tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

"Exploring techniques to build efficient and robust TinyML deployments"

Ashutosh Pandey - Infineon Technologies

January 18, 2022

TINY ML

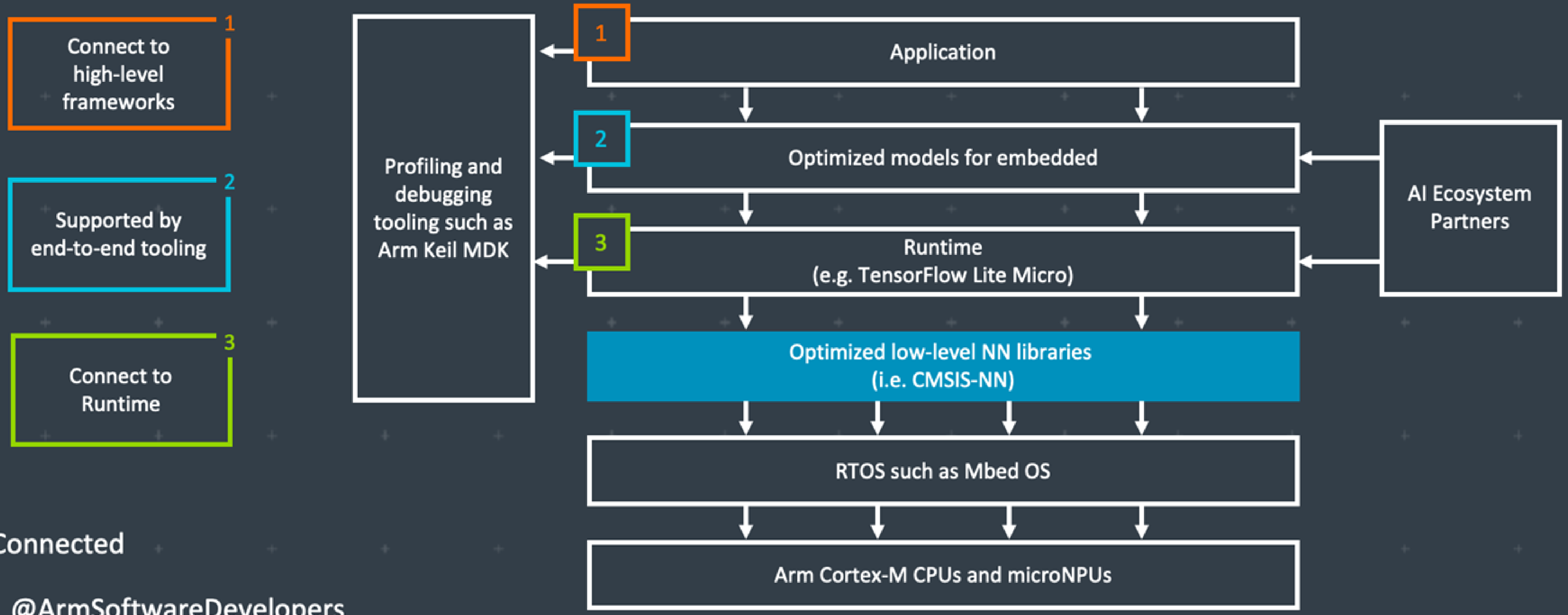
www.tinyML.org

tinyML Talks Strategic Partners

Additional Sponsorships available – contact Olga@tinyML.org for info

Executive Strategic Partners

Arm: The Software and Hardware Foundation for tinyML



Stay Connected

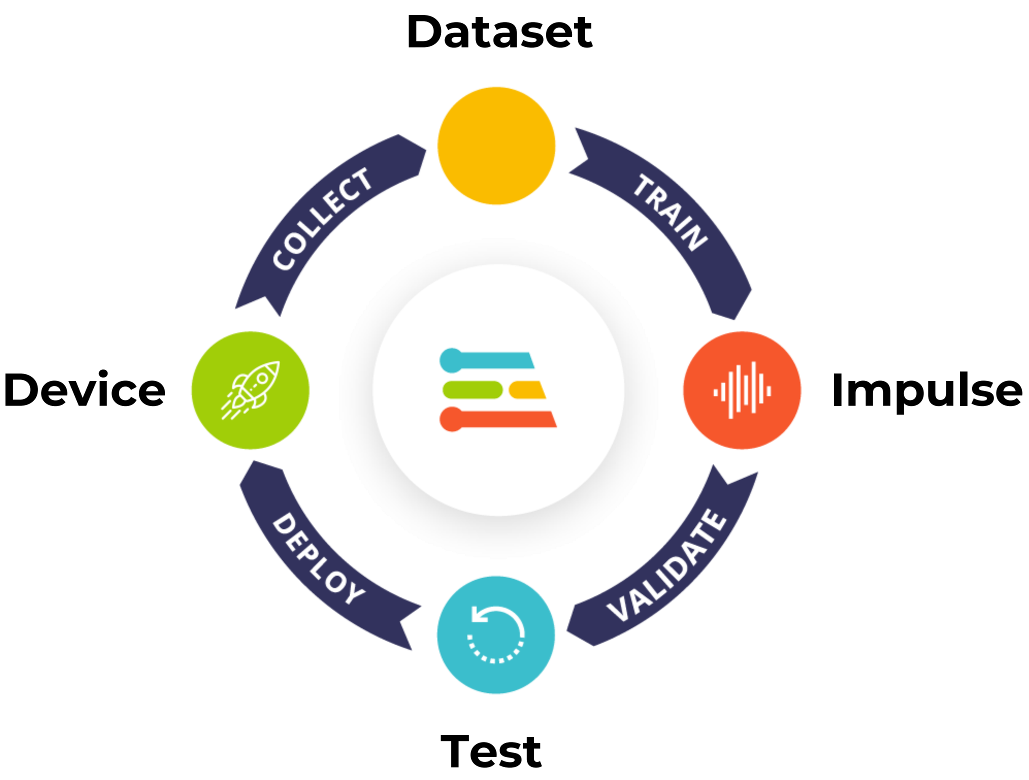
@ArmSoftwareDevelopers

@ArmSoftwareDev

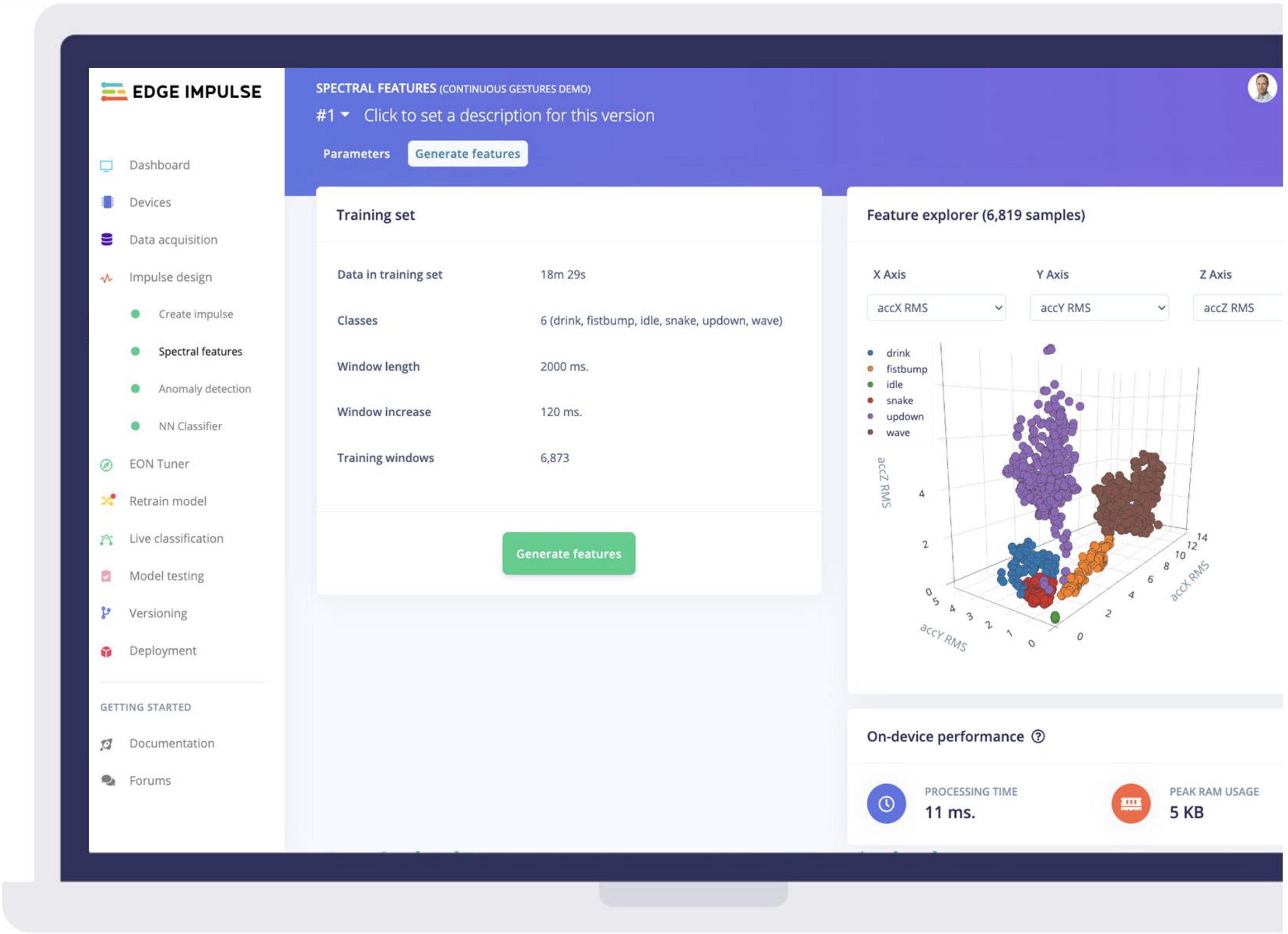
Resources: developer.arm.com/solutions/machine-learning-on-arm

arm

EDGE IMPULSE The leading edge ML platform



www.edgeimpulse.com



Qualcomm AI research

Advancing AI research to make efficient AI ubiquitous

Power efficiency

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

Personalization

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

Efficient learning

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

A platform to scale AI across the industry

Perception

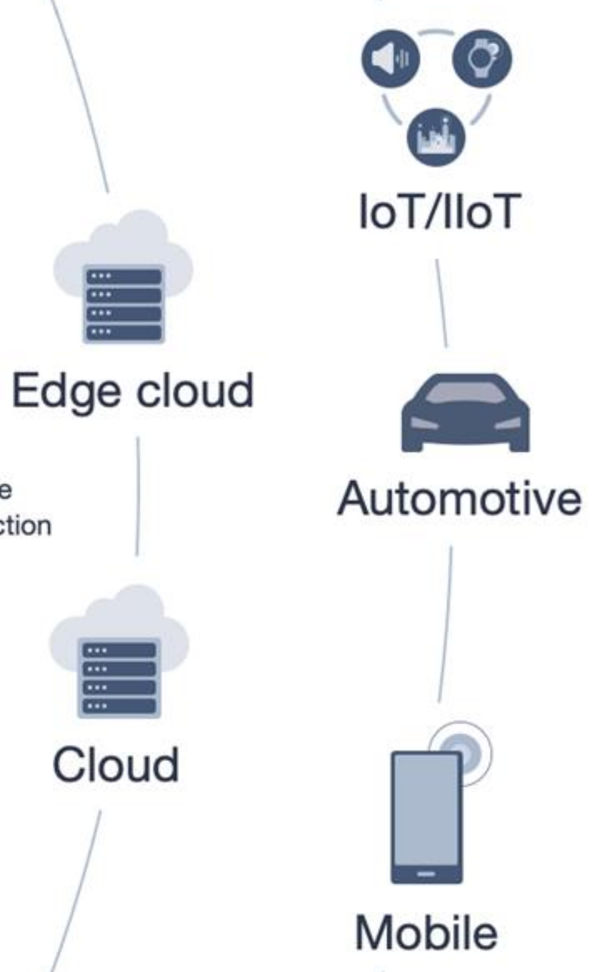
Object detection, speech recognition, contextual fusion

Reasoning

Scene understanding, language understanding, behavior prediction

Action

Reinforcement learning for decision making



SYNTIANT

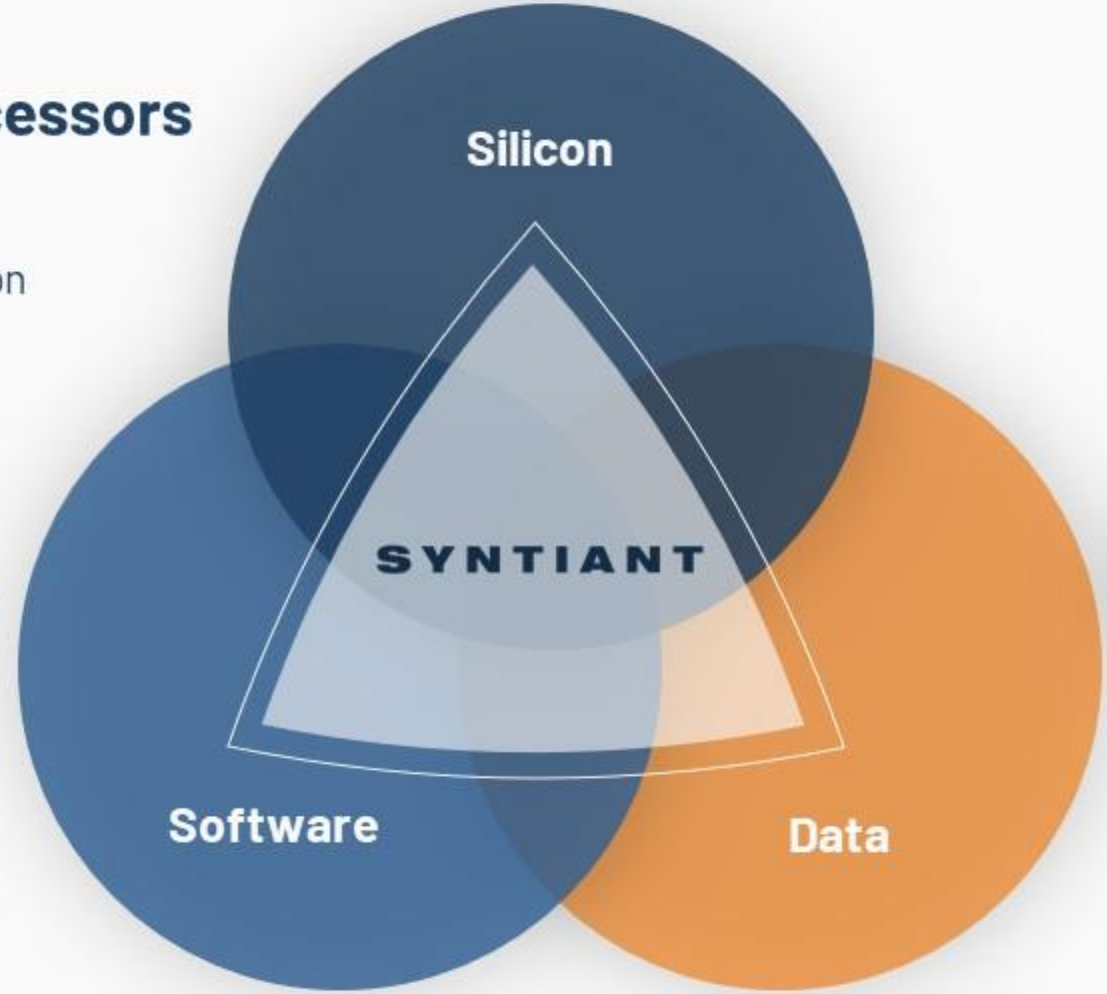
End-to-End
Deep Learning
Solutions

for

TinyML & Edge AI

Neural Decision Processors

- At-Memory Compute
- Sustained High MAC Utilization
- Native Neural Network Processing



ML Training Pipeline

- Enables Production Quality Deep Learning Deployments

Data Platform

- Reduces Data Collection Time and Cost
- Increases Model Performance

Platinum Strategic Partners

Deeplite

WE USE AI TO MAKE OTHER AI FASTER, SMALLER AND MORE POWER EFFICIENT

Automatically compress SOTA models like MobileNet to <200KB with **little to no drop in accuracy** for inference on resource-limited MCUs

Reduce model optimization trial & error from weeks to days using Deeplite's **design space exploration**

APPLY NOW

Deploy more models to your device without sacrificing performance or battery life with our **easy-to-use software**

BECOME BETA USER **bit.ly/testdeeplite**

mobilityXlab arm AI 100 CBINSIGHTS

TINY ML TALKS webcast

KLIKA·TECH

GLOBAL IOT SOLUTIONS

Reality AI®

Add Advanced Sensing to your Product with Edge AI / TinyML

https://reality.ai info@reality.ai @SensorAI Reality AI

Pre-built Edge AI sensing modules, plus tools to build your own

Reality AI solutions

- Prebuilt sound recognition models for indoor and outdoor use cases
- Solution for industrial anomaly detection
- Pre-built automotive solution that lets cars “see with sound”

Reality AI Tools® software

- Build prototypes, then turn them into real products
- Explain ML models and relate the function to the physics
- Optimize the hardware, including sensor selection and placement

BROAD AND SCALABLE EDGE COMPUTING PORTFOLIO

Microcontrollers & Microprocessors

Arm® Core

RENESAS RA — Arm® Cortex®-M 32-bit MCUs
Arm ecosystem, Advanced security, Intelligent IoT

RENESAS RZ — Arm®-based High-end 32 & 64-bit MPUs
High-resolution HMI, Industrial network & real-time control

RENESAS RE — Arm® Cortex®-M0+ Ultra-low Power 32-bit MCUs
Innovative process tech (SOTB), Energy harvesting

Renesas Synergy™ — Arm®-based 32-bit MCUs for Qualified Platform
Qualified software and tools

Renesas Core

RENESAS RL78 — Ultra-low Energy 8 & 16-bit MCUs
Bluetooth® Low Energy, SubGHz, LoRa®-based Solutions

RENESAS RX — High Power Efficiently 32-bit MCUs
Motor control, Capacitive touch, Functional safety, GUI

RENESAS RH850 — 40nm/28nm process Automotive 32-bit MCUs
Rich functional safety and embedded security features

Core technologies

AI

A broad set of high-power and energy-efficient embedded processors

Security & Safety

Comprehensive technology and support that meet the industry's stringent standards

Digital & Analog & Power Solution

Winning Combinations that combine our complementary product portfolios

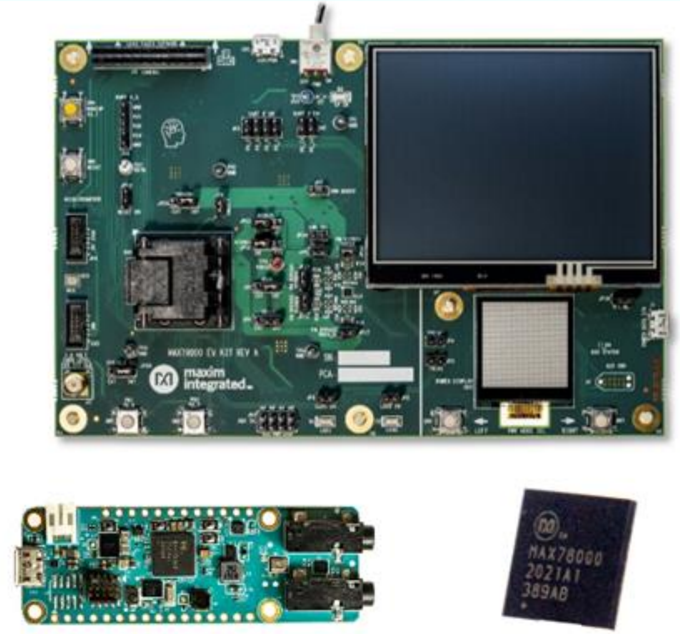
Cloud Native

Cross-platforms working with partners in different verticals and organizations

Gold Strategic Partners

Maxim Integrated: Enabling Edge Intelligence

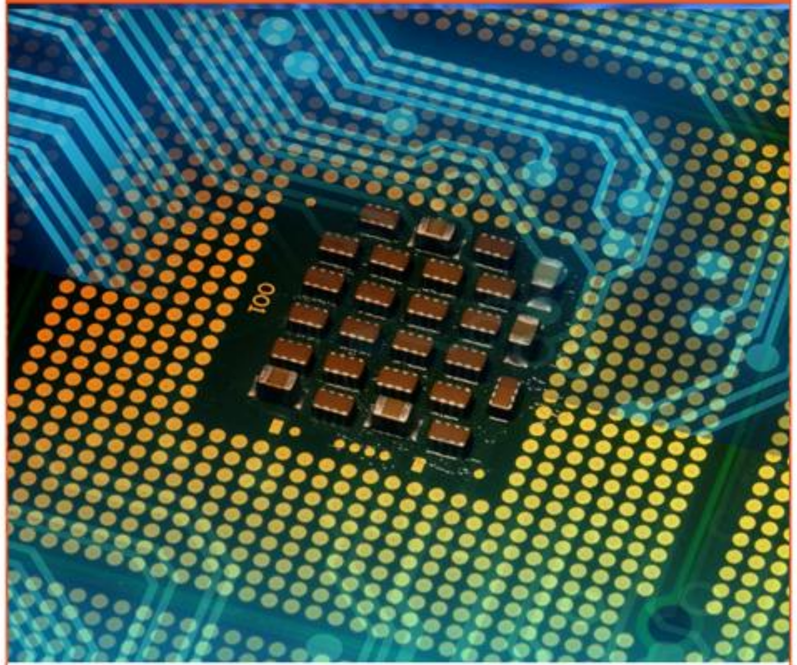
Advanced AI Acceleration IC



The new MAX78000 implements AI inferences at low energy levels, enabling complex audio and video inferencing to run on small batteries. Now the edge can see and hear like never before.

www.maximintegrated.com/MAX78000

Low Power Cortex M4 Micros



Large (3MB flash + 1MB SRAM) and small (256KB flash + 96KB SRAM, 1.6mm x 1.6mm) Cortex M4 microcontrollers enable algorithms and neural networks to run at wearable power levels.

www.maximintegrated.com/microcontrollers

Sensors and Signal Conditioning

Health sensors measure PPG and ECG signals critical to understanding vital signs. Signal chain products enable measuring even the most sensitive signals.

www.maximintegrated.com/sensors

LatentAI

Adaptive AI for the Intelligent Edge

Latentai.com

Micro.ai

seeed studio

The IoT Hardware Enabler

SensiML™

Build Smart IoT Sensor Devices From Data

SensiML pioneered TinyML software tools that auto generate AI code for the intelligent edge.

- End-to-end AI workflow
- Multi-user auto-labeling of time-series data
- Code transparency and customization at each step in the pipeline

We enable the creation of production-grade smart sensor devices.

sensiml.com

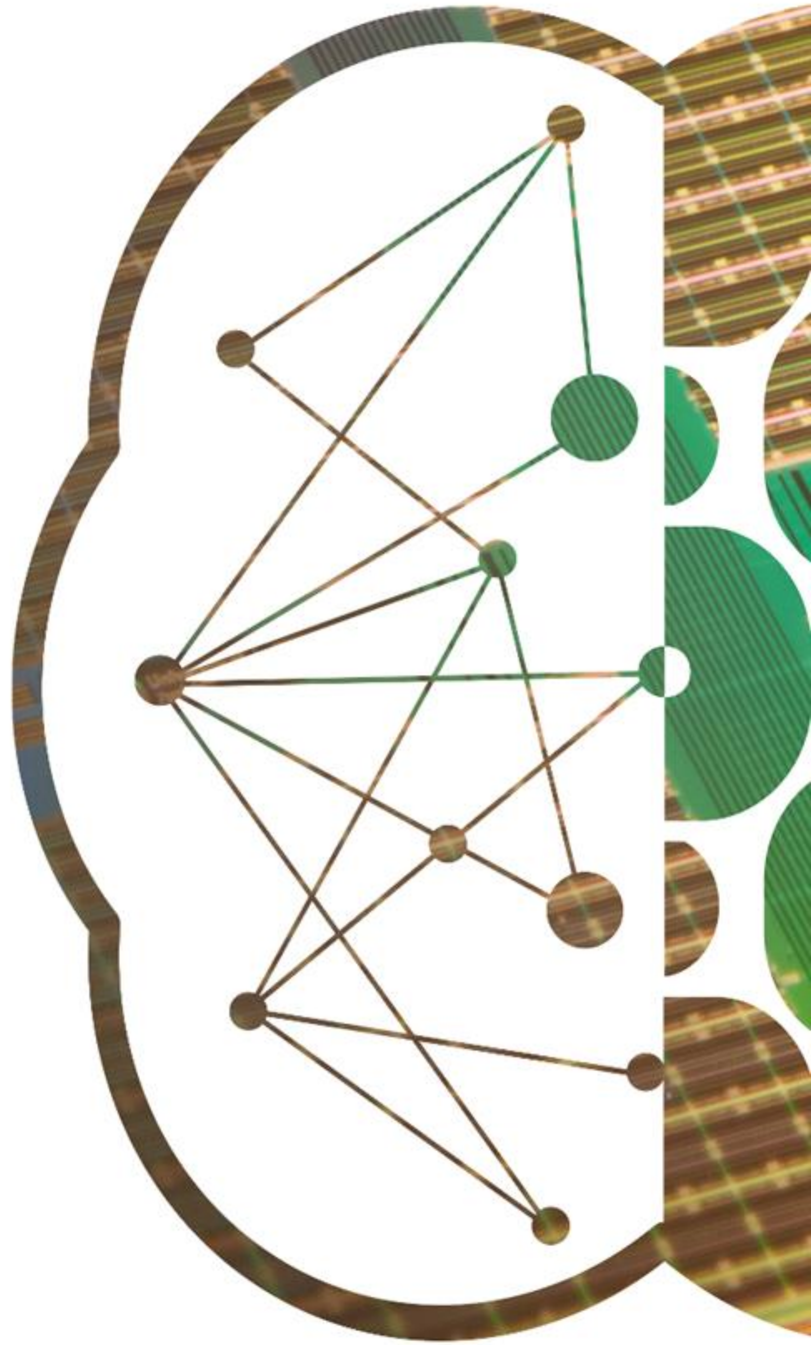
TINY ML TALKS webcast

life.augmented

SynSense

SynSense builds **sensing and inference** hardware for **ultra-low-power** (sub-mW) **embedded, mobile and edge** devices. We design systems for **real-time always-on smart sensing**, for audio, vision, IMUs, bio-signals and more.

https://SynSense.ai



TINY ML TALKS webcast

Silver Strategic Partners

AONdevices

emza visual sense

GREENWAVES TECHNOLOGIES

Grovety Inc.

HOTG

imagimob

PROPHESEE METAVISION FOR MACHINES

Qeexo

REEXEN technology

SAP

SA STREAM ANALYZE

tinyML Summit 2022

Miniature dreams can come true…

March 28-30, 2022

Hyatt Regency San Francisco Airport

https://www.tinyml.org/event/summit-2022/

*The Best Product of the Year and the Best Innovation of the Year awards are open for nominations between **November 15 and February 28.***

tinyML Research Symposium 2022

March 28, 2022

https://www.tinyml.org/event/research-symposium-2022

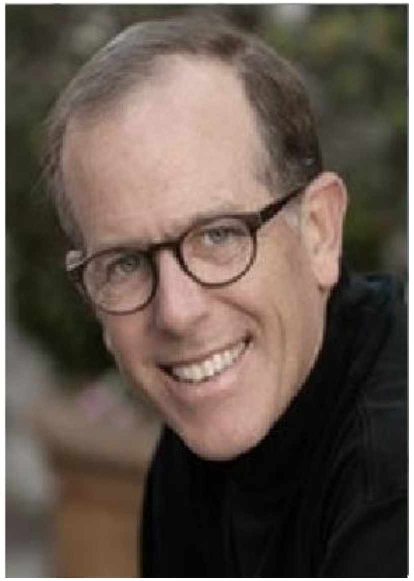
More sponsorships are available: sponsorships@tinyML.org

tinyML Trailblazers Series

Success Stories with Kurt Busch

LIVE ONLINE **January 19th, 2021 at 9 am PST**

TRAILBLAZERS



Register now!

Join Growing tinyML Communities:

7.7k members in
42 Groups in 33 Countries

tinyML - Enabling ultra-low Power ML at the Edge

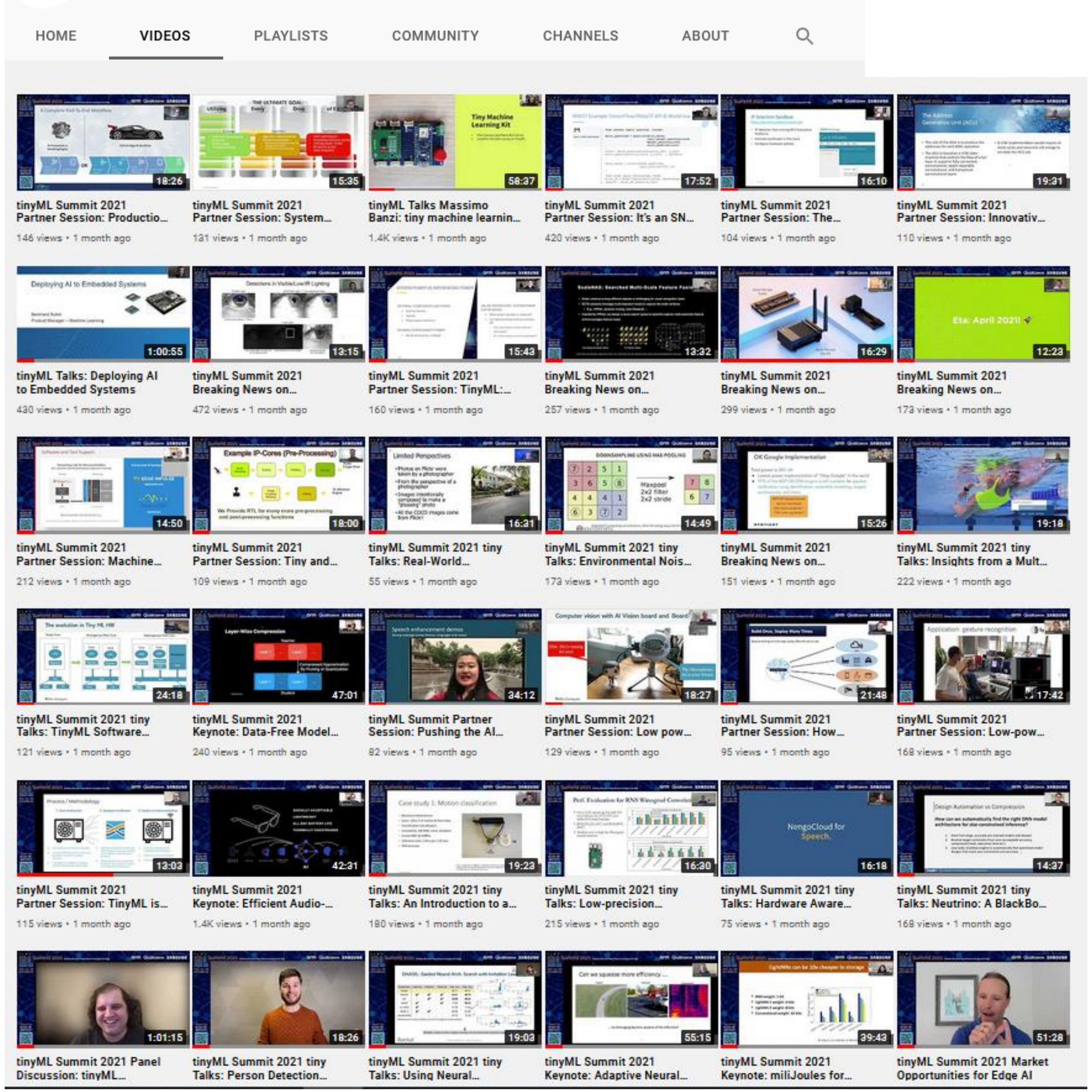
https://www.meetup.com/tinyML-Enabling-ultra-low-Power-ML-at-the-Edge/

2.5k members
&
4.3k followers

The tinyML Community

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

5.8k subscribers, 320+ videos with 166k views



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

www.youtube.com/tinyML

Next tinyML Talks

Date	Presenter	Topic / Title
Tuesday, February 1	Muhammad Shafique, New York University Abu Dhabi (NYUAD), UAE	Energy-Efficiency and Security for TinyML and EdgeAI: A Cross-Layer Approach

Webcast start time is 8:00 am Pacific time

Please contact talks@tinyml.org if you are interested in presenting

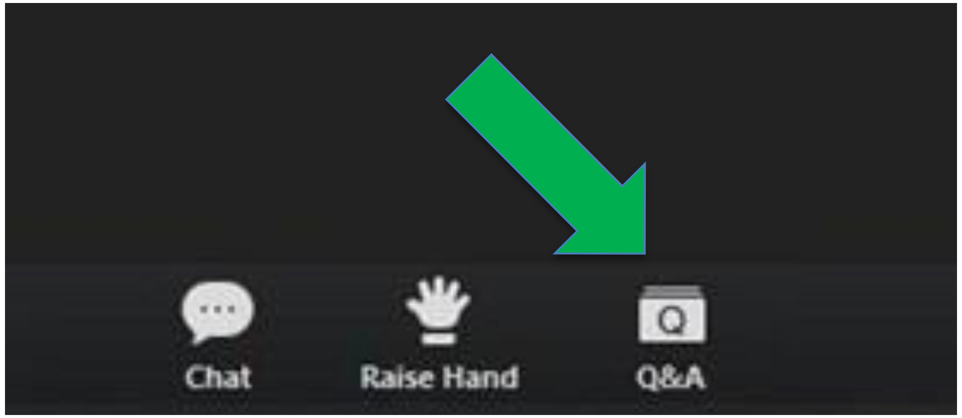
Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums

youtube.com/tinyml

Please use the Q&A window for your questions



Ashutosh Pandey

Ashutosh Pandey is currently a Lead Principal Systems Engineer at Infineon Technologies where he is responsible for Machine learning solutions, architecture, and tooling. He holds a PhD from the University of Utah and has over 40 papers and patents on speech/audio/machine learning systems and algorithms.

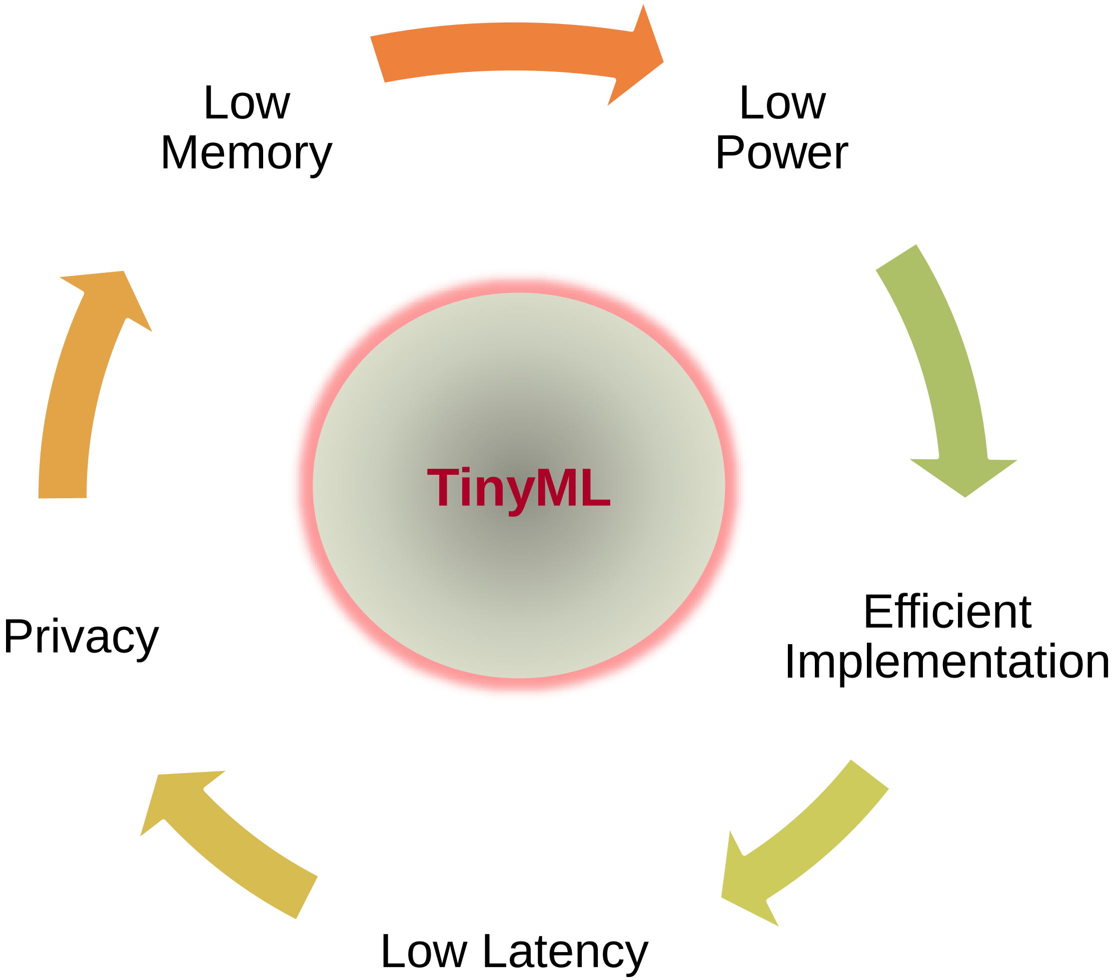
Agenda

- Understanding TinyML attributes, challenges & ambitions
  - Motivation for the talk
  - TinyML Architecture
  - Modified Architecture
- Understanding role of data for tinyML
  - TinyML approach
  - Deep representation
- Understanding Quantization
  - TinyML approach
  - Modified approach
- Understanding Out-Of-Distribution
  - Current Approaches
  - Generalized approach
- Putting it all together
- Conclusion

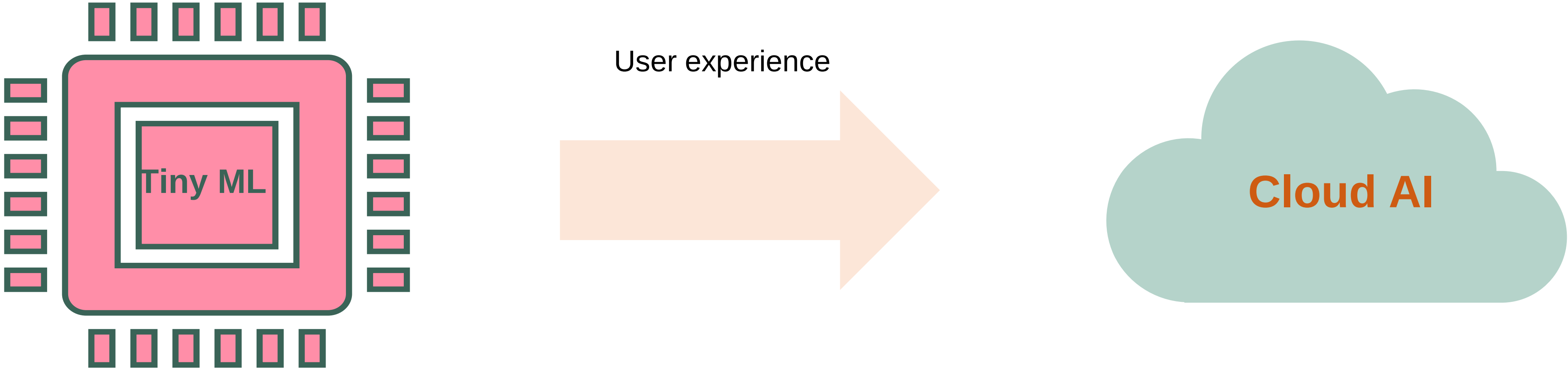
Team

- Aidan Smyth
- Niall Lyons
- Kaiping Li
- Charley Chu
- Sree Harsh Angara
- Vijay Deep Bhatt
- Avik Santra
- Ashutosh Pandey

What is tinyML?



TinyML Constraints and journey towards TinyAI

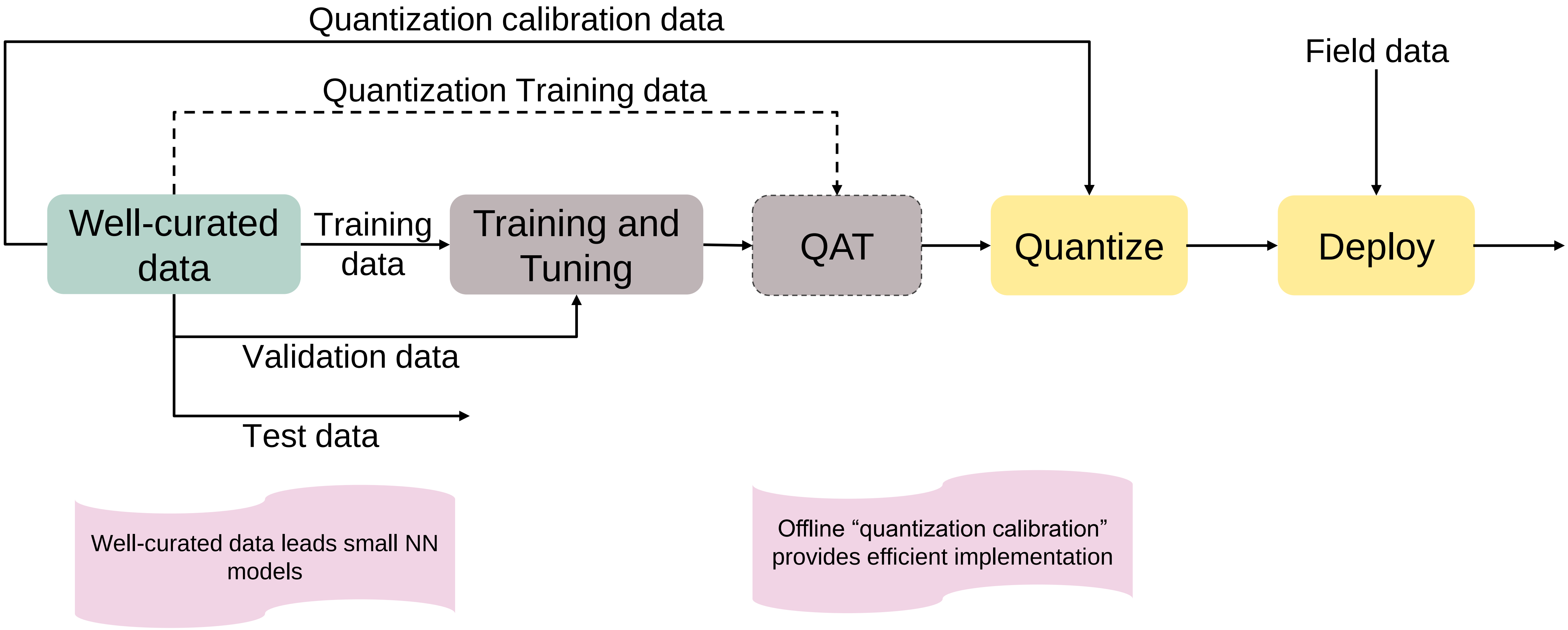


- Limited parallel processing
- Limited precision
- Private
- Non-ensemble learning
- Limited Memory
- Low-latency
- Slow update cycle – every aspect counts for user experience

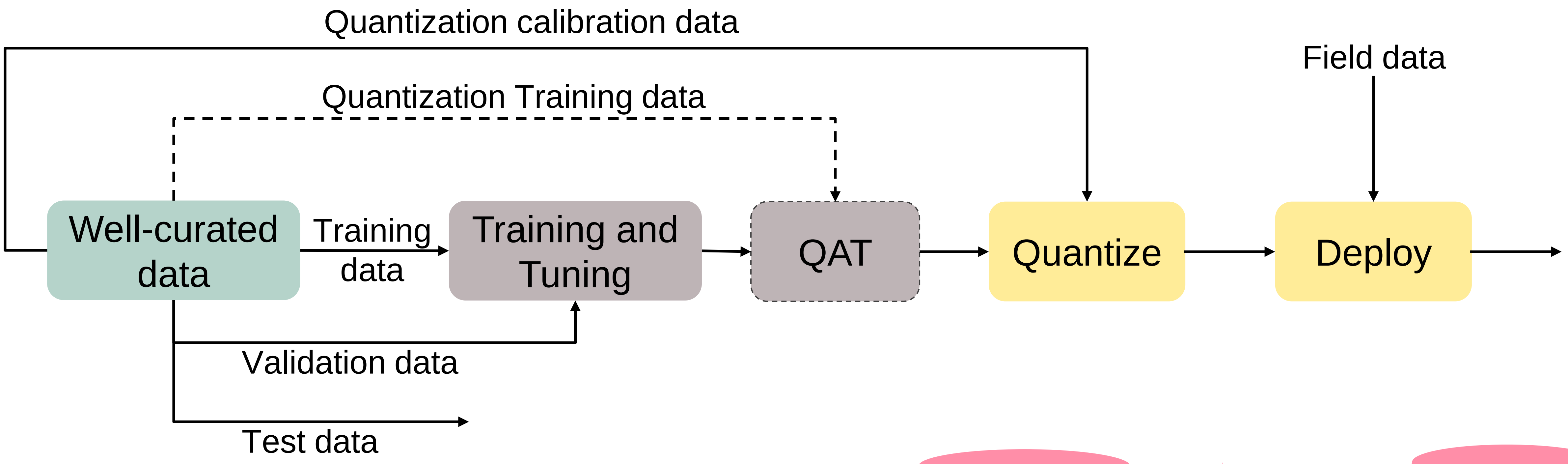
- Parallel, “unlimited” peak performance
- No special requirement
- Semi “private”
- Ensemble learning
- Post analysis through monitoring
- Buffer processing acceptable
- Fast updates – start sub-optimal and improve on-the-go

- Achieving CloudAI performance with TinyML constraints would add great user experience, reliable, scale to multiple applications

Successful prevalent tinyML Solution Architecture



Successful prevalent tinyML Solution Architecture - challenges

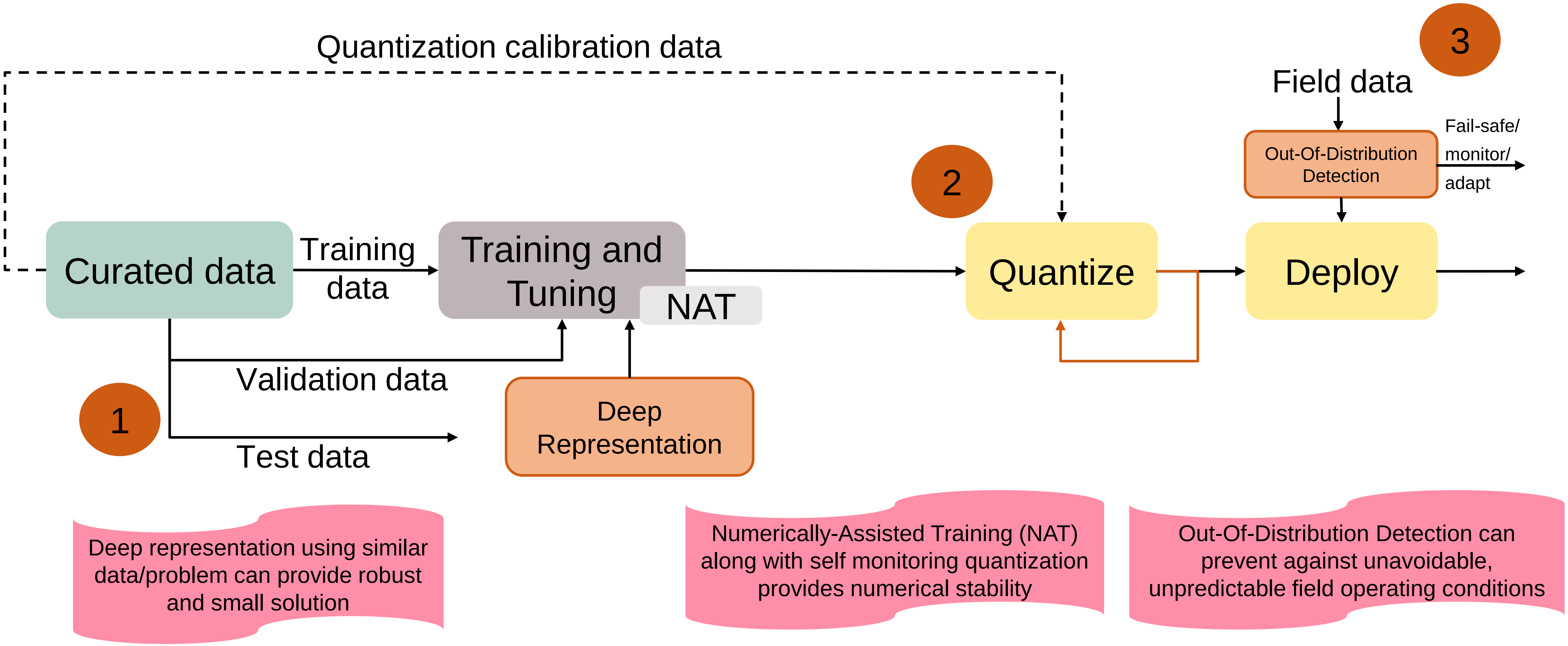


Large dataset needed with balanced statistics at each stage – time-consuming

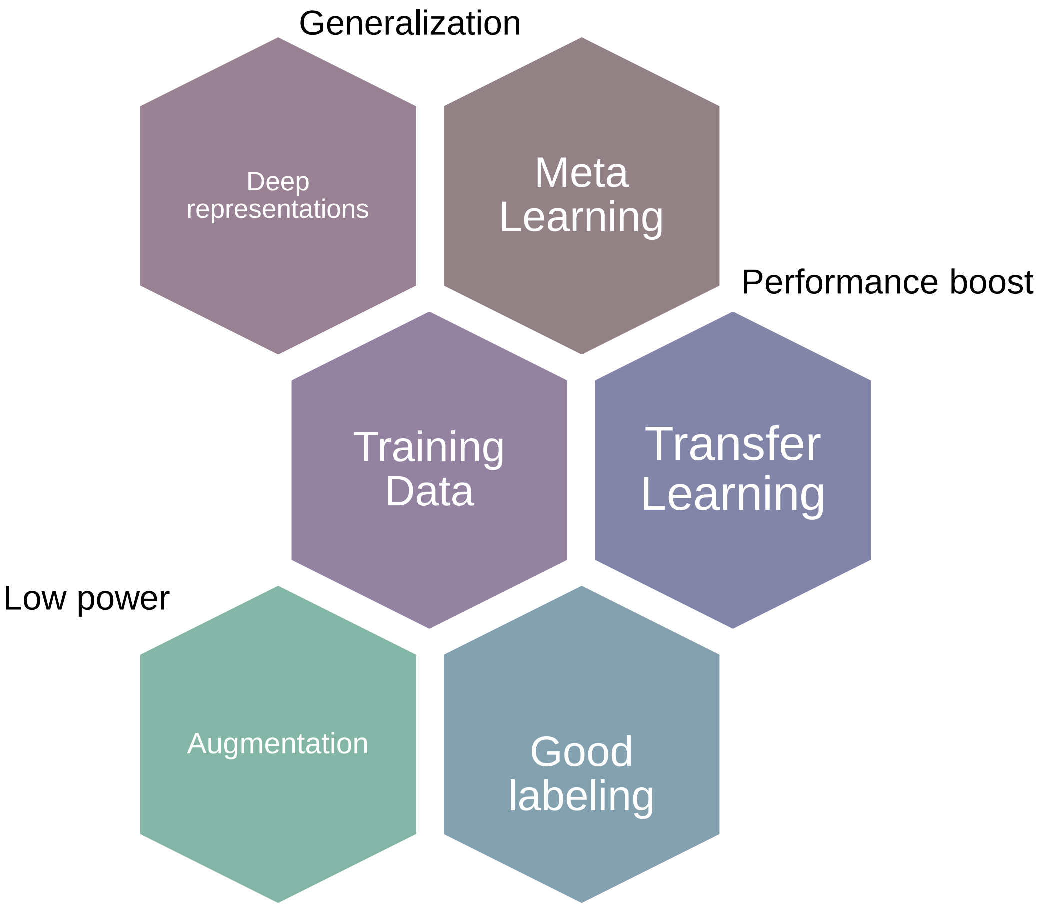
Abrupt changes due to numerical precision if out-of-distribution is observed – security/privacy application may limit “available data”

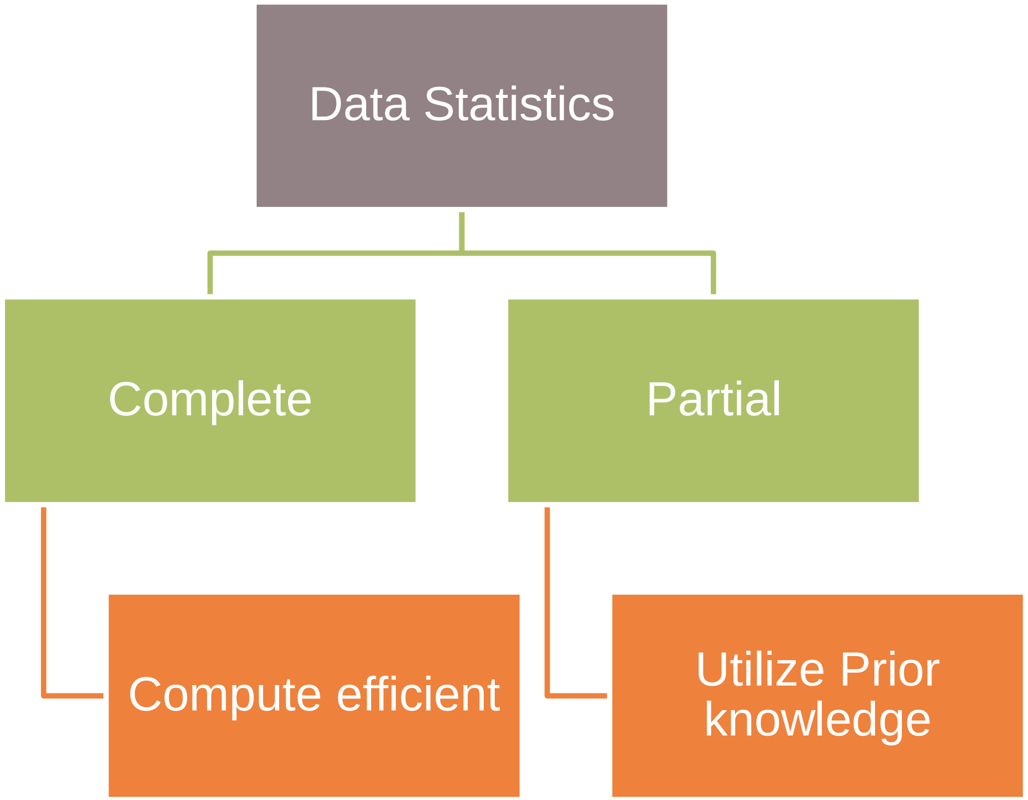
No monitor/feedback mechanism is susceptible to “unsafe” behavior

Proposed tinyML Solution Architecture



Understanding role of data in deep learning Systems

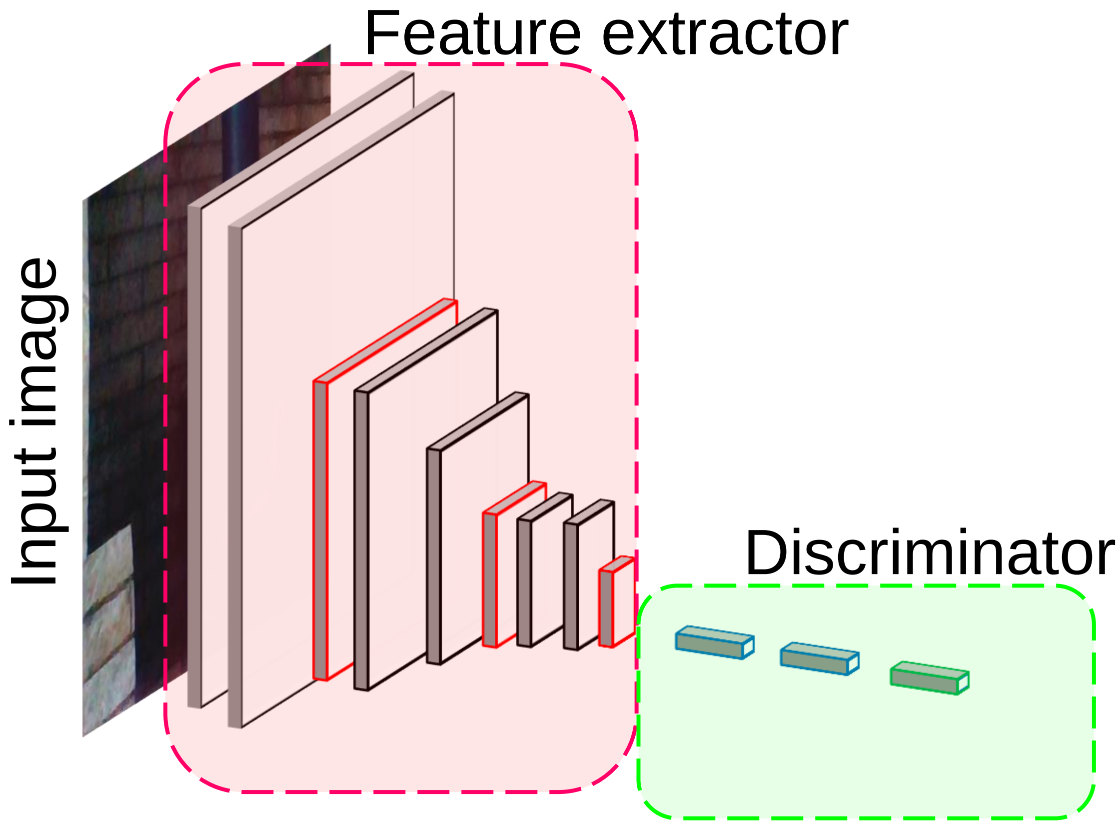




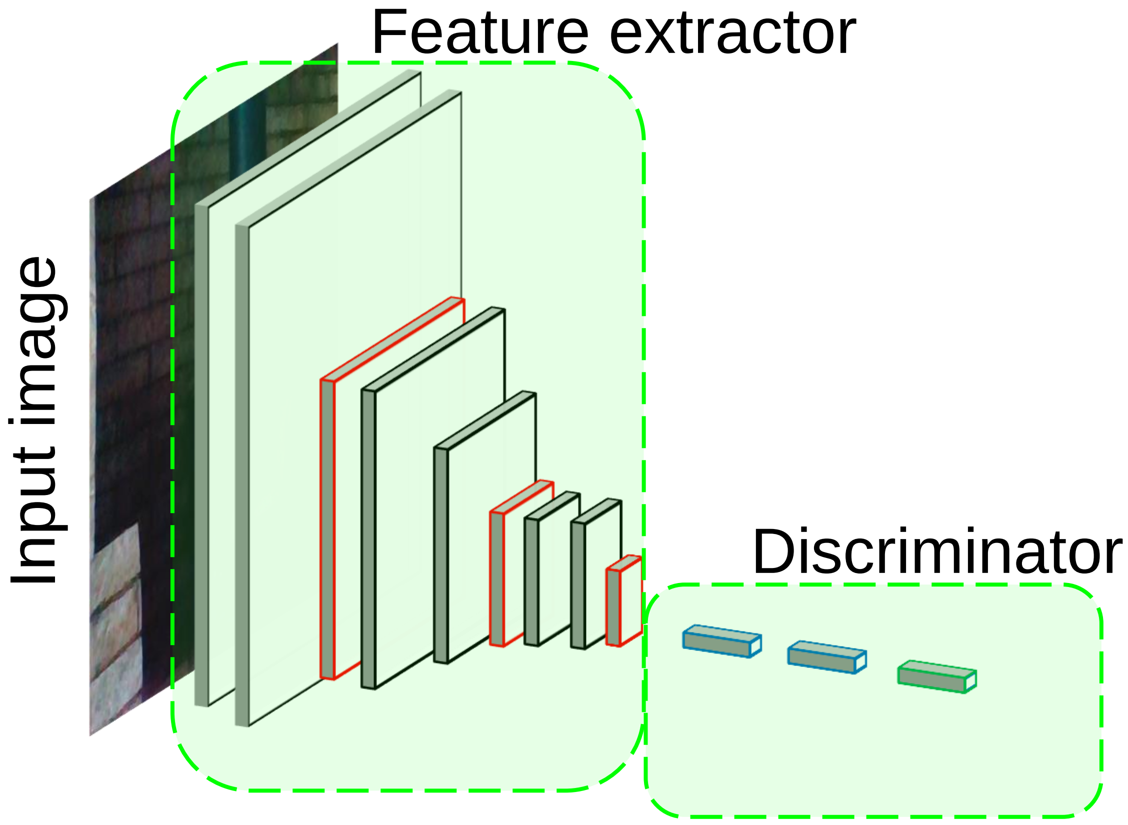
TinyML approach

Transfer Learning - Fine Tuning

- Fine tuning disabled
  - Default
  - Train discriminator and freeze feature extractor
  - Reduces risk of overfit

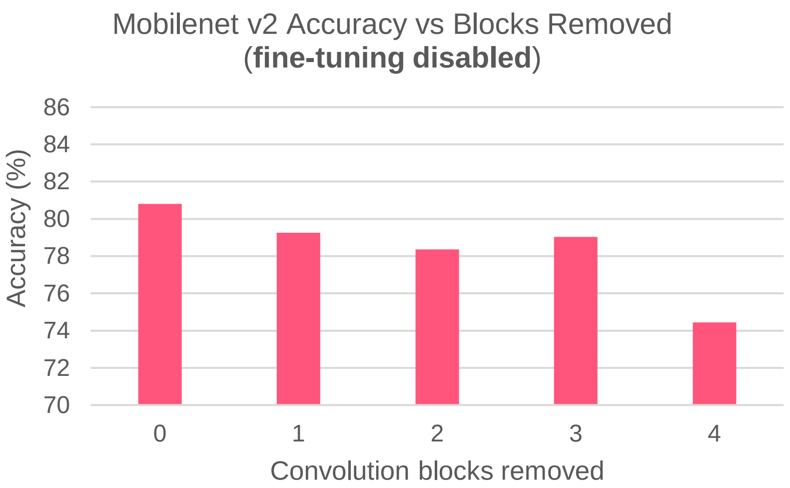


- Fine tuning enabled
  - Train all layers
  - Small dataset -> Risk overfit
  - Large dataset -> Improved accuracy



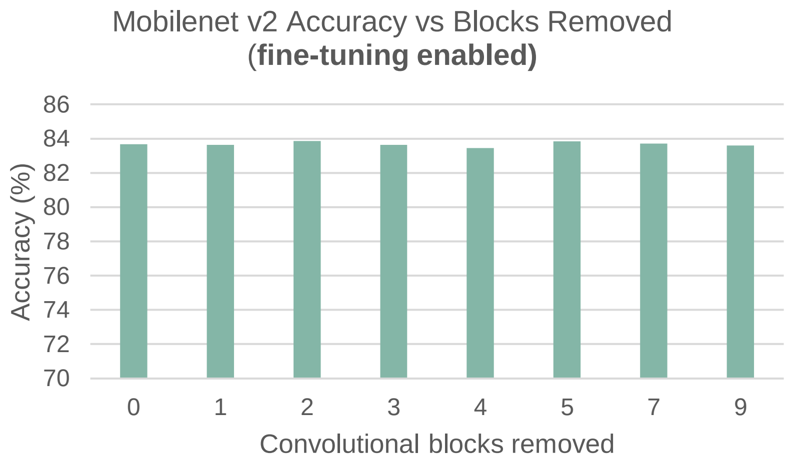
Transfer Learning - Fine Tuning & Model Optimization

› Removing convolutional blocks with fine-tuning of feature extractor **disabled**:
  - significant performance degradation



Transfer learning works but…

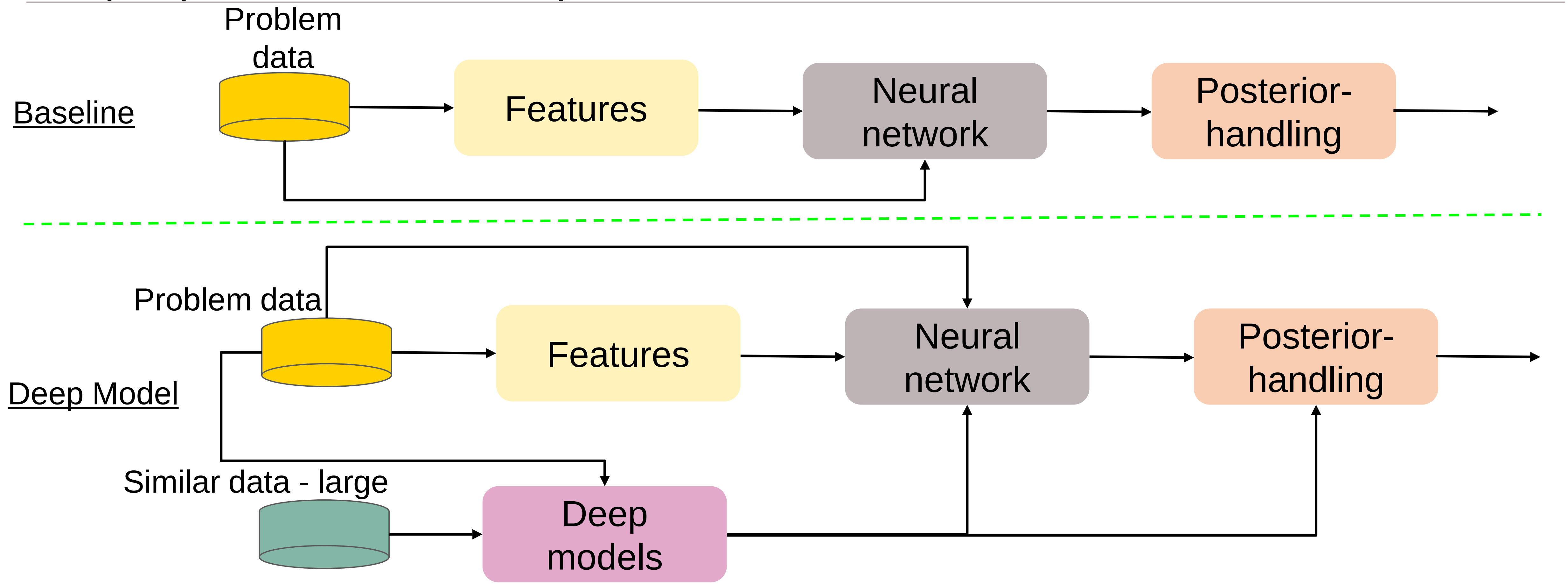
› Removing convolutional blocks with fine-tuning of feature extractor **enabled**:
  - no performance degradation
  - extensive model optimization



Compute efficient

Deep Representation

Deep representations - blueprint



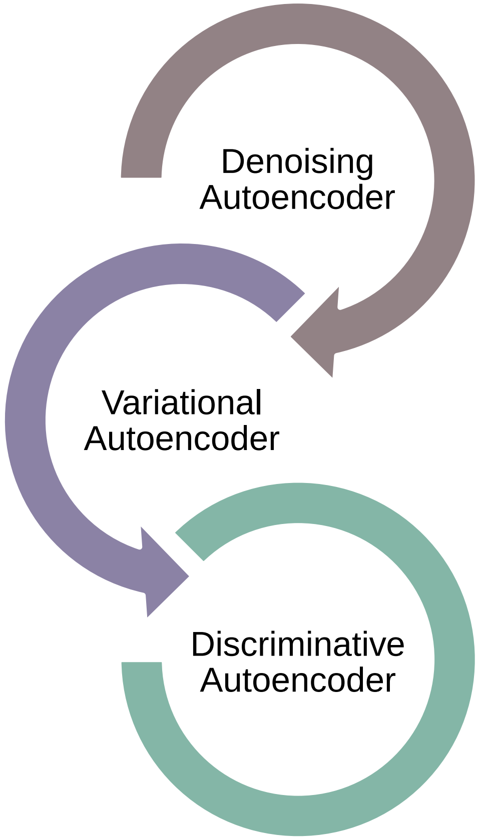
- Deep models can utilize large datasets to provide “additional” performance or utilize small datasets to provide robustness
- Deep models should be able to generalize to everyday problems – denoising, statistical, classification *etc*
- Deep models are designed using tinyML principles and can be utilized across similar problems

Deep Model Design - AutoEncoders

Reconstruction Loss

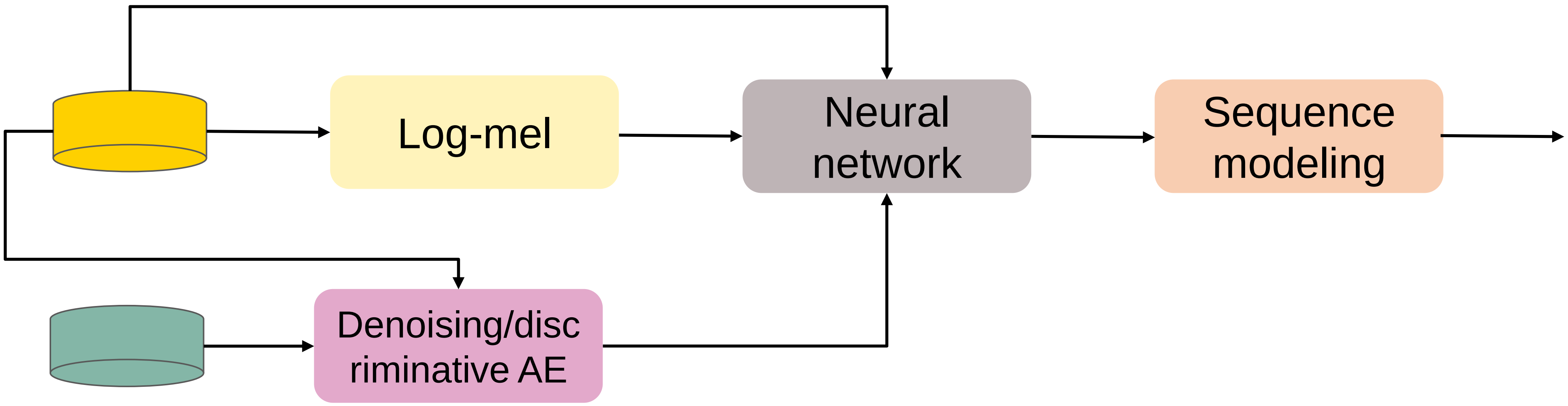
Reconstruction Loss + K-L divergence

Reconstruction Loss + K-L divergence + Classification loss

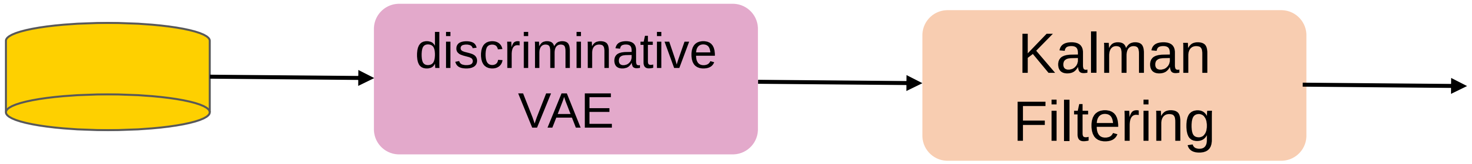


Extending Deep Model methodology to TinyML Applications

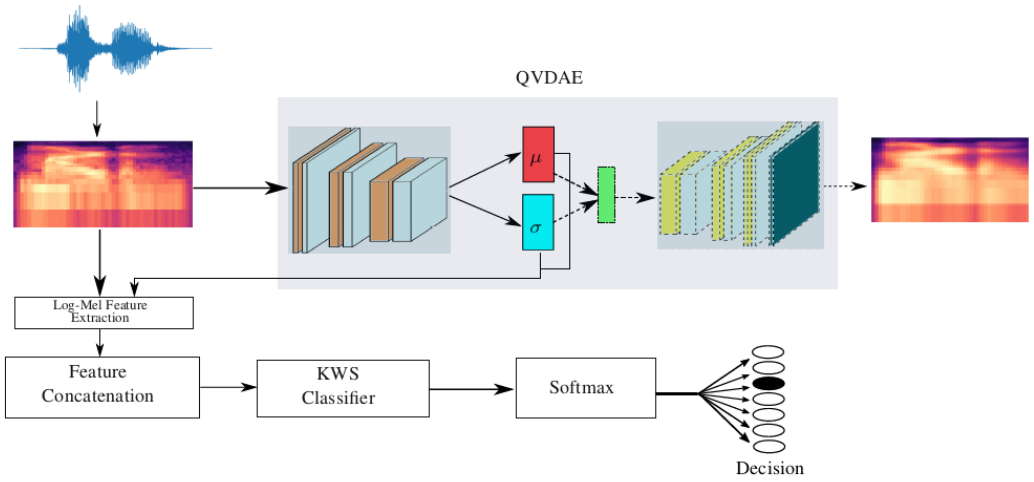
- KeyWord Spotting (KWS) – large dataset



- Human Activity Recognition (HAR) – limited data



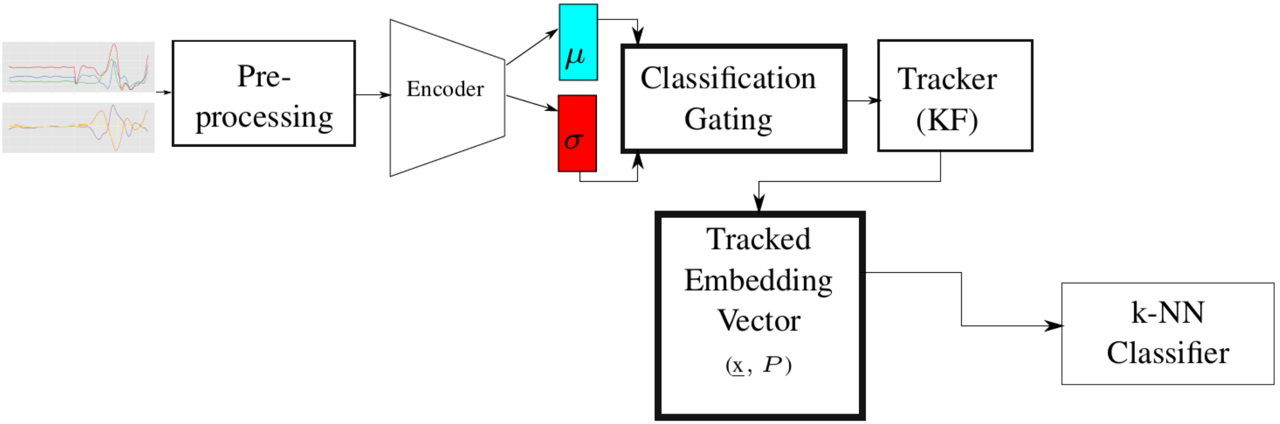
Performance – application of AutoEncoders to KWS



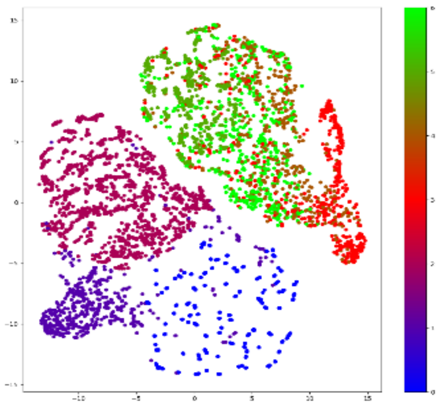
Approach	In-distribution	Out-of-distribution
Baseline	96.3	89.4
Deep - Denoising	98.1	93.3
Deep - Variational	98.0	93.6
Deep – Variational + Discriminative	99.0	94.7

› Deep representation approach provided additional performance
› Deep representations generalized well

Performance - HAR



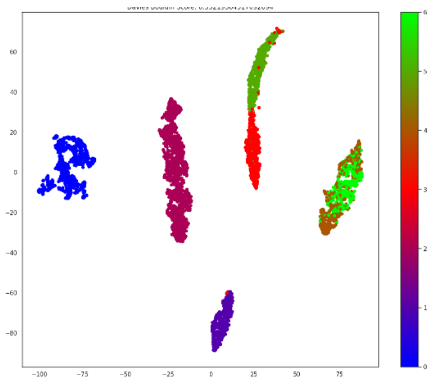
Base classifier



Confusion Matrix QVAE

True label \ Predicted label	Idle	Jump	Sit	Squat	Stairs	Stand	Walk
Idle	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Jump	0.01	0.97	0.02	0.00	0.00	0.00	0.00
Sit	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Squat	0.00	0.00	0.00	0.83	0.07	0.06	0.04
Stairs	0.00	0.00	0.00	0.20	0.61	0.04	0.14
Stand	0.00	0.00	0.00	0.03	0.02	0.95	0.00
Walk	0.00	0.00	0.00	0.25	0.29	0.10	0.36

Deep Bayesian Framework



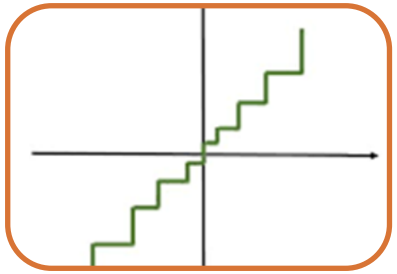
Confusion Matrix QVAE

True label \ Predicted label	Idle	Jump	Sit	Squat	Stairs	Stand	Walk
Idle	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Jump	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Sit	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Squat	0.00	0.00	0.00	0.99	0.00	0.00	0.00
Stairs	0.00	0.00	0.00	0.00	0.98	0.00	0.01
Stand	0.00	0.00	0.00	0.02	0.00	0.98	0.00
Walk	0.00	0.00	0.00	0.00	0.24	0.00	0.76

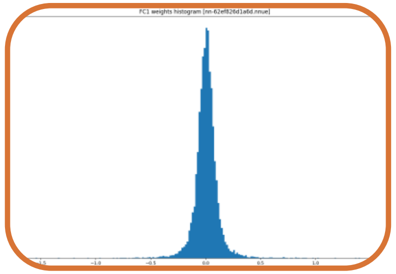
› Deep representation can be utilized with Bayesian framework to create a fundamental building block for tinyML systems

Quantization

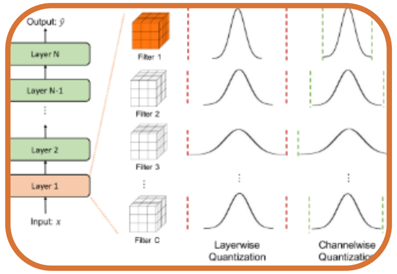
Concepts



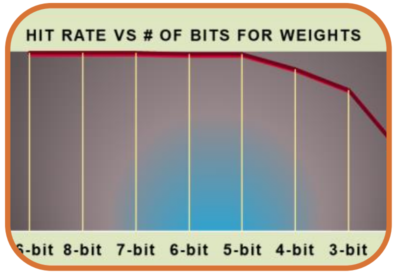
Symmetric, uniform?



Clamp?



Granularity

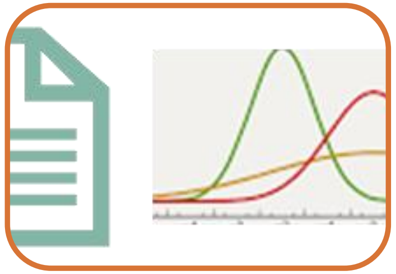


Bits

$$\sum p(x) \log_2 \frac{p(x)}{q(x)}$$

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Optimization



Data statistics

Fine tuning

https://www.arxiv-vanity.com/papers/2103.13630/
https://arxiv.org/pdf/2103.13630.pdf

TinyML paradigm – well understood

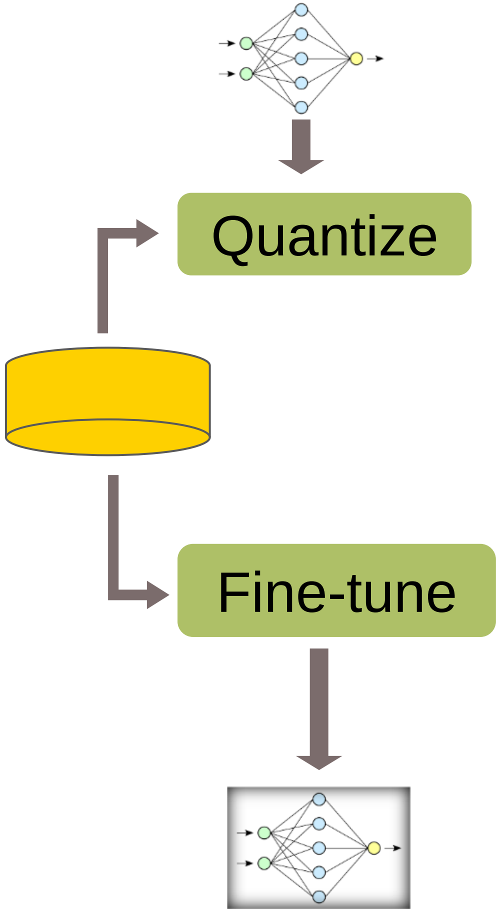
Setup

- Asymmetric uniform quantization
- Per-layer
- 8-bit weights
- Batch normalization

Activations

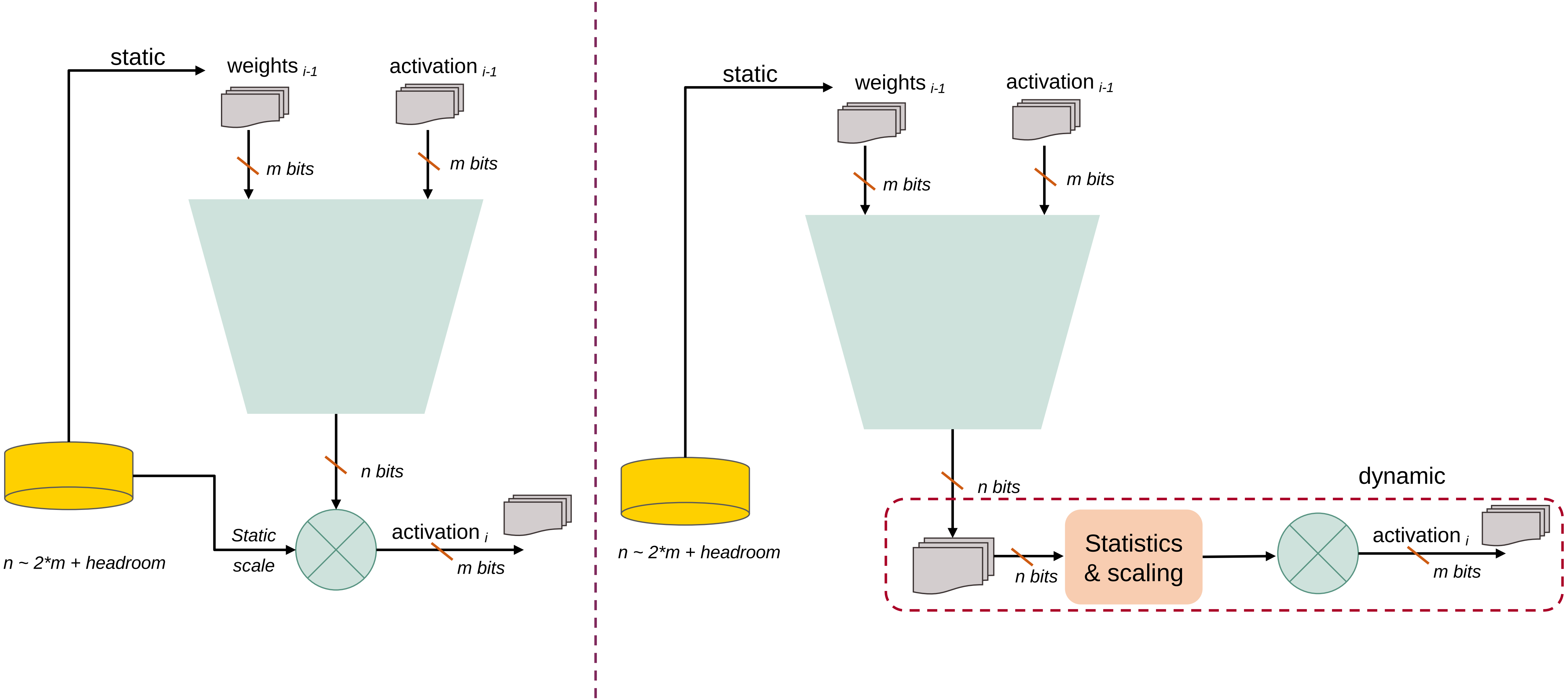
- Well separable  › >4-bit
- discriminative  › 8-bit
- regression, RNNs › >8-bit

Methodology



- Utilizing fine-tuning/calibration statistics alleviate quantization short-comings by pushing out quantization errors to NULL hyperspaces emanating from data statistics

Static *vs* dynamic fixed-point



Performance

Studying Post Training Quantization with limited OOD calibration/test data (asymmetric, uniform, 8-bit weight, 8-bit activation, convolution based)			
Task	Params (k)	Top-1, Relative Accuracy (%)	
		Per-layer Dynamic (MAE)	Per-channel – static (MAE)
HAR	20	100 (0.01)	92 (0.18)
KWS	90	95 (0.08)	91 (0.17)
Image detection – mobilenetv2	224	84 (0.12)	34 (0.64)
Face detection – minVGG2	74	100 (0.02)	95 (0.14)

- Poor quantized accuracy relative to floating-point is indicative of numerical issues – a successful quantized implementation should have graceful degradation
- Accuracy numbers may mask numerical issues that show up when distributions shift

Implementation cost

Extra resources for dynamic fixed-point compared to static quantization (NVM requirements are the same)		
Task	% cycles increased using Scalar ISAs	% Additional memory (RW) required (kB)
HAR	8 %	12% (8 kB)
KWS	12 %	32% (21 kB)
image detection – mobilenetv2	17 %	27%(40 kB)
Face detection – minVGG2	15 %	66%(72 kB)

- Dynamic fixed-point implementation requires incremental logic cost but memory increments can be significant compared to static quantization implementation
- The increase in dynamic quantization implementation are within typical tinyML platform's capability

Out of Distribution Detection

Out of Distribution Detection - Motivation

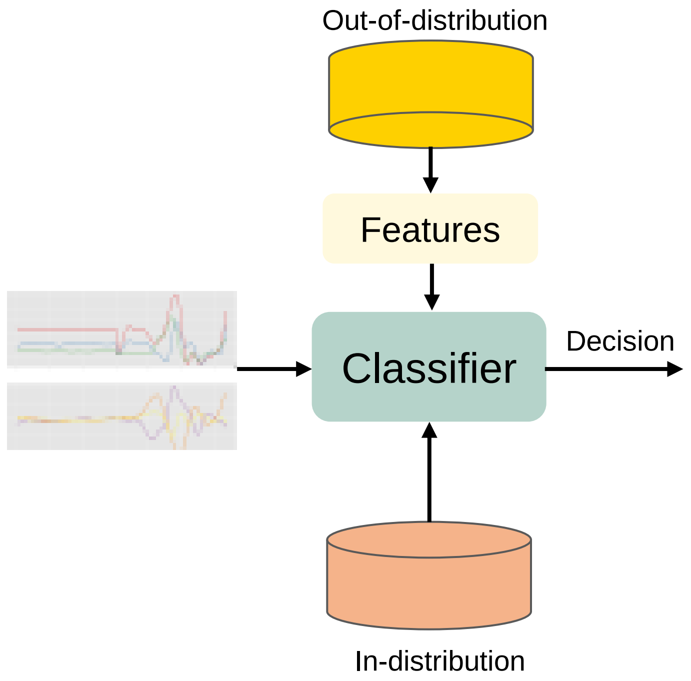
- Measurement noise is invariably present in sensor data and does not vanish with infinite data
- NN models are susceptible to miss-classification under data variability arising due to
  - operating environment,
  - Interferences,
  - sensor degradation,
  - activity transitions
- System should not only predict a class but also handle adverse, out of distribution activities more efficiently
- Decrease power consumption for Neural Network Inference

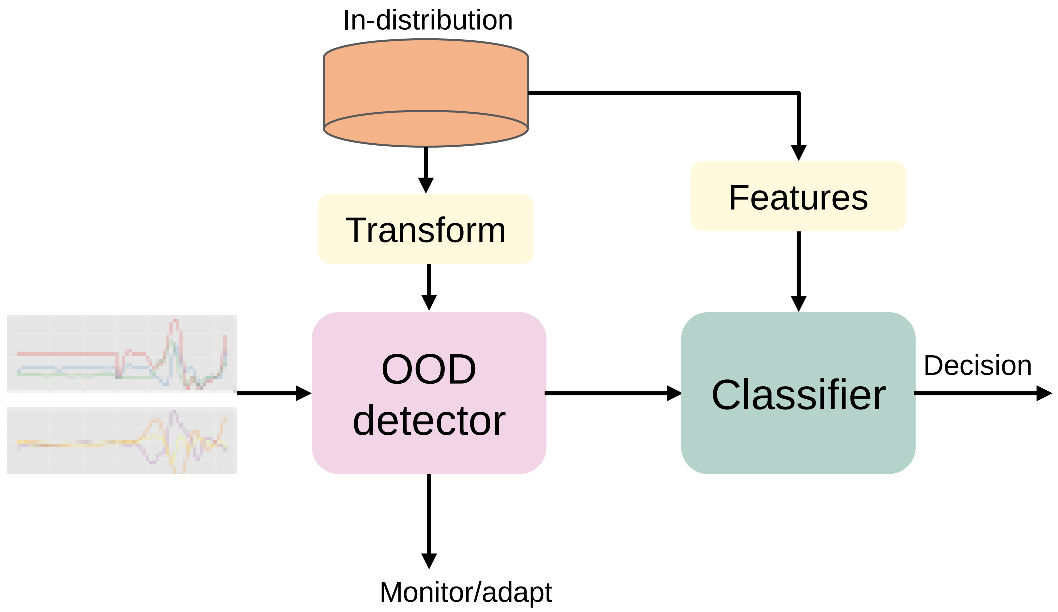
OOD detection approaches

Something is better than nothing

- Implicit OOD detection - Condition the model
  - Model monitoring
  - Model calibration with “limited” out-of-distribution
  - Ensemble
- Explicit OOD detection
  - Open world training
  - OOD detection using In-distribution training

Explicit OOD Architecture





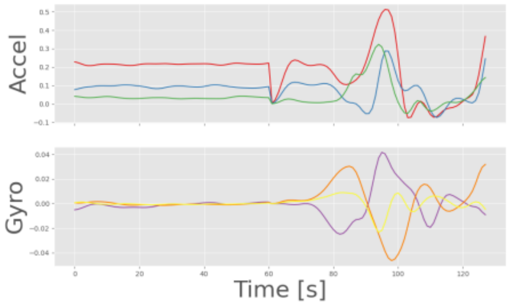
- Requires extensive out-of-distribution data
- Doesn’t account for sensor degradation *etc*
- Bias introduced for in-distribution estimation due to large out-of-distribution data

- Extra processing
- OODD can be designed through tinyML practices
- Shadow implementation builds redundancy into system and provides robustness
- Different transform domain pre-processing on raw samples can be utilized

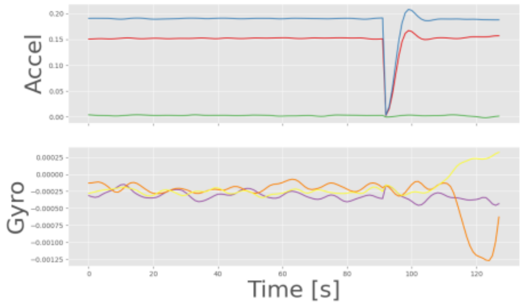
https://arxiv.org/abs/2002.11297

Data Transformation

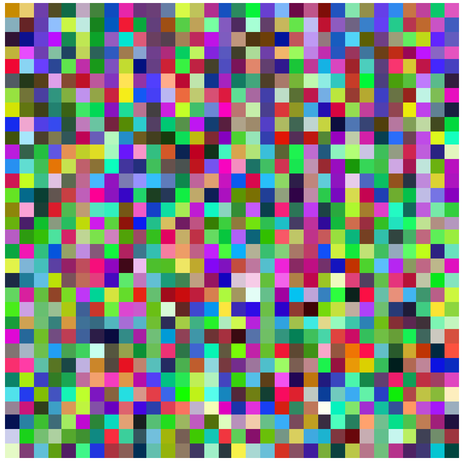
- Step 1:
  - $\hat{f} = \frac{f}{||f||}$
- Step 2:
  - $F = \begin{bmatrix} f_{1,x} & f_{1,y} & f_{1,z} \\ \vdots & \vdots & \vdots \\ f_{n,x} & f_{n,y} & f_{n,z} \end{bmatrix} = [F_x, F_y, F_z]$
- Step 3:
  - $\Omega = \begin{bmatrix} w_{1,x} & w_{1,y} & w_{1,z} \\ \vdots & \vdots & \vdots \\ w_{n,x} & w_{n,y} & w_{n,z} \end{bmatrix} = [\Omega_x, \Omega_y, \Omega_z]$
- Step 4:
  - $R = F_x . \Omega_x^T \in R^{n \ x \ n}$
  - $G = F_y . \Omega_y^T \in R^{n \ x \ n}$
  - $B = F_z . \Omega_z^T \in R^{n \ x \ n}$
- Step 5:
  - $I = [R, G, B] \in R^{nxnx3}$



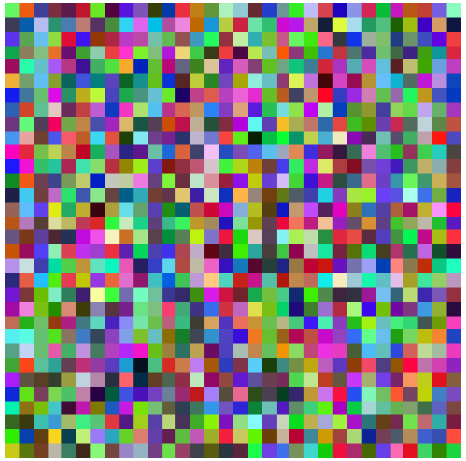
(a) Jumping



(b) Standing



(a) Jumping



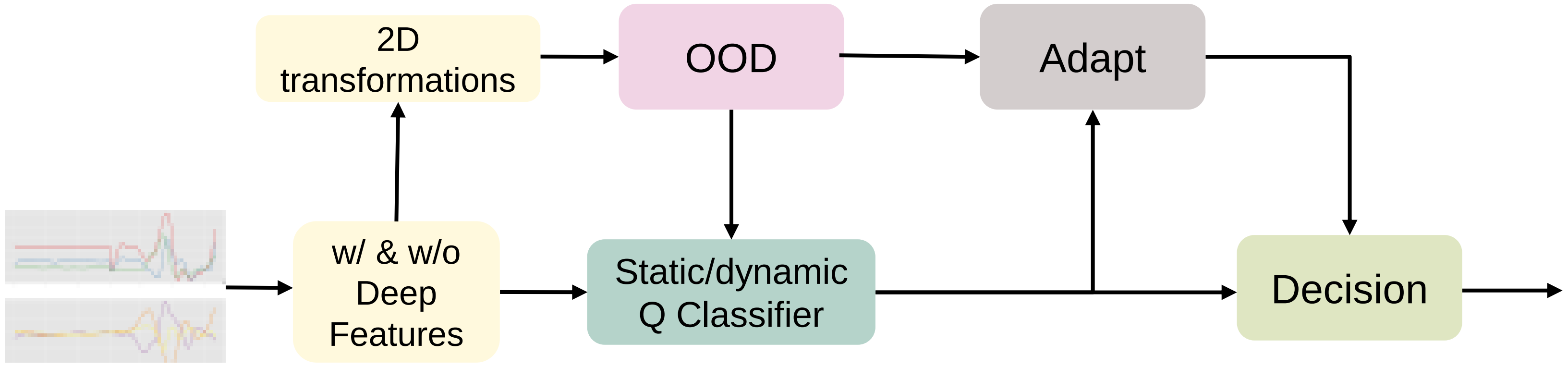
(b) Standing

OOD detector performance

- In Distribution Data
  - Jumping
  - Standing
  - Walking
  - Squatting
  - Stairs
- Out Of Distribution
  - Sitting
  - Kicking
  - Idle Class

Performance Metric	Baseline OoD	Proposed OoD
AUROC	0.87	**0.96**
TNR@TPR95	0.88	**0.99**

Putting all things together



	Top-1 Accuracy, deep features: floating-point, QAT					
	Before Adaptation		After Adaptation			
Adaptation implementation	N/A		Float		Static, 16-bit	Dynamic 16-bit
Inference Implementation	Static-Q, 8 bit	Dynamic-Q, 8-bit	Static-Q, 8 bit	Dynamic-Q, 8-bit	Static-Q, 8 bit	Dynamic-Q, 8-bit
w/ Deep	99	99	95	98	NA	97
w/o Deep	86	86	80	85	NA	85

› Dynamic scaling SW/HW assets would augment learning capability on the edge

Conclusion

- TinyML techniques have matured and established a good working development flow
- By tweaking current prevalent tinyML development flow, we can increase chances of success for tinyML solutions
- This would unlock the next phase of success for tinyML solutions and would help us move one step closer to true “edge” only solutions
- ModusToolbox™ ML supports tinyML friendly tooling to assess model quantization performance and support for dynamic fixed-point

Part of your life. Part of tomorrow.

Copyright Notice

www.tinyML.org