Poster: CATS: Distributed Transformer Inference on Ultra-Low-Power Wireless Devices

Alexander Gräfe
RWTH Aachen University
Aachen, Germany
alexander.graefe@dsme.rwth-aachen.de

Ding Huo
TU Darmstadt
Darmstadt, Germany
ding.huo@tu-darmstadt.de

Vincent de Bakker
RWTH Aachen University
Aachen, Germany
vincent.debakker@dbmc.de

Johannes Berger
RWTH Aachen University
Aachen, Germany
johannes.berger@dsme.rwth-aachen.de

Marco Zimmerling
TU Darmstadt
Darmstadt, Germany
marco.zimmerling@tu-darmstadt.de

Sebastian Trimpe
RWTH Aachen University
Aachen, Germany
trimpe@dsme.rwth-aachen.de

Abstract

Transformer models can capture complex dependencies in IoT sensing data, but individual ultra-low-power wireless devices lack the compute and memory to run them. Distributed inference can pool these resources by partitioning transformer layers across devices, but doing so requires devices to exchange intermediate results wirelessly, introducing communication overhead and exposure to message loss. Building on our IJCAI 2026 paper [2], we present Collaborative Inference at the Sensor-level (CATS), a framework for distributed transformer inference on ultra-low-power wireless devices. To our knowledge, its implementation is the first end-to-end hardware deployment of distributed transformer inference on such devices. CATS combines model partitioning with two complementary techniques: SomeGather exchanges only selected activation columns, while message-dropout trains the model to tolerate message loss. On real hardware, CATS runs a 14M-parameter Vision Transformer across 16 nRF52840 devices—more than 14× the size of the largest model that fits on a single device. In simulations on four time-series benchmarks, SomeGather reduces communication volume by up to 90% while changing mean MSE by less than 1%. Analytical results further show a 67.5% reduction in per-device activation RAM. Message-dropout further improves robustness to message loss.

CCS Concepts

• **Computer systems organization** → **Embedded systems**; *Distributed architectures*; • **Networks** → *Wireless mesh networks*; • **Computing methodologies** → Neural networks.

Keywords

distributed transformer inference, TinyML, ultra-low-power wireless devices, communication-aware pruning,
message-loss robustness

1 Motivation

Transformer models are important for many IoT sensing and prediction applications. Running these models, however, requires substantial computation, flash memory for model weights, and RAM for intermediate activations. These demands often exceed the resources available on typical ultra-low-power IoT devices. Distributed transformer inference addresses this mismatch by pooling resources across multiple IoT devices, allowing them to execute a model that would not fit on any single device.

Doing so requires partitioning the model across devices. Because each device computes only part of a layer, the devices must exchange intermediate activations. Over low-power wireless links, this exchange can dominate latency, increase the RAM needed to buffer activations, and make prediction accuracy vulnerable to message loss.

Existing distributed transformer inference approaches address only parts of this problem. Token-splitting methods such as Voltage [4] reduce activation memory and computation but still require every device to store the full model. Feature-splitting methods such as SiracusaDTI [1] reduce per-device weight storage but incur high RAM and communication costs when using AllReduce-style collectives over wireless meshes. Neither class of approach fully meets the constraints of ultra-low-power wireless devices. CATS [2] closes this gap by co-designing partitioning, sparse activation exchange, and message-loss-aware training. Accordingly, this poster emphasizes the aspects of CATS most relevant to embedded wireless systems: deployment on a physical BLE mesh, communication efficiency, and robustness to message loss.

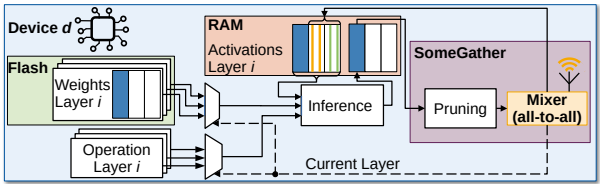

Figure 1: One CATS round [2]. Each device processes its local shards of weights and activations; SomeGather then uses Mixer [3] to broadcast only selected activation columns, while the remaining columns stay local.

2 CATS: Co-Designing Partitioning, Communication, and Training

A transformer layer represents tokens as feature vectors processed by multiple attention heads. CATS partitions both the feature dimension and the attention heads across devices (Figure 1). Each device stores only its assigned weights and computes the corresponding feature slice locally, reducing per-device flash use and distributing computation across the network.

Feature-mixing projections require activation columns from other devices. Exchanging every column would incur substantial network traffic and require more RAM for buffering. SomeGather reduces this overhead by using pruning to select only a subset of columns for broadcast through Mixer's all-to-all primitive [3]. CATS retrains the resulting pruned model to preserve accuracy.

Example. Suppose two devices hold activation columns $\{a_1, a_2\}$ and $\{a_3, a_4\}$. At 50% pruning, each device broadcasts one selected column; with AllGather, it would broadcast both. The unselected columns remain local, and the corresponding cross-device connections are pruned.

Message-dropout improves robustness by simulating three message-loss patterns during training: a sender's message is lost at one receiver; a receiver loses all remote messages; or all other receivers lose a particular sender's message. New message-dropout masks are sampled at each training step.

3 Key Results

Analytical scaling shows that adding devices relaxes per-device flash constraints and supports larger feature dimensions F, whereas sequence length N remains limited by activation storage and exchange [2]. We measure block latency on hardware (Figure 2) and evaluate prediction MSE and robustness to message loss in simulation over ten random seeds. The hardware measurements show that communication can dominate latency once computation is distributed.

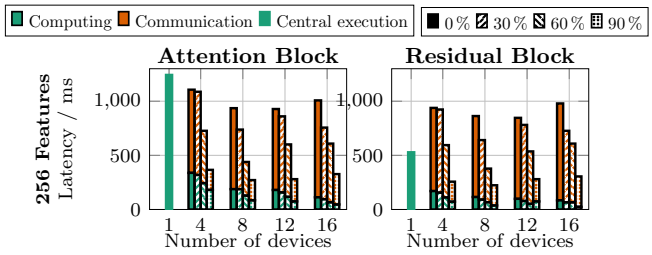

Figure 2: Per-block hardware latency for 256-feature attention and residual blocks on nRF52840 devices [2]. Hatching indicates SomeGather pruning.

Analytical resource results. By transmitting only selected activation columns, SomeGather reduces communication volume by up to 90% and per-device activation RAM by 67.5% relative to unpruned distributed inference.

Simulation results. Across four time-series benchmarks, SomeGather reduces communication by 90% while changing mean MSE by less than 1% relative to unpruned inference. Figure 3 shows representative trade-offs for Traffic and London-smart-meters. On Traffic, SomeGather achieves a mean MSE of 0.1645 at 90% savings, close to the unpruned value of 0.1651; conventional pruning reaches 0.2486 at 87.5% savings.

Under 10% simulated message loss, message-dropout limits the relative increase in prediction error to at most 23.6%, compared with up to 200% without it.

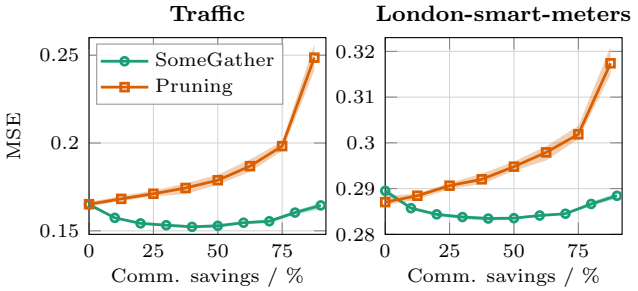

Figure 3: Communication–accuracy trade-offs evaluated in simulation on Traffic and London-smart-meters [2]. Curves show mean MSE, and shaded bands span the minimum and maximum across ten random seeds (lower is better).

Hardware deployment. CATS runs a 14M-parameter Vision Transformer with 8-bit CMSIS-NN kernels across 16 nRF52840 devices (64 MHz Cortex-M4, 1 MB flash, 256 KB RAM) in a 35 m Mixer/BLE testbed with a network diameter of at least two hops. The current one-head-per-device mapping limits the deployment size to the number of attention heads. At larger scales, communication is likely to be the main bottleneck: it already accounts for 69.4–91.4% of unpruned block latency.

4 Conclusion

CATS combines model partitioning, sparse activation exchange, and message-loss-aware training to execute transformer models that exceed the capacity of a single ultra-low-power wireless device. Together, the results demonstrate feasibility and identify wireless communication as the primary bottleneck to larger deployments.

Acknowledgments

This work was supported by the DFG within the priority program 1914 (grant TR 1433/2) and within the Emmy Noether project NextIoT (ZI 1635/2-1), and by the LOEWE initiative (Hesse, Germany) within the emergenCITY center (LOEWE/1/12/519/03/05.001(0016)/72). The authors gratefully acknowledge the computing time provided to them at the NHR Center NHR4CES at RWTH Aachen University (p0021919).

References

[1] S. Bochem, V. J. B. Jung, A. S. Prasad, F. Conti, and L. Benini. 2025. Distributed Inference with Minimal Off-Chip Traffic for Transformers on Low-Power MCUs. In *Design, Automation & Test in Europe Conference (DATE)*. IEEE.

[2] A. Gräfe, D. Huo, V. de Bakker, J. Berger, M. Zimmerling, and S. Trimpe. 2026. Going Beyond the Edge: Distributed Inference of Transformer Models on Ultra-Low-Power Wireless Devices. In *Proceedings of the 35th International Joint Conference on Artificial Intelligence (IJCAI)*. Accepted for publication. arXiv:2605.15694. doi:10.48550/arXiv.2605.15694

[3] C. Herrmann, F. Mager, and M. Zimmerling. 2018. Mixer: Efficient Many-to-All Broadcast in Dynamic Wireless Mesh Networks. In *16th ACM Conference on Embedded Networked Sensor Systems*. ACM.

[4] C. Hu and B. Li. 2024. When the Edge Meets Transformers: Distributed Inference with Transformer Models. In *44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE.