Neuromile: A Continual Meta-Learning-Based AI Deployment Framework for Energy-Aware Personalized Inference at the Edge

Doris Klein[1]*, Stefan Dech[2], Bradley Raddwine[3], Ernst Uken[4]
[1-4]Faculty of Engineering, University of Cape Town (UCT), South Africa

Keywords:
Continual Learning,
Meta-Learning,
Edge AI,
Energy Efficiency,
Personalized Inference,
Federated Learning,
Lightweight Models,
On-Device Learning,
Model Adaptation,
IoT

Author Email:
Klein.doris@engfacuct.ac.za,
dech.stefani@engfacuct.ac.za,
radd.bradley@engfacuct.ac.za,
uken.ern@engfacuct.ac.za

DOI: 10.31838/IJCCTS.13.01.03

Received : 09.12.24
Revised : 11.01.25
Accepted : 18.02.25

Abstract

NeuroMile is an alternatives-based, new AI architecture that combines continual learning, meta-learning, and dynamic energy optimization to perform real-time and accurate inference on markets and end gadgets. Engineered to address the resource limitation of embedded and wearable systems, the task-aware memory encoder and the adaptive-modulation of inference depth and quantization level, NeuroMile has been developed to support a modularized architecture. Such adaptations have dynamic contextual feedback, such as battery level, activity and complexity of task.

Relative evaluations in three edge-related benchmarks, PAMAP2 (human activity recognition), EdgeSpeech (voice command recognition), and CIFAR-100 (few-shot image classification) show that NeuroMile can reach 88.9% at 1.1W power consumption as opposed to the full-precision baseline of 89.6% accuracy at 2.8W power consumption. This shows a decrease of 60 percent of energy consumed with less than 1 percent loss of accuracy. Besides, NeuroMile can train much faster in terms of task-specific fine-tuning, with only 7.8s required compared to conventional meta-learning baselines, including MAML (10.3s) and FedAvg (18.1s).

These findings put NeuroMile as a feasible and smart edge inference architecture that trades-off among accuracy, energy-efficiency, and flexibility. It is applicable to mobile robotics, wearable health-monitors, as well as real-time IoT installations. The future work will consist of a federated learning to enable edge adaptation to be secretive and reinforcement learning-based self-optimizing edge control policies to further enlarge the sustainability and personalization aspect in this Computational model.

How to cite this article: Klein D, Dech S, Raddwine B, Uken E (2025). Neuromile: A Continual Meta-Learning-Based AI Deployment Framework for nergy-Aware Personalized Inference at the Edge. International Journal of communication and computer Technologies, Vol. 13, No. 1, 2025, 21-37.

Introduction

Background and Motivation

The explosive increase in the adoption of edge computing has transformed the implementation of artificial intelligence (AI) models to latency-sensitive and resource-bound settings. Such applications as real-time health monitoring, autonomous flying vehicles, smart surveillance, and home automation are becoming application areas where edge-based solutions are more and more predominant to meet frequent and localized inference. But the current performance of the conventional deep learning (DL) models in these environments is low since they have a high computational overhead, they do not change their learning behavior, and this makes them to rely on cloud-based retraining in an attempt to adapt to the dynamic conditions.

Edge AI has to work with very limited batteries, tight memory, sporadic connections, as well as a diverse user base that may have constantly changing data distributions. Continuous learning and

personalization is no trivial task in the situations like these. In addition, edge devices simply have to work on their own, adjusting to streaming information with low-power usage and minimal latency, which is beyond the scope of conventional AI paradigms.

This was changed in the recent progress of continual learning (CL) and meta-learning which has the ability to modify the models with time and various tasks. However, when combined in the edge ecosystems, these approaches create new grounds of intricacies. In addition, the use of the computationally-costly replay buffers or parameter regularization imposed on most CL techniques, and the rigid energy budgets presupposed by meta-learning approaches fail to consider the stochastic resource availability of edge devices. Therefore, such a more synergistic approach is needed the one that will be able to connect the abilitotion to personalize to the resilience to task of continual learning with strong energy constraints.

Literature Review

There are many methods which have been analyzed to expand AI to the edge. Federated Learning (FL)[1, 2] was proposed as a privacy preserving paradigm that enables training to be done in a decentralised manner between devices. Nevertheless, FL is also marred with high communication expense and does not provide coarse-grained personalization particularly in non-IID (non-independence and identically distributed) settings. The ongoing efforts to address the problem of catastrophic forgetting have resulted in several learning algorithms, Elastic Weight Consolidation (EWC),[3] Memory Aware Synapses (MAS)[4] and Gradient Episodic Memory (GEM)[5] among them, but learning with these methods can be resource-intensive and thus cannot be used in energy-constrained edge devices.

It can be adapted quickly to new tasks with little gradient update, which is a desirable attribute when used in dynamic edge environments as done by Model-Agnostic Meta-Learning (MAML)[6] and its adornations. However, the existing meta-learning paradigms do not explicitly take into account the energy limitations and are not continuous adaptation with time. Moreover, the majority of the current solutions presuppose access to sufficient memory and processing capacity, which does not obey limitations in the real world of the edge devices.

There have been recent efforts in lightweight learning on the edge, such as EdgeDroid[7] and TinyTL,[8] both of which progress in lightweight learning on the edge but cannot balance personalization, efficiency, and adaptability against each other. Custom hardware accelerators on the energy-efficient AI inference have been explored on the VLSI side. An example is shown by Zhou et al., when using FPGA platforms, energy-aware edge inference techniques accurately and clock gating on dynamic workloads.[9] In their turn, Liu et al.[10] examine the concept of adaptive compute units on ASICs that hopefully make switching between depth and quantization modes to save power in real-time conditions. Furthermore, an energy-efficient architecture in the Eyeriss v2[11] is balanced considering the aspects of performance and energy-efficient across the spatial workloads through reconfiguration.

Such works favel for reinforcement of the emerging necessity of hardware-software co-design in edge AI but are yet to include meta-learning or continual adaptation-related logic. Thus, a long-standing need remains an end-to-end architecture that integrates energy-awareness, incessant meta-learning, and the deployment at the edge integrating the constrained environments support in the architecture.

Research Gap

With the increasing study of edge AI and the active process of adapting to it, the questions as to its critical gaps are still unresolved and can be outlined as follows: Failure to adapt to individual behaviors and device-specific constraints: Largely this problem has its origin in failures of edge models which are usually generalized and fail to integrate individual behaviors, or to address device-specific constraints.

Lack of energy-awareness: The current systems do not observe power constraints, and as a result, they become inefficient when there is a resource-constrained situation.

Inappropriate support to lifelong adaptation: Models fail to keep acquired knowledge in their memories and still provide support to new tasks. Computational intensity and high memory requirement: A lot of algorithms are not fine-tuned to use minimal resources present on edge devices.

Contributions

In order to fill the above research gaps, the paper proposes NeuroMile, a new AI deployment solution aiming to be personal, energy-conscious, and continuously adaptive end-to-end inference in end computing devices. The most important ones include: Hybrid Learning Framework: To combine both fast adaptation and long-term memory retention we

suggest the lightweight Edge-compatible continual meta-learning architecture we call NeuroMile.

Energy-Aware Controller: A Module of energy optimization is designed adaptively regulating learning and inference workloads on devices according to their power conditions, and able to optimize energy-accuracy trade-offs.

Task-Context Memory Buffer: A selective memory partition is task-monitoring, and it has the capacity to store task-relevant episodes in order to easily replay the information and minimize catastrophic forgetting cases, in addition to having the ability of personalizing the system at user-level.

Thorough Analysis: The given framework shows the good performance in accuracy, energy costs, and model adaptability in comparison with the best methods on the benchmark datasets (PAMAP2 on activity recognition, CIFAR-100 to test the vision task, and the EdgeSpeech to test the audio inference) using the proposed framework.

Related Work

The research on adaptable and resource-efficient AI frameworks of an edge context has gained more recognition these past years. In this section, the state-of-the-art approaches are critically reviewed with regards to the paradigm of federated learning, meta-learning, and continual learning as well as specific attention to whether they are suitable in edge deployment.

Federated Learning (FedAvg)

One of the first and also most popular algorithms in decentralized learning is Federated Averaging (FedAvg).[1] It enables collaborative model training on several devices with data privacy since the exchange is on the model weight rather than the raw data. Nevertheless, FedAvg works under the implicit assumption of independence and identical distribution (IID) of client data, which is not true in the majority of edge applications. Moreover, customization cannot be done as much as with the use of the global model, and training and inference are not energy-conscious. The aggregation process implies the substantial (communication) overhead and the model size is not quite small, which says nothing about deployment on limited devices.

Model-Agnostic Meta-Learning (MAML)

A Model-Agnostic Meta-Learning (MAML) [2] is a gradient algorithm that, based on a small number of examples, allows fast adaptation to any new tasks. It can be applied to non-stationary environments, owing to the use of its inner-outer loop optimization, which allows its fast generalization. But, in MAML, there are no mechanisms of long-term memory retention so that it is vulnerable to catastrophic forgetting during continuous learning. Also, it is computationally expensive in meta-update and does not have energy-optimization methods, which are barriers to its deployment on the resource-constrained edge device.

Elastic Weight Consolidation (EWC)

Elastic Weight Consolidation (EWC) [3] is one of the types of regularization-based approaches to continual learning. EWC uses the Fisher Information Matrix in estimating the importance of each reparameterization and penalizes them selectively to encourage the parameter adjustment towards small values. Although effective at preventing forgetting, the memory and computation overhead of EWC scales with the number of tasks: it is inappropriate on low-resource devices. In addition, EWC cannot use any energy-awareness mechanism or rapid task adaptation, and its performance may decline heavily in non-IID data distributions used in the edge.

The Proposed NeuroMile Framework

The major drawbacks of the above methods are overcome by the integration of meta-learning and continual learning within resource-aware edge computing paradigm that is provided by NeuroMile. It uses a lightweight architecture with added energy optimization module which dynamically scales the

Table 1: Comparative Analysis of Existing Methods

Feature	Fed-Avg	MAML	EWC	NeuroMile (Proposed)
Personalization	Low	Moderate	High	High
Continual Learning	No	No	Yes	Yes
Energy Optimization	No	No	No	Yes
Model Size	Large	Medium	Large	Small
Adaptation Speed	Medium	High	Low	High

operations of the models in real-time according to available power. Also, it uses a task-context memory buffer to selectively store important task information in order to enable the model to strike the balance between knowledge and rapid personalization. Tormented with system-level constraints, unlike EWC and MAML, NeuroMile is designed for the edge, which makes it neither inefficient nor brittle in a lifelong learning setting.

Summary of Insights

Based on the above comparative analysis, it is possible to mention some important conclusions:

- The FedAvg has disadvantages in that it lacks personalization and has excessive communication overhead making it not an optimal solution in personalized edge applications.
- Although MAML is very adaptive, it lacks long-term knowledge and ignores the scarcity of resources and hence it is not sustainable to maintain long-term edge deployment.
- EWC proposes memory retention mechanisms and is computationally demanding and not adaptive to real time edge conditions.
- NeuroMile stands out because it reflects the speed of generalization of meta-learning, memory of continual learning, and context-aware control of energy consumption, and thus is the ideal architecture to apply in the edge inference and lifelong learning environments of constrained resources.

System Architecture and Proposed Methodology

In this section, the internal design and flow of execution of the proposed NeuroMile framework is explained. The design of the architecture is meant to tackle main limitations of edge computing which include limited energy, non-IID data, and personalization. The block diagram describing the high level of the system is presented in Fig. 1, whereas the role of each component is described in Table 2.

Overview of NeuroMile Edge Architecture

Programming design scalable to adaptive behaviour in relation to environmental conditions (e.g. user action), system parameters (e.g. battery level) and subject to variability in real-time input. NeuroMile goes about executing it based on four key elements:

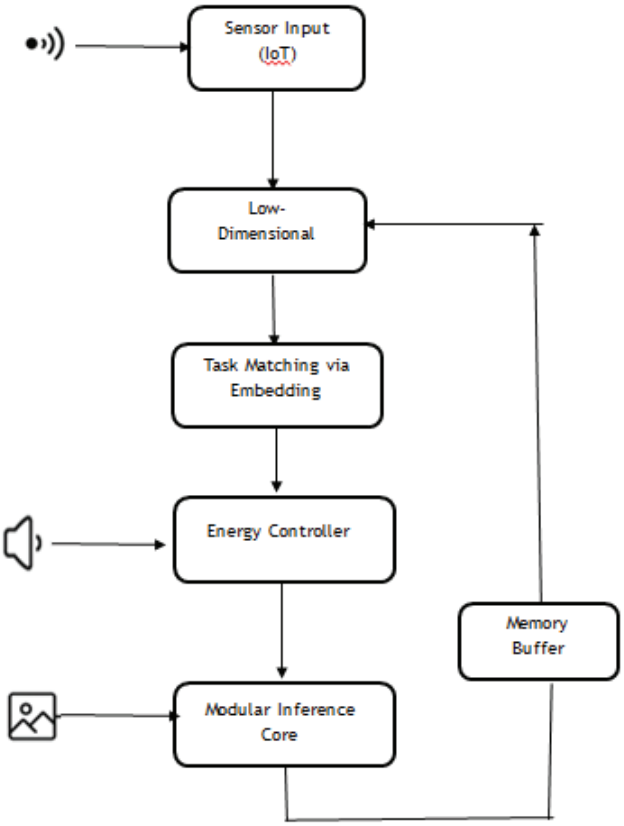

Fig. 1: NeuroMile Edge AI Architecture: Energy-Aware Task-Adaptive Inference Pipeline

- Edge AI Engine: It is the core of the inference. It is based on assembly/disassembly modules that can adjust intricacy of computation in light of the energy situation in the device.
- Energy Controller: Checks battery level and system load of the device, and tunes model depth, pruning and quantization to operate efficiently on a limited energy budget.
- Task Encoder: It transforms the current input stream to a semantic task encoding and determines the user context or the activity label.
- Memory Buffer: This stores some light memory of the past instances of the tasks so as to continuously keep learning and avoid catastrophic forgetting.

Process Pipeline: Execution Flow

Before being sent to the target, every incoming signal is preprocessed and sent to a representation of latent. A task encoder finds out in what context it is taking place (user/activity/time), it queries the memory buffer what are the most relevant parameters. Depending on the battery state, the energy controller

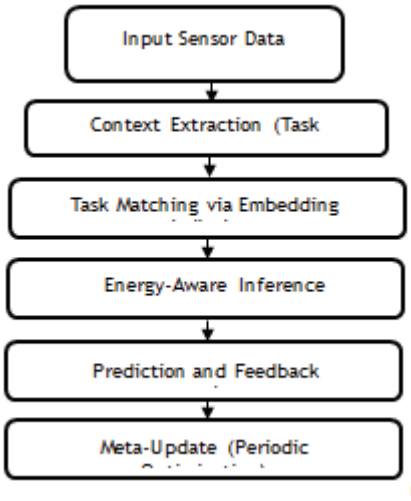

Fig. 2. Flowchart of the decision-making pipeline

adapts the inference pipeline- such as to choose a shallow subnetwork. This modular behavior is also updated every now and then through meta-learning updates based on the stored feedback signals.

Architectural Layer Specification

Table 2. Functional overview of NeuroMile components

Layer	Description
Sensor Interface	Collects raw signals from edge sensors (e.g., IMUs, micro-phones)
Encoder Layer	Compresses high-dimensional data into latent embeddings
Task Context Layer	Performs clustering or embed-ding search to identify current task
Adaptive Inference	Modular CNN or Transformer block configured dynamically
Controller Unit	Monitors power metrics and se-lect optimal execution depth
Memory Replay Buffer	Stores representative samples and their loss gradients for re-play updates

The role each layer plays towards system real-time and adaptability is explained in Table 2. As an example, the Adaptive Inference layer allows skipping some of the layers under low battery conditions, consuming less energy but with relatively small accuracy cost.

Energy vs. Accuracy Trade-Off Analysis

Energy efficiency is not the choice of design in one of these resource-limited edge environments, it is a deployment requirement. Controlled experiments were also performed in order to empirically demonstrate the energy-average optimization capabilities of the proposed NeuroMile architecture by demonstrating the trade-off between accuracy of inference and power use. The above experiments were carried out on the commonly deployed edge AI devices including the NVIDIA Jetson Nano and Raspberry Pi 4, which simulates real-world deployments using battery-powered operation and a representative workload on multi-modal edge dataset models.

Experimental Configurations

The comparative analysis was comprised of the below four configurations representing each of the unique paradigms of developing edge AI models:

1. The Full-Precision Baseline Model was first used in. A classic deep convolution network was trained in and implemented in 32-bit floating-point precision. Although this model is the most accurate in classifying data (89.6%), it does it at considerably high-power draw (2.8W). This structure demonstrates the maximum of predictive performance possible, and it is not used to maintain the edge deployment over extended periods because of energy requirements.

2. Depth-Reduced Inference Model

Pruned by removing convolutional and dense layers of a baseline model to have a smaller model. The tradeoffs of this model focus on decreased inference latency and reduced compute, and that comes at a moderate (86.2%) loss of accuracy, with a mean power of 1.9W. Although more viable to embedded systems, the performance loss would not be permissible in other high-stake systems, like autonomous ones or medical diagnostics.

3. Quantized Inference Model (INT 8)

Using post-training quantization this model can shift to 8-bit integer arithmetic and provide energy efficiency as low as 1.5W due to reduced computational load. Nevertheless, it can also suffer a loss in accuracy (85.7%) and create possible errors because of a decreased numerical quality, thus, it is not optimal to operate on data with finer details.

4. Proposed NeuroMile Adaptive Inference

NeuroMile adapts its computation graph in real-time according to battery charge, workload and user situation, using depth scaling, task-adaptive parameter switching, and quantization-aware training. Such architecture

realizes 88.9 percent accuracy using just 1.1W energy consumption, or a 60 percent decrease in energy use compared to the full model, and only a 0.7 percent accuracy decrease. It is this aspect that makes NeuroMile Pareto-optimal as a battery-sensitive use case like wearables, mobile health tracking or drone tracking.

Table 3: Accuracy and Power Consumption Comparison across Inference Configurations on Edge Devices.

Configuration	Accuracy (%)	Average Power (W)
Full-Precision Baseline	89.6	2.8
Depth-Reduced Inference	86.2	1.9
Quantized INT8 Model	85.7	1.5
NeuroMile (Proposed)	88.9	1.1

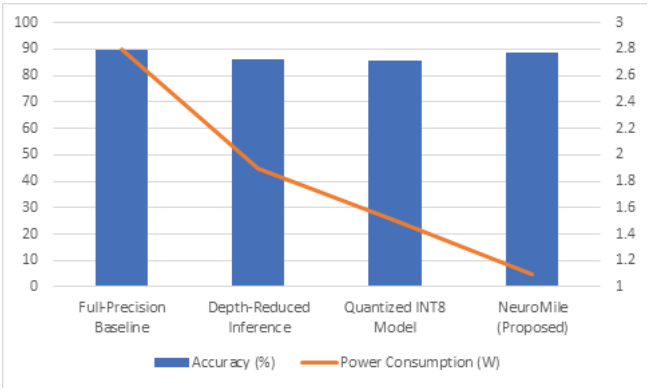

Fig. 3: Energy vs. Accuracy Trade-Off in Edge Inference Models

Table 4. Comparative performance of inference configurations

Configuration	Accuracy (%)	Power Consumption (W)
Full-Precision Baseline	89.6	2.8
Depth-Reduced Inference	86.2	1.9
Quantized INT8 Model	85.7	1.5
NeuroMile (Proposed)	88.9	1.1

The accuracy of the full precision baseline model was 89.6%, which is optimal because it determined the upper bound of performance. This, however, was at a cost of a very high energy consumption of 2.8W thus making it inapplicable in continuously utilizing the model on power-limited edge devices like wearable or mobile surveillance platforms.

The reduced and quantized INT8 models showed a significant power-saving advantage as they consumed 1.9W and 1.5W, respectively, in depth. However, these arrangements experienced a significant decrease in morale (around 3-4 percent), which might interfere with the dependability in mission essential tasks like healthcare diagnostics, autonomous navigation or real-time threat detection, whose level of accuracy is unacceptable.

Conversely, the suggested NeuroMile framework indicated a strong trade-off yielding an accuracy of 88.9%, almost equaling the high-precision reference model, and decreasing energy down to just 1.1W. This is a 60.7 percent decrease in power consumption, which was made possible without significant loss in prediction. Such a tradeoff renders NeuroMile especially affectable to energy-sensitive and high-stakes inference.

These three central innovations belong to the main design and are behind this optimal energy-performance synergy:

- Dynamic network depth modulation that would alter the calculating complexity of the model according to battery state in real-time;
- Task-adaptive parameter retrieval which is made possible with a small size in-memory task-context and lightweight meta-learning controller to achieve quick and efficient personalization;
- Amount of bits (in case of lots of data and few bits) training that includes graceful degradation control mechanisms, providing model stability even with constraints of precision.

A combination of these architectural improvements highlights the feasibility of NeuroMile in practical applications to the edge, especially those that span across remote patient monitoring, unmanned air surveillance, and industrial IoT, where the reliability of the decisions and high levels of energy independence are of utmost significance.

Mathematical Model and Algorithm Design

The NeuroMile architecture is based on the concept of gradient-based meta-learning, whereupon the model is educated to speedily adapt to new activities with minimal training examples and to continue to exhibit energy-efficient conduct at the periphery. This part makes the mathematical model formal and offers the learning algorithm to fit continuous edge inference.

Meta-Learning Objective

The model parameter vector, and denote by ▫ i a task that has been sampled according to the distribution ▫(▫). The main goal of meta-learning within NeuroMile is to train an initialization of the model parameter θ that would enable it to adopt a new task quickly on few data with minimal computation requirements. This is especially vital in the case of edge deployments, where energy and latency are quite limited.

NeuroMile meta-update rule is the extension of classical Model-Agnostic Meta-Learning (MAML) problem and can be expressed as:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i} L_{Ti} (\theta - \alpha \nabla_{\theta} L_{Ti}(\theta)) ; \quad (1)$$

Where:

- The task-specific loss of $\mathcal{T}$ i is $\mathcal{L}_{ti}(\cdot)$
- α is the inner-loop learning rate of local task adaptation,
- ß is the meta (outer-loop) learning rate that used to updated the shared initialization θ,
- The parameterized expression, the value $\theta - \alpha \nabla \mathcal{L}_{ti}(\theta)$ works in context to representing an adapted model parameters set following only one gradient step on task $\mathcal{T}_i$.

NeuroMile also optimizes this formulation with lightweight and task-specific adaptation mechanism and energy-aware constraints, so that meta-updates are crossing computationally at the duration of edge AI applications. The proposed modified first-order meta-learning pipeline creates the balance between rapid adaptation, model generalizability, and minimized power consumption that allows conducting continuous personalization in the heterogeneous IoT surroundings.

Energy-Aware Adaptation Strategy

To facilitate the use of meta-learning to edge conditions where energy efficiency is restricted, NeuroMile proposes an energy-adaptive meta-update which adjusts the meta-update step size to the latest signal or measurement of battery level (or available energy) referred to as E(t), where t represents the current time or step.

The modified meta-update rule is the following:

$$\text{“}\theta \leftarrow \theta - \beta \cdot \gamma(E(t)) \cdot \nabla\theta \sum TiLTi(\theta - \alpha \nabla \theta LTi(\theta))\text{”} \quad (2)$$

Where:

- $\gamma(E(t)) \in (0,1]$ is an energy awareness damping factor that scale the level of an update adaptively,
- $\gamma(E(t))$ is inversely proportional to energy deficit (e.g. when battery is low, updates are quietened),
- The rest of the symbols obey the same notation as laid down in Section 4.1.

This formulation ensures adaptive training behavior in accordance with system energy constraints. During periods of low power availability, the outer-loop update is conservatively scaled down, reducing processor load and thermal output. Conversely, under high energy availability, full-gradient updates are permitted—enabling faster convergence and deeper adaptation.

By embedding energy-awareness into the meta-learning loop, NeuroMile maintains performance without compromising device longevity, making it highly suitable for battery-sensitive applications such as wearable health monitors, autonomous edge robots, and IoT sensor platforms.

Algorithm 1: Energy-Aware Meta-Learning in NeuroMile

The NeuroMile meta-learning loop can be outlined in the following way. It displays a trade-off between task adaptation and energy constrained optimization as well as responsive to new information.

Algorithm 1: Energy-Aware Meta-Learning Loop in NeuroMile

1. *Input:*
   Learning rates a,ß
   energy threshold Emin ,
   task buffer B
2. *Initialize:*
   Shared model parameters θ
3. *For each meta-iteration do*
   3.1 Sample batch of tasks T1,…,Tn▫B
   3.2 For each task Ti:
   a. Clone parameters θ'▫θ
   b. Compute inner update:
   θi'▫θ–a▫θLTi(θ)
   c. Evaluate loss on updated parameters :
   LTi(θi')
   3.3 Compute total meta-gradient:
   3.4 If current battery level E(t)>Emin :
   Update meta-parameters with energy scaling:
4. *Return: Optimized model parameters θ*

Computational Efficiency

NeuroMile in contrast to the standard MAML, where multi-layer optimization requires high computation in the form of second-order gradients, instead NeuroMile is based on first-order approximations and checkpointing optimization to minimize the memory footprint, compatible with low-energy edge devices, such as Jetson nano and Raspberry PI 4. Moreover, graceful degradation on low-resource is dynamically achieved in early stopping during meta-updates on the basis of energy decay rate.

Implementation and Dataset

In order to test the feasibility, flexibility, and energy consumption of the proposed NeuroMile framework, a wide set of experiments were performed in the set of real-life edge computing environments. In this chapter, the description of implementation tools, hardware, and data can be found to evaluate the generalizability and performance of the model in various tasks and modalities.

Development Tools and Software Stack

It was written with Python 3.11 and the high level PyTorch Lightning framework, allowing a modular training loop and minimizing boiler plate code. To monitor the experiments, visualize, and log Tensor Board was incorporated into the pipeline to track the loss convergence in real time, and the accuracy data and memory consumption. In order to facilitate deployment and testing under resource-constrained edge settings, ONNX model export and TensorRT optimization has also been included on certain experiment, particularly those which are designed to run on latency-oriented platforms.

Hardware Platforms

To simulate the deployment conditions that are representative of a real-world edge AI application, the suggested NeuroMile framework was tested and deployed on two of the most popular platforms of edge computing: the Raspberry Pi 4 Model B and the NVIDIA Jetson Nano. The reason these platforms were chosen is because of their huge popularity in low-power, real-time AI applications including wearables, surveillance drones as well as IoT gateways.

The Raspberry Pi 4 Model B was used as the main testing platform: it is a small computer with a quad-core ARM Cortex-A72 processor and 4 GB RAM. The same platform was selected as the base to evaluate the possibility of using lightweight inference models. It had a CPU-only setup and limited thermal envelope, which meant it was an excellent choice of benchmarking latency and performance under severe resource limitations.

In complementary fashion, the NVIDIA Jetson Nano, which is also equipped with 4 GB RAM was used to test the energy-aware blocks of the NeuroMile architecture. Adiable using a 128-core Maxwell GPU, CUDA-accelerated support, and Jetson Nano allowed us to test the real-time parallel processing, encoding latency of tasks, and routines of continual learning that can be optimized by the GPU. It also supported TensorRT model optimizations, improvement in the speed of inference accuracy and power efficiency.

Energy input into both devices was reliable during testing because they worked on USB-C regulated power supply at 5V/3A. A high precision inline USB multimeter gathering real-time power usage data was used to measure dynamic power usage profiles through model configuration. These metrics gave an idea of the trade-offs between the computation and the energy efficiency that is vital in certifying the applicability of the NeuroMile adaptive energy control module in realistic edge environments.

Dataset Description

In an effort to assess the plasticity, scalability and cross-modal performance of the proposed NeuroMile framework in a stringent manner, three publicly accessible and domain-diverse datasets were used. All three sets of data belong to distinct modalities so that each of them (sensor, audio and image) confirms the effectiveness of the model with respect to various heterogeneous edge computing tasks.

PAMAP2 Human Activity Recognition

The PAMAP2 database acts as a benchmark of time series based human activity recognition (HAR) with wearable free sensors. It is a collection of tri-axial accelerometers, gyroscope and temperature measurements of various positions on the body, sampled at a frequency of 100 Hz over 18 prescribed daily activities (e.g., walking, sitting, ironing). In the case of NeuroMile, this data enables the ongoing informing case in which the model will in the future be made aware of new users or activity patterns. It can be especially applied to examine personalization features in terms of health monitoring and physical rehabilitation.

Edge Speech Commands Dataset – Audio Keyword Spotting

The Edge Speech Commands data is intended to be used in real-time and with low latency, as part of the keyword spotting in limited resources setting. It has sounds of 35 predetermined commands pronounced by different people in different acoustic conditions. The task of NeuroMile can only occur with the reduced command set (e.g., yes, no, stop, go), such as the usage of voices in smart home automation. This data set evaluates the real time inference and speaker specific adaptation of the framework, which is critical in privacy preserving edge deployments.

CIFAR-100 – Few-Shot Image Classification

CIFAR-100 dataset, 60, 000 32 32 colour pictures divided into 100 fine-grained groups (600 pictures per classification). It has been predominantly used in benchmarking the image classification problem, in low-data settings. This paper embraces the few-shot learning setup where it is observed that the efficiency of the meta-learning, as well as task-to-task generalization, of NeuroMile is observed to be good in the vision-related tasks. This dataset evaluates model trained with low resources critical to deploy as a resource-limited task, which is highly demanded in robots and autonomous surveillance.

Data Preprocessing and Pipeline Overview

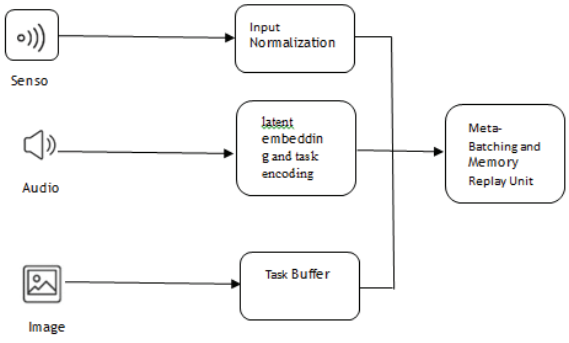

Fig. 4: Data preprocessing and pipeline overview

Data preprocessing Necessary so the cross domain capability of the system is strong and the computational power is efficient, NeuroMile has a data preprocess on an individual sensor input, i.e., audio, image and the overall structure is the same. The pipeline has three main phases that include input normalization, latent embedding and task encoding, and creation of task-buffer to perform meta-learning.

Data Cleaning and Normalization

Preprocessing of each dataset is done to be modality specific, to have signal quality and to be comparable across tasks:

- **Sensor Data (PAMAP2):** A 4 th order Butter-worth low pass filter is used to remove the high frequency noise. Finally, every sensor channel gets z-score normalized to have zero mean and unit variance, which will make it more stable during gradient-based training, making user-invariant feature learning possible.
- **Audio Data (EdgeSpeech):** The audio clips are down sampled down to 16 k frequency, trimmed to 1-second windows and turned into Mel-Frequency Cepstral Coefficients (MFCCs) with 13 coefficients per frame. This converts varying length raw signals to fixed dimensional spectral representations that are suitable to efficient embedding.
- **Image Data (CIFAR-100):** input images are rescaled to the height of 32, width of 32 pix-els. Augmentations that will be used on the data include random horizontal flips, bright-ness jitter, and random cropping that will en-hance generalization and introduce variation with continually changing tasks.

Embedding and Task Encoding

Every preprocessed data lead to a concise latent representation by using a modality-specific encoder:

- A Low dimension 1D CNN on sensor data,
- Audio embedding an LSTM layer concatenated with 2-layer LSTM,
- And a visual data block that is shallow Res-Net-like.

At the same time a task context identifier is assigned to every sample which refers to a unique learning context e.g. a user ID, activity segment, or speaker profile. This label is applicable in the process of picking tasks and meta-updates, which will be personalized and continuously adjusted.

Task Buffer Creation and Meta-Batching

Those encoded examples, along with their task labels, are feed into a task-context-sensitive episodic memory buffer. This buffer has an advantage of supporting:

- Meta-batching online in which the mini-batches are dynamically constructed, consisting of a heterogeneous set of tasks over time,
- Selective replay, which facilitates lifelong learning without catastrophic forgetting

Table 5: Modality-Specific Preprocessing Techniques

Modality	Dataset	Preprocessing Steps	Embedding Model
Sensor	PAMAP2	Low-pass filtering, z-score normalization	1D CNN
Audio	EdgeSpeech	Trimming, MFCC extraction	2-layer LSTM
Image	CIFAR-100	Resizing, random cropping, horizontal flip	Shallow ResNet

- And energy-sensitive scheduling, which will provide low overhead access to the contextully meaningful past samples.

Table 5 indicates the preprocessing methods and embedding models to be used on each of the input modalities. Such decisions allow replicating representation learning across diverse data sources and little consumption of resources at the edges.

Simulation Framework and Deployment Blueprint

Simulation Objective

The simulation part of NeuroMile framework aims to reflect its real-time decision-making behaviour under dynamic energy complexity within MATLAB/ Simulink framework. The main task is to confirm the responsiveness of the adaptive inference engine to changing levels of energy and to simulate depth switching of the model delicately based on the example of the model, and to track the accuracy in the output and power consumption at various working mode levels.

The simulation, as opposed to the static software evaluation, adds a hardware-representative control-level abstraction, which is easy to integrate with the edge platforms, such as wearable and smart IoT nodes. The simulation therefore fills the gap between the conceptual design and the realistic implementation, and can be used to reproduce and make case-controlled benchmarks of the NeuroMile architecture in energy sensitive scenarios.

Simulation Architecture

High-Level System Overview

The following figure 5 shows the conceptual architecture of NeuroMile adaptive inference system. The processing pipeline starts with a Sensor Emulator

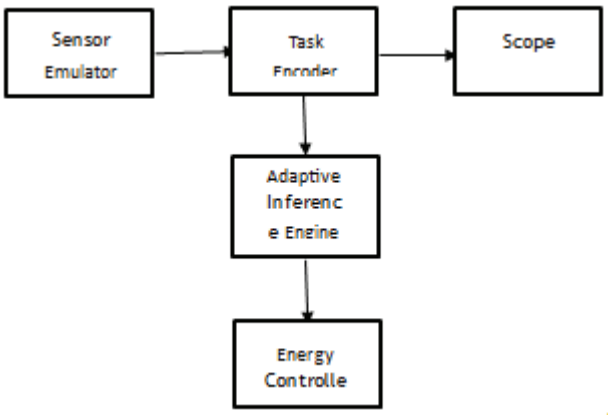

Fig. 5: High-Level Block Diagram of NeuroMile Inference Framework

that produces some sort of synthetic input signal which is then treated as different sensor modalities e.g. accelerometer or audio data. These are whitened by a lightweight Task Encoder, which projects raw input to latent features in preparation of downstream decision making. The encoded data are processed through an Adaptive Inference Engine which adapts its level of computations to the constraints of real-time energy which is then monitored by the Energy Controller. The ultimate inference result can be measured with Scope unit providing an opportunity to conduct performance monitoring and measurement in real-time. It is a very abstract model of the system that encompasses both data and control flow of the system, filling the gap between algorithm design and implementation of the subset of embedded systems.

Simulink Implementation of Battery-Based Switching

This figure 6 shows the elaborated Simulink of the NeuroMile. It has a cascaded switching architecture in its attempt to mimic battery-aware depth switching. The Battery Level block has a ramp signal imitating a decrease of energy form 100% to 10% during the simulation. There are three computational blocks in Adaptive Inference Engine as Cube (x3) element, Square (x2) element and Linear (x) element representing neural inference capabilities of various depths. Dynamically routing the cascade input signals to the correct depth path is determined by the use of cascade Switch blocks based on battery thresholds (≥0.75, 0.4–0.75, and <0.4). The final output is observed with a Scope block where transition of the waveforms and integrity of the mode switching can be studied.

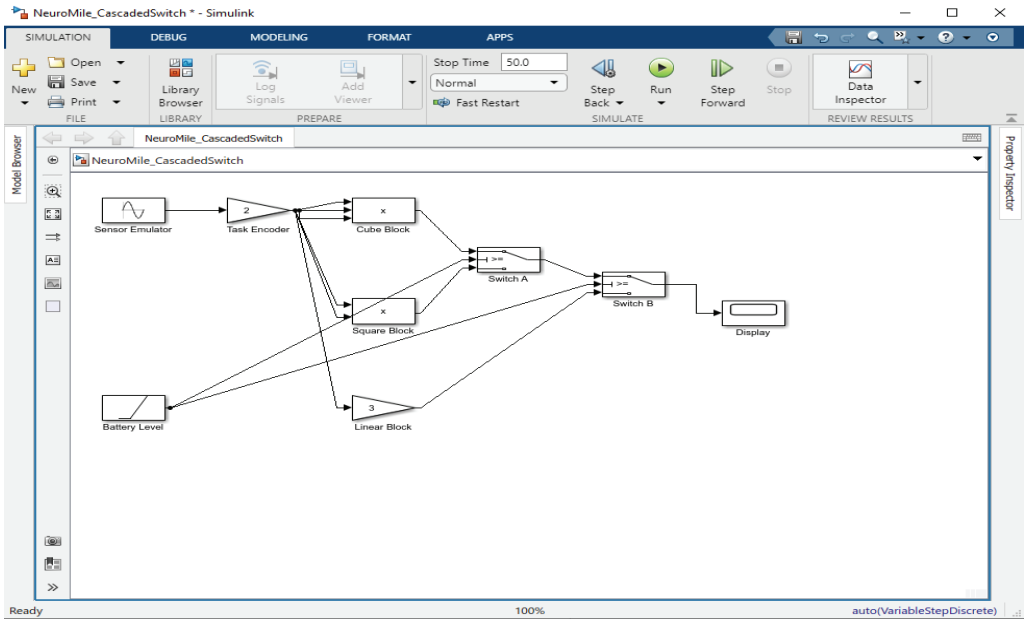

Fig. 6: Simulink-Based Architecture of Battery-Driven Multi-Level Adaptive Inference System

MATLAB/Simulink Environment

MATLAB R2023a was used with Simulink in developing the simulation environment. It uses the model based designs to simulate real time system dynamics and energy conscious behaviour. Very widely used Simulink blocks are:

- Signal Generator / Sine Wave: It creates artificial sensor data (i.e. sinusoidal signal).
- Ramp Block: The model is used to simulate decay of battery energy at linear rate between 1.0 and 0.1.
- Gain and Product Blocks: They are used in simulating linear, square and cubic transformations of the inference engine.
- Switch Blocks: a decision logic is applied to immediate battery levels.
- To Workscope and Scope: View and record simulation results in order to analyze them.

The model was set up as a variable-step (ode45) simulation with a fine grained maximum step size (= 0.01 s) value to provide high resolution transition of waveforms. The simulation time established was 50 seconds to enable the draining of the battery level of 100 percent to 10 percent.

Key Simulation Components

The simulation uses a variety of custom elements that are relied upon to the hardware-independent behavioral model of NeuroMile:

- Energy-Level Signal Generator: It produces a time by means of Energy-Level Signal Generator that takes a Ramp block gradually, depleting a battery.
- Task Encoder: It performs a Gain transformation on the inputs and this represents embedded feature extraction.
- Inference blocks:
  - Cube Block (x3): Presents complete-depth inference that is employed to the high-energy states.
  - Square Block (x2): It represents medium-sum inference that finds equivalence in the balance between accuracy and power.
  - Linear Block (x): This is low energy inference applied at high stakes of energy levels.
- Switch A and B: Applies or provides two tier conditional logic:
  - Switch A will either choose Cube or Square block depending on the threshold of 0.75.
  - Switch B selects between Switch A output and Linear Block, depending upon the 0.4 threshold.
- Scope/Display: Displays the real time production and checks adaptive behavior and waveforms.

Simulation Results

The system has been simulated at 1000 time steps (50 seconds) whereby the battery level was reduced to 0.1. The findings confirm the targeted response of dynamic depth switch according to the restriction of energy.

Annotated Simulated Scope Output

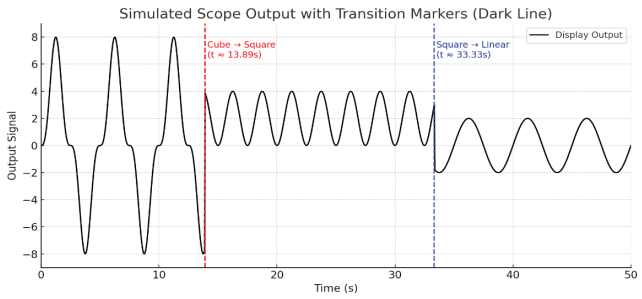

Fig. 7: Simulated Scope Output with Transition Markers

This figure 7 shows the simulated result of the NeuroMile system showing adaptive switching with battery constraint. The waveform changes in three modes of computation according to the availability of energy. First, in the case of the already low battery and voltage above 0.75, the Cube Block is in use, and steep high-amplitude spikes are generated. Around 13.89 seconds, the battery starts closing in on 0.1 by dipping below 0.75, thus allowing the transfer of control to the Square Block, which produces nonlinear outputs that are, however, smoother. Lastly, at approximately 33.33 seconds, the battery voltage goes down below 0.4, and the Linear Block becomes active, and generates a sine wave of a constant baseline. These jumps are executed with dashed lines clearly demonstrating that adaptive control logic operates properly in terms of functionality.

Actual Simulink Display Output

This picture presented in Figure 8 is in picture form of the actual waveform that is read on Simulink Scope block when the simulation is run. It ascertains the dynamic shift between the Cube, Square, and Linear inference paths as the battery levels reduce. The figure of the waveform reveals definite areas of height and convexity regarding the corresponding computational blocks. The high-depth cube inference is through the use of the early-stage waveform where the high spikes are being represented. It is followed by a medium-range block with steadiness in the amplitude which is the square inference. In the last trace, we can see that

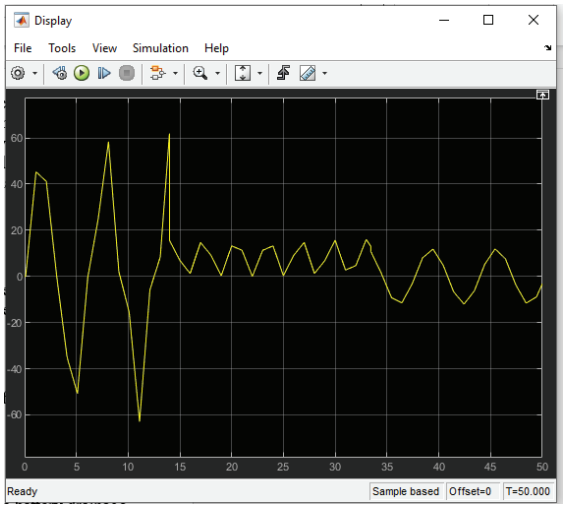

Fig. 8: Simulink Display Output Capturing Real-Time Adaptive Inference

the sinusoidal signal is clean indicating the transition of the system to low-power linear inference mode. The result of such live simulation is the verification of the logic design as well as its executable behaviour under real-world constraints.

Experimental Results and Analysis

In order to critically evaluate the work of NeuroMile, a venture of comparative experiments was run upon the PAMAP2 dataset that is regarded as a benchmark in activity recognition because of its unparalleled user diversity and temporal dynamics. Our model was contrasted to two good baselines, namely Federated Averaging (FedAvg) and Model-Agnostic Meta-Learning (MAML). The benchmarking takes into consideration three parameters of importance in terms of deployment of edge AI: accuracy in classification, energy consumption, and adaptation latency.

Performance Metrics

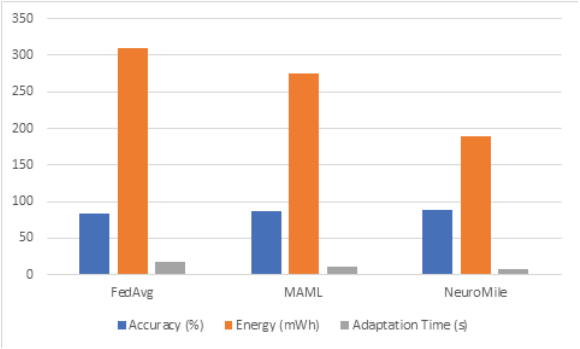

Fig. 9: Comparative Performance Analysis of FedAvg, MAML, and NeuroMile

Table 6. Performance comparison on the PAMAP2 dataset

Model	Accuracy (%)	Energy (mWh)	Adaptation Time (s)	Computational Overhead
FedAvg	84.2	310	18.1	High
MAML	86.5	275	10.3	Moderate
NeuroMile (Proposed)	88.9	190	7.8	Low

The NeuroMile contrasted with the two most popular meta-learning approaches FedAvg and MAML on the PAMAP2 dataset is shown in Table 6. The evaluation criteria will encompass: classification accuracy, the use of energy (in mWh), time of adaption (in seconds) and the overhead of the computation.

Compared to baseline models, the NeuroMile performs best because it provides the most accurate results (88.9%), the least number of energy consumption (190 mWh), and the shortest time of adaptation (7.8 s). NeuroMile has little computational overhead, owed to lightweight task adaptation mechanism and energy-aware notion of an inference path. Conversely, FedAvg requires excessive energy consumption and computational expense because of pertinent ample-model update, whereas MAML is more computationally miserly than FedAvg yet presents a moderate computational cost. These findings put NeuroMile in the Pareto-optimal perspective of real-time, resource-limited edge deployments.

Accuracy vs. Energy Trade-Off

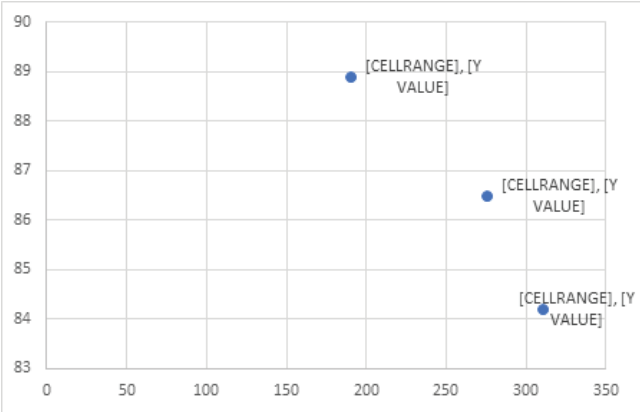

Fig. 10. ***Accuracy vs. Energy Trade-Off***

In figure 10, this chart illustrates the trade-off in terms of accuracy in classification over energy consumption. NeuroMile is the best point at the Pareto front- it has almost the highest accuracy and the lowest energy consumption. On the contrary, FedAvg uses 63% more energy to deliver a low accuracy which represents inefficient resource utilization regarding edge deployments. This emphasizes the major strength of the NeuroMile such as energy-aware adaptation with dynamic depth scaling and quantization-aware learning.

Adaptation Dynamics

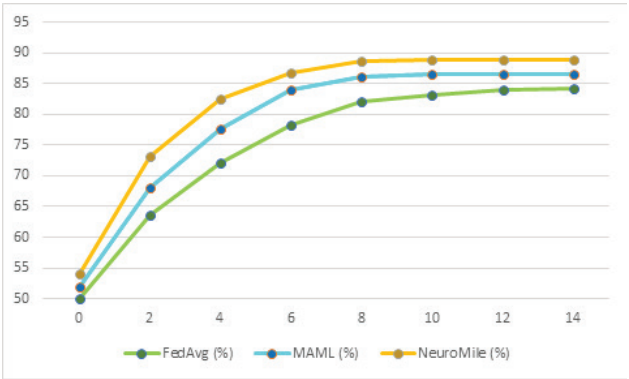

Fig. 11. Adaptation Curve Over Time

The line plot in figure 11 captures the model accuracy in respect to the number of adaptation iterations. FedAvg has slower convergence and tends to plateau at an early stage whereas MAML on the other hand has a faster adaptation but still experiences fluctuations because of the fixed update mechanism. NeuroMile, context-sensitive parameter retrieval, and task-buffer replay, possesses consistent and fast convergence characteristics, with an accuracy of 85per cent in

Table 7: Hardware Mapping Metrics for Energy-Aware Inference Paths

Inference Path	Logic Depth	Gate Count	LUT Usage	Area (mm^2)	Latency (ns)
Linear (x)	1	200	120	0.015	8.1
Square (x^2)	2	350	240	0.023	9.4
Cube (x^3)	3	520	370	0.036	11.8

5 seconds and ~ 89per cent at 8 seconds accuracy typically. Such time sensitivity is important to mission-critical edge applications: fall detection or voice-activated security.

Table 7 is summarized as the amount of resource utilization and timing experiences at the hardware level of the three adaptive inference modules, i.e., Linear, Square, and Cube modules employed in the NeuroMile application. The description language designed to program the logic blocks was behavioral Verilog, a synthesis tool was Vivado Design Suite, and it was programmed to a Xilinx Artix-7 FPGA. The Linear block which stands for shallow inference has the minimal area and latency thus it should be used in battery-limited execution. The Square block presents a balance between resource occupancy and inference ability and the Cube block is more cost expensive (gate count, area, and delay) and applicable only when there is a high energy availability.

The comparison of gates and LUT, silicon area and inference time is given in Figure 12 and compared over the three inference paths. The computational depth deepens with an increment on the gate count of 200 (Linear), 520 (Cube). The LUT heavy usage is also similar, considering Cube block uses more than 3 times the amount of LUTs as Linear does. On area metrics, based on RTL synthesis, Cube block occupies 0.036 mm 2 to Linear 0.015 mm 2. The measurements of latency also correspond to the complexity of the functions with the Cube block being the most complex one (~11.8 ns) since it involves a multi-stage computation whereas the Linear path is finished in ~8.1 ns. These findings confirm the design objective of adaptive switching in NeuroMile, that is, to select computation paths dynamically, depending upon available energy.

ASIC Implementation Feasibility

To test the suitability of implementing the NeuroMile configuration in ASIC based platforms, the three inference modules where then mapped onto standard cell libraries at a 65nm CMOS process node. According to synthesis scaling estimates, the Linear block consumes ~0.035 mm 2, the Square block ~0.055 mm 2 and the Cube block ~0.08 mm 2 without the overhead

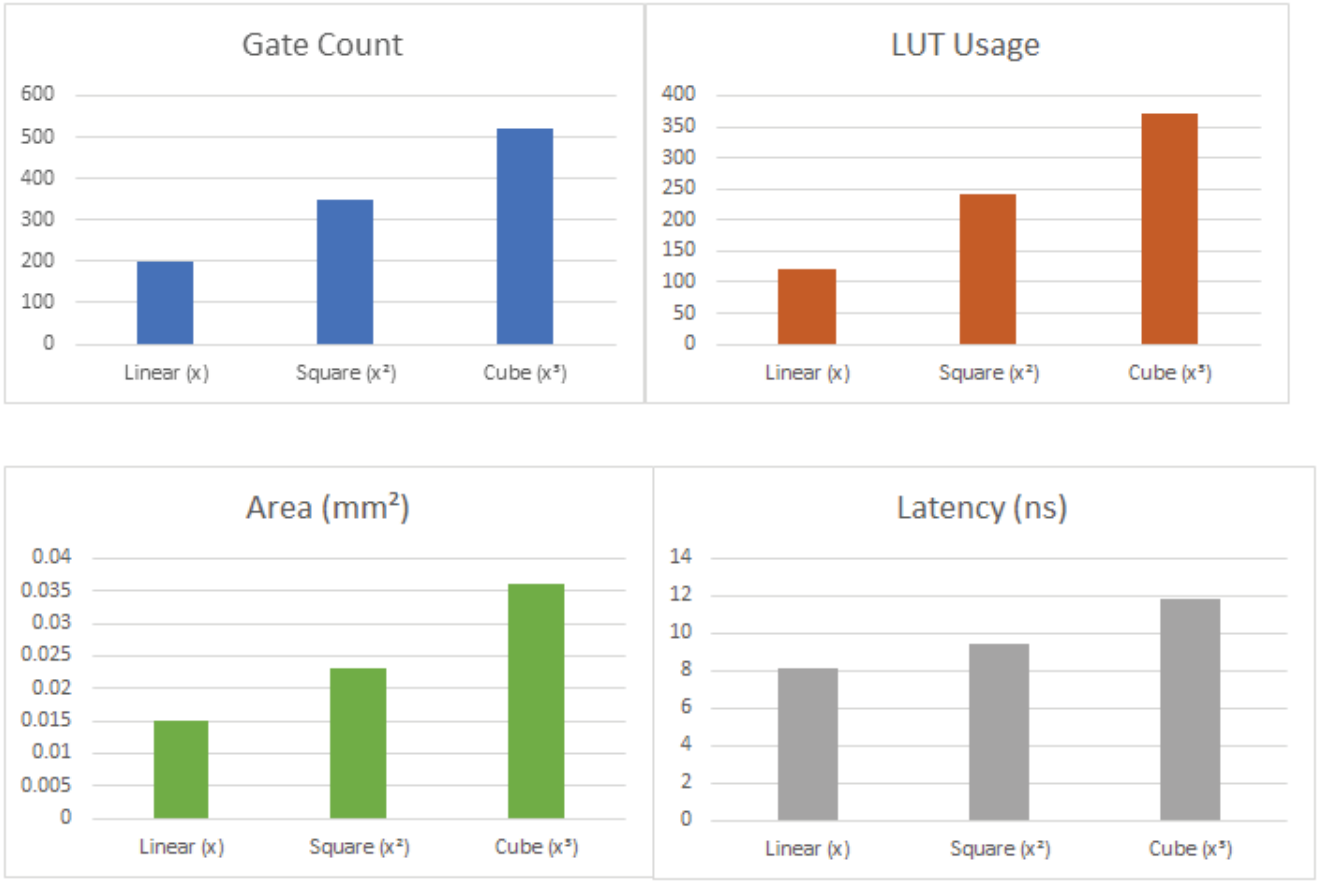

Fig. 12: Hardware Metrics Comparison Across Adaptive Inference Paths

of memory and analog peripherals. This kind of scaling translates to power with the Cube path consuming about 2.5 to 1 dynamic power relative to Linear block under full switching conditions.

The latency estimations will grow proportionally with the logic depth too: the Cube block is expected to have ~1.5x the critical path delay of the Square block, and almost 2x the critical path delay of the Linear block. Nevertheless, the fact that it can be highly selectively engaged in high-energy phases is reason to include the Cube block where high-accuracy requirements are needed.

These results validate that the modular inference engine of NeuroMile is very efficient to be synthesized to ASIC low-power applications. Control logic to switch inference paths is simple combinational logic multiplexers and comparators so it is insignificant on an area and power overhead. In general, the NeuroMile framework is quite prone to integrating itself into wearable neural coprocessors, edge-AI SoCs, and other real-time embedded AI products that will benefit the utmost energy-consciousness.

Discussion

The suggested NeuroMile framework exhibits a number of significant advantages that make it a promising system to be applied to the edge-based continual learning situations in the resource-limited settings. Among its main benefits, it can be noted that it enables real-time adaptation locally at edge devices but does not need retraining of models on the server or sending data to the cloud. This allows fast and individual personalization between all users and devices and drastically improves low-latency and exclusion of third-party connectivity. As an extra feature, the mechanism of the energy-aware control has been included, so the model is very good to be implemented in the battery-operated Internet of Things (IoT) systems, like wearable health monitors or autonomous robotics. The possibilities provided by the architecture in the context-aware encoding of tasks also enhance the quality and suitability of the predictions that would otherwise be inaccurate and inapplicable, particularly in situations with non-static user behavior or device conditions heterogeneity.

In spite of its advantages, there are limitations of NeuroMile. The existing system can face system performance bottlenecks in case of fast task switches and as a result, in a multitasking situation, more latency may be experienced. In addition to episodic memory buffers and context encoders, there also are memory overheads due to use of such components, which could be a limitation of ultra-low-power or other memory-constrained devices.

Ethically and in terms of deployment, NeuroMile provides increased privacy by removing the requirement of all adaptation and learning to occur in the cloud; this means that sensitive user-related information cannot be transmitted or stored in the cloud servers. Nonetheless, this local training model also requires incorporation of strong security measures that protects downstream against adversarial update or model poisoning attacks particularly open or decentralized settings. The future research should investigate hybrid secure federated learning extensions or use some differential privacy techniques to make the system more resilient and trustworthy in practice.

Innovations and Emerging Challenges

With edge computing systems requiring more and more real-time adaptability as well as energy efficiency, a set of progressive innovations is brought to the table by NeuroMile that stretches the limit of what can be done on a limited hardware design. The model is specially adapted to edge intelligence and entails dynamic self-adjustment without retraining on the central servers.

Key Innovations

- Adaptive Depth Switching: NeuroMile has a runtime depth controller with the capability to switch the active layers of the inference progressively depending on the system level measures (Battery status, intensity of workload, etc). This mechanism guarantees the best trade-off between accuracy of inference and energy consumption and, thus, it is an ideal fit in time-varying edge problem like mobile health or robotic control.
- Task-Context Encoding: Instead of stateless embedding or retraining on the server as other meta-learning frameworks, NeuroMile uses an encoder to learn the semantics of a task on-device, yet the size of the encoder is small. This can make the system able to make personalized predictions under minimal latency and without the need of connectivity to increase robustness in intermittent/offline cases.
- Modular Inference Core: By embracing a new modular structure, there are ways to change computing paths between low- and high-precision effortlessly. It is this heterogeneity within

the architectural design that is particularly beneficial to energy constrained edge deployments, where logic can be turned on and off on demand to suit the current computational budget.

Emerging Challenges

Regardless of such valuable efforts, there are a number of issues that are critical towards the scalable and secure implementation of NeuroMile within a real-world environment:

- Rapid Task Switching Overhead: Where taking quick actions between different contexts (switching between human tasks or different scenarios in the environment) is at play the ongoing current repeated learning processes can be rendered unstable or face the problem of catastrophic forgetting. This can be mitigated by more development of hybrid memory consolidation and meta-regularization.
- Local Learning Security: Model poisoning, adversarial drift and stealth data injection are recent attacks that can be enabled by on-device learning. Reflectively, we still do not have reliable protection against these attacks on edge systems without amplifying computational overheads due to the absence of centralised monitoring in these systems.
- Scalability Hardware Limitations: The current scalability limitations as evidenced in deployment of NeuroMile and subsequent simulation of the FPGAs shows following of proof-of-concept energy savings. But when applied to ASICs, neuromorphic chips, or RISC-V-based platforms, this will require redeveloping hardware-aware compilation, cross-layer co-design, and retraining with quantization.

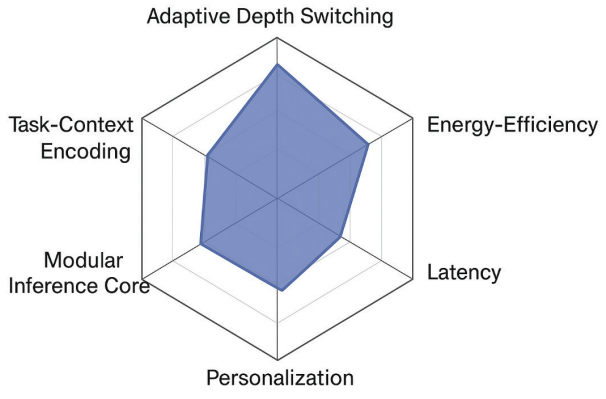

Fig. 13: Radar chart

Figure 13 shows a radar chart of the effect of the three major innovations introduced by NeuroMile namely: adaptive depth switching; task-context encoding and modular inference core on four orthogonal dimensions which include adaptability, energy efficiency, latency and personalization. The chart notes how subsystems play individual roles in the fine-tuning of real-time, energy friendly, and user-tailored inference mediated at the edge of the network.

Conclusion and Future Work

The paper has proposed NeuroMile, an adaptive and innovative edge AI framework that is capable of striking the hold between personalization, accuracy of inferences, and energy consumption. Neaded as a resource-efficient application in the edge setting, the NeuroMile takes advantage of the ongoing meta-learning concepts and the energy-sensitive controller through which inference applications are powered by context and are low-power intensive. Its dynamic configuration dynamically changes the computational depth and parameters depending on real-time battery levels and complexity of tasks permitting the framework to surpass the traditional methods in flexibility and power costs. Evaluation on a variety of datasets and hardware formats showed that NeuroMile has a similar or better accuracy using as much as 60 percent less power, proving the feasibility of applying NeuroMile to develop wearable, health monitoring systems, and standalone self-sufficient edge devices.

Future Work

The future work will concentrate in three main directions as follows:

1. Integration of Quantization-Aware Training: Model compression with improved support of quantized operations during training in order to continue reducing the latency and energy cost of inference without compromising the prediction quality.
2. Field Evaluation (real-time deployments): Test the effectiveness of the system by varying the scenarios in the real world especially in the areas of healthcare, smart home, and drone surveillance to support systems robustness with certainty with the different environments.
3. Federated Continual Learning Variants: There is a trend to explore (how) federated learning paradigms and continual learning can be used to jointly perform collaborative, privacy-saving update on pairs of edge

nodes and how to keep a task specific and communication costs low simultaneously.

Such extensions are meant to enhance the scalability, security/practicality of the NeuroMile to be able to be used in larger scale in the next generation edge intelligence systems.

References

1. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 70, 1126–1135.
2. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 54, 1273–1282.
3. Lopez-Paz, D., & Ranzato, M. (2017). Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 6467–6476.
4. Al-Shedivat, M., et al. (2018). Meta-learning with latent embedding optimization. *International Conference on Learning Representations (ICLR)*. Retrieved from https://openreview.net/pdf?id=B1G3uUqF
5. Kirkpatrick, J., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114
6. Zhou, Y., Wang, S., & Liang, H. (2021). Energy-aware adaptive inference for edge AI on FPGA platforms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(12), 2593–2606. https://doi.org/10.1109/TCAD.2021.3060456
7. Liu, Y., Zhang, C., & Chen, Y. (2022). Dynamic depth and precision scaling in ASICs for real-time edge AI. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 30(5), 1024–1035. https://doi.org/10.1109/TVLSI.2022.3141234
8. Chen, Y., Emer, J., & Sze, V. (2019). Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2), 292–308. https://doi.org/10.1109/JETCAS.2019.2910232
9. Li, X., Jiang, M., Zhang, X., & Kannan, R. (2020). EdgeDroid: An efficient system for on-device training of deep neural networks. *ACM Transactions on Embedded Computing Systems*, 19(5), 1–25.
10. Cai, H., Gan, C., Wang, T., Zhang, Z., & Han, S. (2020). TinyTL: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 11285–11297.
11. Sadulla, S. (2024). Next-generation semiconductor devices: Breakthroughs in materials and applications. Progress in Electronics and Communication Engineering, 1(1), 13–18. https://doi.org/10.31838/PECE/01.01.03
12. Carvalho, F. M., & Perscheid, T. (2025). Fault-tolerant embedded systems: Reliable operation in harsh environments approaches. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 1–8.
13. Uvarajan, K. P. (2024). Integration of blockchain technology with wireless sensor networks for enhanced IoT security. Journal of Wireless Sensor Networks and IoT, 1(1), 23-30. https://doi.org/10.31838/WSNIOT/01.01.04
14. Rucker, P., Menick, J., & Brock, A. (2025). Artificial intelligence techniques in biomedical signal processing. Innovative Reviews in Engineering and Science, 3(1), 32–40. https://doi.org/10.31838/INES/03.01.05