

# An IaC-based Large-scale BACnet Emulation Platform

Yuki Hatanaka <sup>1,\*</sup> and Tetsuya Yokotani <sup>2</sup>

<sup>1</sup> Graduate School of Kanazawa Institute of Technology, Nonoichi, Japan

<sup>2</sup> Department of Electronics and Information Systems Engineering, Kanazawa Institute of Technology, Nonoichi, Japan  
Email: c7400034@st.kanazawa-it.ac.jp (Y.H.); yokotani@neptune.kanazawa-it.ac.jp (T.Y.)

\*Corresponding author

Manuscript received May 12, 2025; accepted August 3, 2025; published October 12, 2026

**Abstract**—Integrating Building Automation Systems (BAS) into Cyber-Physical Systems (CPS) introduces critical challenges in scalability: as network scale grows, traffic characteristics shift and resource consumption increases in ways that are difficult to anticipate. Existing BACnet validation tools, however, focus primarily on single-device conformance testing and lack the means to automatically construct large-scale, realistic network environments with reproducibility. This paper proposes a large-scale BACnet emulation platform that integrates hypervisor-level input emulation with Infrastructure as Code (IaC). By injecting HID events through the QEMU/libvirt stack, the platform enables control of unmodified application binaries without any dependency on guest OS type or security policy. Declarative configuration via Ansible allows network topology and node configuration to be defined as code and instantiated on demand, ensuring reproducible test scenarios across heterogeneous environments. The proposed platform facilitates traffic analysis in configurations where real and virtual BACnet applications coexist, and contributes to seamless CPS integration that preserves and leverages existing BACnet infrastructure.

**Keywords**—BACnet, infrastructure as code, network emulation, cyber-physical systems, scalability, QEMU/libvirt

## I. INTRODUCTION

Building Automation System (BAS) is a system architecture for managing and operating the diverse range of devices installed in modern buildings. Its standard protocol, BACnet, provides a unified communication interface in which all building components are represented as devices, enabling interoperability across complex, multi-vendor building environments [1].

Given that building infrastructure components are typically operated for several decades, there is a growing demand for advanced management platforms capable of supporting future integration with Cyber-Physical Systems (CPS) [2]. A critical barrier to deploying such next-generation platforms, however, is the absence of established methods for accurately reproducing and validating large-scale, multi-vendor operational environments prior to deployment [3].

Existing validation tools, including those distributed through the BACnet Testing Laboratories [4], are designed primarily to verify single-device protocol conformance and are not intended to evaluate the system-wide behavior of large-scale heterogeneous deployments. Realistic validation requires an environment capable of simultaneously emulating a large number of devices with diverse profiles and flexibly composing varied test scenarios. Manual construction of such environments using conventional simulators is, however, impractical at the scale of hundreds of heterogeneous nodes—representing a significant obstacle to ensuring stable operation after deployment.

To fundamentally address these limitations, this paper

proposes a BACnet emulation platform based on Linux and QEMU/libvirt [5]. The proposed platform leverages virtualization to execute real-world application binaries without modification, enabling faithful reproduction of behaviors specific to multi-vendor environments. This work extends the authors' prior work [6], which constructed a large-scale validation environment based on Windows OS and SoftEther [7], by addressing its limitations in reproducibility and vendor dependency through a Linux-based virtualization approach.

The remainder of this paper is organized as follows. Section II reviews the challenges of operational automation in BACnet-based systems and existing approaches. Section III presents the architecture of the proposed BACnet emulation platform, designed to support large-scale, multi-vendor validation. Section IV discusses the effectiveness of the proposed platform through qualitative comparison with the authors' prior work. Section V presents applicable use case scenarios, and Section VI concludes the paper.

## II. CHALLENGES IN BACNET-CPS INTEGRATION AND EXISTING APPROACHES

Next-generation BAS are expected to support sophisticated operational models aligned with the principles of Industry 5.0, including sustainability and human-centric automation [8]. In particular, the realization of digital twins—virtual replicas of physical building infrastructure—and broader CPS integration has become essential for enabling energy optimization and real-time threat analysis [9].

To support these advanced operational paradigms, there is growing interest in applying cloud-native methodologies such as DevOps and Infrastructure as Code (IaC) to the building automation domain [10]. However, applying these approaches to existing BACnet deployments reveals two fundamental technical barriers that cannot be resolved by conventional protocol verification tools.

### A. The Emulation Barrier in Heterogeneous Environments

Real-world building environments consist of multi-vendor configurations in which applications are tightly coupled to specific legacy operating systems, proprietary kernels, or closed binary runtimes. Conventional logic simulators, which emulate only the communication protocol layer, are unable to anticipate runtime errors, library conflicts, or resource contention that arise at the OS level in such environments.

Achieving high-fidelity digital twins therefore requires an emulation platform that can execute unmodified binaries and uniformly control isolated legacy systems through a low-level, vendor-agnostic interface—such as hypervisor-level input injection (e.g., keyboard and mouse event emulation).

### B. The Reproducibility Barrier in Large-Scale Validation

CPS integration presupposes large-scale networks comprising hundreds to thousands of interconnected nodes, which causes the complexity of environment construction and management to grow significantly. Manual setup of such large-scale testbeds is not only labor-intensive but also prone to inter-node configuration inconsistencies — commonly referred to as configuration drift—which substantially undermines the scientific validity of experimental results.

Accurately evaluating nonlinear phenomena that emerge only at scale—such as network congestion and resource exhaustion—requires the application of IaC to the validation process. By defining the entire environment declaratively as code, human error is eliminated and large-scale stress testing can be conducted with complete reproducibility.

These two barriers collectively motivate the design of the platform proposed in this paper, as discussed in the following section.

### C. Related Works

Several approaches have been proposed to address the challenge of validating BACnet-based systems at scale. The BACnet Virtual Test Bed (BCVTB), developed at Lawrence Berkeley National Laboratory, provides a co-simulation framework that connects real BAS control systems with whole-building simulation programs via a BACnet interface [11]. While BCVTB enables hardware-in-the-loop testing, it does not support the execution of unmodified application binaries, and its applicability is limited to environments where simulation models of the target system are available in advance.

The BACnet Testing Laboratories distribute conformance testing tools designed to verify that individual devices correctly implement the BACnet protocol specification [4]. These tools are well-suited for single-device certification but are not designed to evaluate emergent network-level behaviors that arise in large-scale, multi-vendor deployments.

Network emulation frameworks such as NEmu [12], which leverage QEMU as the underlying hypervisor, have demonstrated the feasibility of constructing customizable virtual network topologies with minimal hardware resources. However, these frameworks target general-purpose network research and do not provide mechanisms for automating GUI-driven BAS applications or reproducing OS-level runtime conditions specific to building automation software.

In the domain of CPS testbeds for building systems, prior work has explored the use of digital twin models to replicate physical building behavior in simulation [13]. Such approaches typically rely on abstract models of device behavior rather than executing actual vendor software, which limits their ability to detect implementation-level defects or vendor-specific interoperability issues.

The approach proposed in this paper differs from all of the above in that it combines hypervisor-level input emulation with IaC-based environment orchestration, enabling both OS-agnostic binary execution and fully reproducible large-scale deployment—capabilities that have not been jointly addressed in prior work.

## III. ARCHITECTURE OF THE PROPOSED LARGE-SCALE BACNET EMULATION PLATFORM

This section presents the architecture of the proposed verification platform, which integrates virtualization technology and configuration management tools to overcome the two barriers identified in Section II: the emulation barrier and the reproducibility barrier.

### A. Hierarchical Structure and Role Definition

As illustrated in Fig. 1, the proposed platform comprises two distinct layers: the Infrastructure Director layer, which governs the lifecycle of the entire experimental environment, and the Hypervisor layer, which reproduces the behavior of individual devices.

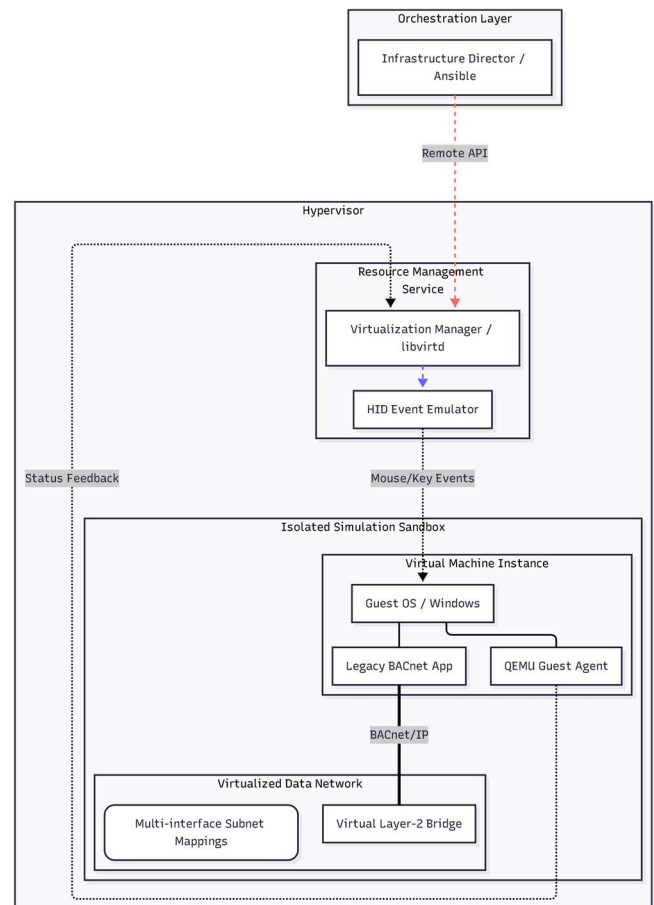


Fig.1 Architecture of the proposed large-scale BACnet emulation platform.

**Infrastructure Director Layer:** This layer employs a configuration management tool (Ansible) to define the verification environment declaratively as code, following the Infrastructure as Code (IaC) paradigm. The number of nodes, network topology, and initial configuration of each node are specified in a declarative manner, thereby ensuring safe and complete reproducibility of experimental scenarios.

**Hypervisor Layer:** Multiple Virtual Machines (VMs) are executed as independent nodes under a Linux-based hypervisor (QEMU/libvirt). A Virtualization Manager manipulates these computational resources via API, realizing a scalable verification environment that is not constrained by physical hardware limitations.

### B. OS-agnostic Application Control via Hypervisor-level Input Emulation

A distinguishing feature of the proposed platform is its ability to automate GUI operations through hypervisor-level input emulation—without relying on any functionality internal to the guest OS.

Conventional approaches to controlling applications within VMs typically rely on OS-specific agents such as the QEMU Guest Agent. However, such agents are susceptible to variations in OS version and security policy, limiting their applicability in heterogeneous environments. To address this, the proposed platform introduces a two-stage control mechanism:

**Input Device Emulation:** Based on instructions issued by the Infrastructure Director, the Virtualization Manager directly injects Human Interface Device (HID) events into emulated virtual input devices (mouse and keyboard). This enables non-intrusive, external execution of GUI interactions—such as button clicks at specified screen coordinates and menu navigation—regardless of the guest OS type.

**Side-Channel State Monitoring:** The QEMU Guest Agent is employed in a restricted capacity, limited to the retrieval of metadata such as application launch status and network interface liveness.

By adopting this hybrid architecture—in which control (input) is performed at the hypervisor layer and monitoring (output) is performed at the guest agent layer—the platform achieves a generalized operational interface applicable uniformly across environments ranging from legacy Windows systems to modern Linux distributions.

The operational sequence of this hybrid architecture proceeds as follows. Upon invocation of the Ansible playbook, the Infrastructure Director layer provisions the target VMs via the libvirt API and configures the virtual network interfaces. Once each VM completes its boot sequence—confirmed by a QGA heartbeat response—the Virtualization Manager begins injecting HID events to launch the target BACnet application on the guest. Subsequent monitoring of application state is delegated exclusively to the QGA, ensuring that the control and monitoring pathways remain decoupled throughout the test lifecycle.

## IV. EFFECTIVENESS OF IAC-BASED SIMULATION

To situate the proposed platform within the context of existing work, this section first compares it with the authors' prior approach, followed by a description of the implementation model. The authors previously reported a large-scale BACnet validation environment based on Windows OS and SoftEther [6]. In that approach, however, virtual network construction was tightly coupled to the host OS network stack, making it difficult to precisely control resource contention under increasing node counts or to characterize packet forwarding latency.

The architecture proposed in this paper advances beyond the authors' prior work in three key respects.

1) **OS Abstraction and Generality:** The prior approach was limited to Windows-specific control mechanisms. By adopting HID emulation at the hypervisor layer, the proposed method establishes a unified automation API that is

independent of the guest OS type.

2) **L2 Transparency and Network Isolation:** Whereas the authors' prior work relied on L2 extension via SoftEther, the proposed method leverages Linux Bridge to reproduce switching behavior closely approximating that of physical network infrastructure. Stable throughput was maintained even when accommodating multiple independent subnets.

3) **Full Automation via Infrastructure as Code (IaC):** The authors' prior work required manual network binding and GUI configuration on a per-VM basis. The proposed method adopts declarative configuration through Ansible, consolidating what was previously a per-node manual procedure into a single definition file. In principle, this enables a dramatic reduction in provisioning steps for deployments on the order of tens of nodes—potentially achieving an order-of-magnitude improvement over the prior approach.

To evaluate the effectiveness of the proposed platform, an implementation model was constructed in which a commercially deployed BACnet simulator (Table 1) was hosted within the virtualized environment. As shown in Fig. 2, the configuration applies a hybrid control scheme to existing binaries running on Windows OS: HID events are injected from the hypervisor layer, while application state is monitored via the QEMU Guest Agent (QGA).

Table 1. Specifications of the BACnet simulator used as the implementation case

| Category              | BACnet Testing Tool-Controller | BACnet Testing Tool-Monitor |
|-----------------------|--------------------------------|-----------------------------|
| Vendor                | Unitec Corporation             | Unitec Corporation          |
| Simulated device type | B-BC                           | B-OWS                       |
| Supported OS          | Windows 10/11                  | Windows 10/11               |

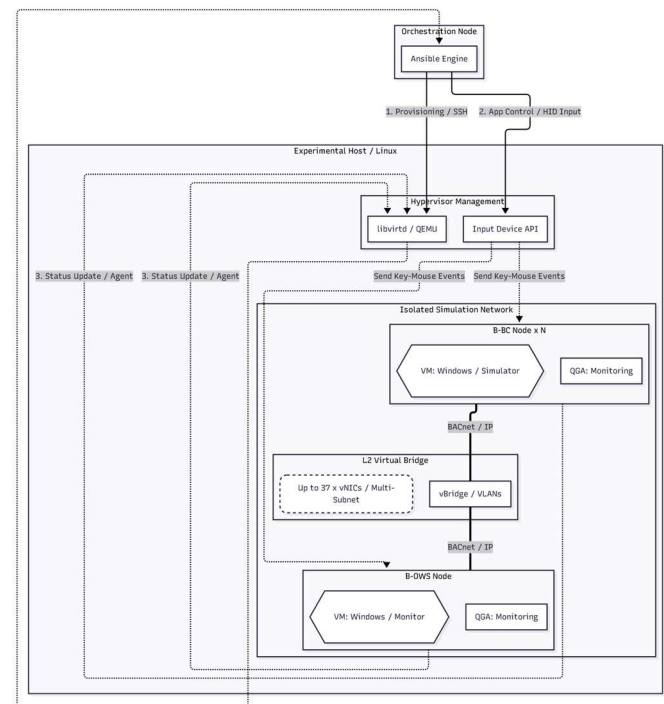


Fig. 2. Detailed configuration of the BACnet validation environment based on the proposed method.

Table 2 presents a qualitative comparison between the Windows-based validation environment reported in the authors' prior work and the IaC-based architecture proposed

in this paper. The prior method relied heavily on manual procedures for VM provisioning and network configuration, which gave rise to configuration errors at scale and a lack of experimental reproducibility. In contrast, the proposed architecture offers the following design-level advantages. First, the adoption of IaC via Ansible formalizes the construction procedure as code, theoretically eliminating human error and enabling complete environment replication. Second, the OS-agnostic HID control mechanism establishes a unified API for automated application control that is unconstrained by guest OS security policies—a capability that was not achievable in the authors’ prior work. These design characteristics are expected to substantially reduce validation overhead during the transition to complex CPS environments. Together, these advantages position the proposed platform as a practical foundation for systematic pre-deployment validation in BACnet-based CPS projects, where the cost of discovering integration failures post-deployment is significantly higher than in conventional IT environments.

Table 2. Qualitative comparison between the conventional method (physical/simple VM) and the proposed method (IaC + QEMU)

| Criterion                   | Conventional method                     | Proposed method                      |
|-----------------------------|-----------------------------------------|--------------------------------------|
| Environment reproducibility | Low (manual operation)                  | High (code-defined)                  |
| Node scalability            | Limited (physical/temporal constraints) | Flexible (configuration change only) |
| OS/binary dependency        | Dependent on specific OS                | Executed without modification        |
| GUI automation              | Not supported                           | Supported (via QGA)                  |

## V. USE CASE SCENARIOS

The proposed platform is applicable to a range of validation scenarios that are difficult or impractical to address with conventional tools.

### A. Pre-deployment Interoperability Testing

In multi-vendor BAS projects, equipment from different manufacturers must be verified to interoperate correctly before physical installation. The proposed platform enables such testing by hosting vendor-specific binaries on isolated VMs and exercising BACnet service exchanges—including object discovery, property read/write operations, and change-of-value notifications—across heterogeneous OS environments. Because each VM executes an unmodified vendor binary, the results reflect actual product behavior rather than idealized protocol models. Furthermore, the IaC-based provisioning allows the same test scenario to be replicated across different software versions or configuration profiles with minimal overhead, supporting regression testing as vendor firmware is updated.

### B. CPS Migration Validation

When integrating legacy BAS infrastructure into a CPS framework, it is necessary to verify that existing applications continue to function correctly under the new management layer. The platform supports this scenario by allowing real application binaries to run alongside newly developed CPS middleware within the same virtual network, enabling early detection of compatibility issues prior to physical deployment.

This is particularly valuable in environments where legacy systems cannot be easily taken offline for testing. The hybrid control mechanism further allows the test orchestrator to simulate supervisory commands from a CPS management layer—such as a digital twin controller—while observing how legacy BACnet devices respond, without requiring any modification to the existing software stack.

### C. Resource Profiling under Scaled Load

A critical challenge in edge-oriented CPS deployments is selecting hardware capable of sustaining the target workload. By incrementally increasing the number of active VM nodes and monitoring host resource consumption via the QEMU Guest Agent, the platform enables systematic profiling of CPU and memory utilization as a function of network scale. This capability directly supports hardware selection decisions for deployments where computational resources are constrained, such as building-edge controllers. Unlike physical testbeds, the proposed platform allows resource allocations—such as vCPU count and memory limits—to be adjusted declaratively between test runs, enabling systematic sensitivity analysis of how resource constraints affect BACnet communication reliability and latency.

### D. Security and Fault Injection Testing

The IaC-based environment definition allows arbitrary network conditions—such as packet loss, latency injection, or node failure—to be scripted and reproducibly applied across the emulated network. This enables structured testing of fault tolerance and anomaly detection mechanisms within BACnet-based CPS architectures, which is otherwise difficult to achieve in production environments without risk to operational systems. Since the entire fault scenario is defined as code, test campaigns can be version-controlled and shared across research teams, improving the reproducibility and comparability of security evaluations in BACnet-based CPS research.

## VI. CONCLUSION

This paper proposed a large-scale BACnet simulation platform that integrates virtualization technology and Infrastructure as Code (IaC) to support advanced operational management and CPS integration in Building Automation Systems (BAS).

The primary contribution of the proposed method lies in the establishment of a hybrid control architecture that decouples HID event emulation at the hypervisor layer from application state monitoring via the guest agent. This design realizes an OS-agnostic API for automated application control, free from the OS-level constraints and binary dependencies that limited conventional approaches. Furthermore, declarative environment definition via Ansible enables the construction of complex, multi-subnet validation environments to be expressed entirely as code, providing a logical guarantee of experimental reproducibility and node scalability.

Because the platform supports the execution of real-world application binaries without modification, it offers substantial practical utility for interoperability testing in multi-vendor environments and for resource profiling during the transition from legacy systems to CPS-integrated architectures.

Beyond the specific contributions described in this paper, the proposed architecture represents a broader shift in how BACnet validation environments are conceived. By treating the test environment as a software artifact—defined, versioned, and deployed through standard DevOps tooling—the platform aligns the building automation domain with contemporary practices in cloud-native systems engineering. This alignment is significant not only for immediate validation tasks but also for the longer-term goal of establishing reproducible, community-shareable benchmarks for BACnet-based CPS research.

Future work will focus on empirical validation of the proposed platform across three dimensions.

1) Provisioning time will be measured as a function of node count, targeting deployments on the order of tens to hundreds of VMs, to quantify the scalability advantage of the IaC-based approach over manual configuration.

2) Computational resource consumption—including CPU utilization and memory footprint—will be systematically profiled under sustained BACnet broadcast traffic to establish baseline hardware requirements for edge-oriented CPS deployments.

3) Cross-vendor interoperability will be evaluated by simultaneously hosting BACnet binaries from multiple vendors within the emulated environment and exercising standard BACnet service exchanges, including object discovery and change-of-value notifications.

These investigations will consolidate the practical validity of the proposed platform and support its adoption as a reproducible, community-shareable benchmark infrastructure for next-generation building automation research.

#### CONFLICT OF INTEREST

The authors declare no conflict of interest.

#### AUTHOR CONTRIBUTIONS

Yuki Hatanaka initiated the research, designed the platform architecture, implemented the prototype system, and drafted the manuscript. Tetsuya Yokotani, as the supervising professor, provided guidance on the research direction and

conducted a comprehensive review of the work. All authors have read and approved the final version of the manuscript.

#### REFERENCES

- [1] ANSI/ASHRAE Standard 135-2012. A Data Communication Protocol for Building Automation and Control Networks. [Online]. Available: <https://www.ashrae.org/technical-resources/standards-and-guidelines/standards-addenda/standard-135-2016-bacnet-a-data-communication-protocol-for-building-automation-and-control-networks>
- [2] T Sauter, S Soucek, W Kastner, and D Dietrich, "The evolution of factory and building automation," *IEEE Industrial Electronics Magazine*, vol. 5, no. 3, pp. 35–48, 2011.
- [3] X. Wu *et al.*, "A hardware-in-the-loop emulation testbed for high fidelity and reproducible network experiments," in *Proc. 2017 Winter Simulation Conference (WSC)*, 2017, pp. 408–418.
- [4] BACnet International. About BACnet Testing Laboratories (BTL). [Online]. Available: <https://btl.org/about-btl/>
- [5] F. Bellard, "QEMU, a fast and portable dynamic translator," in *Proc. USENIX Annual Technical Conference*, 2005, pp. 41–46.
- [6] Y. Hatanaka *et al.*, "A development of BACnet/IP-based platform for monitoring a large-scale building," *IEICE Transactions on Communications (Japanese Edition)*, no. 5, pp. 273–281, 2024.
- [7] D. Nobori. Design and Implementation of SoftEther VPN. [Online]. Available: [https://www.softether.org/4-docs/9-research/Design\\_and\\_Implementation\\_of\\_SoftEther\\_VPN](https://www.softether.org/4-docs/9-research/Design_and_Implementation_of_SoftEther_VPN)
- [8] Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry. [Online]. Available: [https://research-and-innovation.ec.europa.eu/knowledge-publications-tools-and-data/publications/all-publications/industry-50-towards-sustainable-human-centric-and-resilient-european-industry\\_en](https://research-and-innovation.ec.europa.eu/knowledge-publications-tools-and-data/publications/all-publications/industry-50-towards-sustainable-human-centric-and-resilient-european-industry_en)
- [9] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Trans. Ind. Informatics*, vol. 16, pp. 669–680, 2019.
- [10] M. Artac *et al.*, "DevOps: Introducing infrastructure-as-code," in *Proc. IEEE/ACM 39th International Conference on Software Engineering Companion*, 2017.
- [11] T. Nouidui *et al.*, "BacNet and analog/digital interfaces of the building controls virtual testbed," in *Proc. Building Simulation Conference*, 2011.
- [12] V. Autefage and D. Magoni, "NEmu: A distributed testbed for the virtualization of dynamic, fixed and mobile networks," *Computer Communications*, vol. 80, pp. 33–44, 2016.
- [13] A. S. Cespedes-Cubides and M. Jradi, "A review of building digital twins to improve energy efficiency in the building operational stage," *Energy Inform.*, vol. 7, 11, 2024.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).