Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

[a] Rajat Gupta
[a] Yuvaraj Elangovan
[a] Tae Min Hong
[a,b] Stephen Roche
[c] Jim Ignowski
[c] John Moon
[c] Aishwarya Natarajan
[c] Luca Buonanno

[a] University of Pittsburgh
[b] Saint Louis University School of Medicine
[c] Hewlett Packard Enterprise

JM AN TMH

Pheno 2026 Symposium

May 12, 2026

https://indico.global/event/16413/contributions/153957/

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Outline

- Autoencoder — Variational kind, used for compression
- Tabular — Model distillation + tabularization
- Memristive — Analog content-addressable memory

Paper summary

- Real-time analog data on front-end electronics —(i) Tabular ACAM→ Compressed digital data
- Decompressed digital data in DAQ system ←(ii) VAE Decoder— Off detector

Hong
Pittsburgh

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Setup

- e+ on generic 3-layer calorimeter, study for future colliders
- Variables are longitudinal (z) vs. transverse (η-ϕ)

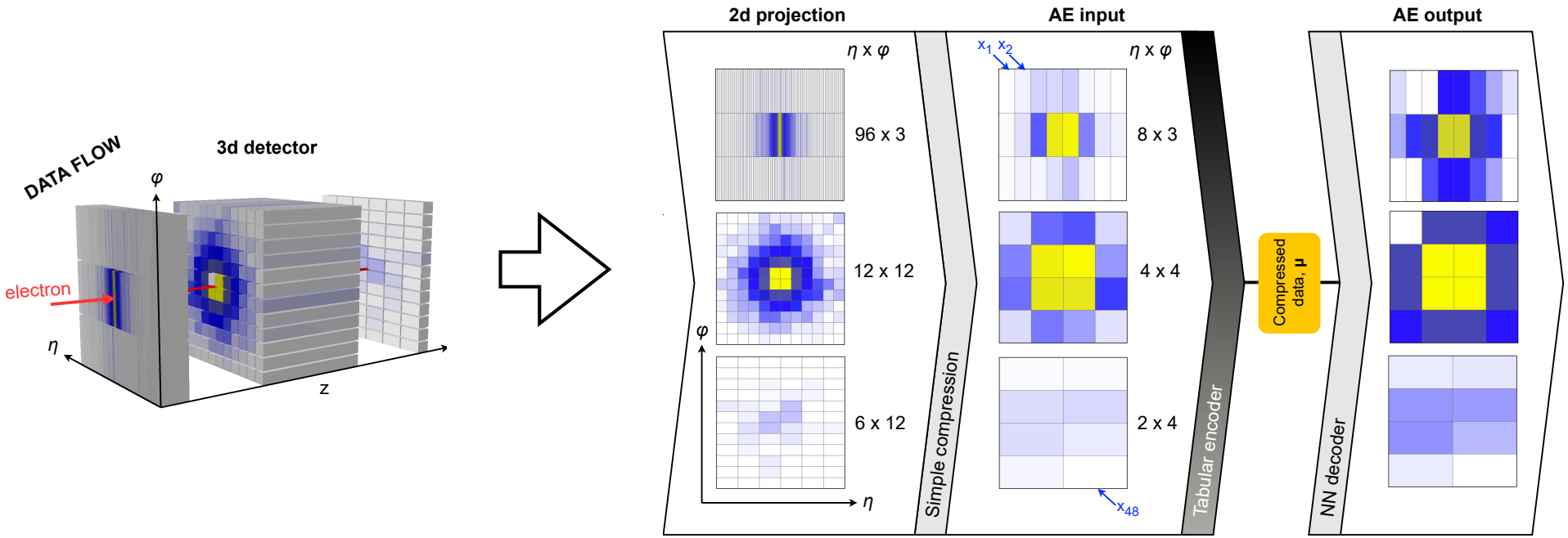


Figure 1: Schematic of an autoencoder for data compression that is distilled into tabular format. Top: The dataflow starts with an incident electron, here with 83 GeV of energy, traversing a three-layer calorimeter. The energy deposits are projected onto the transverse planes, which are then simplified by grouping energies of nearby sensor elements, which serves as input to the tabular AE.

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Variables

Nachman, de Olivera, Paganini
Mendeley Data, V1, 2017

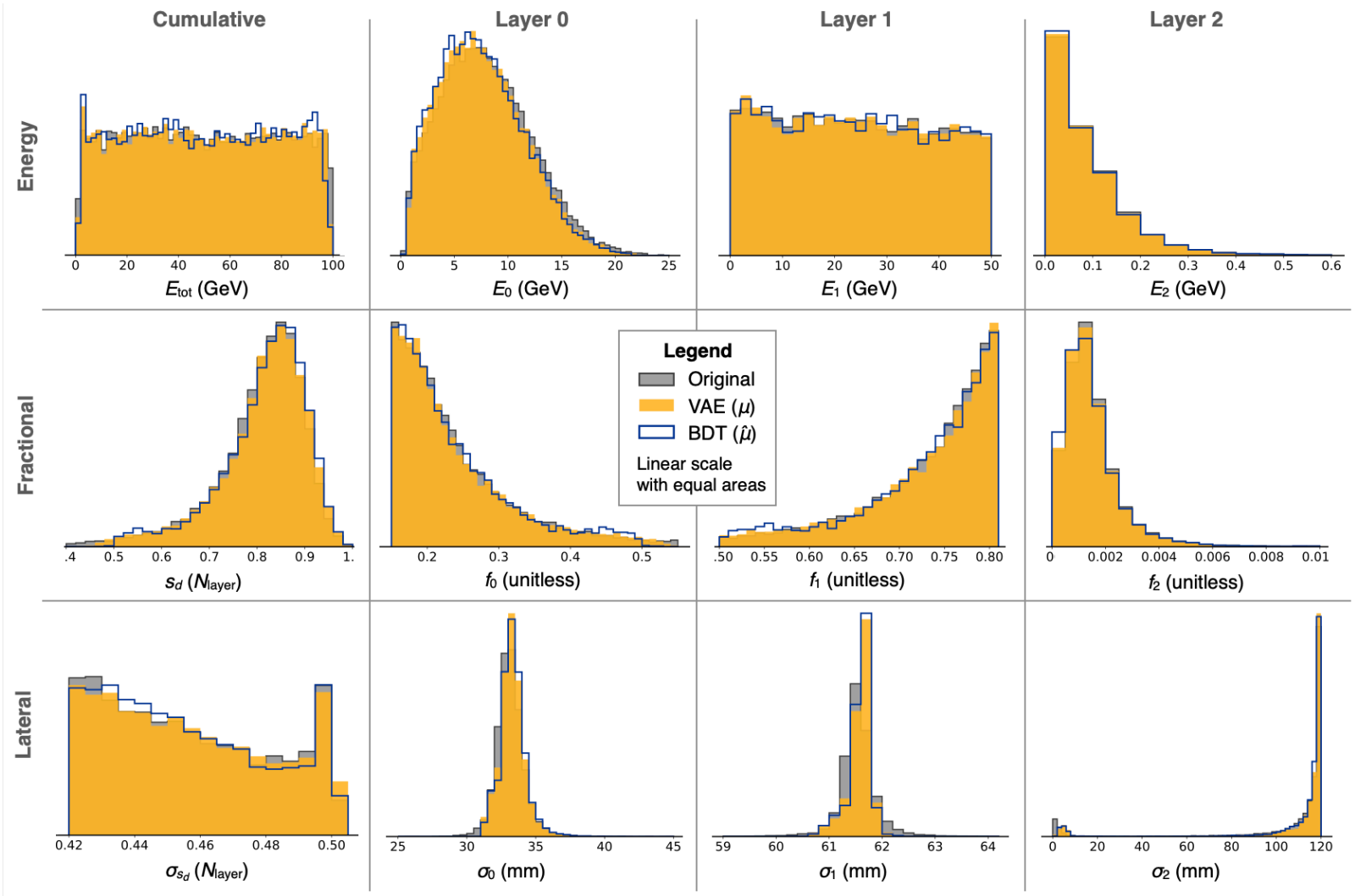


Figure 2: Physics observables before and after compression for the original electromagnetic showers (grey), the VAE encoder ($z = \mu$, gold), the BDT–regressed encoder ($z = \hat{\mu}$, blue line). See text.

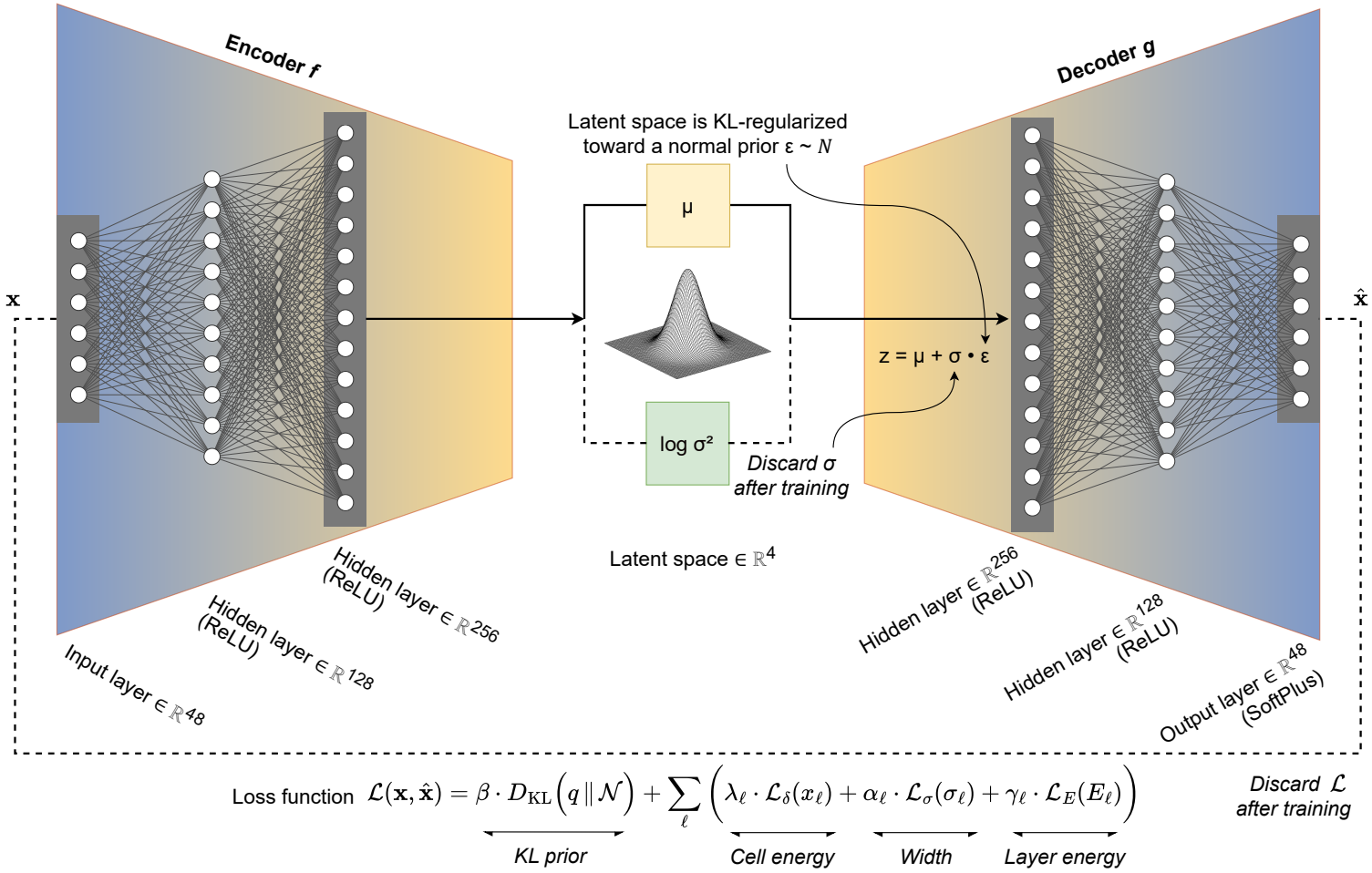
Hong
Pittsburgh

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Autoencoder

- Input-output matching
- Variational with normal constraints

Figure 5: Architecture of the variational autoencoder prior to distillation. The encoder maps the calorimeter input into a latent representation parameterized by the mean and log-variance, $\mu, \log\sigma^2 \in \mathbb{R}^4$. During training, latent vectors are sampled using the reparameterization trick, while inference is performed deterministically with $z = \mu$. The decoder reconstructs the shower representation with a non-negative constraint.

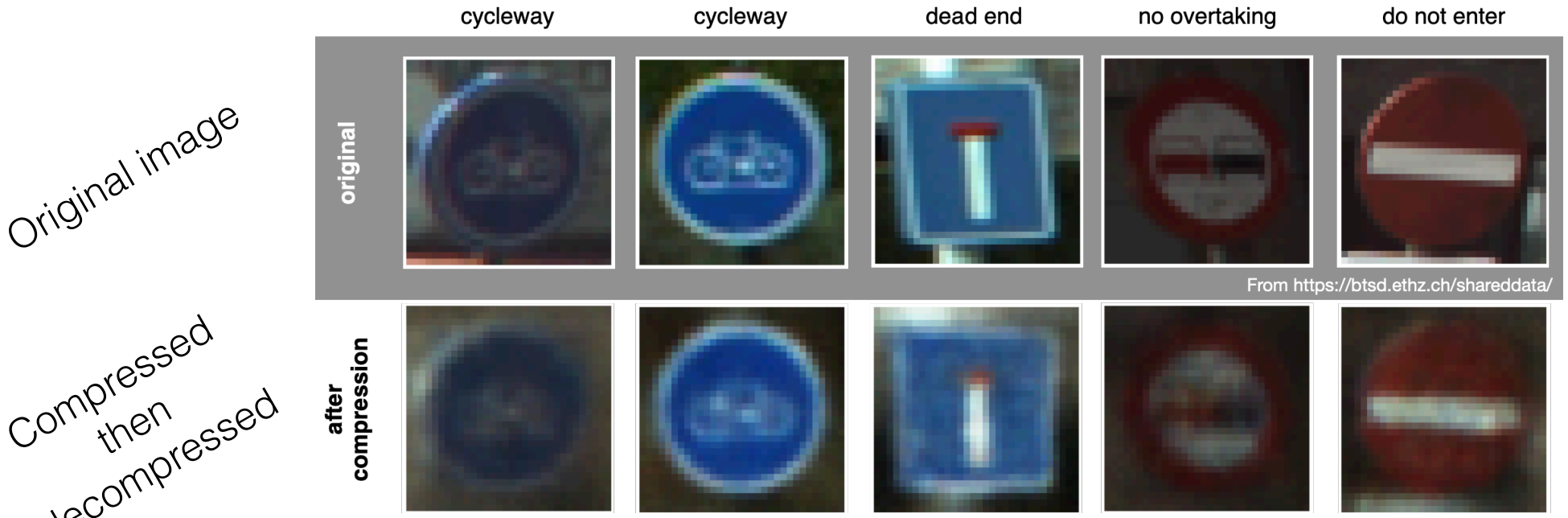


- Physics-based terms

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Variational autoencoder

- 100x compression of RGB images of 32 x 32 pixels

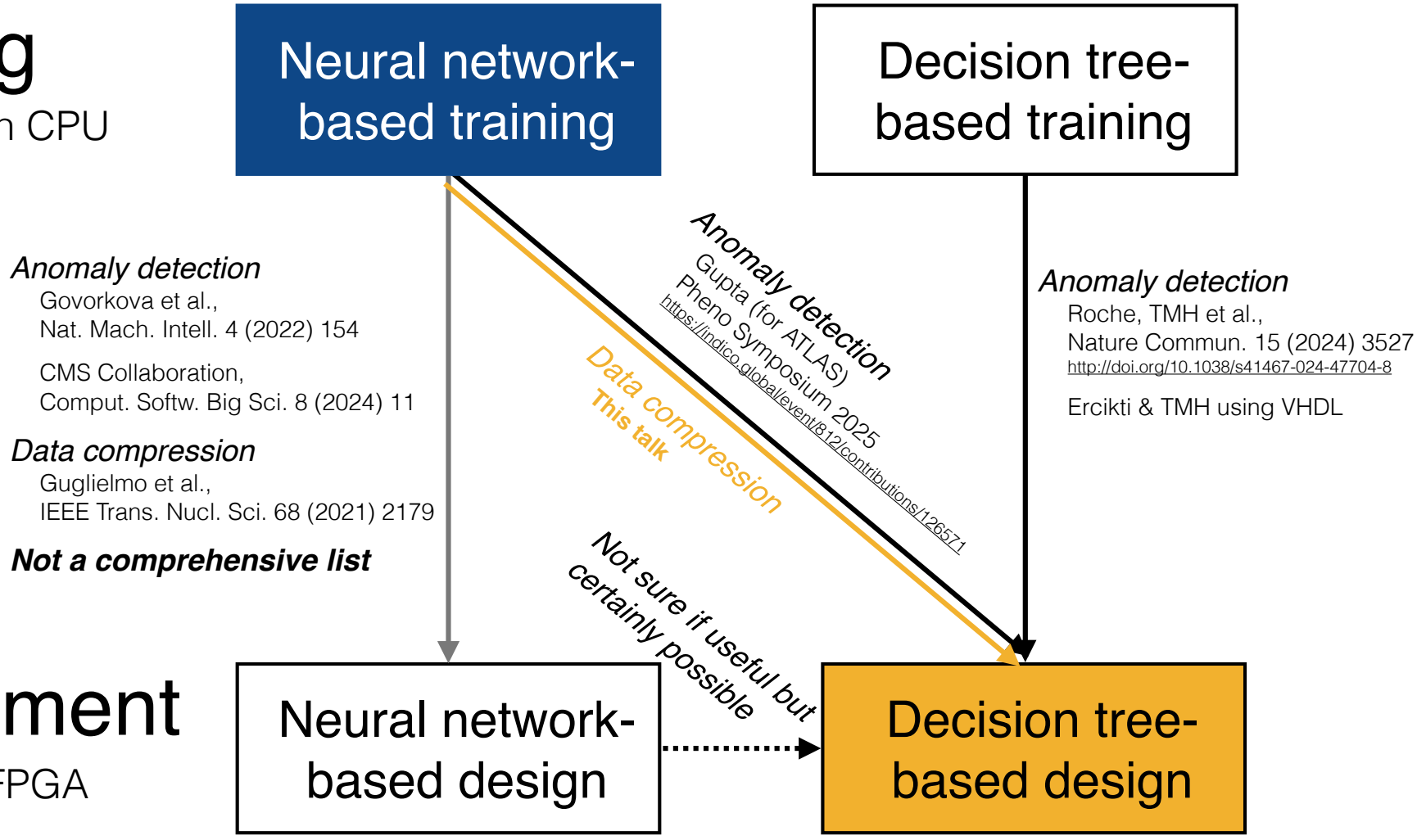


Supplementary Figure 3: Compression and distillation example on color photos of Belgian traffic signs. The images show the same workflow used in this paper. Top row (original images): 32×32 RGB cropped sign images from the BelgiumTS/BTSC dataset [1, 2]. Bottom row (images after compression-decompression): reconstructions obtained by feeding the BDT-predicted latent vectors $\hat{\mu}$ into the VAE decoder. Details: *Model* used is a convolutional encoder with three 4×4 Conv layers, each using stride 2 (with ReLU activations and 3, 32, 64, and 128 feature maps), which maps an input image to a latent representation; this produces the μ and $\log \sigma^2$ parameters of a d=32 dimensional Gaussian latent space. During training a latent code is sampled via the standard reparameterization $z=\mu+\sigma\cdot\epsilon$; a symmetric transposed-convolution decoder (three 4×4 ConvTranspose layers with ReLU, followed by a Sigmoid output layer) reconstructs the image. This corresponds to a compression of a 32×32×3 (3072 pixel intensities) to a 32d latent vector, i.e., 96x reduction.

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Training

done offline on CPU



Deployment

for online on FPGA

Also called “model distillation,” “model compactification” and/or “teacher-student learning” depending on the method

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Model distillation

- Regression

  Regress latent space variables with decision trees

  Carlson, Bayer, TMH & Roche JINST 17 P09039 (2022)

Drama

- Had trouble

  Wide E range so tried log, but lateral quantities were blurred

- Found solution

  Linear E, but scale-up the loss function weight for layer 2

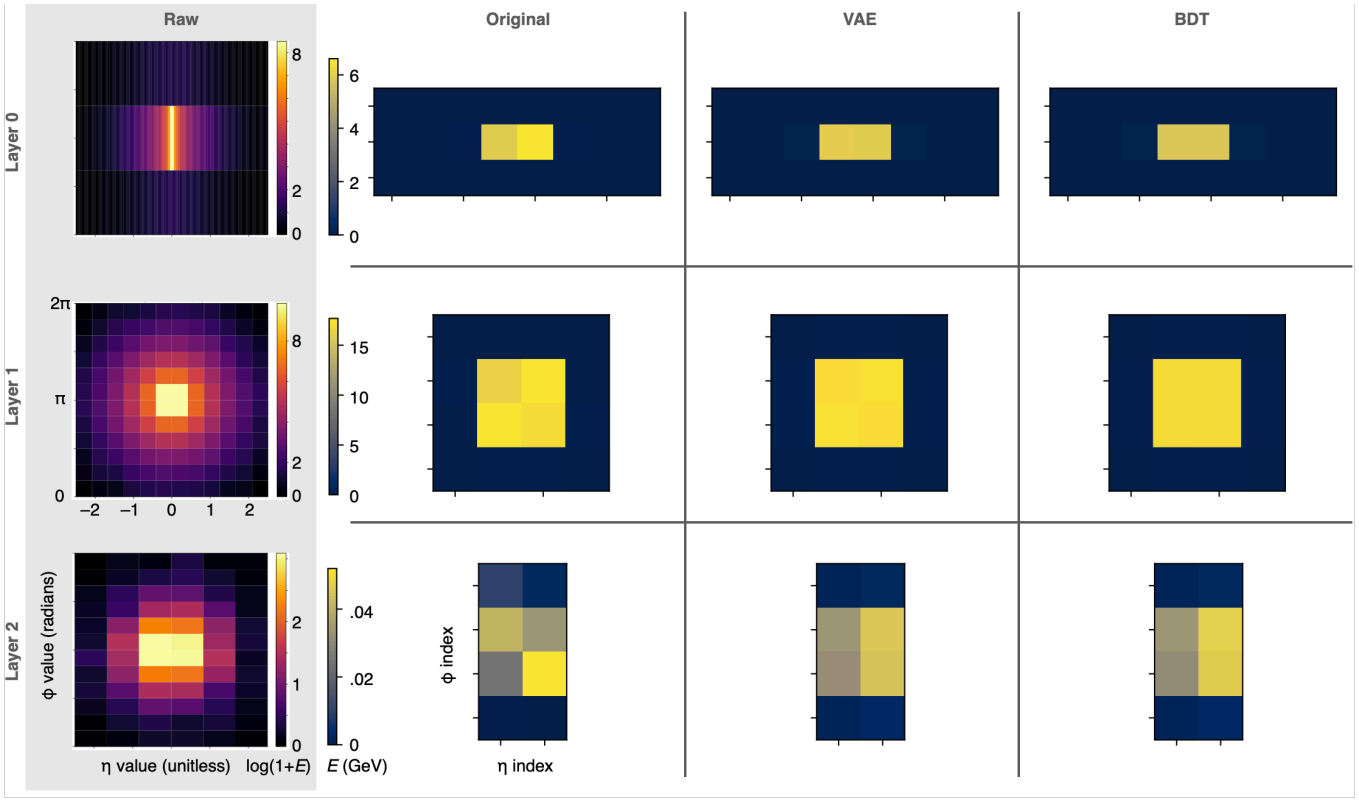


Figure 4: Visualization of energy deposition in η-ϕ for the ECAL layers in rows. Leftmost column in gray: average energy deposition from 100 simulated electron shower events with the color scale corresponding to $\log(1+E)$, where E is in GeV. Right three columns show one representative shower after rebinning with color scale corresponding to linear E: *Original* designates the input value to the VAE, *VAE* is the reconstruction using the encoder mean μ, and *BDT* is the reconstruction using the regressed mean $\hat{\mu}$.

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Correlation

- Near-unity

Residual

- Small unbiased

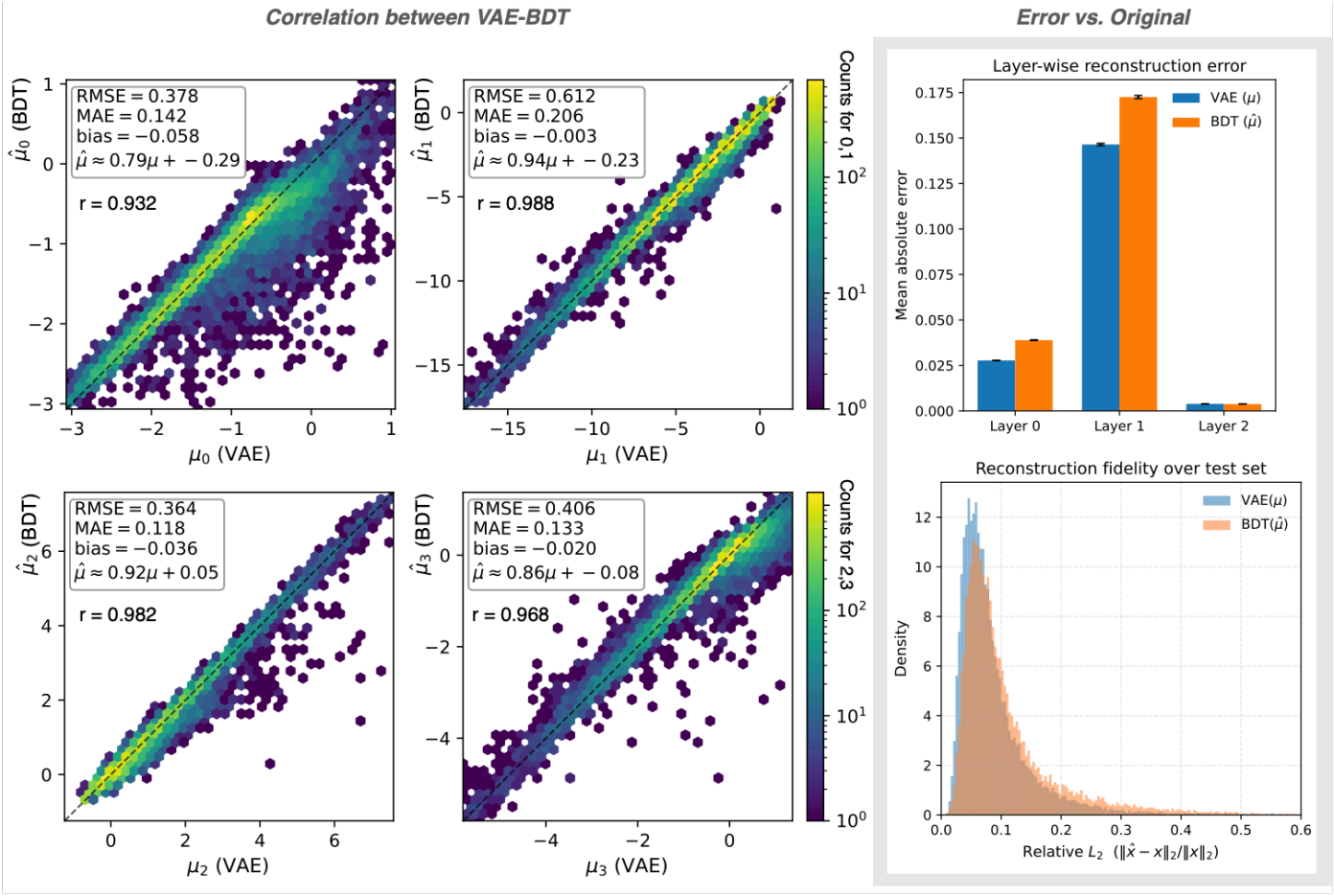


Figure 7: Model distillation results. Left group of four plots compares one latent component from the VAE encoder (μ_i) with the corresponding BDT-predicted value ($\hat{\mu}_i$) for incident electrons in the test sample. BDT exhibits strong linearity ($r = 0.93$ to 0.99) and negligible bias, reproducing the encoder outputs with sub-percent mean absolute deviation. Right group of two plots shows quantitative reconstruction fidelity over the test set. Top-right: Mean absolute error, between VAE and BDT, per ECAL layer; error bars indicate the standard error of the mean across events. Bottom-right: Distribution of the relative L_2 error over all cells in all layers.

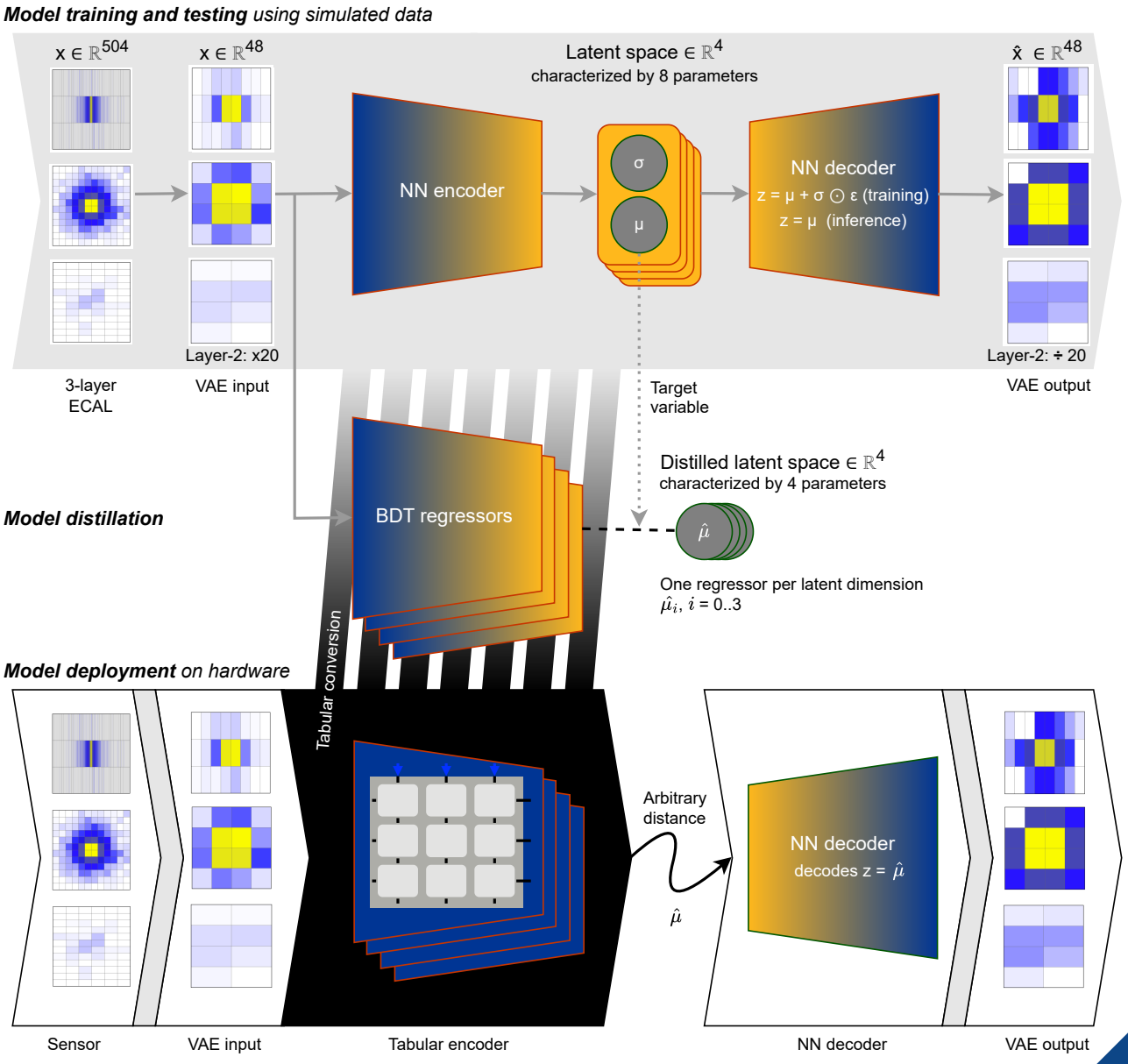
Hong
Pittsburgh

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Tabularize

- Root-to-node parallelization
- Convert model to table of inequalities

Serhiayenka, Roche, Carlson & TMH,
NIM-A 1072 (2025) 170209



Hong
Pittsburgh

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Memristive analog CAM

- Programmable resistor

  Theorized by Chua & Kang, *Proc. IEEE* **64**, 209–223 (1976)
  Realized by Strukov, Snider, Stewart & Williams, *Nature* **453**, 80–83 (2008)

- Analog CAM

  Pedretti et al. Nature Commun. 12 no. 1, 5806 (2021)

  Content-addressable memory (CAM)
  Current flows if matched

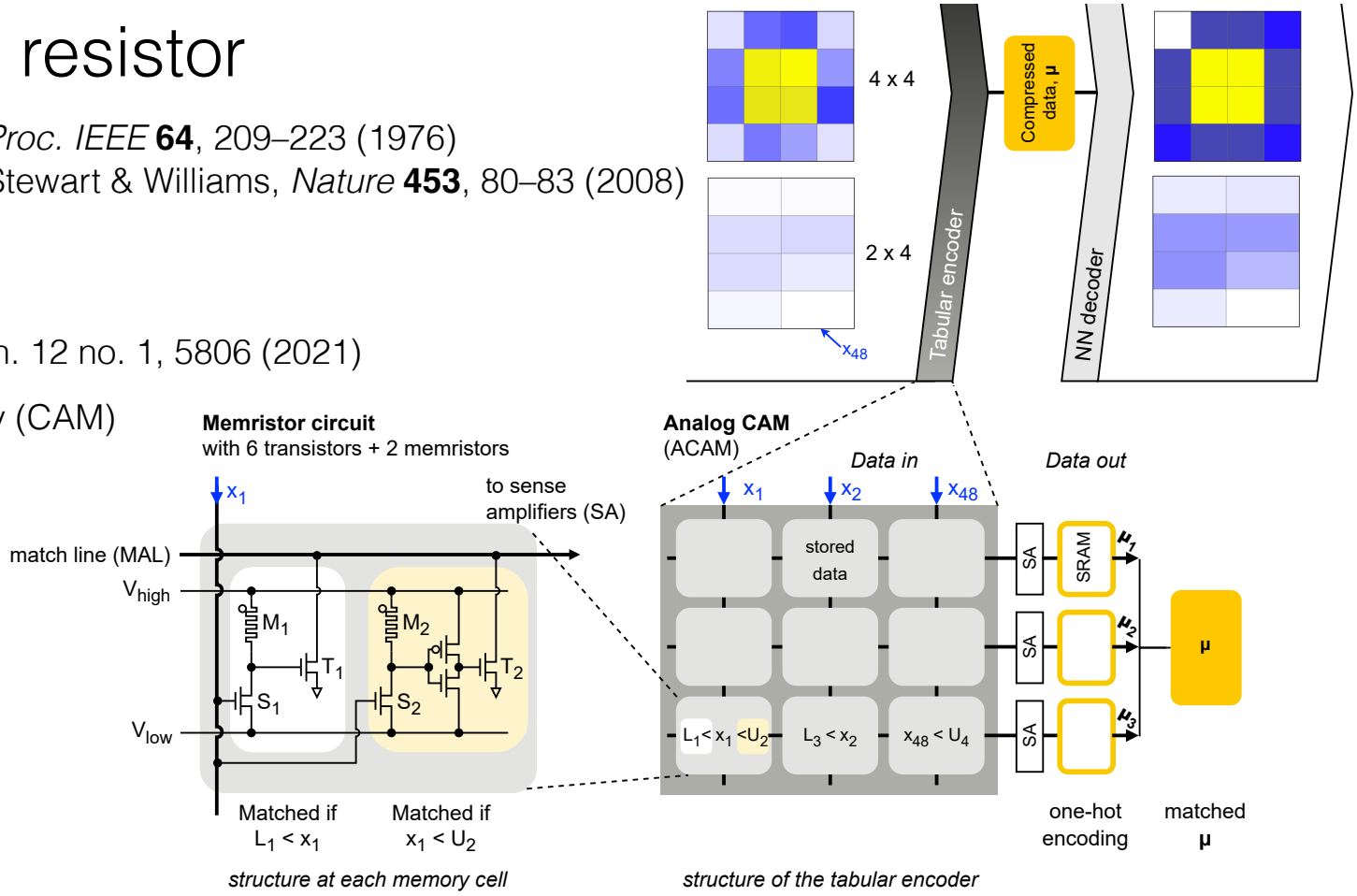


Figure 1: Bottom: Close-up of a memristor-based analog content-addressable memory shows the crossbar structure of the input data (x) crossing the match line to read-out into static RAM. Further close-up at each crossbar shows the memristor circuit architecture to produce a binary output. The latent data is transmitted and decompressed.

Hong
Pittsburgh

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Architecture

- 4-bit analog
- Nested for > 4

Fabrication

- Custom by HPE

Match

- Ideal vs. physical

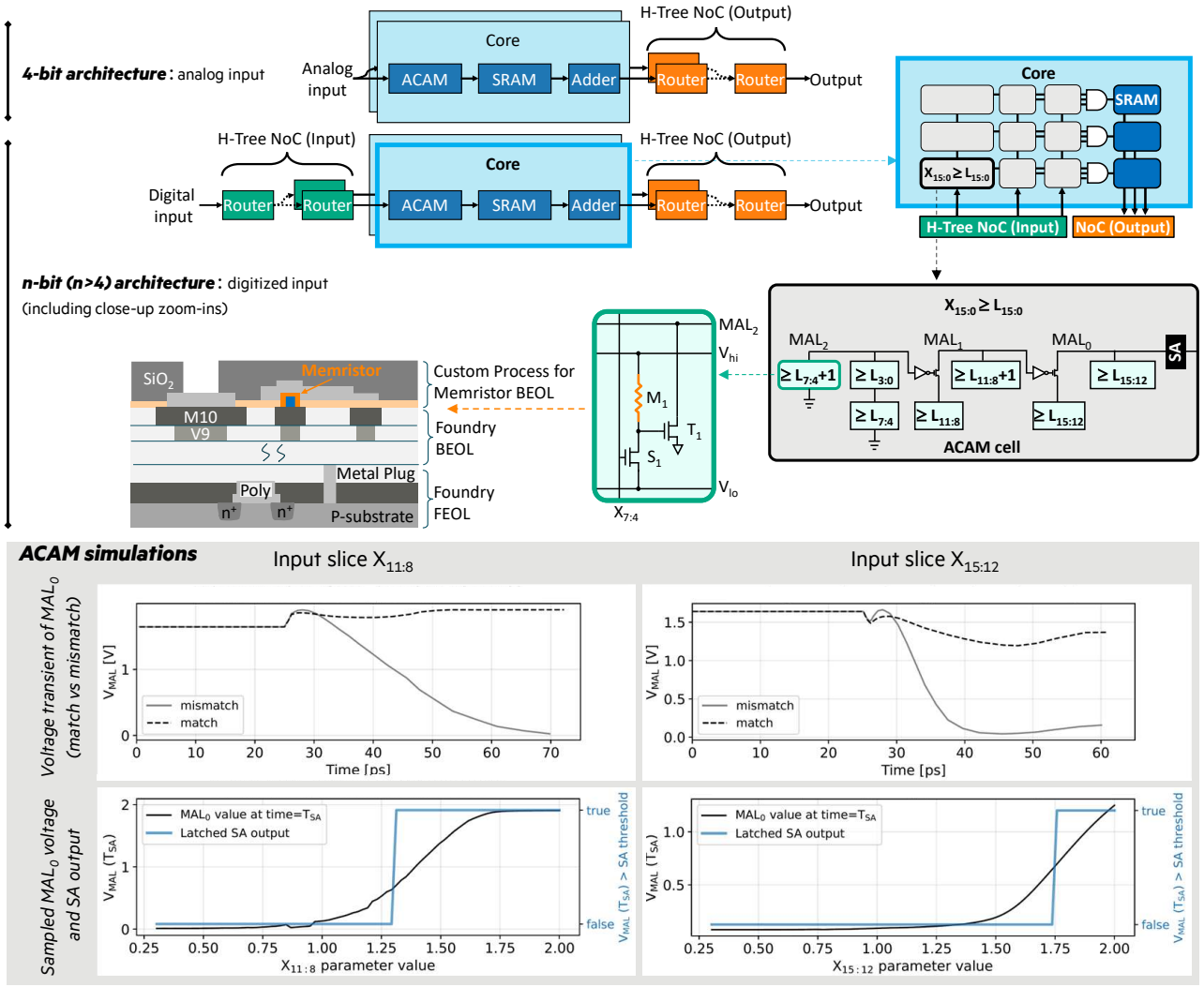


Figure 8: ACAM details and results. *4-bit architecture*: Simulation schematic for 4-bit inputs from direct analog sensor data. *n-bit architecture*: Simulation schematic for quantized n-bit inputs width for higher precisions (here 16-bit) and successive close-up zoom-ins. Inside the ACAM cell, the MAL_0 stays charged if, and only if, $X_{15:0} \geq L_{15:0}$; if all the inequalities in the same row of ACAM cells are satisfied, the data in the associated SRAM—corresponding to the selected leaf of the decision tree—is provided as result of the inference. We additionally show the schematic design of an ACAM based on ReRAM [27], and the typical material stack cross-section for a custom BEOL manufacturing process of ReRAM [45]. *ACAM simulations*: The top row of plots show the time-domain MAL waveforms for match and mismatch, with left and right panels corresponding to input slices $X_{11:8}$ and $X_{15:12}$; the mismatch induced by the $X_{15:12}$ swept values reflects on the MAL state with a smaller time delay due to the ACAM self-loading of the MAL_i and of the pull-down devices, with an $X_{11:8}$-induced mismatch propagating in ~20 ps. The bottom row of plots show the MAL_0 at the sense-amplifier sampling instant (black box) and the resulting SA decision relative to a fixed threshold; left sweeps $X_{11:8}$ (other input slices are fixed) and right sweeps $X_{15:12}$, with the threshold crossing set by the programmed memristor conductances.

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Comparisons

- **Speed**
  10 ns latency
  350 MHz thruput
- **Efficiency**
  4.1 nJ per compression
  2.1 mm² area of silicon
- **Vs. FPGA**
  Better if < 16 bits

Comments

- **IMC**

  In-memory computing stores coefficients on-chip to reduce data movement on/off chip

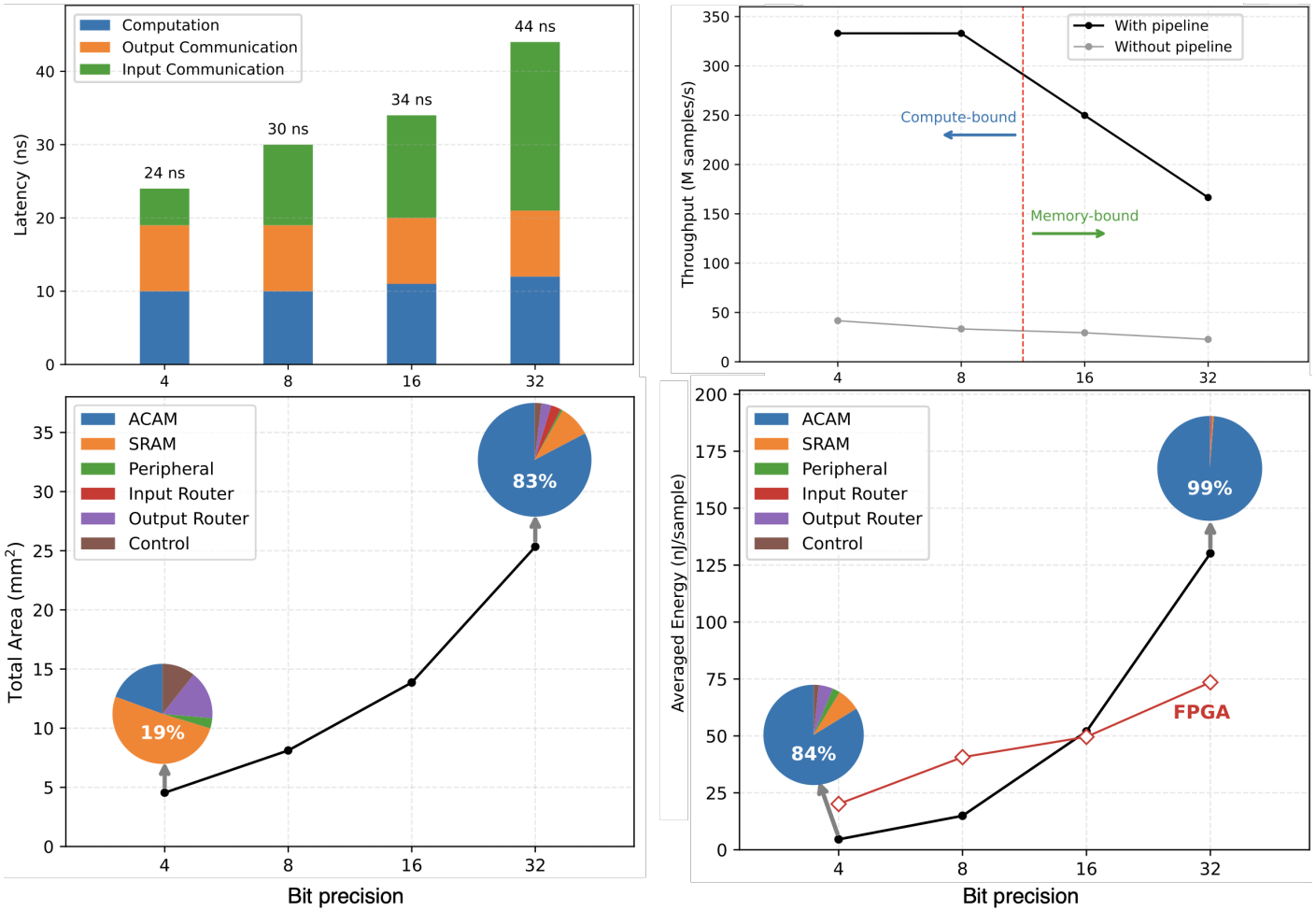
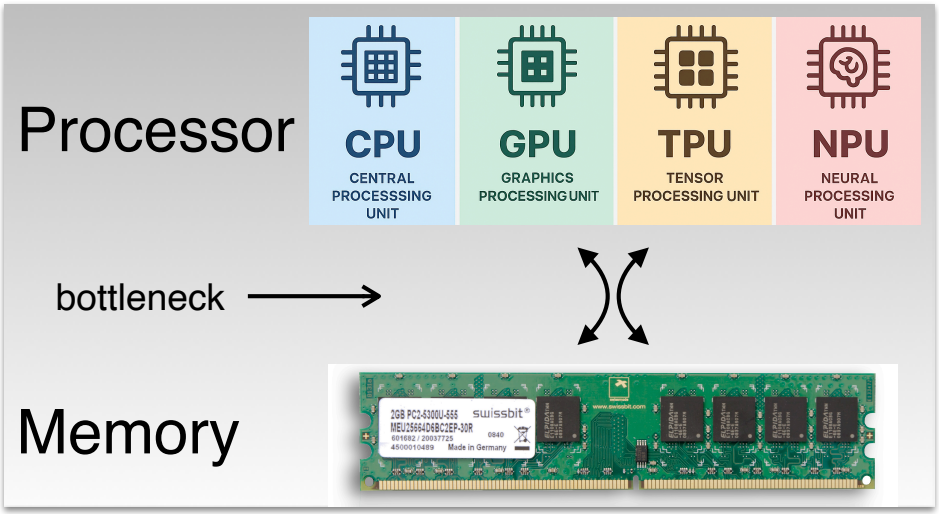


Figure 3: Latency (top-left), throughput (top-right), total area (bottom-left) and energy per compression (bottom-right) of the encoding part simulated on the ACAM-based architecture. *Computation* latency at about 10 ns is due to the ACAM alone. See *Discussion* for the comparison to FPGA in the bottom-right plot.

Memristive tabular variational autoencoder for compression of analog data in HEP [2602.15990]

Standard paradigm

- Von Neumann architecture →
  Classic processor-memory computing paradigm
  Spend a lot of time & energy to move data



https://resources.l-p.com/knowledge-center/cpu-vs-gpu-vs-tpu-vs-npu-architecture-comparison-explained

Our design summary

- In-memory computing (IMC)
  Save energy & time not moving coefficients around
  All the coefficients are baked in-memory!
- Analog, not digital
  Memristors - see *Nature* **453**, 80–83 (2008)
  Optimal for 4-bits, can be nested for higher bits
- Autoencoder
  Train NN-based variational autoencoder
  Distill by decision trees
  Tabularize by root-to-node
  Execute on cross-bar arrays