Grade the Learning, Not the Implementation: AI-Aware Assessment in Internet Measurements Education

Oliver Hohlfeld
University of Kassel
Germany

1 Introduction

Generative AI tools are reshaping what students can produce with modest programming effort. A student can now obtain, process, and visualize a dataset in a fraction of the time it previously required. This shift creates a new opportunity for networking education, but it also exposes a fragility in how we assess learning: if grading focuses on implementations and written documentation, AI trivially satisfies those criteria without any understanding on the student's part. We argue that this challenge is not a threat but a forcing function that refocuses on a course's learning objectives. Drawing on our experience in a graduate-level Internet Measurements course, we highlight a needed shift: assess the learning experience made, not (only) the artifact produced.

2 AI Opportunity in Internet Measurements

Learning objectives. Internet Measurements is a domain where the central learning objectives are deeply conceptual. Students must learn to *i)* identify, understand, and reason about the biases and limitations of measurement approaches and datasets, and *ii)* select and combine active and passive measurement approaches or data sets to study a given research problem. This learning also requires sustained engagement with real data—without being constrained by programming effort that is required to realize measurements.

Pre-AI challenge. Practically reaching these objectives involves a substantial programming prerequisite: students must write scripts to collect, process, and visualize data before they can engage with the actual measurement problem. As a result, most time is often spent on enabling work rather than addressing the learning objectives themselves.

Example exercise. One exercise in our Internet Measurements course studies Internet topology and requires to combine BGP and traceroute data to reconstruct a topology. The *intended* learning outcome is to experience firsthand how hard this is and how incomplete the resulting maps are. In practice, however, students spend a significant part of their time on Python scripting the required groundwork to enable this exercise, not the approach design and data interpretation that provides insights and contributes to the learning objectives. Groundwork involves writing scripts to parse traceroute outputs, integrate traceroute with reverse DNS data, construct AS-level paths from BGP, and router-interface-level paths from traceroute data, while merging both perspectives. It also involves writing extensive plotting code using `matplotlib` in Python scripts to quickly visualize various aspects of the data and enhance understanding. This is a typical situation in Internet Measurements and highlights the potential for AI: helping to re-focus coursework on conceptual work and producing insights.

AI opportunity. If used well, AI lowers the barrier to engaging with the conceptual parts of an exercise by absorbing implementation work that is necessary but not the learning objective. Students still need to understand the concepts well enough to *i)* select approaches and defend their choice, *ii)* direct AI tools to realize them into code, *iii)* interpret the output of their analysis scripts, and *iv)* orally present and discuss the chosen approach and the obtained outcomes.

The `matplotlib` plotting code in the topology exercise illustrates this well. Asking an AI tool to *"plot the topology data"* produces generic and often useless output. But a student who understands the measurement problem can write a precise prompt: *produce a CDF plot with five lines, one per vantage point dataset, using these specific colors and dashing schemes, where each CDF shows the distribution of AS path lengths computed as follows from the BGP dumps...*—and quickly obtain a publication-quality figure. The domain knowledge required to write that prompt is exactly the conceptual understanding the course targets: knowing what a CDF communicates, why path length distributions differ across vantage points, and how AS paths are derived from BGP data. A student without that knowledge cannot write the prompt—a student with it saves hours of implementation effort and can iterate on visualizations rapidly. AI is most productive where implementation serves as an enabler and the assessed work is conceptual.

AI policy. AI is thus a helping *tool* and its use can be made explicit in an AI policy: students are responsible for understanding what they submit and expect to be questioned on in the exam—AI is no different than any other tool. AI use is encouraged where it automates work they already understand, and—mirroring industry norms—they remain solely accountable for the correctness and validity of their results.

Paper presented at the ACM SIGCOMM Education Workshop 2026: Networking Education for the AI Generation (A4NE)

Need for domain experience. AI does not remove the need for domain expertise—it makes that expertise the bottleneck, which is where it should be. In our experience with (advanced) networking courses, naively submitting an exercise to an AI tool often yields only superficial results. Without the necessary context and careful prompting, the output often falls short of a complete and correct solution. Knowing *what* to ask, *how* to ask it, and *when* to distrust the answer requires exactly the conceptual understanding that (graudate-level) courses target. Yet, this needs to be made explicit to students and the course needs to provide the right incentives to them.

3 Rethinking Assessment: Grading Experience

If AI can answer (simple) networking questions [1], produce implementations [3] and polished write-ups, then grading those artifacts no longer reliably measures learning. We therefore reorient assessment around two mechanisms: *experience reports* and *oral interaction.*

Experience reports. Each exercise requires students to submit a brief experience report documenting what they did, why they made specific choices, and what they observed or struggled with. We apply a binary grading scheme (adapted from [2]): **0** (fail) if the student did not demonstrate sufficient effort or engagement with the learning objectives; **1** (pass) if the learning objectives of the sheet are fulfilled; and **2** (bonus) to reward work that goes meaningfully beyond the expected outcome. Crucially, a technically correct implementation that lacks documented learning experiences does not receive a **1**. The report is the evidence of learning, not the code. A good report should thus reference specific intermediate results/screenshots only obtainable through genuine engagement. Importantly, it is not independently trusted (as it could also be produced by AI) and understanding is verified in oral interactions.

Oral interaction. Students present and discuss their results in exercise sessions. This format surfaces understanding—or its absence—quickly and fairly. The oral exam reinforces this: it explicitly probes the same exercises, design choices, and conceptual understanding.

We design the oral exam (30 minutes per student) based on a philosophy that shifts the responsibility of demonstrating competency from the teacher to the student, dividing the exam into two parts [2].

(1) The First 7 Minutes: The student is given completely uninterrupted time. The explicit prompt given to them at the start of the course is: *"The first 7 minutes are yours. Present something to show me that you have reached the learning objectives of this course."* [2].
(2) The Q&A Follow-up: The remaining 23 minutes consist of asking deep, reactive follow-up questions. This allows the examiner to assess the student's comprehension, ensuring they did not merely memorize a script. It also evaluates their ability to apply learned competencies to specific scenarios and observes their problem-solving approaches.

In our experience, students that use AI without achieving the learning objectives will not even be able to present a solid 7-minute presentation.

We tell students upfront that polished submissions that cannot be explained in the exam lead to course failure. We have observed exactly this: students who used AI to produce attractive hand-ins but engaged superficially with the material and could not defend their work. The role of exams as a means of ensuring integrity has thus grown with the introduction of AI.

4 Broader Implications

This approach generalizes beyond Internet Measurements, and we apply it to all of our graduate-level courses where implementation serves conceptual objectives.[1] Wherever a course has conceptual or experiential learning objectives that are mediated by implementation work, AI creates the opportunity to decouple the two. We do not need to resist AI in networking education—we need to let it expose where our assessments were/are misaligned. By shifting from (only) grading implementations and documentation to grading learning competencies and learning experiences through experience reports, oral presentations, and exams, we can make AI integration a net positive: it removes routine barriers, surfaces conceptual depth as the central challenge, and produces graduates who can reason about their tools rather than merely operate them. Yet, this necessitates a willingness to reassess the course design.

Broadening access. AI also has a meaningful inclusiveness benefit. Our experience with non-CS students in our courses illustrates this directly: by opening our courses to Electrical Engineering students, who typically have limited programming background, we observed that AI assistance allowed them to clear the implementation barrier and focus on the conceptual work the course targets. Before, EE students often dropped out of our courses due to the required programming load. AI, in this sense, democratizes access to advanced networking courses by decoupling the learning objectives from (substantial) programming proficiency.

[1]Note that undergraduate courses differ fundamentally in their learning objectives. When the goal is to master basic skills that LLMs can trivially produce (e.g., in beginner programming or data structure courses), AI integration presents different challenges and opportunities than in graduate-level courses focused on conceptual understanding.

5 Limitations and Tradeoffs

We acknowledge that this approach is not without cost. Oral interaction is more demanding per student than automated or rubric-based grading, and experience reports require careful reading rather than automated checking. Binary grading helped to mitigate both concerns: it reduced TA grading workload and shifted student focus toward the learning objectives themselves, eliminating unproductive disputes over fractional point deductions on sub-questions that are ultimately irrelevant to whether a student has achieved the course's learning outcomes.

Can oral assessments realistically scale to large classes? Oral examinations are, in our experience, significantly more effective at surfacing genuine understanding than written exams—but they do not scale. For the small graduate courses this paper targets, typically below thirty students, the time investment is manageable and the pedagogical return justifies it. Beyond that threshold, the per-student time cost becomes prohibitive, and written exams—despite their limitations in an AI-rich environment—scale more gracefully. We do not have a satisfying solution to this trade off: oral examination and scale are, at least for now, in tension.

Is this concept universally applicable? Finally, the approach described in this paper is not universally applicable. Deliberately de-emphasizing implementation in assessment only makes sense when implementation is not itself a learning objective. In our Internet Measurements course, producing `matplotlib` plots or parsing scripts is groundwork, not the goal—and offloading it to AI is pedagogically defensible. The same logic does not transfer to courses where writing correct, efficient, or well-structured code *is* the learning objective, such as algorithms, systems programming, or software engineering courses. In those settings, allowing AI to absorb the implementation is not removing a barrier to learning—it is removing the learning itself. The experience reported here should therefore be read as specific to courses where conceptual understanding and reasoning are the primary objectives, and implementation is only the means to reach them.

Should learning objectives be redesigned? The answer depends on the course. For undergraduate courses, we argue they should not: learning to implement things that AI can trivially produce is often precisely the point. This is not new—data structures have long been available in standard libraries, yet implementing them from scratch remains a core learning objective in introductory courses. AI makes this tension more visible, but does not change the underlying pedagogy. For graduate-level courses, however, the question is worth asking seriously: is the coding genuinely a learning objective, or merely the path to one? If it is the latter—as in our Internet Measurements course—then redesigning assessment around the actual objective, and letting AI absorb the implementation path, is not a concession but a clarification. The course becomes more honest about what it is trying to teach.

Acknowledgements

We thank the anonymous reviewers for their detailed and constructive assessments, the many concrete suggestions for improvement, and the thoughtful questions they raised—all of which substantially shaped the final version of this paper.

References

[1] Mubashir Anwar and Matthew Caesar. 2025. Understanding Misunderstandings: Evaluating LLMs on Networking Questions. *ACM SIGCOMM Computer Communication Review* 54, 4 (Oct. 2025).

[2] Kristina Edström. 2023. *The Teaching Trick: How to improve student learning without spending more time teaching.* https://www.youtube.com/watch?v=eImIeynO7pE

[3] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2026. A Survey on Large Language Models for Code Generation. *ACM Transactions on Software Engineering and Methodology* 35, 2 (Feb. 2026).