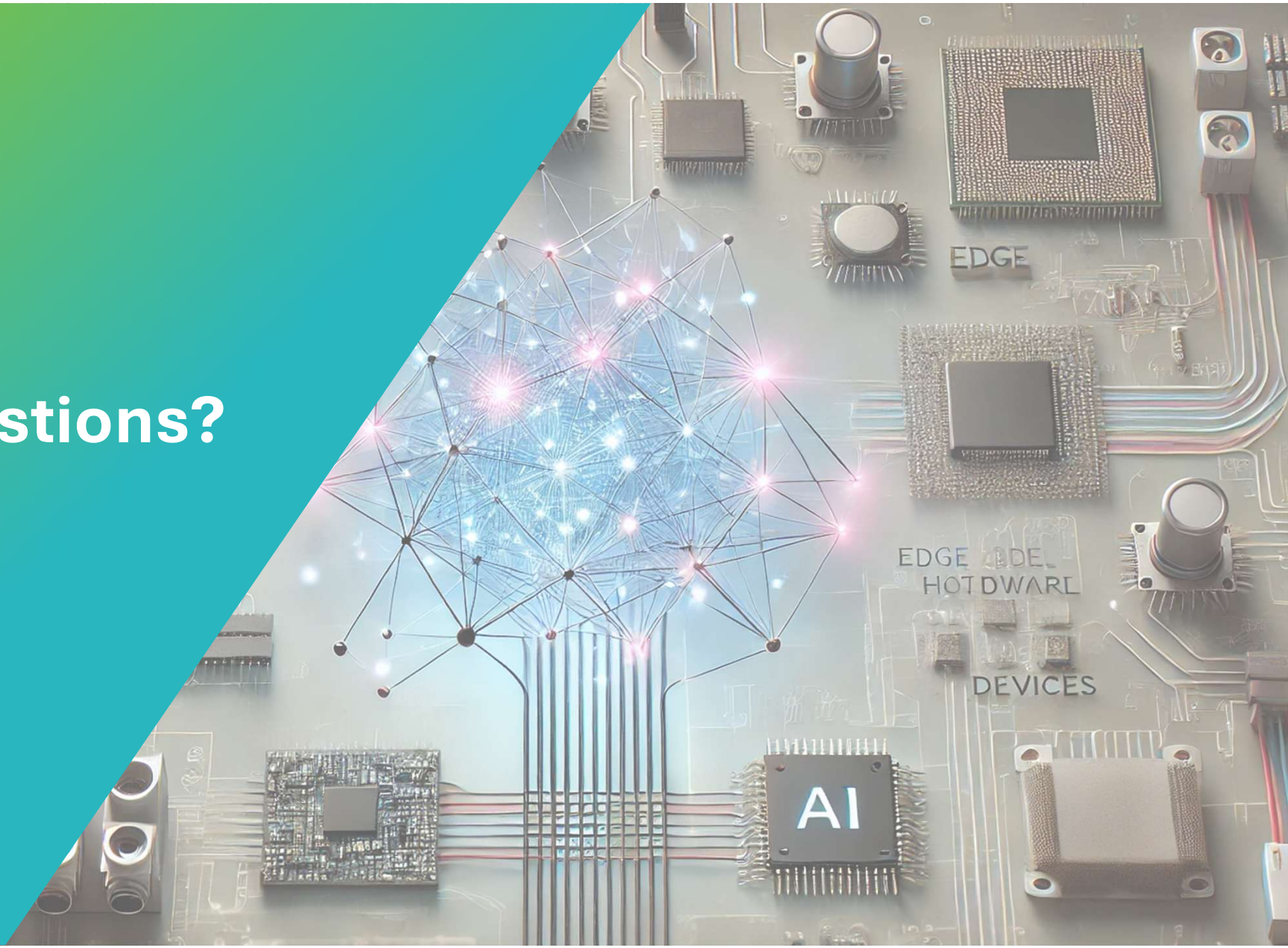
Devices:
- → Raspberry Pi 5
- → Nvidia Jetson Orin Nano CPU
- → Nvidia Jetson Orin Nano GPU (cuda)
- → Apple MacBookPro M1

ApacheTVM vs. ONNX runtime

- ONNX Runtime vs. ApacheTVM: Inference time on OS devices.



Questions?



WiFi access

Network: Flanders Make Guest
Login: Guest
Password: LetsM4kelt

KU LEUVEN

FLANDERS MAKE

DRIVING INNOVATION IN MANUFACTURING

VLAIO

DTAI

Declarative Languages &
Artificial Intelligence

Agenda

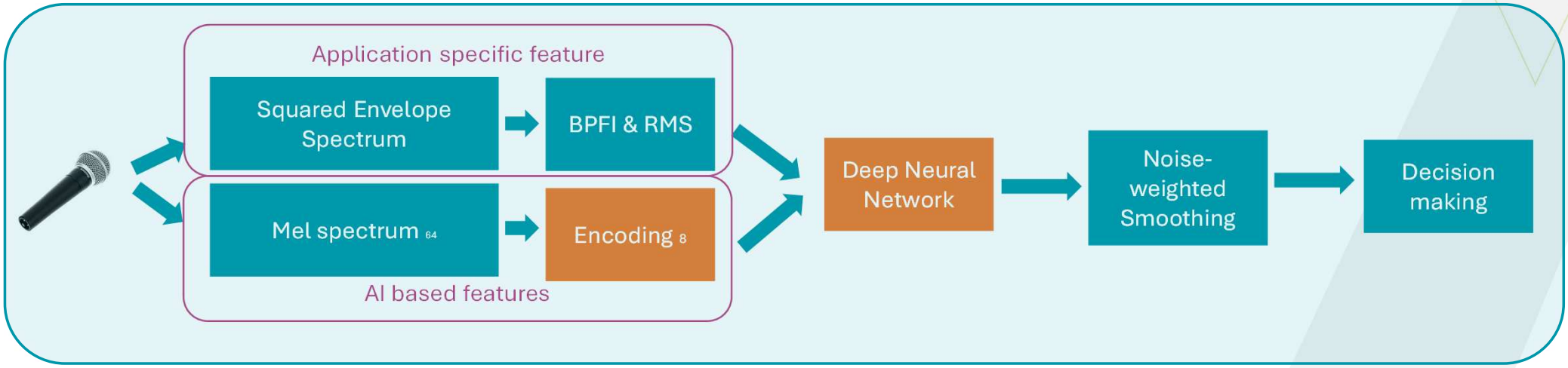
13u30	Welcome & meeting objectives
13u40	Overall project goals
14u25	Tutorial : From pretrained model to edge deployment
14u55	Coffee break
15u05	**Demo's: Bearing failure monitoring & edge tower**
16u05	Knowledge transfer & implementation
16u35	Planning & next steps
16u50	Closing
17u	Reception

Time-series use case: Acoustic monitoring

- Accelerated lifetime tests (ALT) of ball bearings
- 66 bearings tested
  - Some run until failure
  - Some did not fail
  - Some were completely healthy
- Constant load
- Vibrations, acoustics, temperatures, RPM
- The microphones naturally pick up ambient noise, speech, …

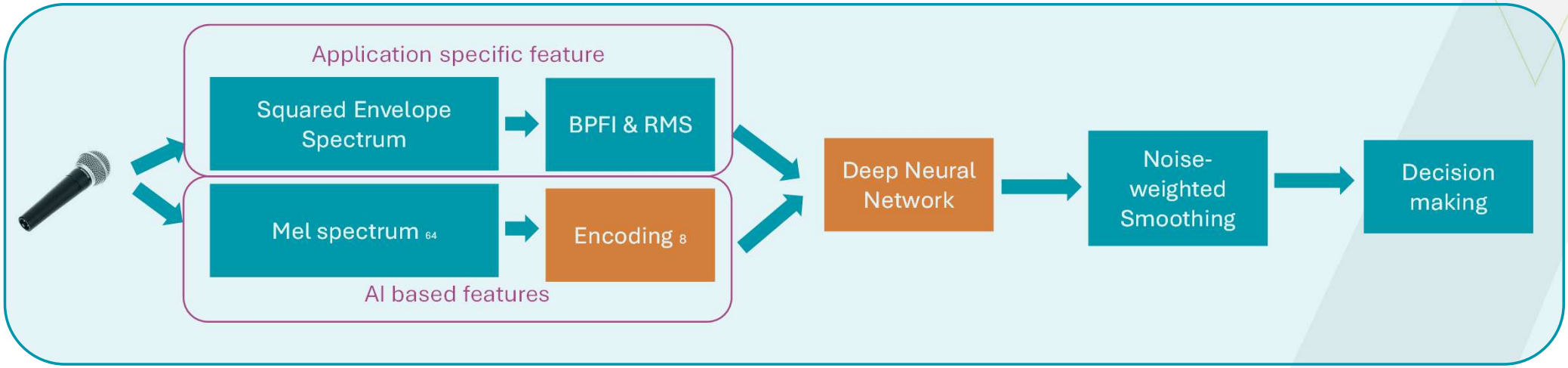
ALT setup @ FM Leuven

Healthy **Faulty**



Time-series use case: Acoustic monitoring

- Mel-spectrum calculation is not real-time
- Two AI-models
- Python logic required for orchestrating the models and postprocessing the estimates
- Option: **Integrate feature calculations in model**
  - Unified deployment, potential hardware acceleration, simplified maintenance
- Option: **Separate optimized DSP**
  - Preserved domain knowledge, minimal code changes, potentially leverage specialized CPU/DSP instructions



User group input

- Challenges
- Use case interests
- KPI requirements
- Frameworks
- Devices

Scan QR-code or
join at menti.com w/ code 2587 9014

Agenda

13u30	Welcome & meeting objectives
13u40	Overall project goals
14u25	Tutorial : From pretrained model to edge deployment
14u55	Coffee break
15u05	Demo's: Bearing failure monitoring & edge tower
16u05	**Knowledge transfer & implementation**
16u35	Planning & next steps
16u50	Closing
17u	Reception

COOCK+ : Collective R&D and collective knowledge distribution

Goal: Valorization of (basic) research results by accelerating the introduction of knowledge and/or technology

A COOCK+ project consists of 2 complementary parts

Part A focuses on application-oriented knowledge building, translation research and knowledge dissemination activities	**Part B** encompasses all company-specific actions designed to evaluate and implement Part A within organizations.

COOCK+ : Collective R&D and collective knowledge distribution

Active involvement of the target group before, during and after the project is essential. In order to stimulate interaction with the target group, a representative user group must be set up for each project.

Part knowledge building, translation research and knowledge dissemination activities	specific actions designed to evaluate and implement Part A within organizations.

Part A: Collective actions

Demonstrations: a setup at a live event with an interactive character that does not require specialist understanding.

Created by okta from Noun Project

Written and digitalized knowledge dissemination: any unique content item that can be consulted publicly and without obligation (blog, newsletter, articles, videos, books, informative e-learning modules, etc.)

Workshops: any targeted, substantively specialist, highly technical and interactive event that is specifically set up for the target group and by the applicants of the COOCK+ project (interactive character that requires specialist understanding).

Created by zum rotul from Noun Project

Events for broad dissemination: any event format that (partly) focuses on the target group of the COOCK+ project and in which the dissemination of the results of the project forms an active part. (a presentation, a poster, etc.)

Part A: Collective actions

Demonstrations:

2 Generic case studies

- Computer vision
- Time series

Written knowledge dissemination:

Best practice manuals :

- for the selection of HW/SW combination
- to design edgeAI with pretrained models
- For monitoring of edge DL software

Academic papers

Workshops:

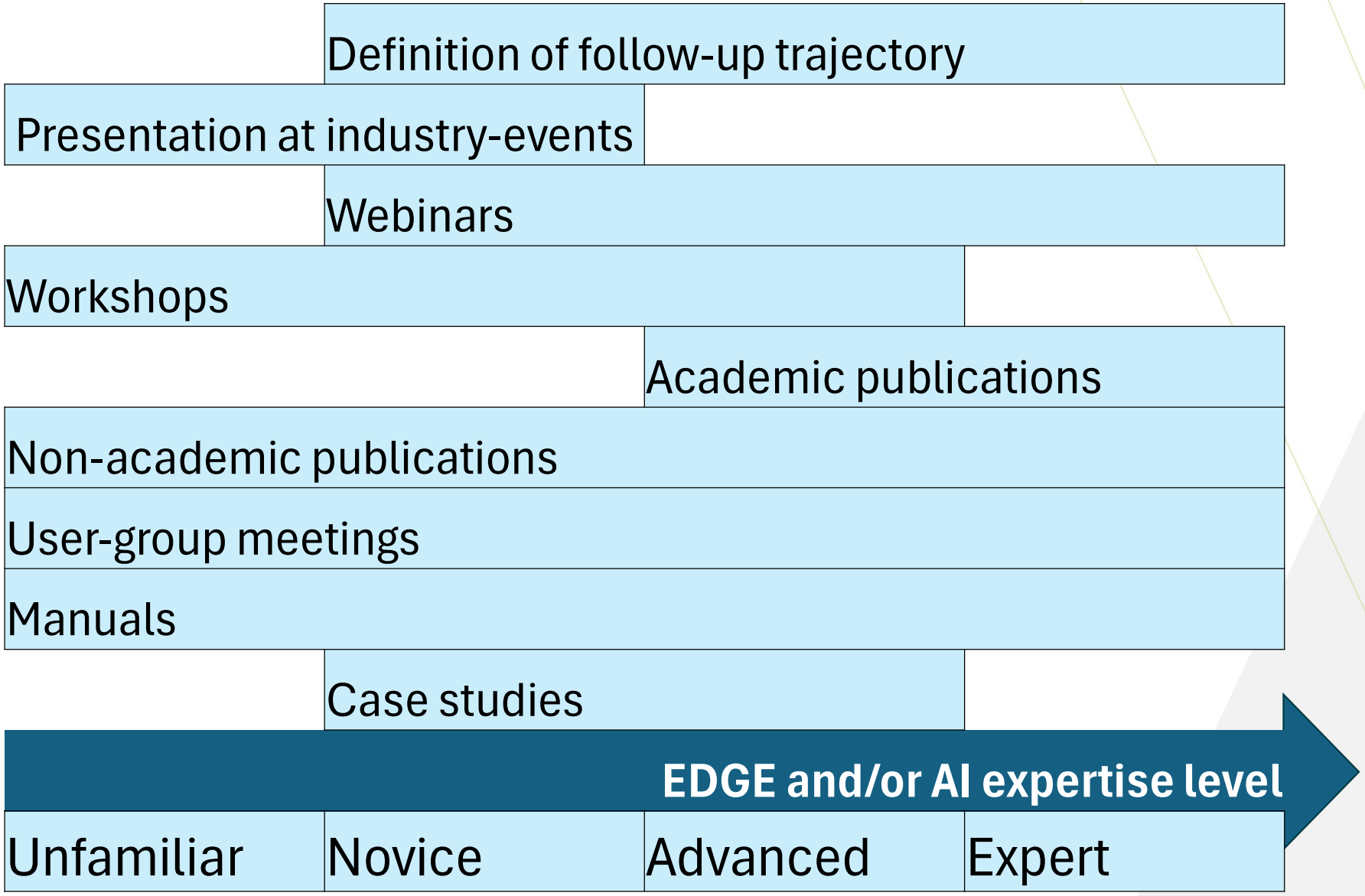
Based on the generic case studies and tailored to your feedback

Events for broad dissemination:

Flanders Make symposium

Part B: company specific actions

- **Advice:**
  - (technical) substantive consultation and/or support of an innovation trajectory within a company
- **Knowledge transfer:**
  - knowledge transfer from part A of the COOCK+ project to another (collective) project;
  - following a targeted training by a target group company at a real market price
- **Exploration:**
  - any form of preparatory study carried out by or for a target group company to evaluate the potential of its application within the company.
- **R&D:**
  - an in-depth development carried out by or for a target group company using the (generic) results from part A of the COOCK+ project
- **Integration:**
  - an integration or implementation of a solution beyond TRL 7, carried out by or for a company


Definition of follow-up trajectory
Presentation at industry-events
Webinars
Workshops
Academic publications
Non-academic publications
User-group meetings
Manuals
Case studies
EDGE and/or AI expertise level
Unfamiliar
Novice
Advanced
Expert

Projectwebsite: https://medlicoock.github.io/

MEDLI

Managing Edge Deployment of Large Deep Learning Models in Industry

Contact:

marjolein.deryck@kuleuven.be

Project manager

Reglement van orde van de begeleidingsgroep

- Doelstelling
- Leden en waarnemers
- Intellectuele eigendom
- Werking
- Signature
- Annex: lijst leden en waarnemers

VLAIO

Closing

Questions?
Concerns?
Feedback?

		Year 1				Year 2			
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
WP1: Edge model specialisatie en compressie	T1.1: Edgemodel ontwerp via een HW-specifieke doelmodel databank T1.2: Edgemodel ontwerp via voorgetrainde modellen en vrij doelmodel T1.3: Ontwerp v/e GUI voor modelcompressie en specialisatie SW								
WP2: HW specifieke optimalisatie & deployment	T2.1: Selectie en beschrijving van relevante edge HW T2.2: Compilerselectie T2.3: Deployment tools T2.4: Simulatie van edge performantie T2.5: Data streaming management voor edge devices								
WP3: Edge model monitoring	T3.1: Overzicht van monitoring tools voor DL modellen T3.2: Studie van monitoring tools voor edge toepassingen								
WP4: Demonstratie via twee gevallenstudies	T4.1: Verzamelen industriële gebruikersvereisten T4.2: Validation and demonstration for vision use-case T4.3: Validation and demonstration for time-series use-case								
WP5: Disseminatie en valorisatieopvolging	T5.1: Handleidingen T5.2: Disseminatieevents T5.3: Publicaties T5.4: Valorisatieopvolging								

UGM UGM UGM UGM UGM

Omschrijving leverbaarheden en mijlpalen	Voorziene timing
Projectspecifieke kennisontwikkeling	maand
M1: Database met edge-HW specifieke doelmodellen	*6 en 15*
M2: SW suite met modelspecialisatie- en compressiealgoritmes	*12*
M3: Grafische gebruikersinterface om de SW-suite (M2) gemakkelijk te kunnen gebruiken	*14*
L1: Beslissingsbomen voor de ondersteuning bij de keuze van edge-HW en compiler combinatie en overzicht van deployment tools, simulatietools en SW voor het beheer van data streaming	*15*
L2: Vergelijkende studie van bestaande monitoringtools voor edge AI	*20*
L3: Twee generieke gevalsstudies (visie en tijdsreeks) met demonstratoren	*12 en 21*
Collectieve/generieke kennisoverdracht	maand
L4: Handleiding met best-practices rond het selecteren en gebruiken van een geschikte combinatie van edge-HW en SW voor de edge AI toepassing	*11*
L5: Handleiding met best-practices rond het ontwerpen van edgeDL oplossingen mbv voorgetrainde modellen	*15*
L6: Handleiding met best-practices rond het monitoren van DL software op de edge	*20*
L7: Hands-on workshop en webinars gebaseerd op generieke gevalsstudies	*22*
L8: Zes publicaties	*24*

Tabel 1: Overzicht van Milestones (M) en Leverbaarheden (L)