

Poster: Road to Tiny Reality: Digital Twins for Decentralized AI on Microcontrollers

Navidreza Asadi*
Technical University Munich
Munich, Germany

Halil Ibrahim Bengü*
Technical University of Munich
Munich, Germany

Lars Wulfert*
Fraunhofer IMS
Duisburg, Germany

Hendrik Wöhrle
University of Duisburg-Essen
Duisburg, Germany

Wolfgang Kellerer
Technical University Munich
Munich, Germany

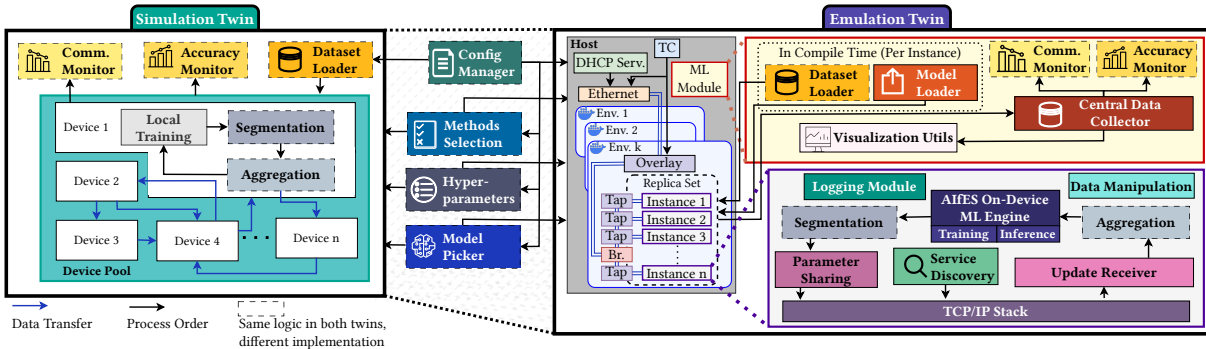


Figure 1: Overview of the digital twin methodology for developing and validating Decentralized FL (DFL) algorithms. Stage 1 (Simulation) enables rapid prototyping and algorithmic exploration, while Stage 2 (Emulation) provides hardware-aware validation with realistic network conditions and resource constraints.

Abstract

This work presents a two-stage digital twin methodology for developing and validating DFL algorithms on resource-constrained microcontrollers. The first stage, our simulation-based twin, enables rapid prototyping and algorithm exploration without hardware constraints, while the second stage, based on leveraging several hardware emulation instances in a containerized environment, provides hardware-aware validation under realistic conditions including network delays, resource limitations, and communication protocols. This approach bridges the critical gap between research and deployment, enabling performance analysis at a pace impractical

with physical hardware alone. We demonstrate how this digital twin pipeline is essential for robust Machine Learning Operations (MLOps) in IoT environments, allowing for scalable, cost-effective testing of decentralized tiny ML. Our results across simulation, emulation, and a cluster of real ESP32-S3 microcontrollers show that our twins faithfully reproduce physical device behavior, making it a valuable framework for advancing tiny, decentralized AI.

CCS Concepts

• Computing methodologies → Distributed artificial intelligence; Distributed computing methodologies; • Applied computing → Microcomputers; • Computer systems organization → Embedded systems.

Keywords

Digital Twin, Decentralized Federated Learning, TinyML

ACM Reference Format:

Navidreza Asadi, Halil Ibrahim Bengü, Lars Wulfert, Hendrik Wöhrle, and Wolfgang Kellerer. 2025. Poster: Road to Tiny Reality: Digital Twins for Decentralized AI on Microcontrollers. In *The 31st Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '25)*, November 4–8, 2025, Hong Kong, China. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3680207.3765668>

*Equal contribution

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM MOBICOM '25, Hong Kong, China

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1129-9/2025/11

<https://doi.org/10.1145/3680207.3765668>

1 Introduction

The proliferation of tiny Internet of Things (IoT) devices has created unprecedented opportunities for distributed intelligence. However, developing and validating robust DFL algorithms for these resource-constrained microcontrollers presents significant challenges.

Motivation. Traditional federated learning approaches face several barriers when deployed on capable IoT devices [2], let alone constrained microcontrollers. DFL eliminates the need for central servers, enabling direct model sharing among devices. This not only addresses privacy concerns but also reduces single points of failure, making it a promising paradigm for decentralized intelligence on tiny devices [3, 4, 7].

Challenge. Unlike centralized federated learning that relies on powerful edge or central servers, DFL on microcontrollers operates under severe constraints: extreme memory limitations (typically <1MB RAM), limited computational capacity, and unreliable network connectivity [8]. Real-world IoT deployments suffer from intermittent connectivity, frequent hardware failures, and unpredictable network conditions [5], making algorithm validation particularly challenging.

Problem Statement. Existing evaluation methodologies fall short of addressing these challenges. Pure simulation, while fast and scalable, abstracts away hardware constraints and real-world network behaviors. Physical testbeds provide measurement results but are limited in scale, expensive to deploy, and difficult for systematic experimentation. This methodological gap hinders the development of DFL for tiny devices.

Contribution. We propose *Zwe*³, a systematic two-stage digital twin methodology that combines the speed of simulation with the fidelity of hardware emulation¹. This approach provides a complete development and evaluation pipeline for tiny DFL systems, enabling developers to progress from algorithm design through hardware-aware validation to deployment-ready systems. Our methodology enables MLOps practices for tiny IoT, facilitating faster development cycles while ensuring thorough algorithm testing before physical deployment.

2 Two-Stage Digital Twin Methodology

Our methodology comprises two digital twins: one based on simulation and another on hardware emulation (Fig. 1). This approach provides a systematic pathway from algorithm conception to deployment-ready systems through two complementary stages, each serving a distinct but interconnected purpose in the development pipeline.

► Stage 1: Rapid Prototyping with Simulation.

The first stage of *Zwe*³ is a simulation-based digital twin that enables rapid algorithm development and exploration without hardware constraints. This stage allows researchers to quickly

¹*Zwe*³ is an abbreviation for “Zwei Zwergzwillinge”, combining (1) “Zwei” (two), (2) “Zwerg” (tiny) and (3) “Zwillinge” (twins).

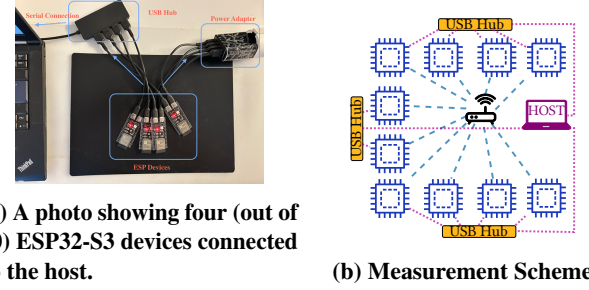


Figure 2: Physical testbed setup.

iterate on new DFL approaches, test different configurations, and optimize hyperparameters across diverse data distributions. Our simulation stage uses a Python-based environment built around PyTorch and NumPy, helping developers create and evaluate their algorithms without the challenges of low-level coding and hardware limitations. This stage enables:

- ❶ *Algorithmic Exploration*: Rapid iteration on new DFL approaches, including training algorithms, segmentation strategies, and aggregation methods. Researchers and developers can test different configurations in hours rather than days.
- ❷ *Hyperparameter Optimization*: Systematic tuning of learning rates, batch sizes, local epochs, and communication frequencies across diverse data distributions.
- ❸ *Scalability Analysis*: Testing with hundreds or thousands of virtual devices to understand algorithmic behavior at scale, identifying potential bottlenecks before hardware implementation.
- ❹ *Statistical Validation*: Multiple independent runs enable robust statistical analysis of algorithm performance, providing confidence intervals and significance testing.

► Stage 2: High-Fidelity Hardware Emulation.

Our emulation digital twin creates networks of virtual microcontrollers running firmware identical to that on physical devices. This provides:

- ❶ *Hardware-Realistic Constraints*: Accurate memory limitations, processing delays, and timing behaviors that match physical microcontrollers. Each emulated device shares the same microarchitecture and operates under the same resource constraints as real hardware.
- ❷ *Networking-in-the-Loop*: Full TCP/IP stack implementation with realistic network delays, packet loss, and bandwidth limitations. Devices communicate through virtual network interfaces that approximate real conditions. Furthermore, the traffic control (TC) component available in the Linux kernel helps emulate different networking scenarios for each instance separately.
- ❸ *Firmware Validation*: Identical C programs run in both emulation and on physical hardware, ensuring software compatibility and eliminating abstraction gaps.
- ❹ *Systematic Experimentation*: Controlled network conditions enable reproducible experiments while maintaining hardware fidelity, supporting performance evaluation.
- ★ *Scalability*: The containerized environments not only facilitate MLOps but also enable vertical (resource) and horizontal (instance

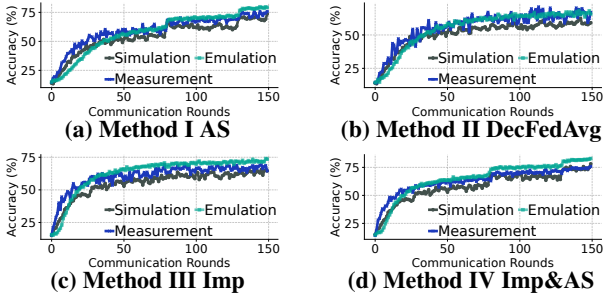


Figure 3: Comparison of convergence through time for our two digital twins and real measurements on ten devices.

count) scaling. This allows developers to evaluate their methods at a scale not easily achievable with real hardware. The ability to emulate cross-silo FL using separate environments is an added benefit. Our emulation environments run inside a host machine. We implemented our second twin in C, including components with shared logic such as the Segmentation, Monitoring, and Aggregation modules. We use AIFES [8] for on-device training and inference.

► Final Stage: Physical Hardware Validation.

Our physical testbed consists of up to 10 ESP32-S3 microcontrollers (Fig. 2) running identical firmware to the emulated environment. The final stage acts as a validation step to ensure that emulation accurately represents real-world behavior and provides deployment confidence.

3 Results

Early Outcome. With the help of our *Zwe*³ methodology, we could quickly iterate over different ideas and algorithms. This led us to develop an improved segmentation algorithm that shares sparse parts (segments) of the model in each aggregation round based on their importance [1]. It outperforms previous methods, not only on the digital twins but also in real measurements. This process would have been significantly more difficult and time-consuming without the twins.

Fidelity: Simulation vs. Emulation vs. Real Measurement.

We compare convergence over time on our two digital twins and real measurements over 150 communication rounds. We implemented four different DFL methods: Decentralized Federated Averaging (DecFedAvg) [6], AdaStair (AS) [7], our importance-based method (Imp), and its combination with AS (Imp&AS). We use ten microcontrollers and a deep neural network model adapted from [7]. Figure 3 shows the comparison across the four methods. While both twins follow the same trend as the real devices, the simulation twin sometimes undershoots the accuracy. The emulation twin, on the other hand, closely resembles the results from physical hardware.

Scalability Analysis. To answer the question of how the number of participating devices affects convergence, we ran DFL on 10, 100, and 200 emulated microcontrollers. Performing such a large-scale experiment on physical hardware would

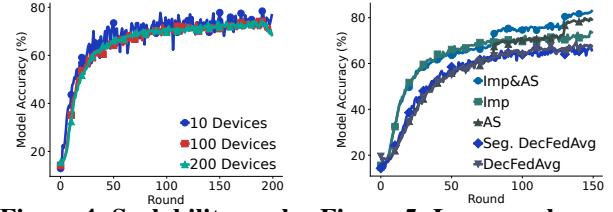


Figure 4: Scalability analysis. Figure 5: Large-scale emulation (200 devices) comparing different methods.

be impractical and time-consuming, demonstrating the utility of our twins. Figure 4 depicts the results for our importance-based method on the emulation twin. As anticipated, a larger number of devices leads to smaller fluctuations.

Performance Analysis at Scale. We compare the DecFedAvg, AS, Imp, and AS&Imp DFL methods on 200 participating microcontrollers to inspect their behavior at a larger scale (Fig. 5). This again highlights the effectiveness of *Zwe*³ for scalable and robust evaluation.

References

- [1] Navidreza Asadi, Halil Ibrahim Bengü, Lars Wulfert, Hendrik Wöhrle, and Wolfgang Kellerer. 2025. Gist - Optimizing Segmentation for Decentralized Federated Learning on Tiny Devices. In *Federated Learning and Edge AI for Privacy and Mobility (FLEdge-AI '25)* (Hong Kong, China) (FLEdge-AI '25). Association for Computing Machinery, New York, NY, USA. doi:10.1145/3737899.3768527
- [2] Yunming Liao, Yang Xu, Hongli Xu, Min Chen, Lun Wang, and Chunming Qiao. 2024. Asynchronous decentralized federated learning for heterogeneous devices. *IEEE/ACM Transactions on Networking* (2024).
- [3] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Jérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celadrán. 2023. Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges. *IEEE Communications Surveys and Tutorials* 25, 4 (2023), 2983–3013.
- [4] Minh K Quan, Pubudu N Pathirana, Mayuri Wijayasundara, Sujeewa Setunge, Dinh C Nguyen, Christopher G Brinton, David J Love, and H Vincent Poor. 2025. Federated learning for cyber physical systems: a comprehensive survey. *IEEE Communications Surveys & Tutorials* (2025).
- [5] Stefano Savazzi, Monica Nicoli, and Vittorio Rampa. 2020. Federated Learning With Cooperating Devices: A Consensus Approach for Massive IoT Networks. *IEEE Internet of Things Journal* 7, 5 (May 2020), 4641–4654. doi:10.1109/jiot.2020.2964162
- [6] Tao Sun, Dongsheng Li, and Bao Wang. 2023. Decentralized Federated Averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2023), 4289–4301. doi:10.1109/TPAMI.2022.3196503
- [7] Lars Wulfert, Navidreza Asadi, Wen-Yu Chung, Christian Wiede, and Anton Grabmaier. 2023. Adaptive Decentralized Federated Gossip Learning for Resource-Constrained IoT Devices. In *Proceedings of the 4th International Workshop on Distributed Machine Learning (Paris, France) (DistributedML '23)*. Association for Computing Machinery, New York, NY, USA, 27–33. doi:10.1145/3630048.3630181
- [8] Lars Wulfert, Johannes Kühnel, Lukas Krupp, Justus Viga, Christian Wiede, Pierre Gembaczka, and Anton Grabmaier. 2024. AIFES: A Next-Generation Edge AI Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 6 (2024), 4519–4533.