Friedrich-Alexander-Universität
Technische Fakultät

Unsupervised Learning of Variational Autoencoders on Cortex-M Microcontrollers

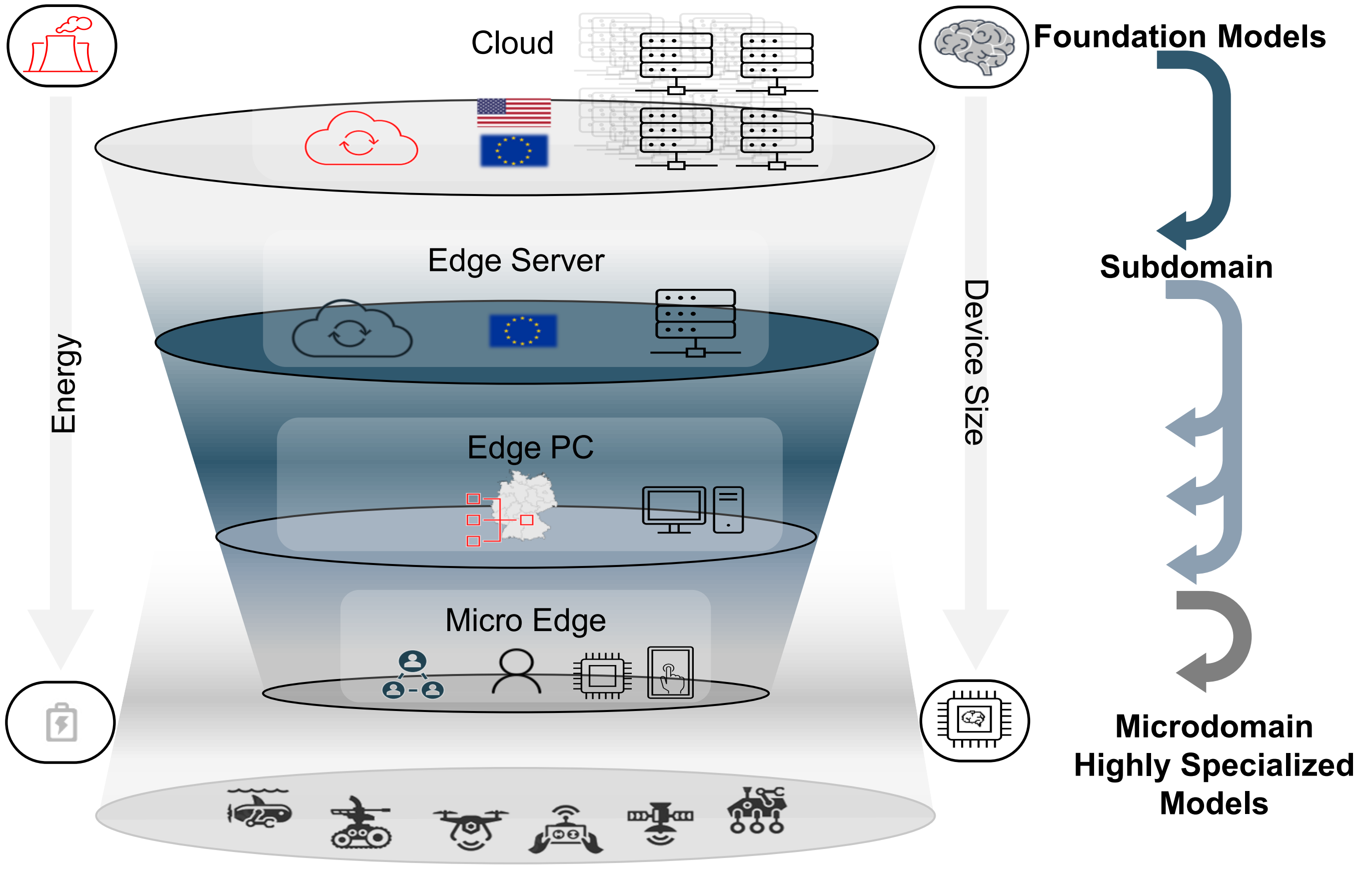
IEEE International Symposium on Embedded Multicore/Manycore SoCs (MCSoC-2025)

Mark Deutel[1,2], Axel Plinge[2], Dominik Seuß[2], Christopher Mutschler[2], Frank Hannig[1], Jürgen Teich[1]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg
[2] Fraunhofer Institut für Integrierte Schaltungen, Fraunhofer IIS

Motivation

The Cloud-Edge-Continuum



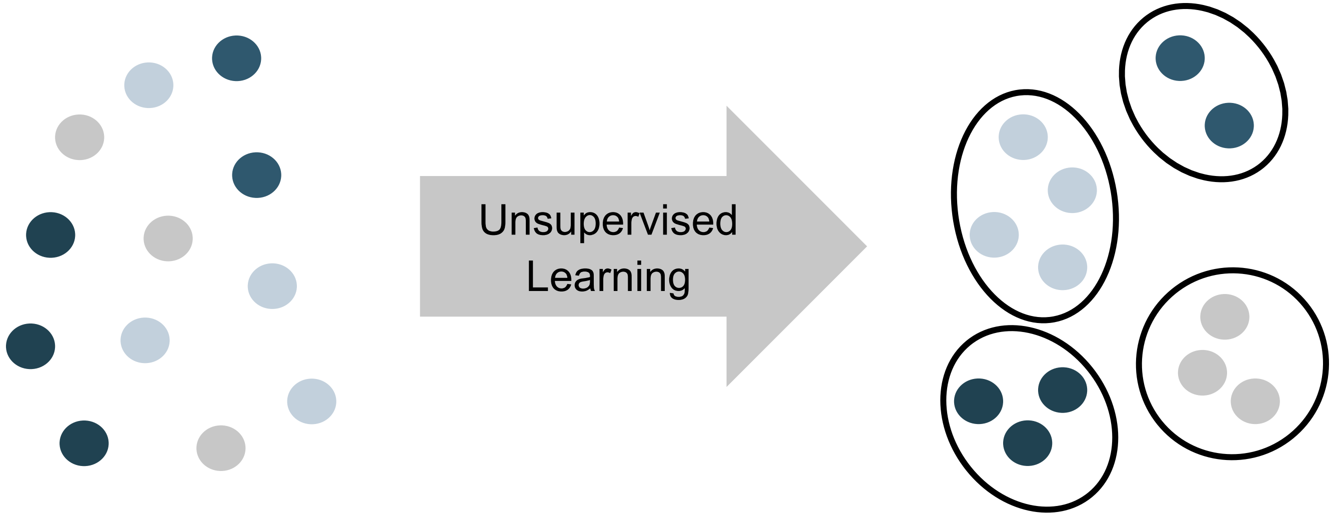
Quelle: https://stock.adobe.com/de/images/robotics-industry-glyph-icon-set-with-robot-and-bot-technology-artificial-intelligence-ai-machine-learning-ml-automated-and-remote-control-smart-chip-android-toy-and-more-tech-symbols/265786856

How can labels required for training on the edge be acquired or substituted?

Motivation

Unsupervised Training of DNNs

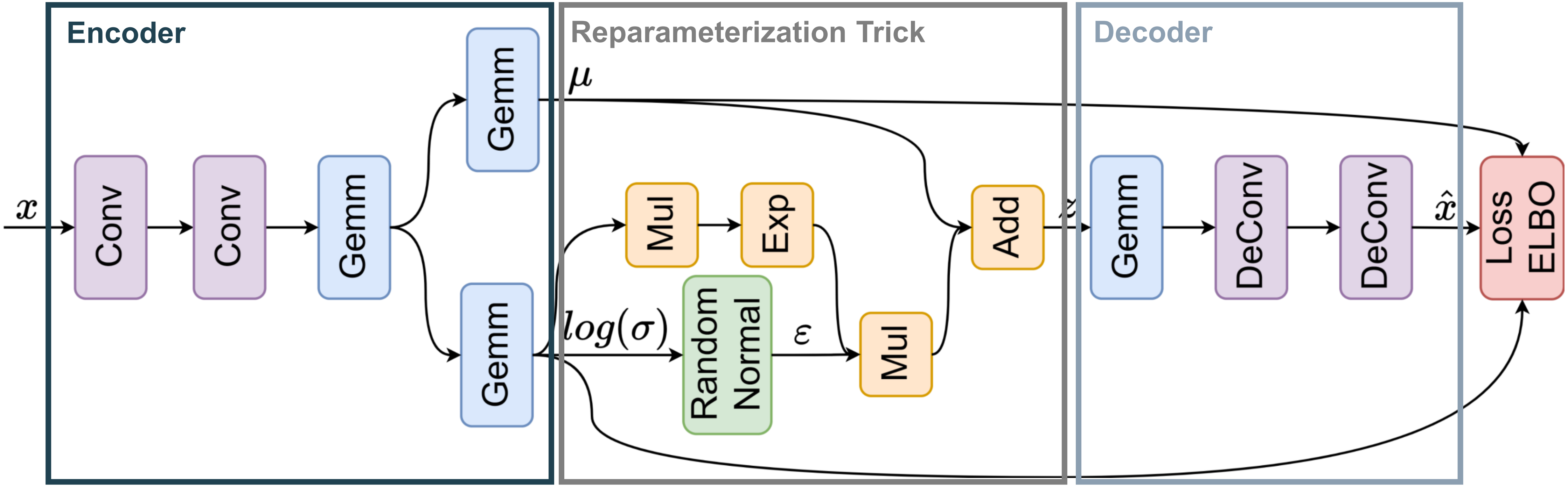
- **Learn** patterns exclusively from **unlabeled data**
- Clustering algorithms, k-means, dimensionality reduction
- When **DNNs** are **trained unsupervised**, they use the **data** itself to **generate supervisory signals.**
  - Often called **self-supervised learning**
  - Common methods: **autoassociative learning**, contrastive learning
  - **Contribution of this work**: On-device training of variational autoencoders on Cortex-M MCUs



Training of Variational Autoencoders on Microcontrollers

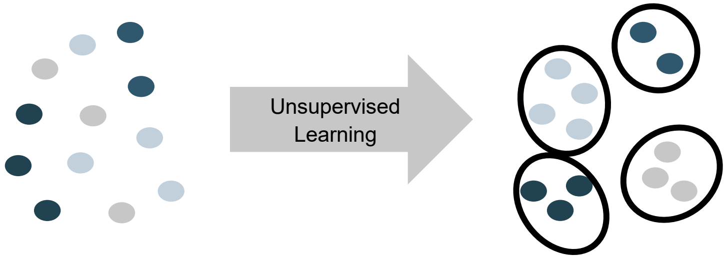
Autoassociative Unsupervised Learning

Variational Autoencoders[1] (VAEs)



Gaussian distributions are not differentiable as discrete functions
→ use reparameterization trick for backpropagation

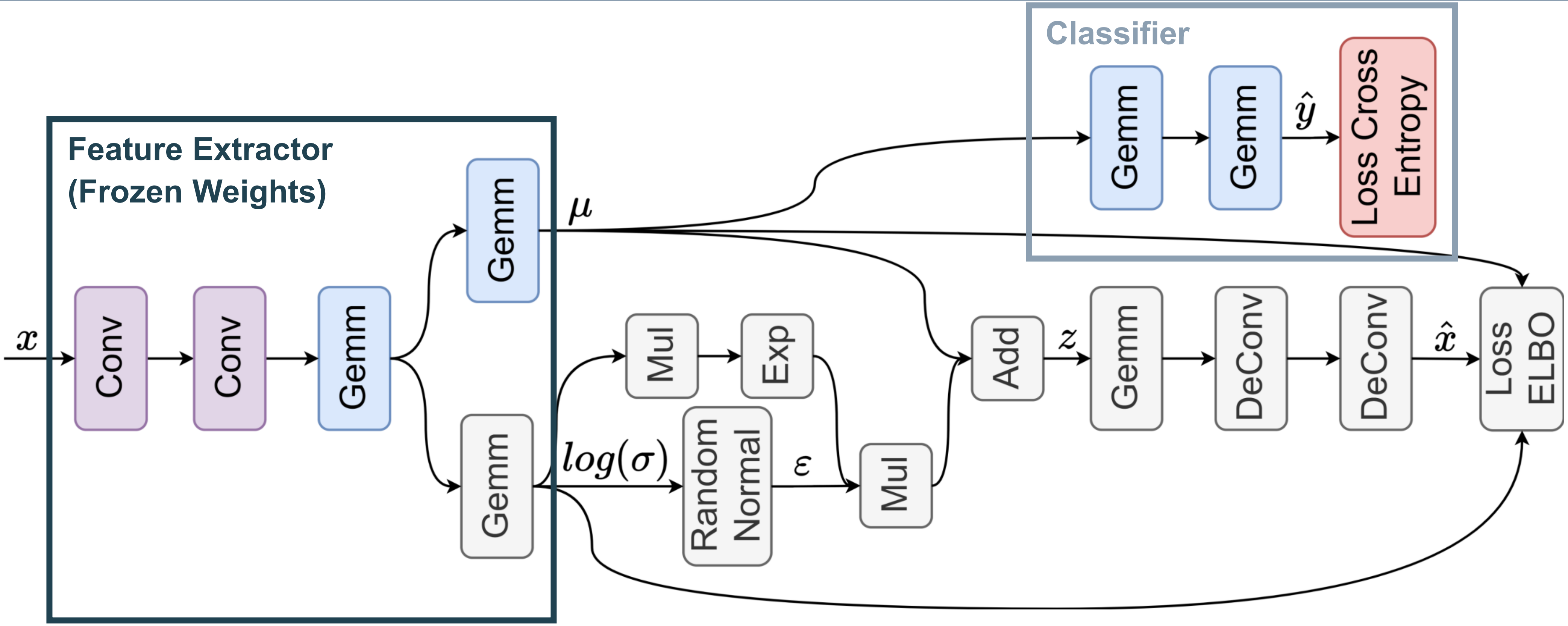
$$z_i = \mu_i + \sigma_i \epsilon_i \text{ with } \epsilon_i \sim \mathcal{N}(0,1)$$



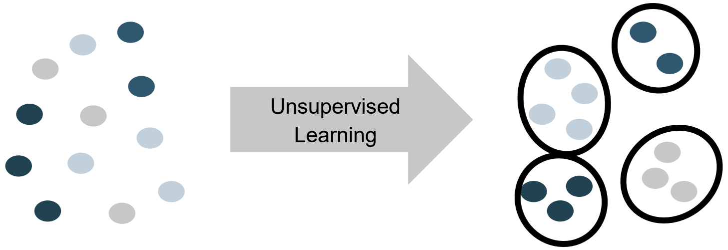
1. Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." 20 Dec. 2013

Autoassociative Unsupervised Learning

Supervised Training using the Encoder of a VAE as a Feature Extractor

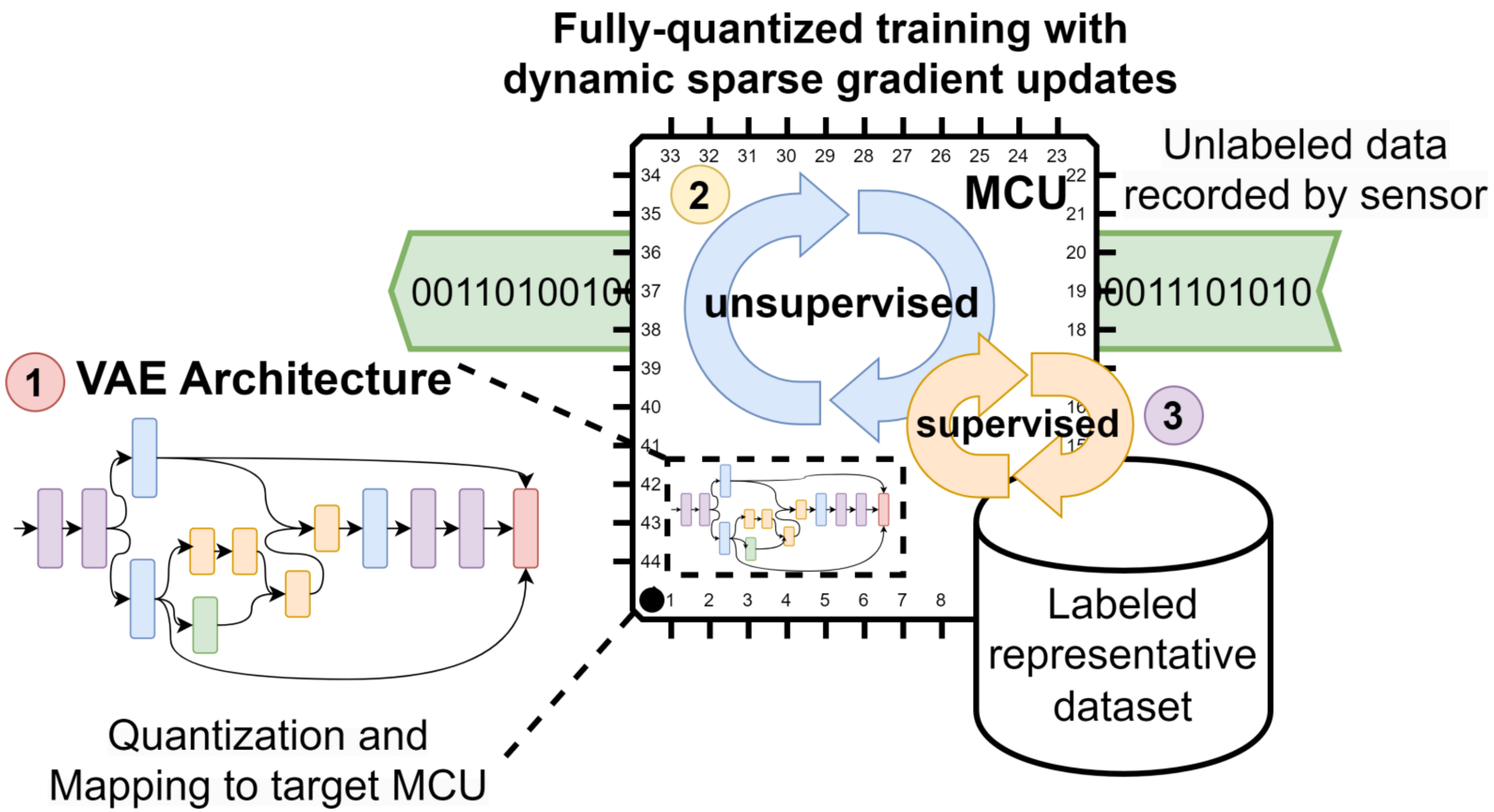


- **Freeze** the **encoder network** of the VAE that was trained unsupervised and use it as a **feature extractor**
- Attach a **classification head** to the **encoder's output** and **train** it using a small **labeled dataset.**



Training of Variational Autoencoders on MCUs

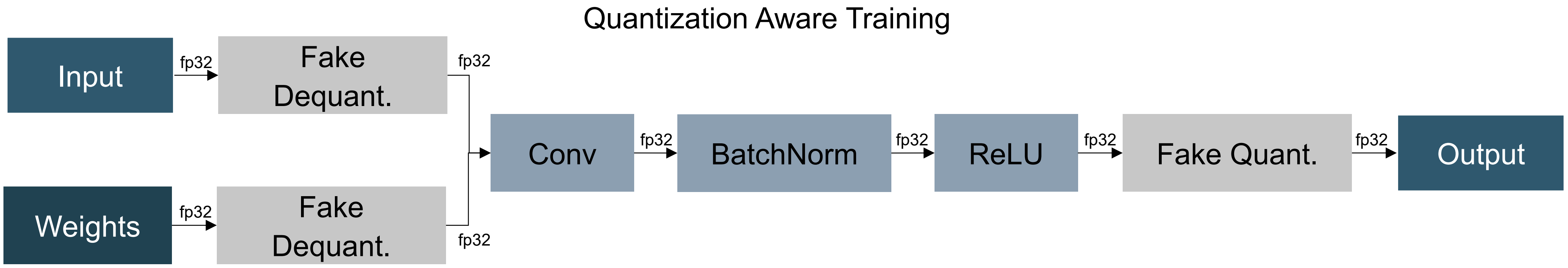
Framework

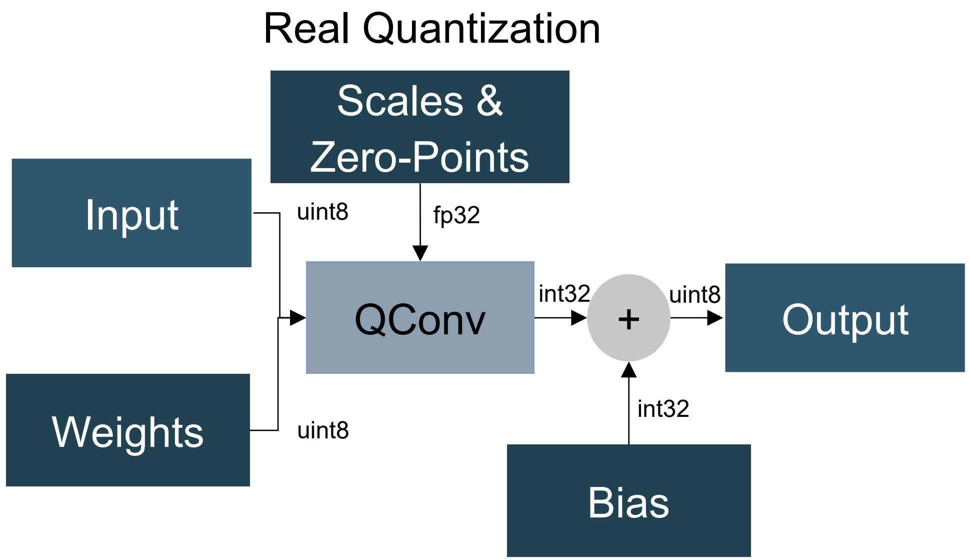


1. **Quantize** and **map** the **untrained VAE** architecture to the target **microcontroller** platform
2. **Use unlabeled data** recorded by a **sensor** to **train** the **quantized VAE** unsupervised **on-device.**
3. **After** the ELBO **loss has converged**, **freeze the encoder network**, and **train the attached classification head supervised** using a small, externally stored, labeled dataset

Training of Variational Autoencoders on MCUs

Quantized Backpropagation[1]





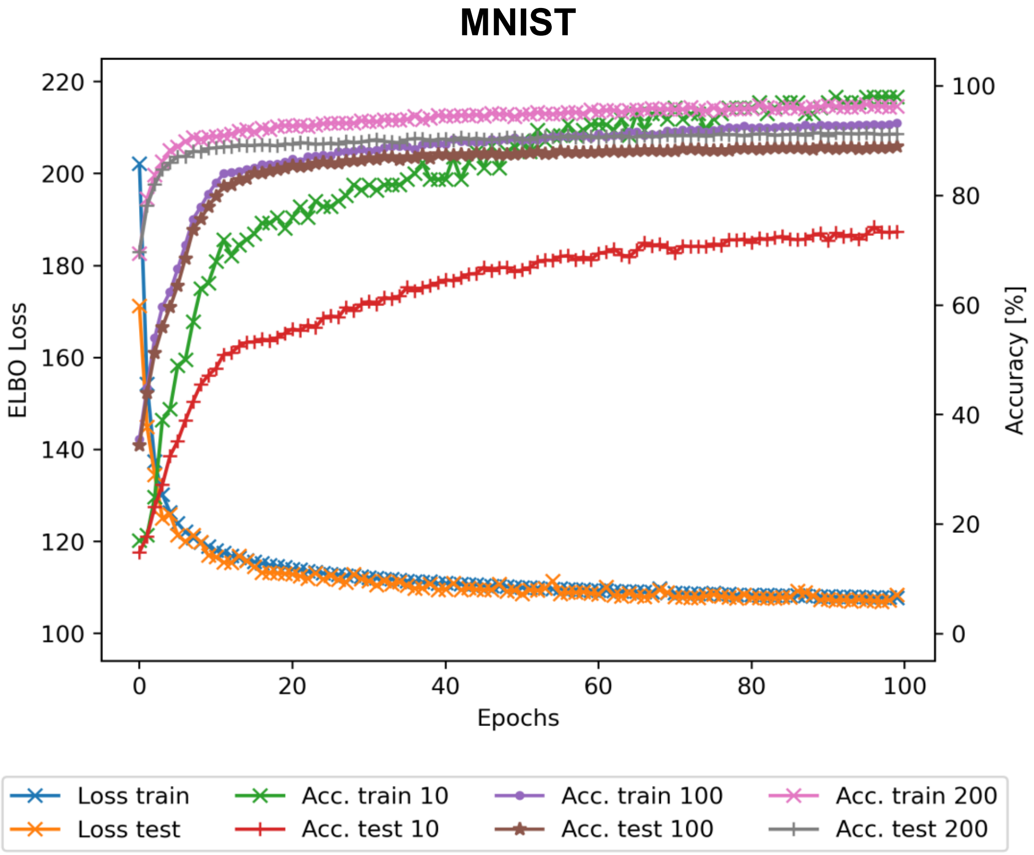
- By using **Quantization Aware Training**, all memory and algorithmic optimizations used to enable deployment on MCUs would be lost
- **Real Quantization** fixes this issue, however …
  - … using quantized tensors for BP and SGD increase chance of training failure
  - … the scaling rates used for quantization must be adapted during training without any floating-point reference weights

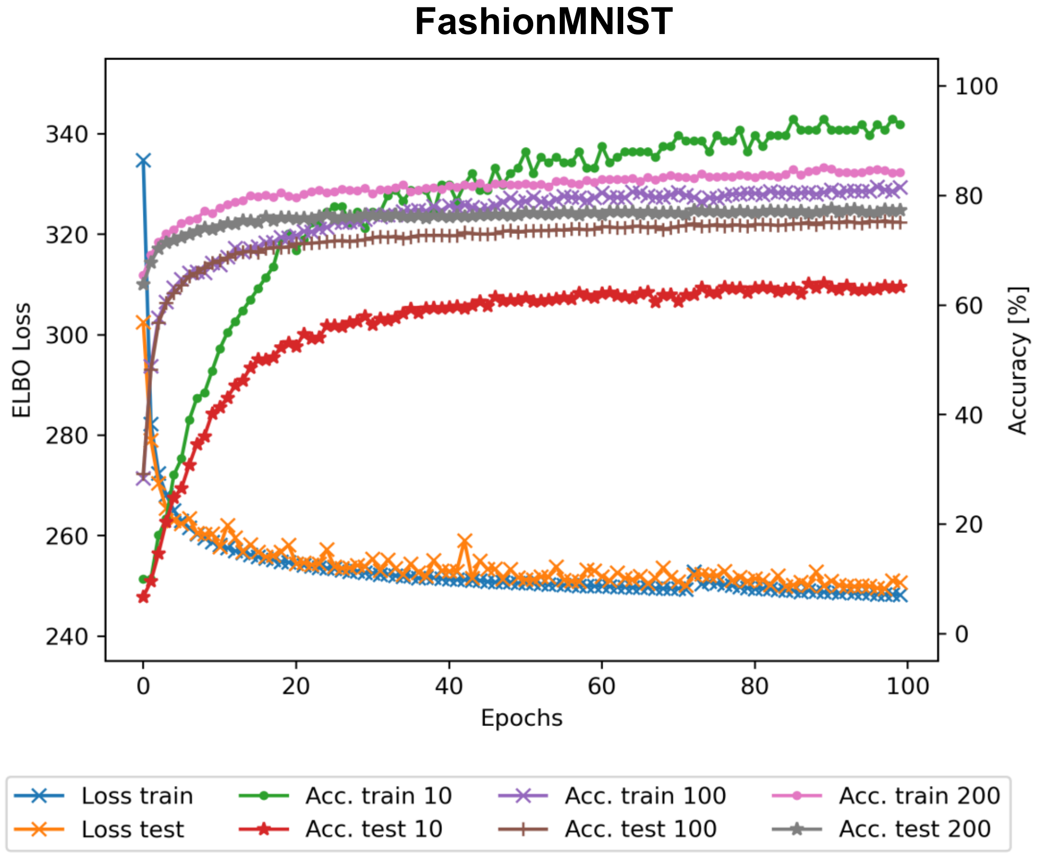
1. M. Deutel, F. Hannig, C. Mutschler, and J. Teich. "On-device Training of Fully Quantized Deep Neural Networks on Cortex-M Microcontrollers". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 44(4), (pp. 1250–1261), 2024.

Evaluation

Evaluation

On-Device Training Results – Floating Point (fp32)

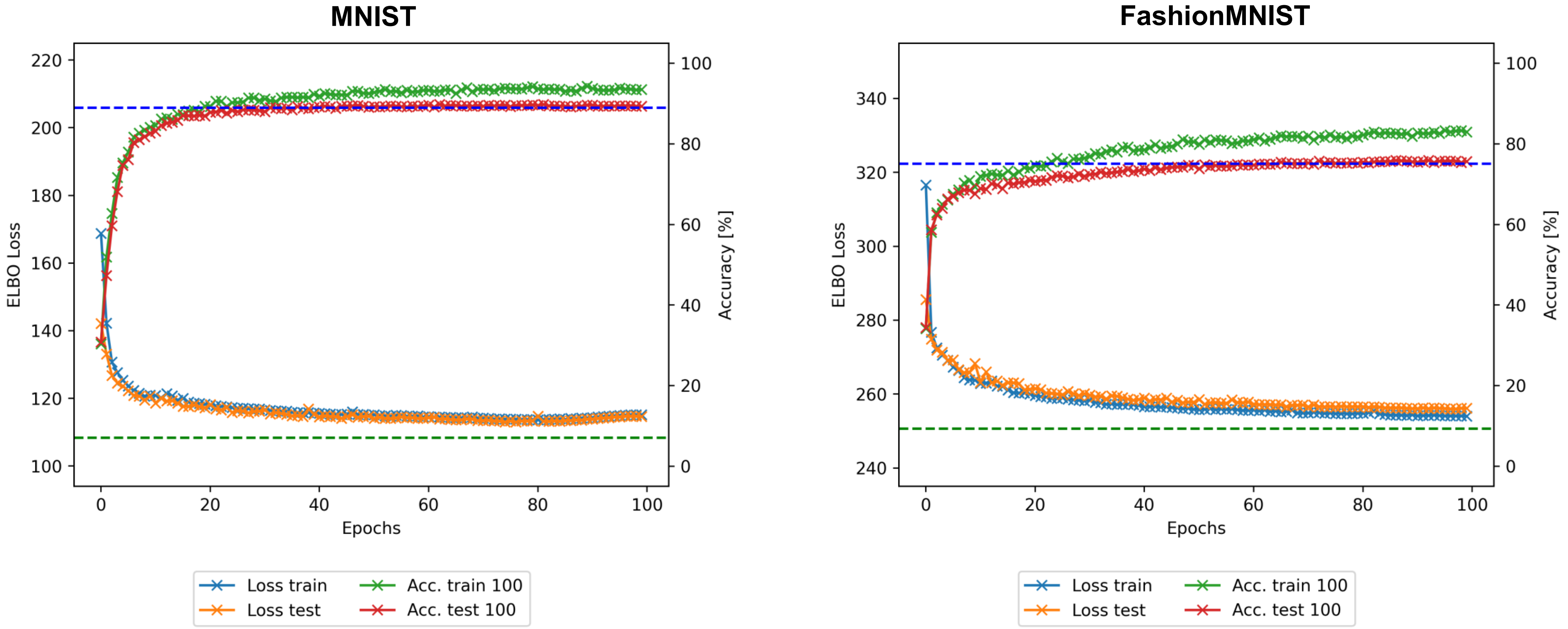




- **ELBO** loss when **training the VAE unsupervised** on the MNIST and FashionMNIST datasets (no labels, 60000 samples)
- **After each epoch**, three different **classification heads were trained** using a **labeled subset of the training datasets** (with 10, 100, and 200 samples per class), and then **evaluated using the labeled test split of the datasets**

Evaluation

On-Device Training Results – Quantized (u8)



- ELBO loss and accuracy when **repeating** the **same experiment** as on the previous slide, but with **quantized VAEs**
- The **dashed lines** show **the best loss and accuracy** achieved by the **floating-point VAEs** from the previous slide

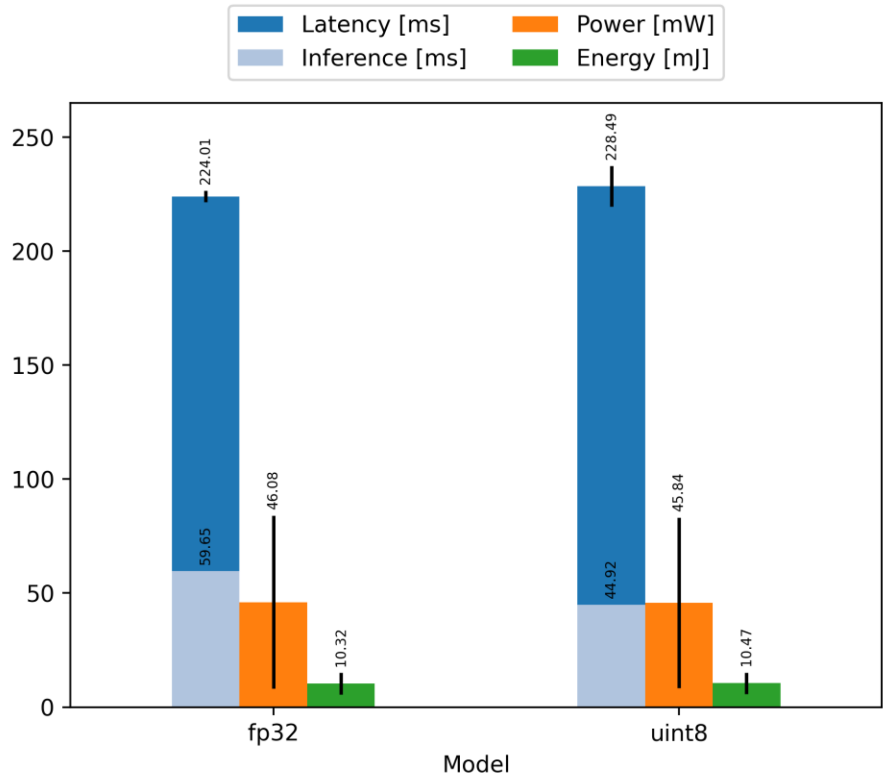
Evaluation

Memory, Latency, and Energy

Memory Requirements (SRAM)

Model	**Activations**	**Weights**
fp32	45.3 kB	484.1 kB
u8	23.0 kB (↓49.2%)	221.0 kB (↓54.3%)

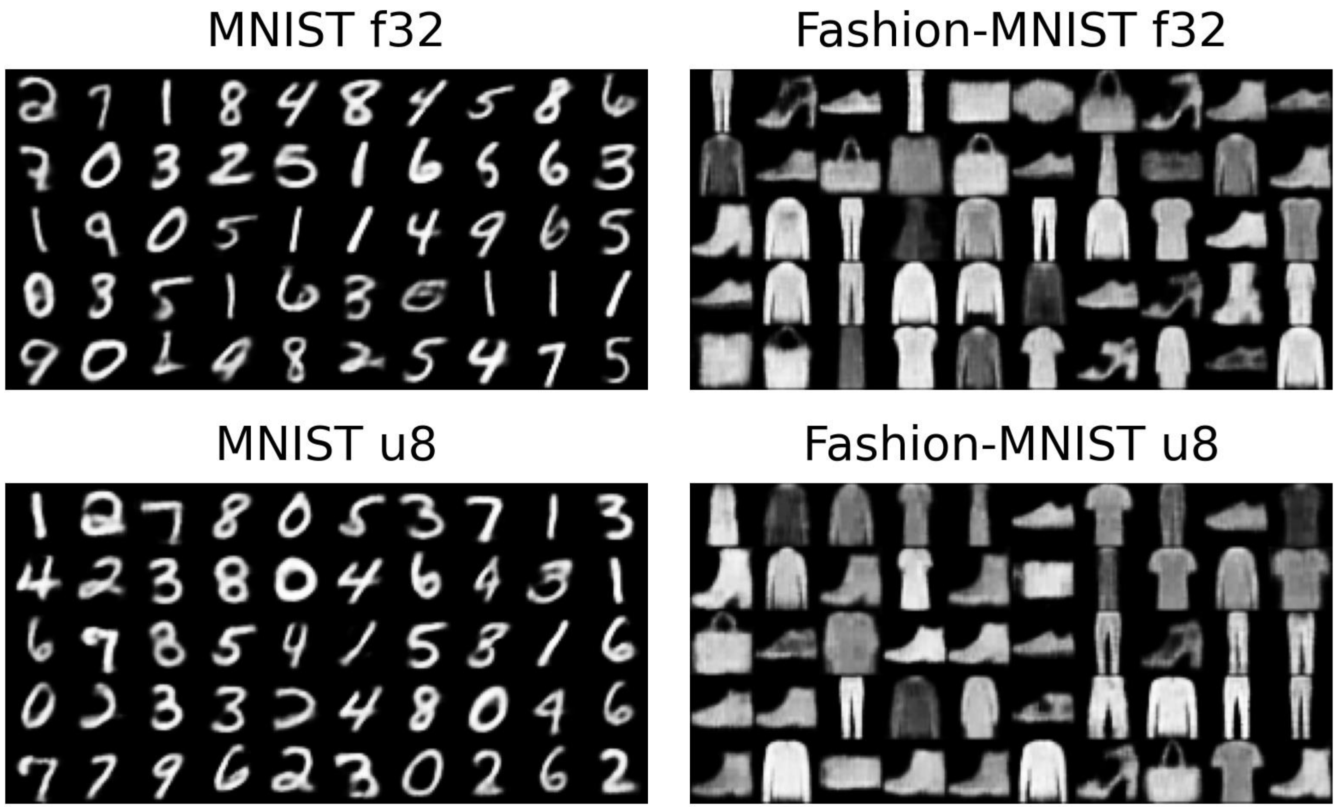
Average **Latency**, **Power**, and **Energy** per training sample

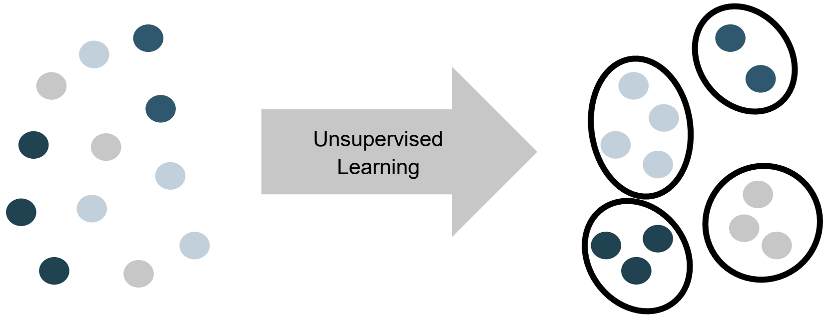


STM32 L4R5ZI Cortex-M4, 120MHz

Evaluation

Qualitative Results





- Random samples in no particular order, generated using the **decoders of VAEs** trained on MNIST and Fashion MNIST.
- For both datasets, the latent spaces of the on-device trained **quantized VAEs** allow the generation of samples with the **same visual quality** as the latent spaces of regularly trained **floating-point VAEs.**

Conclusion

Conclusion

- **Variation Autoencoders (VAEs)** can be used for **unsupervised training** on microcontrollers
  - The encoder network of a VAE can be frozen and used as a feature extractor after unsupervised training
  - Then, a classification head can be trained using a small, representative dataset (~100 labeled samples per class)
- **Fully-quantized training** allows for **memory efficient training** of VAEs on microcontrollers
  - The quantized VAEs achieved the same accuracy as the unquantized VAEs for classifying both MNIST and Fashion-MNIST

Thank you for your attention!

Contact: mark.deutel@fau.de