The Fourteenth International Conference on Learning Representations

SNAP-UQ: Self-Supervised Next-Activation Prediction for Single-Pass Uncertainty in TinyML

Ismail Lamaakal, Chaymae Yahyati, Khalid El Makkaoui,
Ibrahim Ouahbi, Yassine Maleh

Multidisciplinary Faculty of Nador, Mohammed First University, Oujda 60000, Morocco
Laboratory LaSTI, ENSAK, Sultan Moulay Slimane University, Khouribga 54000, Morocco

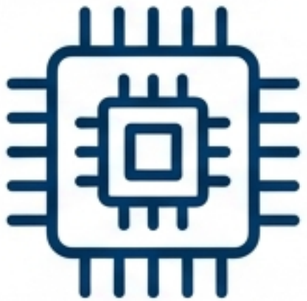
SNAP-UQ: Self-Supervised Next-Activation Prediction for Single-Pass Uncertainty

Reliable, State-Free Monitoring for Edge AI

The Core Definition	The Value Proposition & Impact
SNAP-UQ is a label-free uncertainty estimation mechanism designed specifically for microcontrollers (MCUs). It derives uncertainty not from the final softmax output, but from the network's internal feature dynamics.	■ **Single-Pass**: No ensembles, multiple passes, or Monte Carlo sampling. ■ **State-Free**: Zero temporal buffers; minimizes RAM usage. ■ **Integer-Friendly**: Fully compatible with Int8 quantized inference (CMSIS-NN). ■ **Impact**: Enables autonomous monitoring on milliwatt-scale hardware (Cortex-M4/M7).

Motivation: The TinyML Constraint Conflict

The Hardware Envelope



- **Target Devices:** Low-power MCUs (STM32L4, STM32F7)
- **Memory Budget:** <256KB Flash, <64KB SRAM
- **Compute Budget:** Milliwatt power; Real-time inference

The Safety Requirement

- **Corrupted-in-Distribution (CID):** Sensor noise, gain shifts, lighting changes.
- **Out-of-Distribution (OOD):** Anomalous inputs (unknown objects/speech).

The Conflict: Standard reliability methods (Ensembles/Bayesian) require multiple passes, state buffers, or large footprints.

TinyML needs a metric that is intrinsic to the single forward pass.

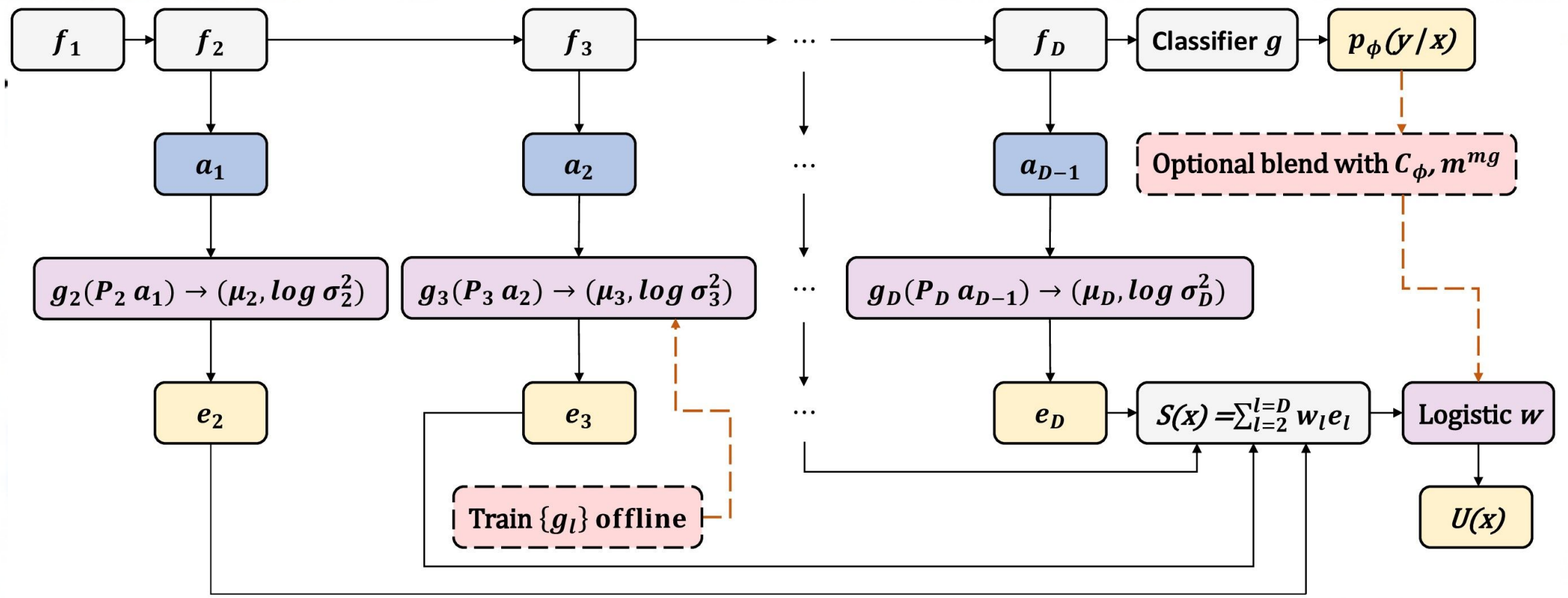
Problem: Limitations of Current Uncertainty Methods on MCUs

Deep Ensembles & MC Dropout	• **Latency Tax:** Requiring M forward passes multiplies latency/energy by factor M. • **Memory Tax:** Ensembles require storing M sets of weights. • **Failure:** Triggers Out-Of-Memory (OOM) errors on Small-MCUs.
Softmax Confidence (The Baseline)	• **"Peaky" Behavior**: Deep networks are overconfident on OOD samples in high-density regions. • **Late Detection**: Captures errors only at the output. Fails to detect internal feature distortions early.
Early-Exit Ensembles	• **Overhead**: Branching architectures and control logic add 10-25% overhead. • **Complexity**: Hard to optimize for linear instruction streams.

Architecture: The SNAP-UQ Pipeline

Concept: The "Tap" System

SNAP-UQ attaches lightweight side-branches to a frozen backbone at specific depths (mid-block and penultimate).



1. **Tap:** Extract activation $a_{\ell-1}$.
2. **Project (P_ℓ):** Compress using 1x1 conv + GAP to low-rank summary z_ℓ.
3. **Predict (g_ℓ):** Tiny MLP predicts statistics of the *next* activation.
4. **Compare:** Calculate mismatch between predicted and actual a_ℓ.

Methodology: Next-Activation Prediction

The Core Hypothesis: If a network "understands" an input, the transformation from layer $\ell - 1$ to ℓ is predictable.

- **In-Distribution (ID):** Features evolve smoothly.
- **Corrupted/OOD:** Internal evolution is disrupted. Actual a_ℓ deviates from the prediction.

Self-Supervised Supervision:

- **Signal:** The network's *own* future activation.
- **Query:** "Given features at $\ell - 1$, what should ℓ look like?"

The Predictor Heads (g_ℓ):

- **Input:** Low-rank projection z_ℓ.
- **Output:** Mean (μ_ℓ) and Log-Variance ($\log \sigma_\ell^2$).
- **Cost:** Approx. 2% of backbone FLOPs.

Math I: Conditional Density Modeling

$$p_\theta(a_\ell | a_{\ell-1}) = \mathcal{N}(\mu_\ell, \mathrm{diag}(\sigma_\ell^2))$$

Components

- $\mu_\ell = g_\mu(P_\ell a_{\ell-1})$: The expected value of the next activation.
- $\sigma_\ell^2 = \mathrm{softplus}(g_\sigma(P_\ell a_{\ell-1})) + \epsilon$: The expected variance (uncertainty).

Why Diagonal Gaussian?

- **Efficiency**: Full covariance requires $O(d^2)$. Diagonal requires $O(d)$.
- **Independence**: Sufficient for detecting feature disruptions without expensive matrix ops on MCUs.

Math II: Surprisal Calculation

Measuring the “Surprisal” (Negative Log-Likelihood) of the actual activation.

$$u_\ell(x) = (a_\ell - \mu_\ell) \odot \sigma_\ell^{-1}$$

Formula of Quadratic Energy (Surprisal):

$$e_\ell(x) = \|u_\ell(x)\|_2^2 = \sum_{i=1}^{d_\ell} \left(\frac{a_{\ell,i} - \mu_{\ell,i}}{\sigma_{\ell,i}} \right)^2$$

Mean Mismatch: Feature value isn’t what was predicted.

Scale Normalization:
Penalize errors in stable channels; be lenient in noisy ones.

Result: $e_\ell(x)$ spikes when the feature transition is anomalous relative to the training distribution.

Math III: Aggregation and Uncertainty Mapping

Step 1.
Dimension Normalization

$$\bar{e}_\ell(x) = \frac{1}{d_\ell} e_\ell(x)$$

Normalize energy by layer dimension d_ℓ for comparison.

Step 2.
Aggregation

The SNAP score $S(x)$ is a weighted sum across tapped layers $\mathcal{S}$:

$$S(x) = \sum_{\ell \in \mathcal{S}} w_\ell \bar{e}_\ell(x)$$

Where w_ℓ are layer-specific weights for combining scores.

Step 3.
Uncertainty Mapping ($S(x) \to U(x)$)

Mapping Recipe

- **Logistic:** $U(x) = \sigma(\beta_0 + \beta_1 S(x) + \beta_2 m(x))$ (Recommended for in-distribution calibration; uses sigmoid function σ.)
- **Isotonic Regression:** Non-parametric monotone mapping for strict calibration. (Recommended for fixed coverage monitoring; preserves ordering.)
- **Optional:** $m(x)$ blends classifier confidence (C_ϕ) for added separability. (e.g., top 2 logit margin or max softmax. When m is unavailable, set $\beta_2 = 0$.)

Theory I: Relation to Depth-wise NLL

Proposition 2.1 (Surprisal–Likelihood Equivalence)

Under the diagonal-Gaussian model, the negative log-likelihood of activation a_ℓ given $a_{\ell-1}$ is:

$$-\log p_\theta(a_\ell|a_{\ell-1}) = \frac{1}{2}e_\ell(x) + \frac{1}{2}\sum_{i=1}^{d_\ell}\log\sigma_{\ell,i}^2 + \text{const}$$

Implications:

- **Equivalence:** Minimizing the SNAP-UQ objective ≈ Maximizing likelihood of feature transitions.
- **Affine Transform:** The Score $S(x)$ is an affine transformation of the depth-wise NLL.
- **Inference:** High $S(x)$ strictly implies a lower conditional likelihood of the observed activations.

Theory II: Relation to Conditional Mahalanobis

Proposition 2.2

If feature dynamics are approximately linear ($\mathbf{a}_\ell \approx \mathbf{W}_\ell \mathbf{a}_{\ell-1} + \mathbf{b}_\ell$), then $e_\ell(\mathbf{x})$ is the squared Mahalanobis distance to the conditional mean:

$$e_\ell(\mathbf{x}) = \mathrm{MD}^2_{\Sigma^2_\ell}(\mathbf{a}_\ell, \mathbf{W}_\ell \mathbf{a}_{\ell-1} + \mathbf{b}_\ell)$$

Class-wise Mahalanobis (State-of-the-Art Baseline)	SNAP-UQ (Conditional)
• Compares $\mathbf{a}_\ell$ to static class centroid $\bar{\mu}_c$. • Fails if features shift but remain near centroids.	• Compares $\mathbf{a}_\ell$ to dynamic sample-specific prediction. • Sensitive to shifts that break layer-to-layer mapping.

Affine Invariance: Score is robust to Batch-Norm-like per-channel rescaling.

Implementation: TinyML Optimization

Int8 Quantized Weights

CMSIS-NN Kernels (Int32 Accumulators)

LUT Approximation

- Use small 256-entry Look-Up Table for $\exp\left(-\frac{1}{2}\log\sigma^2\right)$.
- Avoids expensive Floating Point math.

Footprint

- **Flash:** +10-30 KB (weights for tiny heads).
- **RAM:** Zero temporal buffering. Stream-based computation.

Experimental Setup

Hardware

- **Small-MCU:** STM32L4 (*Computer Modern*, Cortex-M4, 256KB Flash, 64KB RAM)
- **Big-MCU:** STM32F7 (*Computer Modern*, Cortex-M7, 512KB Flash, 216 MHz)

Backbones & Data

- **Vision:** ResNet-8 on **CIFAR-10 / TinyImageNet**
- **Audio:** DS-CNN on **SpeechCommands**

Baselines

- **Single-Pass:** Softmax Entropy, Max-Prob, Class-wise Mahalanobis
- **Multi-Pass:** MC Dropout (5x), Deep Ensembles (3-5x)
- **Early-Exit:** EE-Ensembles

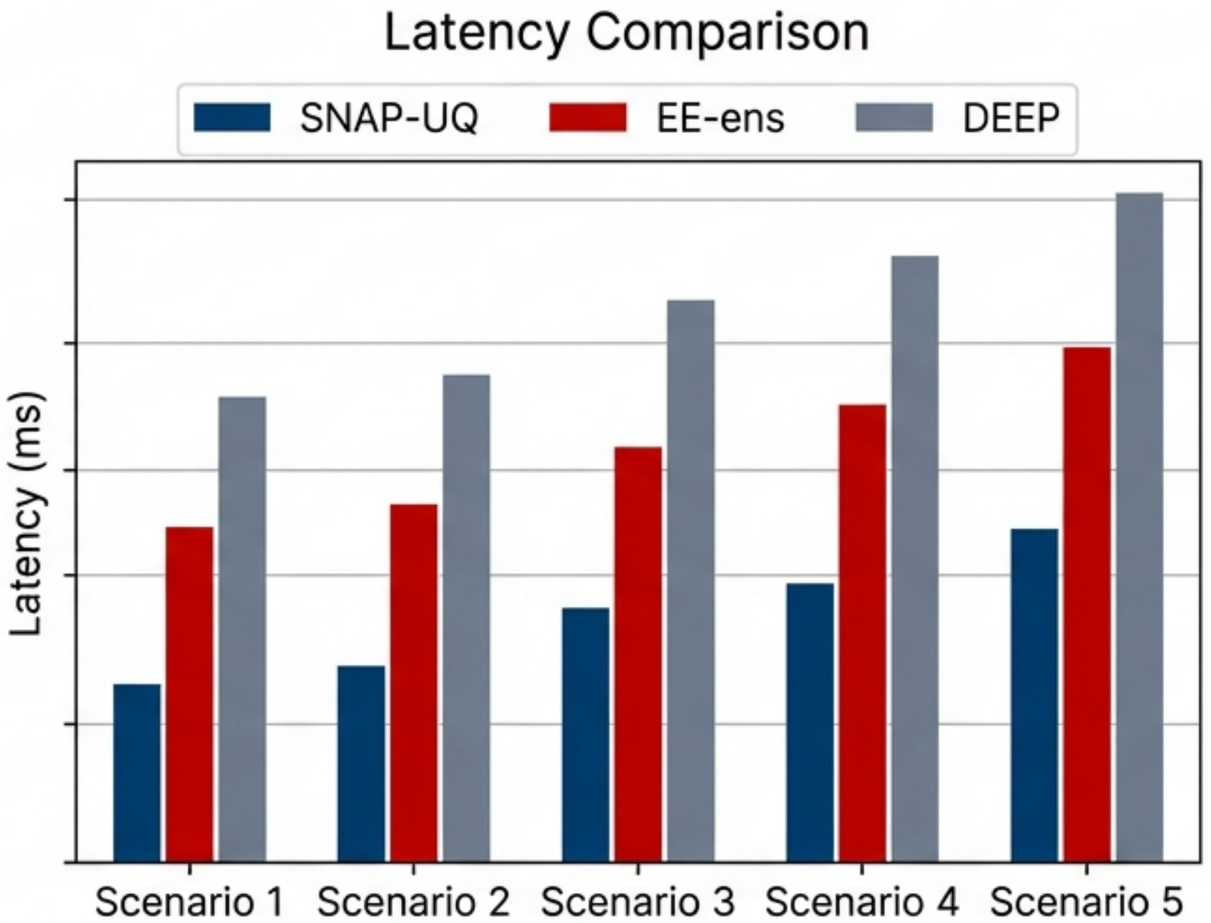
Results: Efficiency and Deployability

Flash & Memory:

- **Small-MCU**: Deep Ensembles & MC Dropout trigger **OOM (Out-of-Memory)**.
- SNAP-UQ fits comfortably (40-60% smaller than Ensembles).

Latency:

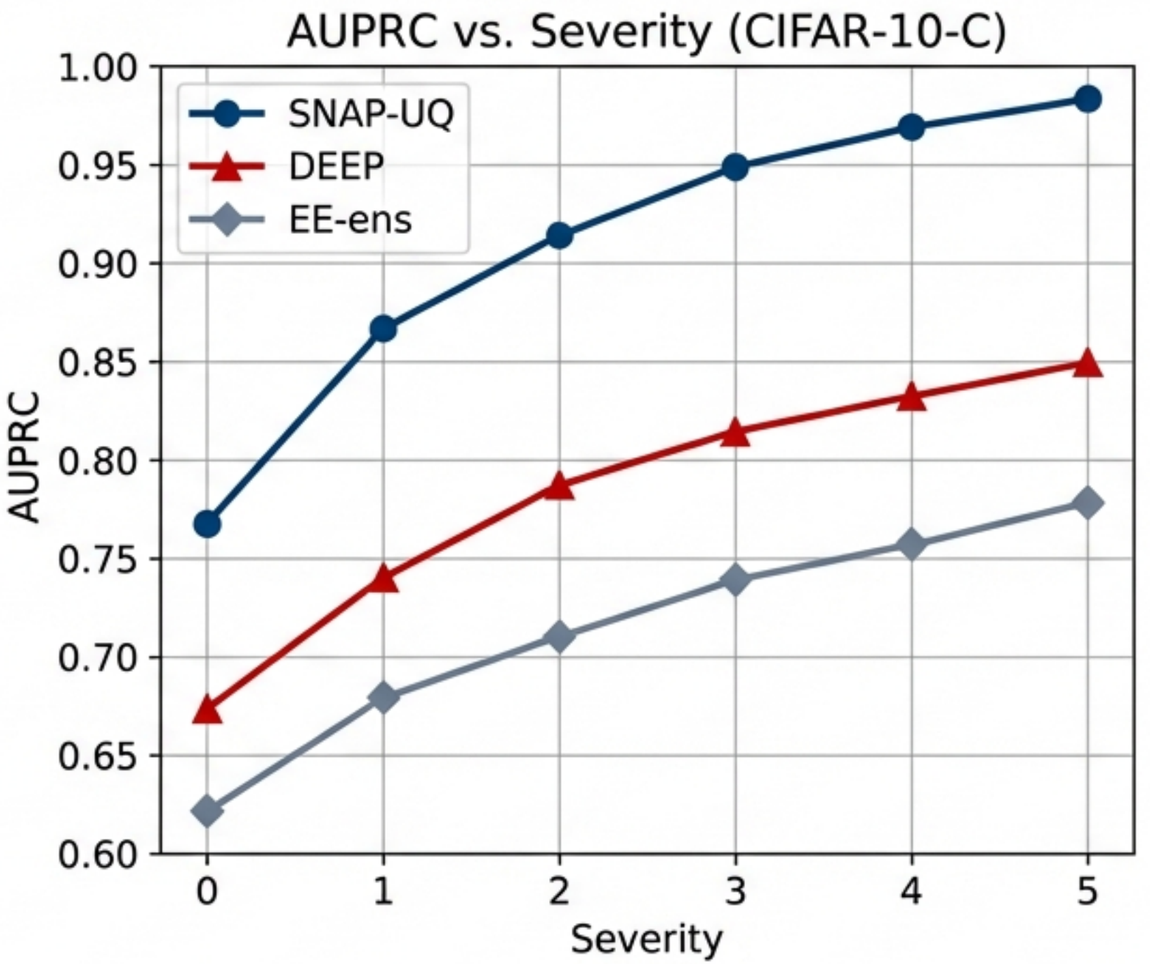
- **Speedup:** 25–35% faster than Early-Exit Ensembles.
- **Overhead:** <10% added to vanilla backbone.



Results: Robustness on Corrupted Streams

Metric: AUPRC (Area Under Precision-Recall Curve) on corrupted streams (CID).

- **Severity Scaling:** SNAP-UQ performance scales best as corruption severity increases.
- **Detection Delay:** Detects failure events **15-20 frames earlier** than Softmax Entropy.
- **Comparison:** Outperforms Deep Ensembles on failure detection despite single-pass constraint.



Conclusion: Certainty in Constraints

Key Contribution:

SNAP-UQ provides the first viable path for reliable, uncertainty-aware monitoring on ultra-constrained hardware (Cortex-M4).

The Paradigm Shift:

From: **Output Confidence** (Softmax) or **External Checks** (Ensembles)

To: **Internal Consistency** (Next-Activation Prediction)

Summary:

- **State-Free**: No buffers.
- **Single-Pass**: Minimal latency.
- **Label-Free**: Detects anomalies without online supervision.

Enabling autonomous edge devices to know *when they don't know*.

Thank You

- Acknowledgments
- Code: https://github.com/Ism-ail11/SNAP-UQ

ICLR 2026