ISSN 2186-7437

NII Shonan Meeting Report

No. 251

Green Intelligence: Sustainable Development and Operations of Intelligent Systems

Heng Li
Marin Litoiu
Lei Ma

August 31 - September 4, 2026

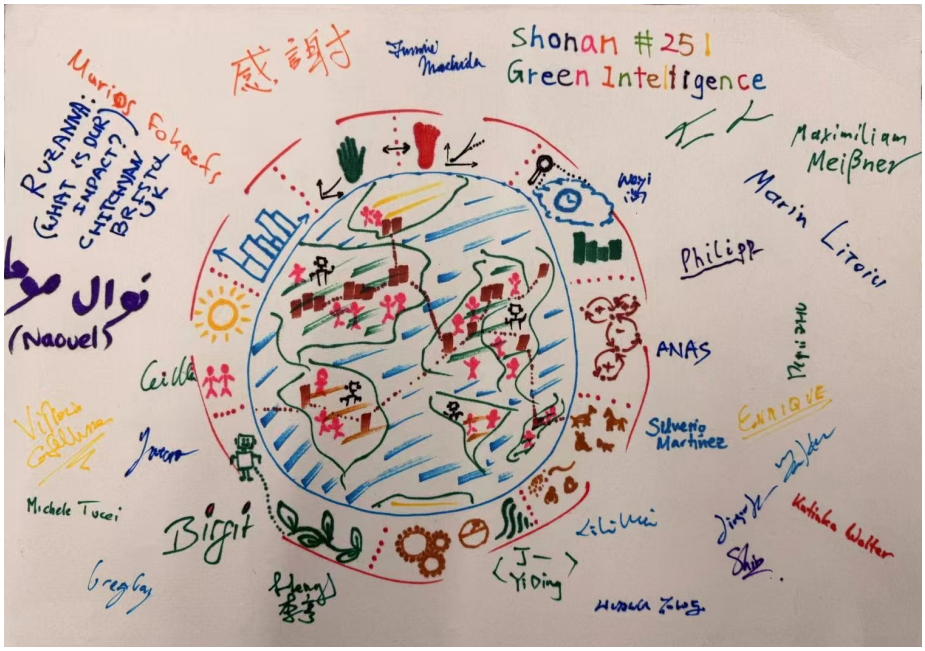


National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Green Intelligence: Sustainable Development and Operations of Intelligent Systems (NII Shonan Meeting No. 251 Report)

Organizers:
Heng Li (Polytechnique Montréal)
Marin Litoiu (York University)
Lei Ma (University of Tokyo)

August 31 - September 4, 2026



NII Shonan Meeting No. 251 brought together experts in software engineering, performance, sustainability, and AI to explore challenges, emerging research directions, and engineering practices for sustainable intelligent systems. The meeting focused on measuring and optimizing the environmental impact of AI and agentic systems, understanding sustainability trade-offs, and identifying new approaches to integrating sustainability throughout the software development and operations lifecycle.

Background and introduction

The broad influence of Artificial Intelligence (AI) and Machine Learning (ML) has led to a massive new software paradigm of Intelligent systems. For these systems, key components are either AI-based or AI-generated (i.e., AIWare). Among the key attributes driving research and industry efforts to govern and regulate intelligent systems, sustainability is becoming a critical factor. In particular, the ever-growing energy consumption and carbon emissions from AI-based software play a central role in the sustainability of intelligent systems, from large-scale data centers to personal devices. Thus, improving software energy efficiency is critical to reducing the energy footprint of intelligent systems and supporting sustainable development.

The challenge of ensuring the energy efficiency of software in the era of AI is twofold. First, despite the rich body of performance and energy research on traditional software, AI-based software, such as ChatGPT, may not fit into well-established engineering processes. The best practices distilled through decades of experience may not apply. Second, AI-generated software may not prioritize efficiency, leading to suboptimal software that wastes energy during runtime. The proposed research toward more energy-efficient AIWare will contribute theories, methods, and tools to support the development, testing, and auditing of energy-efficient AIWare. The outcome not only benefits the software industry by providing more cost-effective AI solutions across industries but also protects the environment by reducing global electricity demand and CO2 emissions.

The connection between development and operations, i.e., DevOps, has become a core principle for developing and operating modern software systems, aiming to increase the velocity of releasing high-quality software through automation and by integrating development and operations responsibilities. On the one hand, the fast DevOps release cycles pose significant challenges for ensuring the energy efficiency of intelligent software and its deployment environment. On the other hand, DevOps offers strong opportunities to enhance intelligent software sustainability, since integrating development and operations enables continuous collection and analysis of energy-consumption data, which can be used as feedback to optimize the energy efficiency of software and its deployment environment.

The goal of the meeting is to bring together software engineering, performance, and AI experts from academia and industry, featuring and taking a special focus on the sustainability of intelligent systems, to discuss how to design new engineering practices, especially for intelligent systems, that would allow us to effectively engineer an AI-based system in a more energy-efficient way. The long-term goal is to make the activities across the whole development lifecycle of intelligent systems sustainable (Green Intelligent Systems Engineering) as it is nowadays for traditional software. The AIWare and green software community has grown rapidly in recent years, with early-stage results from researchers worldwide. The meeting enables the exchange of state-of-the-art research and technology, promotes sustainability as an important research direction, and emphasizes its potential industrial applications, thereby contributing to the urgent demand for sustainable intelligent systems.

Overview of the meeting

NII Shonan Meeting No. 251 (*Green Intelligence*) hosted 28 participants from all over the world. These participants contribute diverse expertise in software engineering, artificial intelligence, and sustainability.

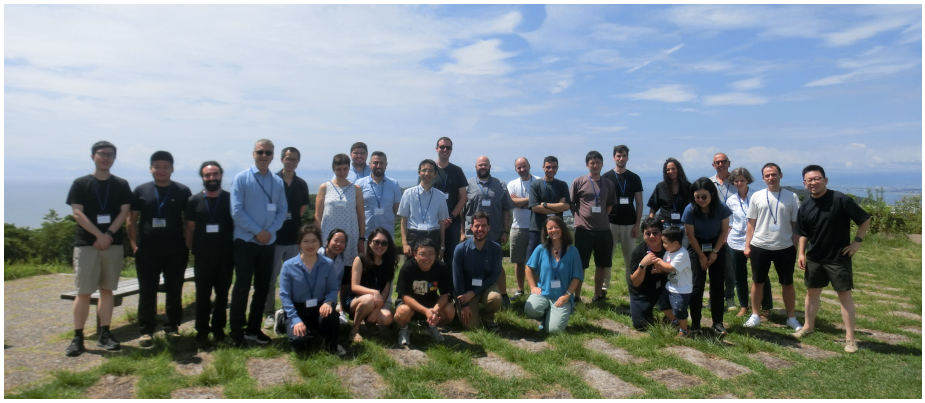


Figure 1: Group picture

This five-day meeting (and beyond) consists of the following five phases:

- Introductions (pre-meeting). Before the meeting, each participant prepared one introductory slide describing their research interests and proposed discussion topics. At the start of the meeting, each participant had 2 minutes to present their introduction slide.
- Presentations (Day 1). Nine participants were invited or volunteered to make a 20-min presentation about their work related to *Green Intelligence*.
- Exploratory discussions (Day 2). An initial set of topics was identified from the topics proposed by the participants (the one-slide introduction). Guided by the questions of "What? So what? Now what", the participants then discussed the list of topics and proposed new ones. In the end, five topics were selected for in-depth discussion in the next phase. Details about these topics are described in the "Summary of discussion" section.
- Deepening discussions (Day 3-5). Participants voluntarily selected one of the five topics and formed five discussion subgroups. Each subgroup had extensive free-form discussions around the focused topic in multiple sessions, with several cross-subgroup synchronization presentations in between. On the last day, each subgroup made a final presentation of their investigation and discussion outcomes.
- Continuing activities (post-meeting). Most subgroups have described their plan to continue the investigations and discussions. A few subgroups planned to submit a paper to the New Ideas and Emerging Results (NIER) track of the International Conference on Software Engineering (ICSE 2027). We also created an email thread for coordinating post-meeting activities.

In addition to the above-mentioned activities, we had an excursion (visiting Kenchoji Temple, learning about "Zazen") and a banquet on Day 3. Some participants formed small hiking/touring groups.

Overview of Talks

Planetary boundaries: Oops, humans.

Birgit Penzenstadler, Chalmers Technical University and Gothenburg University

Birgit Penzenstadler introduced the planetary boundaries as a reference framework for human impact on the planet. She contrasted the sustainable development goals and the UN member nations limited reach within the timeframe, and outlined the problems of limits to growth, rebound effects, and techno-solutionism. She advocated to integrate planetary boundaries into software design modelling as it makes environmental dependencies explicit in design, helps reveal conflicts and synergies with ecological limits, and helps trace distinct ecological claims and measure impacts. Via artefact-based requirements engineering, Sustainability Goals can be modelled, impacts projected within the Sustainability Awareness Framework, and Nature as a Stakeholder presented as central concept. For Green Intelligent Systems Engineering, the goal is to integrate the planetary boundaries as constraints, and to integrate causal network perspectives and identify leverage points to reduce the environmental footprint of the systems and make sure their handprint serves humanity and the environment.

Water as a New Constraint for Sustainable Intelligent Computing

Yi Ding, Purdue University

The rapid growth of AI is increasing pressure on energy, water, and natural resources. Yet sustainable computing has focused mainly on carbon emissions. This talk asks a broader question: Is carbon enough? It shows why water must also be considered and explains how the location, timing, and source of water use can greatly change its real-world impact. Drawing on recent research, the talk presents methods for measuring direct, indirect, operational, and embodied water use across computing systems. It also introduces the Software Water Intensity specification, an emerging standard for consistent and transparent water accounting. Together, these efforts can help researchers, technology companies, and policymakers identify hidden tradeoffs, avoid shifting environmental burdens, and make computing and AI systems more sustainable in water-stressed regions.

Digital Twins for sustainable IT Systems

Katinka Wolter, Freie Universitat Berlin

High-performance computing (HPC) systems keep getting faster, and their power demand keeps rising. Beyond better hardware and buildings, data centres can save energy by operating in response to their surroundings, such as weather, energy prices, renewable supply and limits on available power. This talk presents a Digital Twin for energy-aware HPC operation. It pairs a Digital Shadow, which collects internal and external monitoring data, with a Digital Master

made up of a node power model and a scheduling simulator. From each job's runtime estimate, the twin calculates what the job will cost in energy. It then picks the best start time within a lookahead window and switches idle nodes off when they aren't needed, adapting to large seasonal swings in energy prices, renewable supply and CO_2 intensity. The experimental results show limits as well as benefits: PUE closely tracks outside temperature and has to be read critically, and power capping that works in simulation is not always met in practice. The talk ends by looking at networks of Digital Twins for complex industrial processes such as battery production.

Performance Optimization With LLMs

Philipp Leitner, Chalmers | University of Gothenburg

Performance (e.g., execution time efficiency) is a unique non-functional property of software, in that it inherently depends on the interplay of three dimensions: software (code), systems (infrastructure and platforms), and usage (workloads). We can rarely assess the efficiency of a program or software system without taking into account where it is going to run, or how it will be used. In my talk, I illustrated how generative AI struggles with producing efficient code, exactly because it reasons about code in isolation, blind to the systems it runs on and the workloads it serves. Bringing in runtime data can serve as a way forward for future agentic coding platforms.

The Hidden Sustainability Debt of AI

Mohammad Hamdaqa, Polytechnique Montréal

AI systems are increasingly judged by what they can do, but much less attention is paid to the resources they consume along the way. In this talk, I introduce the concept of AI sustainability debt. I will show how this debt can emerge across three layers of an intelligent system; from implementation and framework decisions, to workflows and pipelines, to deployment and runtime infrastructure; and how small inefficiencies at one layer can propagate and accumulate into larger costs elsewhere. The talk will conclude with a broader research challenge: how do we distinguish necessary computation from waste, trace resource costs back to the engineering decisions that caused them, and build intelligent systems that are not only capable, but also efficient, observable, traceable, and governable?

Beyond Green AI: Sustainability as a Property of Socio-Technical Systems

Ruzanna Chitchyan, University of Bristol

Most work on sustainable AI looks at how efficiently AI models run and how much energy and compute they use. This talk argues that efficiency is necessary but not enough: how AI affects sustainability depends on the whole socio-technical system it is deployed in, not on the AI component alone. The talk uses the shift to smart local energy systems as its example. There, AI

forecasts demand, optimises the use of renewable energy and enables peer-to-peer trading, so extra AI compute can bring a net sustainability benefit by replacing centralised fossil-fuel generation. Yet the UK's slow uptake of heat pumps (about 1% of homes, compared with over 60% in Norway) shows that a technically sound solution can still fail in practice. Social housing residents report poor handover and "don't touch it" instructions, along with worries about bills, and some stop using their heat pumps altogether. Able-to-pay households, meanwhile, accept poor performance and high installation costs and still say they are satisfied. Building on these studies, the talk argues that a technically sustainable AI system can be part of an unsustainable socio-technical system. Whether it succeeds also depends on people wanting the technology, the skills to use it, trust and privacy, expectations, regulation, business decisions and supporting infrastructure. The talk closes by showing how the Sustainability Awareness Framework (SusAF) can help software engineers take these wider effects into account.

Including AI Workloads in Industry-Standard Server Efficiency Benchmarks: Why It Is Necessary Yet Challenging

Maximilian Meißner, University of Wuerzburg

The rapid adoption of GPU-accelerated AI and HPC workloads has driven a sharp rise in data center energy consumption, making reliable energy efficiency measurement critical for research, system design, and operational decisions. Unlike conventional performance benchmarking, energy efficiency benchmarking introduces additional sources of variability (e.g., power-domain selection, sampling strategy, thermal and power-management policies, and workload non-determinism) that must be explicitly controlled and documented. Small changes in drivers, firmware, or tuning can produce large, misleading differences in reported efficiency, particularly on modern GPU platforms where hardware/software co-optimization and vendor-specific features are pervasive. In this talk, we summarize best practices and lessons learned from implementing AI workloads for the latest SPEC efficiency benchmarks. We discuss current challenges in creating standardized efficiency benchmarks that yield reliable, reproducible, and relevant results while ensuring fair comparisons across heterogeneous servers, from single-GPU entry-level systems to multi-node multi-GPU high-end platforms, with divergent hardware capabilities and vendor-specific optimizations. Our contributions aim to standardize methodology, reduce ambiguity, and enable trusted energy efficiency benchmarking of GPU-accelerated systems.

Shadow Software Development: An Agentic Software Development Lifecycle for Domain Experts

Marios Fokaefs, York University

AI has made it easy for anyone to write code, but that does not mean everyone can ship software. Domain experts often write one-off code that solves one immediate problem but is hard to maintain, reuse or transfer, and the knowledge behind it stays with the person who wrote it. The next expert then starts

over, wasting effort and energy on repeated work. This talk presents an agentic software development lifecycle that keeps domain experts in control while specialised agents supply the engineering practices around their code. Across the lifecycle, agents turn natural-language requirements into UML designs, recalling more than 80% of entities and relationships in a two-pass generate-and-refine process. They build performance models (LQN), both from design artifacts and from runtime logs, metrics and traces. They maintain threat models and explain, patch and mitigate vulnerabilities. They generate CI/CD workflows that pass quality gates over 90% of the time across 39 projects. For MLOps, agents recover the ML context hidden in repositories so that models are retrained only when that context changes. For GreenOps, a digital twin supports vehicle self-localization when sensor data is missing, choosing between a small purpose-built transformer and a small language model by accuracy per joule rather than accuracy alone. The overall message is that AI's promise is not just writing code, but delivering engineering quality at the speed of domain expertise, through practices that are transparent, explainable and quality-aware.

Challenges of addressing sustainability of machine learning systems

Silverio Martínez Fernández, Universitat Politècnica de Catalunya BarcelonaTech

Training and running ML systems uses a growing amount of energy, so software engineering has a part to play in reducing their carbon footprint. This talk takes a Green AI view, which values energy, time, reproducibility and reuse alongside accuracy. It presents a structured approach to environmental sustainability across the ML lifecycle based on the Quality Improvement Paradigm and GQM+Strategies. The approach covers setting energy-related goals, for example in line with the EU AI Act, measuring and reporting energy and carbon in model cards, and assessing energy efficiency through energy labels such as GAISSALabel. It also builds a catalog of green tactics that apply to data, models and systems. The talk summarises empirical findings on such tactics. During training, accuracy and energy for the next epoch can be predicted well enough to choose the best trade-off between them. For model optimisation, dynamic quantization pays off only in systems that run over a million inferences. For deployment, the choice of execution provider and runtime engine significantly affects energy use when serving small code language models. Finally, the talk raises open challenges for discussion. Practitioners need practical, comparable benchmarks and energy labels, as well as accurate predictions of what tactics will save, to meet emerging standards and regulations. Researchers need an experience factory that combines empirical results into models and theories, and ways to manage how ML assets evolve. Educators need to teach that performance is not the same as efficiency.

List of Participants

- Anas Bouziane, Polytechnique Montréal
- Tse-Hsun Chen, Concordia University
- Ruzanna Chitchyan, University of Bristol
- Vittorio Cortellessa, University of L'Aquila
- Yi Ding, Purdue University
- Marios Fokaefs, York University
- Gregory Gay, Chalmers University of Technology, Göteborgs Universitet
- Mohammad Hamdaqa, Polytechnique Montréal
- Yuheng Huang, The University of Tokyo
- Yutaro Kashiwa, Nara Institute of Science and Technology (NAIST)
- Philipp Leitner, Chalmers | University of Gothenburg
- Heng Li, Polytechnique Montréal
- Marin Litoiu, York University
- Lei Ma, The University of Tokyo
- Fumio Machida, University of Tsukuba
- Silverio Martínez-Fernández, Universitat Politècnica de Catalunya
- Maximilian Meissner, University of Wuerzburg
- Naouel Moha, École de technologie supérieure
- Birgit Penzenstadler, Chalmers Technical University and Gothenburg University
- Enrique Barba Roque, TU Delft
- Weiyi Shang, University of Waterloo
- Shin Hwei Tan, Concordia University
- Luca Traini, University of L'Aquila
- Michele Tucci, University of L'Aquila
- Lili Wei, McGill University
- Katinka Wolter, Freie Universitat Berlin
- Jinqiu Yang, Concordia University
- Derui Zhu, Rochester Institute of Technology

Meeting Schedule

Check-in Day: August 30 (Sun)

- Welcome Banquet

Day1: August 31 (Mon)

- Introduction and Presentations
- Group Photo Shooting

Day2: September 1 (Tue)

- Exploratory Discussions
- Identification of focused discussion topics and subgroups.

Day3: September 2 (Wed)

- Deepening Discussions
- Excursion and Main Banquet

Day4: September 3 (Thu)

- Roundtable on Sustainability and Education
- Deepening Discussions

Day5: September 4 (Fri)

- Final Presentations and Discussions
- Wrap up

Summary of discussions

In the exploratory discussion phase, we identified five focused discussion topics: 1) **Environmental Impact of Agentic Systems**, 2) **Optimization of Agentic AI Efficiency**, 3) **AI Sustainability Tradeoff**, 4) **Measurement and Benchmarking for the Sustainability of Agentic AI**, and 5) **People and Governance**. The participants formed five discussion subgroups according to these topics; each participant voluntarily joined one subgroup.



Figure 2: The "AI Sustainability Tradeoff" subgroup in discussion

Environmental Impact of Agentic Systems

Yi Ding, Shin Hwei Tan, Jinqiu Yang, Lei Ma

The subgroup discussed how to understand the environmental impact of LLM-based agentic systems. Unlike a normal LLM request, an agent may make many model calls, retrieve data, use tools and external services, store information, and run generated code. Its impact therefore comes from the entire agentic pipeline, not only the LLM. We identified major impact sources, including model inference, memory and retrieval, tools and external services, agent orchestration, communication, and supporting infrastructure. Important challenges include defining the system boundary and functional unit, tracing impacts across distributed services, and measuring systems whose execution paths change from one task to another.

We paid particular attention to water use. Agentic systems can consume water directly through data center cooling and indirectly through electricity generation, while the amount and impact vary by location, time, grid mix, and cooling technology. Water use from third-party tools and services is especially

difficult to observe. Future work should develop end-to-end measurement methods, study coding agents as an initial case, and build prediction tools that estimate environmental impact across different models, tools, and configurations. This could support decisions that balance task quality, performance, energy, carbon, and water use.

Optimization of Agentic AI Efficiency

Luca Traini, Philipp Leitner, Gregory Gay, Yutaro Kashiwa, Mohammad Adnan Hamdaqa

The subgroup discussed how LLM-based agentic systems could be optimized for energy efficiency. We started from the view that energy should not be optimised on its own. Practical sustainability is a trade-off between energy consumption (E), output quality (Q) and time to completion (T). We framed this as a weighted multi-objective problem, optimising αE + βQ + γT.

We then mapped the main levers that determine an agent's energy footprint. The first is the prompt: the context (short- and long-term memory, external knowledge via RAG or MCP, tool output) and the instructions. The second is the workflow, meaning the structure of the agentic system itself: which tools are used and how agents interact. The third is the model: family, size, routing, parameters and infrastructure configuration. The main idea that came out of the discussion is a self-optimizing agentic system. A monitoring component computes sustainability indices from the running task agents. A decision-making agent combines these with a quality signal and adapts the prompt, workflow and model at runtime. In effect, this is a MAPE-K-style feedback loop with the three levers as actuators. The decision-making agent could be implemented in different ways: (1) as a reinforcement learning agent, which treats different configurations and their impact as a black box, or (2) as an LLM agent, which reasons about the impact of different configurations and saves lessons learned in a textual knowledge base.

We also tried to instantiate this idea in a concrete scenario involving the use of agentic systems for code generation in a hypothetical industrial setting. In this scenario, signals related to code quality and non-functional characteristics represent the quality dimension (Q). These signals can be continuously monitored and used by the agentic system to maintain or improve the quality of the generated code while adapting the underlying configuration. The overall goal is therefore to make the code generation process sustainable not only in terms of output quality, but also in terms of time to completion and energy consumption.

Our next steps will include developing multiple independent instantiations of this scenario, with different case studies, actuators, and decision-making approaches.

AI Sustainability Tradeoff

Weiyi Shang, Lili Wei, Katinka Wolter, Vittorio Cortellessa, Michele Tucci, Fumio Machida

The subgroup discussed how engineers can take sustainability into account when designing and running AI-enabled software, focusing on the decisions they

face in practice. Such decisions rarely affect a single quality. A smaller model may reduce resource use but limit what the system can accomplish, and changing an agent's memory or retrieval strategy may alter both inference cost and result quality. The subgroup therefore proposed studying design choices together with their effects on software quality, recognising that the best choice depends on the task, hardware, workload, deployment environment and user requirements. A related question was whether decisions should be made for the whole system or for individual components and agents, since improving one part does not necessarily benefit the whole.

The literature showed that environmental goals can reinforce some software qualities and conflict with others. It also showed that machine-learning-enabled systems involve many potential quality trade-offs, and that sustainability research still lacks measurement and empirical evidence. Results also tend not to carry over between settings. In one study, the relative energy efficiency of LLM-generated code differed across a server, a PC and a Raspberry Pi. In another, the energy savings from microservice tactics depended on the workload. Building on this, the subgroup organised its discussion as a framework with three steps: identify the decisions and attributes to examine, measure their consequences, and use the results to support engineering choices. Cross-cutting concerns include matching the level of analysis to the decision, turning long-term sustainability goals into concrete targets that stay useful as AI systems change, recording the conditions behind each result, and making uncertainty explicit.

The subgroup chose MIMIC, an LLM-agent system for game testing, as its first case study. In MIMIC, a planner, an action executor and an action summarizer are connected through a shared memory, and the summarizer feeds critiques back to the executor so it can correct itself. This makes MIMIC a good setting for studying how model calls, feedback and memory retrieval together shape a complete workflow. One observation from the discussion motivated the choice: a 7B model handled individual tasks well but struggled when the agents had to work together. This shows that savings at the component level may not hold for the system as a whole.

The subgroup produced an experiment plan. The planned experiments, run on Shattered Pixel Dungeon with locally hosted models and personality-based prompting turned off, will compare several model sizes and memory strategies. The memory strategies range from no memory and full history to retrieving only relevant or recent memories. Each configuration will be repeated three times. The experiments will measure energy and token consumption alongside coverage, time per attempt, and the number of failures and attempts, to check whether savings on individual calls survive once retries and corrections are counted. Token use is treated as an indicator of resource use rather than a direct measure of carbon. Other choices raised in the discussion, such as orchestration, API access versus local hosting, scheduling and deployment, and qualities such as price, privacy and reproducibility, are left for a broader study.

The results will feed a dataset of observed trade-offs, used to identify Pareto-efficient configurations and to quantify how much energy a configuration saves for a given loss of coverage under different workloads and hardware. A later prototype would let engineers state their requirements, such as a minimum coverage and a time limit, and compare the energy use and uncertainty of the configurations that meet them, with the final choice left to the engineer. Future work will

extend the study to other systems and environments, examine design choices beyond model size and memory, adopt sustainability measures beyond energy and tokens, and explore whether the same evidence can guide reconfiguration during operation.

Measurement and Benchmarking for the Sustainability of Agentic AI

Anas Bouziane, Tse-Hsun Chen, Yuheng Huang, Heng Li, Naouel Moha, Enrique Barba Roque, Derui Zhu

The subgroup started from a simple observation: two agents can reach the same goal by very different paths and at very different energy costs. This leaves a key question unanswered: which component of an agent caused the cost, and did it earn its benefit? Agents are hard to measure and benchmark because their workflows are non-deterministic, their behaviour depends on state such as the KV cache, and they run across mixed model calls and tool calls, CPU and GPU, and local and remote execution.

The subgroup's main outcome is a proposal for a sustainability cost function for agentic AI. It estimates the energy needed to accomplish a goal from factors that can be observed, such as the context size, turns, tools, retries, and hardware. The idea is to fit the function where energy can be metered, then use it where it can't. The unit of measurement is the trajectory, meaning one successfully accomplished goal. The energy spent on failed attempts before that success is added to its cost, so benchmark results are normalised by success rate and trajectory length. The function splits energy into three terms: 1) model calls: driven by context size, output tokens and per-call overhead, with KV-cache hit rate and the share of activated parameters as refinements; 2) tool calls: driven by each tool's average energy and its execution time, with CPU, memory and network use as candidate inputs; 3) idle and waiting time.

The subgroup sees the cost function as what makes agent efficiency something developers can engineer, through three steps: 1) Measure: how expensive was a goal, in a way that can be compared across agents, models and harnesses? 2) Explain: where did the energy go (tokens, tool calls, memory, context growth, failures, retries)? 3) Improve: what should change (cheaper models, fewer calls, better planning, caching, forgetting)?

Planned applications include catching energy regressions in CI, root-cause diagnosis, tradeoff analysis and benchmarking. The longer-term vision is closed-loop energy management: estimating before execution, monitoring and steering during it, and optimising afterwards.

The subgroup's proposed experiment is to measure trajectories over many combinations of tasks, agent configurations and hardware (e.g. Terminal-Bench), then test whether trajectory features predict energy better than tokens or time alone. Validation would use unseen models, harnesses and hardware. As an immediate next step, the subgroup plans a proof of concept that checks how well estimated energy correlates with measured energy, for an ICSE 2027 NIER submission, followed by a full paper. Finer modelling of the KV cache and extension to multi-agent systems are left for future work.

People and Governance

Ruzanna Chitchyan, Marios Fokaefs, Marin Litoiu, Silverio Martínez-Fernández, Maximilian Meissner, Birgit Penzenstadler

The subgroup began by looking at AI's social and environmental impacts on developers, users and affected communities, and at governance questions such as ethics, policy and planetary boundaries. It then narrowed to one concrete question: how does the way users provide context to an LLM affect its resource use? The hypothesis is that the more an LLM has to assume, the more wasteful it is. Underspecified queries need several rounds of clarification before they produce a useful answer.

The subgroup illustrated this with a medical example. A query that gave all the clinical context up front reached an equivalent answer with about a quarter of the tokens of a vague query clarified over five turns. Based on this, the subgroup proposes client-side query optimisation: a tool on the user's device that finds missing or ambiguous context in a prompt, asks the user short clarifying questions, and sends one consolidated query to the LLM. The questions could come from rules, NLP techniques or a lightweight local model. The evaluation will use existing conversation datasets to compare token use for multi-turn exchanges, single full-context queries and the tool, and will check whether the answers are equivalent.

Open issues are who defines a "good" query, the risk of biasing exploratory questions, and whether token counts from local and commercial models can be compared. As a first step, the subgroup plans a short SEiS paper built on 10–15 examples from the medical, legal and energy-trading domains. A user study, and a study on whether showing resource-use information in the interface changes user behaviour, are left for future work.

Summary of new findings

Our discussions in the entire group and across different subgroups converged on several shared findings.

The agentic pipeline, not the model, is the unit of concern. An LLM-based agent may make many model calls, retrieve and store information, invoke tools and external services, run generated code, and retry failed steps. Its environmental cost therefore comes from the whole pipeline. Today, most measurements are either end-to-end totals or cover only the LLM, which hides where the cost arises. Several subgroups independently arrived at the *trajectory*, i.e., one accomplished goal including the failed attempts that preceded it, as a natural functional unit for measuring, comparing and optimising agents. This offers a first answer to the open question raised in the talks of what the right unit of efficiency is for an agentic workflow (Mohammad Hamdaqa, *The Hidden Sustainability Debt of AI*).

Sustainability is a trade-off, not a standalone objective. Energy and token savings compete with output quality, coverage, correctness, time, price, privacy and reproducibility. The subgroups identified a common set of design levers that move this balance: the prompt and its context (instructions, short- and long-term memory, retrieved knowledge, tool outputs), the workflow (task decomposition, agent interaction, tool selection), the model (family, size, routing, local versus remote hosting), and deployment and scheduling. The amount and form of *context* stood out as a lever that recurs at every level: in agent memory management, in prompt and workflow design, and in how human users phrase their queries. In one illustrative medical example, a query that supplied the full context up front reached an equivalent answer with about a quarter of the tokens needed by a vague query clarified over several turns.

Measured costs can be turned into engineering feedback. The subgroups moved beyond reporting costs towards making them actionable:

- a cost function that estimates the energy of a trajectory from observable factors such as context size, output tokens, tool calls, retries and waiting time;
- self-optimising agents that monitor sustainability indicators and adapt the prompt, workflow and model at runtime (a MAPE-K-style loop);
- context-aware trade-off analyses that help decision-makers choose between design alternatives;
- client-side tools that improve user queries before they reach the LLM.

Together, these point to closed-loop resource management for agentic systems: estimating cost before execution, monitoring and steering during execution, and diagnosing and optimising afterwards.

Efficiency is necessary but not sufficient. The talks and discussions repeatedly highlighted that the sustainability of AI is a property of the **socio-technical system** in which it is deployed, not of the AI component alone. Efficiency gains can be offset by rebound effects (Jevons paradox), and a technically efficient system can fail in practice if people do not adopt it, lack the skills

to use it, or lack supporting regulation and infrastructure. **Planetary boundaries** were proposed as a reference frame that makes such environmental limits explicit in software design.

A research gap beyond energy. Water use depends strongly on location, time and local water stress. Its adjusted impact can differ by orders of magnitude between regions, and water used by third-party tools and services is particularly hard to observe. Yet, when moving from vision to concrete next steps, most subgroups fell back on energy or tokens as the main sustainability indicator. This is partly because agent developers mainly observe tokens, while water, carbon, embodied impacts and e-waste are determined by infrastructure they do not control. Social and ethical dimensions were even less represented. Closing this gap between the broad notion of sustainability and the narrow metrics that are measurable today emerged as a key finding of the meeting.

Key research questions. Each subgroup distilled its discussion into guiding research questions:

- *Environmental Impact of Agentic Systems:* How can the energy, carbon and water footprint of an agentic pipeline be attributed across models, tools, external services and infrastructure, and predicted for different configurations?
- *Optimization of Agentic AI Efficiency:* How can an agentic system adapt its prompt, workflow and model at runtime to balance energy, quality and time to completion?
- *AI Sustainability Tradeoff:* How much can energy and token use be reduced without losing performance, and what side effects do the underlying design decisions cause?
- *Measurement and Benchmarking:* Can trajectory-level features explain and predict the energy of an agent better than tokens or time alone, in a way that is comparable across agents, models, harnesses and hardware?
- *People and Governance:* How does the way users provide context to an LLM affect its resource use, and can client-side support reduce it without degrading the answers?

Identified issues and future directions

Pushing the boundary beyond energy. In the final discussion, participants observed that the outcomes were strongly focused on energy. The community should deliberately broaden its scope:

- treat water, carbon, embodied impacts, land use and e-waste as first-class constraints alongside energy;
- account for where and when workloads run;
- avoid shifting environmental burdens from one dimension to another;
- address rebound effects and the social and governance dimensions of sustainability.

Concrete directions include multi-layer impact models for agentic systems (starting with carbon and water), integrating planetary boundaries as design constraints, and causal models linking socio-technical impacts to ecological limits.

Measurement, observability and benchmarking. Several issues remain open:

- Agentic systems are non-deterministic, stateful (e.g., the KV cache) and heterogeneous.
- Many run on opaque commercial APIs that expose only token counts.
- Token counts from different models are not directly comparable.
- Existing benchmarks rarely reflect realistic agent workloads.
- Experience from industry-standard efficiency benchmarks (e.g., SPEC) shows that results are highly sensitive to hardware, drivers and software stacks.

Future work should develop portable, reproducible sustainability benchmarks for agentic AI, report measured or estimated values according to what can be observed, and make traceability, i.e., what an agent did, why, and whether it could have done so with less waste, a built-in property of agents.

From measurement to engineering practice. Sustainability information is only useful if it reaches decisions. Open directions include:

- integrating energy and resource checks into CI/CD pipelines to detect regressions;
- making the cost of operations visible to agents so that they can plan in a cost-aware manner;
- providing decision-makers with context-specific trade-off analyses;
- studying whether exposing resource information to end users changes their behaviour.

Distinguishing necessary computation from waste will also require developers to state their intent explicitly.

Education for sustainability. A roundtable on sustainability and education identified two connected issues. First, generative AI is changing what software engineering students need to learn. They must be able to judge, test, explain and repair AI-generated artefacts, which requires solid foundational knowledge. Second, sustainability is rarely part of the curriculum. Participants agreed that a single elective course is not enough and that sustainability should be integrated across all courses. Proposed practices include:

- showing instructors that they often already teach sustainability aspects (e.g., maintainability as technical sustainability, accessibility as social sustainability);
- introducing the dimensions of sustainability at the start of each course, and asking instructors to state in the course plan how the course addresses them;
- using AI and SE4AI topics as motivation.

A major barrier is that such initiatives are usually voluntary. A position paper comparing curricula was suggested as a next step.

Community actions. Participants agreed on several actions to sustain the collaboration:

- each subgroup continuing its work towards publications, including submissions to the ICSE 2027 NIER track and a short paper at SEiS;
- guidelines for integrating and reporting sustainability considerations in software engineering research, analogous to the ACM empirical standards;
- a mailing list for post-meeting coordination and continuing collaborations;
- lasting collaborations through jointly funded projects.