Openness as a pedagogical affordance: free and open-weight large language models in the preparation of informatics teachers

Vladyslav Ye. Velychko

Donbas State Pedagogical University, 19 Henerala Batiuka Str., Sloviansk, 84116, Ukraine

Abstract. In most school subjects a large language model is something a class consumes; in the preparation of informatics teachers it is something the class must be able to open, inspect, adapt, and govern – the very competencies the discipline obliges its graduates to teach. This dual relation makes the openness of free and open-weight models not a cost-driven substitute for proprietary services but the decisive pedagogical affordance. This conceptual, framework-building paper develops that claim from a structured review of a 202-record Web of Science pool, extended by targeted, DOI-verified supplementation. It re-conceptualizes openness as a vector of six affordances: four descend directly from the software freedoms to run, study, modify, and share – inspectability, adaptability, deployability with data sovereignty, and shareability with licensing – and two are reconstructions demanded by a stochastic, versioned artifact: reproducibility and governability. These affordances are arrayed against four enacted teacher roles – designer, facilitator, monitor, and evaluator – as a claim-graded openness–agency matrix whose every cell is marked as empirically evidenced, transferred from adjacent evidence, or argued as a proposition. Four deployment tiers, T0 through T3, locate where each affordance becomes actualizable; the decisive threshold is the T0-to-T1 transition, where a hosted endpoint gives way to a locally run model and three constrained affordances switch on together. A four-axis governance scaffold – configuration, boundaries, logging, and adjudication – makes deployment documentable and doubles as student-authored coursework. Five falsifiable propositions frame the framework for empirical test. The organizing maxim follows: model selection in teacher education is a curriculum decision, not a procurement decision.

Keywords: open-weight large language models, free software, informatics teacher education, pedagogical affordances, teacher agency, TPACK, data sovereignty, AI competency frameworks

1. Introduction

Informatics teachers occupy a position that no other teaching specialism shares: they stand in a dual relation to the artificial-intelligence systems now entering classrooms. Like every teacher, they may reach for a large language model as a pedagogical instrument – a tutor, a drafting aid, a source of worked examples – and so *teach with* it. Unlike other teachers, they are also obliged to *teach about* it, because generative models are themselves curricular subject matter in computing: their architecture, training, and characteristic failure modes belong to the discipline. A third relation, *teaching for* life in an algorithmic society, adds civic and ethical preparation to the mandate [10, 34, 41]. This triple charge arrives amid documented pressure on informatics teacher education, a field in which the pace of technological change repeatedly outruns both curricula and the readiness of the teaching workforce [36, 48, 68].

When the field asks which models to adopt, it tends to compare open and proprietary systems along three axes: cost, capability, and safety. Recent syntheses of large language models in computing and in education largely inherit this frame, weighing hosted proprietary services against open-weight

0000-0001-9752-0907 (V. Ye. Velychko)
vladislav.velichko@gmail.com (V. Ye. Velychko)
https://ddpu.edu.ua/cc/velychko/ (V. Ye. Velychko)

Received	Accepted	Published	Version of record
2026-02-03	2026-03-18	2026-03-21	2026-03-21

alternatives as broadly interchangeable providers of the same capability [32, 51, 67]. The comparison is a *procurement* framing: it treats a model as a service to be sourced at an acceptable price, quality, and risk, and it asks whether an open substitute is good enough to displace a proprietary default [16]. For most subjects that framing may suffice. For the preparation of informatics teachers it carries a blind spot, because it evaluates the model only as something the class *consumes* and never as something the class must be able to open, inspect, and govern – precisely the competencies these teachers must later teach.

This paper advances a different claim. In the preparation of informatics teachers – uniquely among teaching specialisms – the openness of free and open-weight models is not a cost-driven substitute for proprietary services but is itself the decisive pedagogical affordance. Only open systems let future informatics teachers *enact*, rather than merely be told about, the inspection, adaptation, and governance competencies their discipline obliges them to teach: one cannot rehearse fine-tuning a model one may not modify, cannot audit data flows one cannot see, and cannot govern guardrails one does not own. Openness here is a graded property of a released system rather than a slogan, and most open-weight releases are not, in the strict sense, open source [47]. Reframed this way, the adoption question shifts from sourcing to formation – from what a model costs to what future teachers can *do* with it. In teacher education, then, model selection is a curriculum decision, not a procurement decision.

The lens that makes this visible is not new. A sustained program of free-software didactics has argued that the freedoms to run, study, modify, and share software are not merely economic conveniences but conditions for the kind of professional agency that teacher education should cultivate [13, 62–65]. Open-weight language models extend that argument to a new class of artifact, and the four software freedoms supply the missing vocabulary for what openness affords pedagogically. Methodologically, this is a conceptual and framework-building contribution grounded in a structured, two-stream review of the informatics-teacher-preparation and computing-education-with-LLMs literatures; it synthesizes existing evidence and reasons from it rather than reporting a new empirical dataset. The framework it proposes is therefore offered as an analytic and design instrument – to be instantiated and tested in practice – rather than as an already validated intervention.

The paper makes four contributions. **C1** re-conceptualizes openness as a vector of six pedagogical affordances for informatics teacher preparation (section 5.1, table 2). **C2** arranges these affordances against four enacted teacher roles as an openness–agency matrix of concrete, gradeable practices (table 3). **C3** maps the evidence across two review streams and names the gap the matrix confronts (figure 1). **C4** supplies deployment tiers and a governance scaffold that make the framework operable in real programs (tables 4, 5). The remainder proceeds through the literature review (section 2), the objectives (section 3) and methods (section 4), the results that develop C1–C4 (section 5), and the discussion (section 6), limitations (section 7), and conclusions (section 8).

2. Literature review

2.1. Informatics teacher preparation as a mature, technology-turbulent field

Informatics teacher preparation is not a young field meeting a disruptive technology for the first time; it is a mature discipline whose defining condition has, for four decades, been technological turbulence. Its literature exhibits a persistent duality between the tools of instruction and the content of the discipline, together with a recurring habit of absorbing paradigm shifts by reformulating the competencies a teacher must hold rather than by discarding its foundations. Early work already framed the preparation of informatics teachers as a distinct professional problem, analysing the tension between programming skill and pedagogical understanding [44] and asking which applications and roles should define the teacher's expertise [61]. Problem-based and integrative models followed, aligning instruction with authentic engineering and disciplinary practice [25, 26], while broader surveys mapped the directions and challenges the field would repeatedly confront [24].

A second strand foregrounded fundamental concepts as the stable core beneath shifting tools: functions, objects and states as the conceptual spine of school informatics [23], concept-based pedagogy [20], and the deliberate exposure of core informatics ideas to young learners and their teachers [43]. What should constitute advanced, master's-level preparation in the discipline was posed directly [4]. As the field matured, its attention turned to competency description and curricular modelling: competence frameworks were articulated to make teacher knowledge explicit and assessable [33], informatics education was reframed for "new millennium" learners around contest-based and tinkering didactics [7], and design- and evaluation-oriented laboratory concepts were proposed for teacher training [58]. Studies of professional formation examined how future informatics teachers are prepared [14] and surfaced the questions educators themselves most wanted answered [8]. The same tensions reappear most recently in specifying K–12 computing curricula [68], in the knowledge lower-secondary teachers bring to informatics [36], and in what primary and secondary teachers actually know about the discipline they are asked to teach [48].

Across these eras a single dynamic recurs. Each technological wave – structured programming, objects, parallelism, contest informatics, and now generative AI – is absorbed not by abandoning the field's commitments but by reformulating the competencies expected of the teacher; the field's habitual response to disruption is curricular rather than merely instrumental. This history matters for the present argument, because it establishes informatics teacher preparation as the one specialism whose competencies are defined by the very technologies it must both teach with and teach about. It is against this backdrop that free and open software, and now open language models, arrive as the next paradigm to be absorbed.

2.2. Large language models in computing education: what is known

A rapidly growing body of work examines how large language models affect learning and teaching in computing, and recent syntheses attempt to consolidate it [51, 56, 67]. Two findings are comparatively robust. The first is that the cognitive effect of a model on a learner is task-dependent rather than uniform. When students delegate the generative core of a task – obtaining complete programs or automated repairs on demand – the evidence points to weakened learning and heightened dependence: trivial program generation displaces the productive struggle on which programming pedagogy relies [49], and unstructured use correlates with poorer outcomes and over-reliance [27, 69]. Patterns of help-seeking show learners drifting toward answer-extraction when nothing constrains the interaction [55]. When, by contrast, the model is confined to conceptual clarification – explaining an error, unpacking a concept, withholding the solution – the effect is more favourable: guardrailed tutors that refuse direct answers [31] and context-aware explanations of compiler errors [59] support rather than supplant understanding. The distinction is pedagogical rather than technical: what matters is whether the interaction preserves or removes the reasoning work the learner is meant to do.

The second finding is that models are increasingly useful on the teacher's side of the desk. The corpus documents assistance with code grading [6], the training of teaching assistants through simulated students [39], the generation of practice questions and formative exercises, and broader whole-learner support functions [38], alongside systematic attempts to evaluate whether a model can act as a competent teacher across pedagogical tasks [16]. These uses locate the model as an assistant to professional judgement rather than a replacement for it. Running through both strands is a persistent concern with academic integrity and assessment security, since the same fluency that aids explanation also enables undetectable misconduct [49, 69].

Yet this literature carries a structural blind spot that matters decisively for teacher preparation. Almost without exception, the studies that establish these effects were conducted on proprietary, hosted models accessed through an API. Even the reviews that explicitly foreground open-source alternatives find the empirical base thin and dominated by closed systems [32, 51]. The field therefore knows a great deal about what models *do* to and for learners, and almost nothing about what difference the *openness* of a model makes to that pedagogy; openness is treated, when treated at all, as a matter of cost or deployment convenience rather than as a variable with pedagogical consequences of its own. For most instructional uses this omission is harmless. For the preparation of informatics teachers – who must not only use models but open, inspect, adapt and govern them – it removes from view precisely the property on which their professional formation depends.

2.3. Free software in teacher education and the openness question for models

The conceptual resources for taking openness seriously as a pedagogical property already exist in the free-software tradition. The Free Software Definition names four freedoms – to run, study, modify and share a program [13] – and a sustained program of free-software didactics has argued that these are not merely economic conveniences but conditions for the professional agency that teacher education should cultivate: future teachers who can study and modify their tools relate to technology as authors rather than as consumers [62–65]. That lineage supplies the vocabulary this paper extends from software to language models.

The extension is not automatic, because openness in language models is graded and frequently overstated. The Open Source AI Definition makes the crucial distinction explicit: most releases marketed as "open" distribute only weights, and open-weight is not the same as open source [47]. Weights without training data, code and documentation permit use and adaptation but not full study or reconstruction. It is therefore more accurate to speak of a spectrum of license classes than of a binary, and to ask which of the four freedoms each class actually delivers (figure 3). Five classes organise the field, named here by license and grounding property and illustrated with model *families* as exemplars only. At the open end sit fully-open releases that publish weights together with training data and code, restoring the freedom to study in full; the OLMo, Pythia and BLOOM efforts exemplify this commitment [3, 19, 30]. Next come permissive open-weight releases distributed under standard software licenses such as Apache or MIT [1, 46], whose weights may be used, adapted and redistributed with few restrictions – the Mistral, Qwen, Phi, DeepSeek and Falcon families occupy this class. Restricted open-weight releases attach custom community terms that add acceptable-use or scale conditions [17, 40], as with the Llama and Gemma families, constraining the freedom to share. Research-only releases under responsible-AI licenses permit study but bar commercial classroom deployment [53]. At the closed end, proprietary API models expose no weights at all and confine the teacher to prompt-level use.

The pedagogical consequence is that different license classes preserve different subsets of the four freedoms, and therefore different educational affordances. This is exactly what the existing literature does not ask. Across both the free-software and the LLM-in-education strands, openness is evaluated as cost, capability or deployment convenience: the literature evaluates open LLMs in education by what they *cost and do*, never by what they *let future teachers become*; this paper supplies that missing lens for the one population where it matters constitutively.

3. Research objectives

Existing scholarship appraises free and open-weight large language models chiefly by what they cost and what they can do, rarely by what they enable a future informatics teacher to become. As section 2 established, the pedagogical literature tends to treat openness as an economic or infrastructural attribute and stops short of connecting the concrete properties of these models to the competencies and enacted roles that informatics teacher preparation must cultivate. The result is a missing analytic lens for precisely the population in which those properties matter constitutively.

This paper addresses that gap as a conceptual and framework contribution built on a structured-review base. It synthesizes and reasons from existing evidence rather than reporting a new empirical dataset, and the framework it develops is offered as an analytic and design instrument to be instantiated and tested in practice rather than as an already validated intervention. Four research questions organize the inquiry:

RQ1. Which properties of free/open-weight LLMs function as pedagogical affordances specifically in informatics teacher preparation?

RQ2. How do these affordances map onto the competencies and enacted roles of future informatics teachers?

RQ3. Under which deployment tiers are the affordances practically actualizable by teacher-education institutions?

RQ4. Which governance conditions (integrity, privacy, licensing) must accompany deployment – and how do they themselves become teachable content?

The questions are answered in sequence and map onto the paper's contributions. RQ1 is addressed in section 5.1, which delivers contribution **C1**; RQ2 in section 5.2 (**C2**); RQ3 in section 5.3 (the deployment tiers of **C4**); and RQ4 in section 5.4 (the governance scaffold of **C4**). The cross-cutting evidence gap (**C3**) is delivered not by a single results section but by the review itself – the methods of section 4 and the two-stream flow of figure 1 – together with the gap statement that closes section 2.3.

4. Research methods

The evidence base for this framework was assembled through a *structured review*. In the review typology of Grant and Booth [18], the design sits between a critical review and a scoping review: like a critical review it appraises and interprets a heterogeneous literature in order to build a conceptual model, and like a scoping review it maps the extent and distribution of the evidence through an explicit, reproducible search. The procedure is reported against the applicable items of the PRISMA-ScR reporting guidance [60] and follows the stagewise logic of Arksey and O'Malley's scoping method [2] – identifying the question, locating relevant studies, selecting them, charting the data, and collating the results. It is deliberately *not* presented as a full systematic review: there is no registered protocol, screening was performed by a single reviewer (the sole author), no risk-of-bias grading was applied, and no effect estimates were pooled. These departures are stated openly so that each claim is read at the strength the method warrants and no more.

The primary corpus came from the Web of Science Core Collection. Two topic searches were run and exported: query 1, `TS=(Free OSS LLMs) AND TS=(training) AND TS=(informatics teachers)`, returned 100 records, and query 2, `TS=(large language models) AND TS=(training informatics teachers)`, returned 102 records. Merging the two exports identified 202 records spanning 1971–2026; 2 duplicates were removed, leaving 200 records for screening. Title-and-abstract screening against the criteria below retained 84 records and excluded 116. Figure 1 traces this flow from the two exports through de-duplication, screening, targeted supplementation, and synthesis.

Records were included when they addressed large language models in relation to education or teacher preparation with content bearing on informatics, computer science, or programming pedagogy, and when a full text or a resolvable DOI was available. Records were excluded when they fell outside this scope or when they clustered into the recurrent noise themes that the topic queries inevitably surfaced. The 116 exclusions resolved into five reason clusters: out-of-scope items (54); computer-assisted language learning and foreign-language teaching (24); model-engineering and systems work with no pedagogical framing (23); isolated medical applications (11); and records with no retrievable full text or DOI (4). These counts sum to the 116 excluded and are recorded cluster by cluster in figure 1, keeping the boundary between the framework's concern and its adjacent literatures auditable.

Because a database-bounded search under-represents fast-moving open-model scholarship, the two Web of Science streams were supplemented by a third, targeted stream. This supplementary stream drew on academic-search and preprint (arXiv) channels through documented queries and contributed 43 candidate items, each admitted only after its DOI had been verified through Crossref;

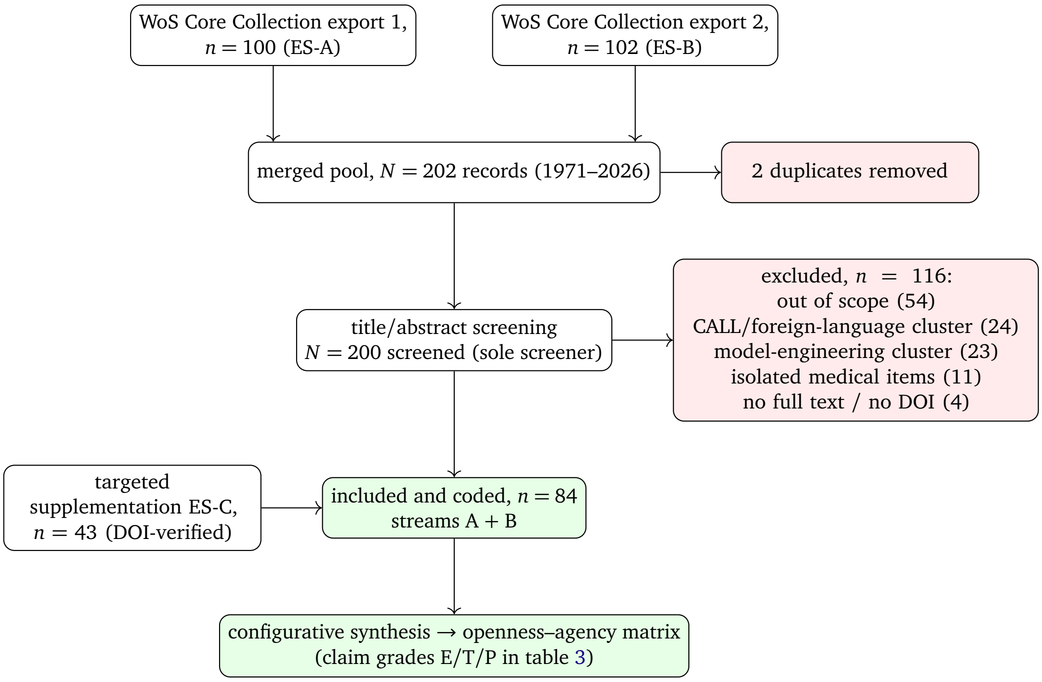

Figure 1: Two-stream structured-review flow: provenance, screening with named noise-cluster exclusions, targeted supplementation, and configurative synthesis. The design is reported against PRISMA-ScR items where applicable without claiming a systematic review [18, 60].

items without a verifiable DOI were not carried forward. The supplementation widened coverage without relaxing provenance, so that every additional source meets the same citation standard as the database corpus. The two-stream configuration is therefore configurative by construction: the database corpus establishes the mapped extent of the field, while the targeted stream repairs its known temporal blind spots.

Included records were charted with the coding scheme summarized in table 1, which combines deductive families – affordances, enacted teacher roles, curricular axis, education level, and deployment tier – with inductively derived emergent themes such as integrity, cognitive offloading, equity, and multilinguality. The synthesis is *configurative* rather than aggregative: evidence is arranged to build and interrogate concepts, namely the affordance vector and the openness–agency matrix, rather than to pool comparable outcomes into a single estimate. Each cell of the resulting matrix carries a claim grade recording the strength of its support – E for cells anchored in direct empirical evidence, T for theoretically motivated but not yet empirically demonstrated claims, and P for practice-reported or plausible expectations. Grading every cell keeps the framework's confident claims and its conjectures visibly distinct.

Finally, the review applied an explicit quarantine protocol to generative tooling. Two AI-generated research digests were consulted as scoping aids only: they helped orient the search and suggest structure, but they were never cited, never paraphrased as fact, and contributed no facts and no citations to this paper. Every factual claim was traced back to a DOI-verified primary source, and any system or result named only in a digest – but not confirmable in a primary – was dropped or re-labelled as this paper's own synthesis rather than repeated as established fact. This step ensures that the provenance of each assertion is a verified, human-authored primary source and never the scoping aid itself.

Table 1
Coding scheme of the structured review and screening summary.

Code family	Values	Type
Affordance	A1–A6 (openness–agency matrix rows)	deductive
Enacted role	Designer / Facilitator / Monitor / Evaluator	deductive
Curricular axis	about-AI / with-AI / for-AI	deductive
Education level	school CS / pre-service / in-service	deductive
Deployment tier (context)	T0–T3	deductive, optional
Emergent themes	integrity; cognitive offloading; equity; multilinguality	inductive

Screening summary (from the deposited log): from two Web of Science exports, 202 records were identified; 2 duplicates were removed, leaving 200 screened; of these, 84 were included and 116 excluded across five reason clusters – out of scope (54), CALL/foreign-language (24), model-engineering (23), isolated medical (11), and no full text or DOI (4). Every included record carries at least one code.

5. Research results

5.1. The openness construct: from four freedoms to an affordance vector

An affordance is not a property of a thing but a relation between a thing and an agent acting toward a goal. Gibson [15] introduced the term to name what an environment offers an animal, insisting that the offer is neither purely objective nor purely subjective but relational; Norman [45] carried it into design, distinguishing the affordances a system really has from those it merely signals; and Kirschner [28], followed by Hammond [22], transposed it to education, where a technology's affordances are the action possibilities it opens for a learner or teacher pursuing a pedagogical goal. The lesson these authors share is that an affordance is a three-place predicate – property × agent × goal – and therefore cannot be read off a system in the abstract. The same released model affords different things to a data-entry clerk, a physician, and a computing teacher, because each brings a different goal and a different competence to the encounter. This is why the present thesis is deliberately population-specific. The claim is not that openness is pedagogically decisive for every teacher, nor for every subject; it is that for the preparation of informatics teachers – whose disciplinary goals include inspecting, adapting, and governing the very systems they use – openness becomes the property that converts a tool into an object of professional formation. Stated for any other population the claim would overreach; stated for this one it is exact.

What, concretely, does an open release make available for such a relation to form? The framework distinguishes a layer of grounding properties – the objective, agent-independent facts about a released system – from the affordances those properties can support (figure 2, left). The grounding layer is inherited directly from the free-software tradition, whose didactic program argued that the freedoms to run, study, modify, and share a program are not economic conveniences but the conditions of an authorial, rather than consumerist, relation to technology [13, 62, 63, 65]. For a language model the corresponding properties are the released weights, tokenizer, and architecture; the model and data cards; the license class; commodity runnability on ordinary hardware; the ability to pin a checkpoint; and the rights attached to derivatives. None of these is yet an affordance: weights on a disk afford nothing until an agent with a goal acts on them. But without them no inspection, adaptation, or governance is possible at all, and a closed API withholds every one.

Mapping the four freedoms onto these properties yields four of the six affordances directly. The freedom to run grounds A3, deployability and data sovereignty; the freedom to study grounds A1, inspectability; the freedom to modify grounds A2, adaptability; and the freedom to share grounds A6, shareability and licensing (table 2). The transfer, however, breaks in two revealing places, and the breakage is not a defect but the source of the framework's two remaining affordances. First, weights are not source code. A released model may be run, adapted, and redistributed while its training data and training code remain undisclosed, so the freedom to study is delivered only partially: one can

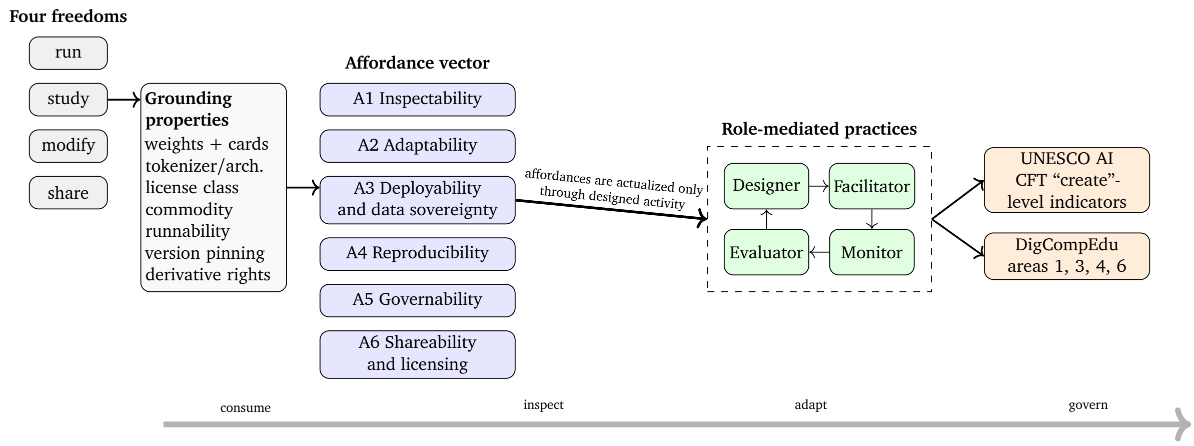

Figure 2: The openness–agency framework: the four software freedoms ground the openness properties of open-weight models; these afford the six-dimensional affordance vector A1–A6; the affordances are actualized only through designed activity (relational affordance theory [15, 28]) enacted through four teacher roles, producing competency outcomes; the whole is crossed by the consume → inspect → adapt → govern progression.

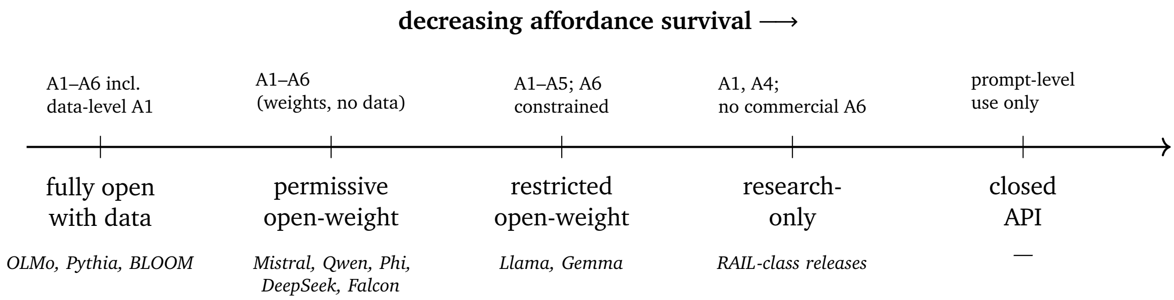

Figure 3: The openness spectrum by license class, annotated with the affordances that survive each threshold; model *families* appear as class exemplars only (no versions, no scores). Per the Open Source AI Definition, most open-weight releases are not open source [47].

probe behaviour and read cards but cannot reconstruct the artifact. The Open Source AI Definition makes exactly this distinction, and it is why most releases marketed as open are open-weight rather than open source [47]; the fully-open efforts that publish data and code alongside weights – OLMo, Pythia, BLOOM – are the exception that restores study in full [3, 19, 30], and they anchor the open end of the license spectrum (figure 3). Second, a language model is stochastic and versioned in ways a compiled program is not. Running the same source twice yields the same behaviour; querying a model twice need not, and a hosted model may change silently beneath a stable name. The four freedoms never had to address decoding randomness or version drift because software did not exhibit them. Two LLM-era reconstructions close the gap: A4, reproducibility – the capacity to pin a checkpoint and its decoding parameters so that classroom conditions repeat – and A5, governability – the capacity to own guardrails, logs, and policies at institutional scale rather than accept vendor-set ones. Both are recognisably descendants of run, study, and modify, rebuilt for a stochastic artifact operated at the scale of a program or a faculty.

It follows that "open-source LLM", used as a binary label, is analytically useless for education. The label collapses a graded spectrum of license classes onto a single bit and, worse, says nothing about what a given release lets a given agent do. Two models bearing the same badge may differ in whether their data is inspectable, their weights adaptable, or their derivatives shareable; two teachers facing the same model may actualize entirely different affordances. The pedagogical weight is therefore carried not by the label but by the affordance vector – the six-dimensional, standards-anchored profile of table 2, which ties each affordance to specific UNESCO, DigCompEdu, and TPACK

Table 2
The six openness affordances for informatics teacher preparation: grounding, freedom lineage, and standards anchoring (UNESCO AI CFT dimension/level [41]; DigCompEdu area [52]; EDEH axis; TPACK locus [42]).

Affordance	Grounding property / freedom lineage	Pedagogical meaning for future informatics teachers	Standards anchoring
A1 Inspectability	released weights, tokenizer, architecture, model/data cards; *study* (partial: weights without training data)	the model as object of study: probing, tracing tokenization, reading cards critically (teach-about)	UNESCO: foundations, deepen→create; DigCompEdu 3, 6; EDEH: about; TPACK: TK∩CK
A2 Adaptability	fine-tuning/LoRA, quantization, system prompts on an owned stack; *modify*	the model as constructionist object-to-think-with: building course-scoped assistants; localization including Ukrainian	UNESCO: AI pedagogy, create; DigCompEdu 3; EDEH: with/about; TPACK: TCK
A3 Deployability and data sovereignty	local/offline execution; *run*	governing where learner data lives; exam-safe, air-gapped, connectivity-resilient operation	UNESCO: ethics, deepen; DigCompEdu 1; EDEH: with/for; TPACK: TPK
A4 Reproducibility	pinned checkpoint + decoding parameters; reconstruction of *study* under stochasticity and silent API drift	scientific method applied to stochastic systems; repeatable classroom conditions	UNESCO: foundations, create; DigCompEdu 4; EDEH: about/with; TPACK: TPK
A5 Governability	ownable guardrails, logs, policies; reconstruction of *run/modify* at institutional scale	institutional agency: course-owned rules instead of vendor-set ones; governance as teachable content	UNESCO: ethics + pedagogy, create; DigCompEdu 1, 4; EDEH: for/with; TPACK: TPK + ethics
A6 Shareability and licensing	permissive licensing of derivatives; license-class literacy; *share*	commons-based professionalism: publishing fine-tunes, prompts, curricula; rights and duties by example	UNESCO: professional learning, create; DigCompEdu 1, 6; EDEH: for; TPACK: TK + ethics

targets [41, 42, 52]. Reframing openness as this vector, rather than as a badge or a price, is the first contribution and the precondition for everything that follows.

5.2. The openness–agency matrix

If the affordance vector answers what openness offers, the openness–agency matrix answers who actualizes it and how (table 3, figure 2). Its rows are the six affordances; its columns are four roles a future informatics teacher enacts – Designer, Facilitator, Monitor, and Evaluator. The four-role scheme is this paper's synthesis rather than an imported taxonomy. It draws together two established lines of scholarship: the tradition, articulated most fully by Laurillard [29], that casts teaching as a design science in which the teacher is the designer of learnable activity; and the orchestration literature, in which Dillenbourg [9] analyses the teacher's real-time management of a multi-plane classroom. Designer and Facilitator descend from these precedents directly; Monitor and Evaluator extend them to the specific obligations a stochastic, governable model imposes – the continuous auditing of behaviour and the adjudication of automated decisions. The four are not disjoint job titles but a cycle a single teacher moves through: designing an activity, facilitating it live, monitoring its conduct, and evaluating its products, then redesigning. Crossing the six affordances with the four roles yields a 6 × 4 grid of twenty-four cells, each naming one concrete, gradeable practice rather than a disposition, so that the matrix functions as a design instrument and not merely as a description. Each cell, moreover, carries a claim grade – E, T, or P – recording how far its practice is yet backed by evidence, so that the grid's confident claims and its conjectures remain visibly distinct throughout the walkthrough that follows.

Table 3
The openness–agency matrix: enactable teacher-education practices by affordance (rows) and enacted role (columns). Claim grades: E = empirically evidenced by at least one verified study; T = transferred from adjacent evidence; P = proposition, argued conceptually. Bold cells are the four flagship practices instantiated in the vignette (section 6).

	Designer	Facilitator	Monitor	Evaluator
A1 Inspectability	select models by critically reading model/data cards; anticipate capability limits and bias in lesson design (T)	demonstrate the mechanism (tokenization, sampling) live on the actual artifact (T)	**hallucination-audit lab: trace failure cases to mechanism, not folk theories (T)**	judge whether students' explanations of model behaviour are mechanistically correct (P)
A2 Adaptability	**fine-tune/prompt-engineer a small open model into a syllabus-scoped Socratic tutor that refuses full solutions (T)**	switch scaffolding modes mid-course (Socratic / no-direct-solution) (T)	run regression probes: compare base vs. adapted behaviour for side-effects (P)	build rubric-aligned evaluator configurations; calibrate against human grading (T)
A3 Deployability and data sovereignty	architect deployment so learner data never leaves the institution (T)	run offline/air-gapped class instances; use authentic data without third-party exposure (T)	audit the deployment's data flows; verify zero telemetry/egress (P)	assess under connectivity-independent, equitable conditions (same local model for every student) (P)
A4 Reproducibility	build repeatable lab configurations (pinned weights, decoding params, seeds) (T)	guarantee all students face identical model behaviour (task fairness) (P)	re-run flagged interactions to verify misconduct or hallucination claims (P)	**make automated feedback re-executable and contestable; documented rubric overrides (T)**
A5 Governability	encode course policy as an executable configuration (system prompt + filter + logging rules) (T)	**students draft and defend the class usage policy, implemented as an actual system prompt and filter (P)**	audit anonymized logs for integrity risks and over-reliance; tighten course-owned guardrails (T)	adjudicate flagged cases under human-override rules; retain pedagogical authority over automated decisions (T)
A6 Shareability and licensing	choose license-compatible models/datasets for the intended educational use (T)	teach reuse rights and duties by example: publish course configs under open licenses (T)	maintain a license inventory; detect violations (P)	assess student artifacts (fine-tunes, datasets, prompts) for license compliance and attribution (P)

Read across its columns and down its rows, the matrix traces a progression of deepening agency: consume → inspect → adapt → govern (figure 2, lower band). Generic frameworks for teachers' digital competence, and the professional-development offerings built on them, concentrate their mass at the first two stations. They prepare a teacher to use a digital tool responsibly and, at their most ambitious, to look inside it and explain it – to consume and to inspect. That is appropriate for specialisms in which the model is only an instrument. For informatics teachers it stops precisely where the discipline's obligations begin. The centre of gravity of the present matrix lies deliberately at the far end of the progression, in adapt and govern: in modifying a model to serve a pedagogical purpose (the Adaptability row) and in owning the rules, records, and remedies under which it operates (the Governability row). These are not enrichment activities layered on top of consumption; they are competencies the informatics curriculum obliges the teacher to teach, and the matrix's claim is that they can be enacted – not merely described – only on an open artifact. The progression thus doubles as a diagnostic: a deployment that permits only the first two stations is, for this population, under-resourced by definition.

Consider one cell per role, beginning with the Designer. The flagship Designer practice sits at the intersection with Adaptability: the teacher fine-tunes or prompt-engineers a small open model into a syllabus-scoped Socratic tutor that withholds full solutions and instead provokes the student's own reasoning (table 3, bold). The pedagogical value of such a guardrailed tutor is not speculative – guardrailed tutors that refuse direct answers and error-explanation aids that unpack rather than repair have been shown to support programming learning [31, 59]. But that evidence was gathered on proprietary, hosted models configured through their APIs; the practice of building the tutor on an owned, modifiable model – of the teacher exercising the freedom to modify rather than renting a vendor's configuration – has not itself been studied. The cell is therefore graded T, transferred: its pedagogical logic is evidenced, its instantiation on an open artifact is not. The Designer's work here is architectural – deciding what the model may and may not do before any student meets it.

The Facilitator acts in the live classroom, and the flagship Facilitator practice sits at the intersection with Governability: students draft and defend the class usage policy, which is then implemented not as a poster on the wall but as an actual system prompt and content filter running on the class's own deployment. Governance ceases to be a lecture and becomes an artifact the students author, negotiate, and see take effect. No verified study yet demonstrates this practice, on open or closed models alike; it is graded P, a proposition argued from the affordance and from the orchestration literature's insistence that classroom rules are enacted in real time rather than declared in advance [9]. It is nonetheless among the framework's sharpest claims, because it is precisely the practice that a proprietary deployment forbids: one cannot hand students the guardrails of a model whose guardrails belong to a vendor.

The Monitor watches the system in operation, and the flagship Monitor practice sits at the intersection with Inspectability: a hallucination-audit laboratory in which students trace a model's failure cases back to mechanism – tokenization, sampling, distributional gaps – rather than to folk theories about a machine that "lies" or "knows". The activity turns the model's errors into the curriculum, and it presupposes inspectable internals: one audits what one can open. Evidence that learners drift toward answer-extraction and over-reliance when nothing constrains the interaction [27, 55] motivates monitoring as a standing practice rather than an occasional check, but the specific audit-to-mechanism lab has not been evaluated, so the cell is graded T. The Monitor role is where the informatics teacher's dual mandate is sharpest: the same failure a general user would route around becomes, in this classroom, the object of study.

The Evaluator judges products and closes the cycle, and the flagship Evaluator practice sits at the intersection with Reproducibility: automated feedback is made re-executable and contestable, with every model-assigned assessment reproducible from a pinned checkpoint and decoding parameters and every rubric override documented. A grade a student cannot re-run is a grade a student cannot appeal; reproducibility is what makes automated evaluation compatible with due process. The use of models to assist grading and to train assessment judgement through simulated students is documented in the corpus [6, 39], which is why the cell is graded T rather than P; what the open setting adds – and what studies conducted on drifting hosted models cannot guarantee – is the pinned, repeatable substrate that makes a feedback decision auditable after the fact. Evaluation then feeds back into design, and the role cycle begins again.

Two features of the completed matrix deserve emphasis. The first is its standards anchoring: every cell is tied through its row to specific targets in the UNESCO AI competency framework for teachers, the DigCompEdu areas, and the TPACK knowledge types [41, 42, 52], so that the practices are not idiosyncratic to this paper but map onto instruments a program is already accountable to. The second, stated plainly, is that no cell in the matrix reaches grade E. Fifteen cells are graded T and nine P; not one rests on direct empirical evidence, because peer-reviewed evidence for these practices conducted on open-weight models does not yet exist. Every E-eligible study in the corpus was run on a proprietary, hosted system. This is not a weakness the framework must apologise for; it is the framework corroborating, from the inside, the very gap the review identified – the field knows what models do to learners and almost nothing about what their openness affords teacher formation. The empty E column is that gap made visible cell by cell, and it is what the falsifiable propositions of section 6 are designed to convert into evidence. The grading also makes the matrix falsifiable by construction: because each cell carries its own claim and its own grade, a reviewer may contest any single practice without collapsing the whole. The matrix survives the loss of individual cells; what it cannot survive, and what would refute it, is a demonstration that openness makes no difference to any of them.

5.3. Deployment tiers as degrees of affordance actualization

Affordances are potential until a deployment actualizes them, and deployments differ. Table 4 arranges four tiers, T0 through T3, defined not by the hardware they name but by which affordances each one switches on. The capacity classes are described qualitatively and deliberately without products, prices, or benchmark figures: a hosted open-weights endpoint (T0); a personal device with at most a single consumer GPU running a quantized small-model class through a local runner of the llama.cpp or Ollama kind (T1); a department lab server serving a mid-size class to many users at once through a batched server of the vLLM kind (T2); and an institutional cluster serving the largest model classes faculty-wide (T3). The economic logic tracks the same ladder: T0 trades a capital outlay for a per-call operating cost and a standing dependency, while T1 and above substitute an owned, largely one-time capacity for recurring fees and external reliance. The point of the table is not to recommend a machine but to show that each affordance has a threshold – a minimum deployment below which it simply cannot be enacted, however open the model's license.

The decisive transition is from T0 to T1. A hosted open-weights endpoint already delivers shareability and a partial inspectability – one can read the cards and study the license – and it is a reasonable site of first contact with open artifacts. But three affordances that a hosted endpoint leaves dark or only partial – A3 and A5 dark, A4 only partial at T0 – reach full or partial actualization together at T1: A3, deployability and data sovereignty, because only local execution puts learner data under institutional control; A4, reproducibility, because only an owned deployment can pin a checkpoint against silent drift; and A5, governability, at least in part, because only an owned stack can carry course-set guardrails and logs. This is the empirical heart of the tier analysis and it carries a conceptual payload: the transition proves that openness is not the same as locality. A model's license may grant every freedom, yet a hosted endpoint leaves the freedoms that depend on controlling the running system unexercised. Openness is necessary but not sufficient; a tier that actualizes it is

Table 4
Deployment tiers as degrees of affordance actualization (● actualized, ◐ partially actualized, ○ not actualized). Capacity is described only as classes; no products, prices, or benchmark scores.

Tier	A1	A2	A3	A4	A5	A6	Typical teacher-education use
T0 hosted open-weights endpoint	◐	◐	○	◐	○	●	first contact with open artifacts; card-reading exercises
T1 personal device (CPU or single consumer GPU; small-model quantized class; llama.cpp/Ollama-class runner)	●	●	●	●	◐	●	individual labs: deploy, probe, adapt, pin
T2 department lab server (mid-size class; multi-user batched serving, vLLM-class)	●	●	●	●	●	●	whole-class concurrent use under course policy
T3 institutional cluster (large/MoE class)	●	●	●	●	●	●	faculty-wide serving; program audits

required as well. The practical consequence is reassuring rather than utopian. The minimum viable tier for the teacher-education curriculum is modest: T1 suffices for the individual laboratories in which a future teacher deploys, probes, adapts, and pins a model on a personal device, and T2 – a single departmental server – suffices to add whole-class concurrency under a shared course policy. Neither T3 nor exotic hardware is a precondition for the framework; the affordances that matter most are actualized at the low end of the ladder.

These tier economics acquire a particular sharpness in the setting for which the framework was written. Ukrainian pedagogical universities have, since 2022, conducted teacher preparation under conditions of disrupted connectivity, displacement, and recurrent interruption to campus infrastructure – circumstances documented in studies of higher education under the pandemic and under martial law [54, 57]. For such institutions the low-tier affordances are not a matter of convenience but of continuity. A model that runs on a personal device or a departmental server keeps working when the network does not, so that an air-gapped or offline laboratory can proceed through an outage that would suspend any hosted service; the same locality that secures data sovereignty also secures connectivity-resilient instruction. And because learner data never leaves an institutionally controlled machine, deployment aligns with the data-protection obligations these universities carry, without routing the work of future teachers and their students through infrastructure beyond national jurisdiction. The framework's minimum viable tier and the operational realities of Ukrainian teacher education thus point to the same modest, owned deployment – a convergence of pedagogical principle and practical necessity rather than a compromise between them.

5.4. Governance: integrity, privacy, licensing – and governance as curriculum

Governance is the fourth affordance made operational, and it is where the mismatch between a general-purpose safety layer and a computing classroom is most acute. A vendor's guardrails are tuned to a global average user and optimised to avoid categories of harm that have little to do with a programming course; they will refuse or hedge on legitimate material – exploit code discussed for defensive purposes, say – while remaining indifferent to the one behaviour a computing educator most needs to shape: the student who asks the model to do the assessed work. Off-the-shelf safety under-fits the classroom in both directions at once. What the setting requires instead is a domain taxonomy of prompts, and the matrix's integrity concerns suggest a three-class one, offered here as this paper's synthesis of the academic-integrity findings in the reviewed corpus. A prompt is *irrelevant* when it falls outside the course's scope; *relevant-safe* when it seeks legitimate help – a concept explained, an error unpacked, a design discussed – of exactly the kind the guardrailed-tutor evidence shows to support learning; and *relevant-unsafe* when it seeks to offload the graded work itself: a demand for a full solution, the completion of an assessed task, or a jailbreak crafted to defeat the tutor's restraint. The boundary that governance must police is not safe-versus-unsafe in the vendor's sense but this discipline-specific line between help that teaches and help that substitutes.

Policing that line calls for documentation, and the framework supplies it as a four-axis orchestration scaffold (table 5): Configuration records the pinned checkpoint and decoding parameters; Boundaries records the system prompts, the no-direct-solution guardrails, and the three-class taxonomy as applied; Logging records the local, anonymized, minimized interaction records and their retention rules; and Adjudication records the human-override rules and the appeal path for any automated decision. Together the four axes are the documentation norm a serious deployment should meet. Their distinctive feature in a teacher-education setting is that the same scaffold report doubles as a student-authored artifact. The future teacher does not merely receive a governance document; the future teacher writes one – specifying a configuration, drafting boundaries, defining a logging and retention policy, and designing an adjudication procedure – as an assessed product of the course. Governance thereby closes an isomorphism loop: the artifact a teacher must one day maintain over a classroom deployment is the very artifact the teacher authors while learning, so that the competence and its documentation are acquired in a single act. Each axis, as the table records, trains specific matrix cells, tying the scaffold back to the practices it operationalizes.

Table 5
The four-axis governance/orchestration scaffold: the documentation norm for teacher-education deployments, doubling as a student-authored artifact.

Axis	What is documented	Who exercises it	Matrix cells it trains
Configuration	pinned checkpoint id/hash + decoding parameters (temperature, top-p, seed)	instructor; students in Designer role	A4×Designer, A4×Facilitator
Boundaries	system prompts, no-direct-solution guardrails, the three-class prompt taxonomy	instructor + students (policy drafting)	A2×Facilitator, A5×Designer, A5×Facilitator
Logging	local anonymized interaction logs; retention and minimization rules	course team; students in Monitor role	A5×Monitor, A3×Monitor
Adjudication	human-override rules for automated decisions; appeal path	instructor; students in Evaluator role	A5×Evaluator, A4×Evaluator

Crucially, all four axes are course-owned only under open deployment. A hosted service exposes, at best, a configuration dial and a usage dashboard; it does not surrender its guardrails, its raw logs, or its adjudication authority to a course. This is what the framework calls *artifact-level access*: possession of the model's own weights, its own guardrails, its own logs, and offline reproducibility – the condition under which every governance axis becomes an object the course can inspect, modify, and own rather than a setting a vendor exposes. Artifact-level access is the deployment counterpart of the openness construct, and it is the precondition for governance to be taught by enactment rather than by description. The governance content is not confined to pedagogy: it is where the framework meets law. Local, minimized logging operationalizes the data-minimization and purpose-limitation principles of the General Data Protection Regulation [11]; the open-source provisions of the European Union's Artificial Intelligence Act bear directly on how open-weight models may be deployed and documented in an educational setting [12]; and for the institutions in view, Ukraine's Law on Personal Data Protection governs the handling of learner data on owned infrastructure [66]. Treated as legal primaries rather than as background, these instruments make the governance scaffold auditable against external obligations – and, in the process, turn compliance itself into teachable content, the final demonstration that in this specialism governance is not a constraint on the curriculum but part of it.

6. Discussion

The argument of this paper can be stated as three premises and a conclusion. The first concerns the object of the competency. Informatics teachers stand in a dual relation to generative systems: they may *teach with* a model as a pedagogical instrument, and they are obliged to *teach about* it because the model's architecture, training, and characteristic failure modes are themselves curricular subject matter, to which a third, civic charge of teaching for life in an algorithmic society is added [10, 41]. Call this the *dual-relation premise*. Its force is that the teach-about and teach-for functions, unlike teach-with, cannot be discharged by fluent use alone; they require the model to be available as an object of study and control, not merely as a source of outputs.

The second premise concerns how the competency is acquired. Teacher education works by modelling: future teachers teach, to a considerable degree, as they were themselves taught, having absorbed the practices they witnessed across years of schooling long before formal preparation begins [35], and teacher educators exert their strongest influence when they make their own pedagogical reasoning explicit and enact the practices they advocate [37]. Call this the *isomorphism premise*. It entails that a program cannot install a disposition it does not itself enact: a cohort taught to consume a sealed service will carry consumption, not stewardship, into their own classrooms, whatever the syllabus asserts about openness.

The third premise concerns the material conditions of enactment. The practices the first two premises demand – inspecting an artifact's internals, adapting its weights and behaviour, deploying it under one's own governance, reproducing a fixed configuration, and sharing a derivative – are afforded only by open, or at least open-weight, systems; a closed interface affords consumption and prompt-craft but structurally withholds the remainder [47]. Call this the *affordance-asymmetry premise*. The three together yield the conclusion: openness is not a procurement preference but a pedagogical affordance constitutive of informatics-teacher competency. And because affordances are relational rather than intrinsic – a property realised only in the coupling of an artifact with an agent's designed activity [15, 28] – the conclusion carries a corollary: openness enables the competency but does not deliver it. It must be actualised through deliberately designed activity, which is exactly what the openness–agency matrix (table 3) specifies.

Four objections deserve a direct answer. The first is *capability*: proprietary systems are more capable, so surely they serve learners better. But for the teach-about function capability is secondary to inspectability – a smaller model whose mechanism can be opened teaches more about how such systems work than a stronger one that cannot be examined – and scoped, fine-tuned small open models already reach task-sufficient quality in computing-education settings such as context-aware compiler-error explanation [59]. The tier scheme, moreover, is not abolitionist: it retains closed tools at the consumption tier for the teach-with function they serve well. The apparent counterexample of a vendor fine-tuning API, which seems to offer adaptation without openness, dissolves under the definition of *artifact-level access* established in section 5.4: uploading data to tune a hosted endpoint yields neither the model's own weights and guardrails, nor its raw logs, nor offline reproducibility, so it leaves the adapt, govern, and reproduce affordances as dark as before.

The second objection is *gradient*: openness is a matter of degree, not a binary, so a clean instrument/service dichotomy misdescribes it. This is conceded and absorbed rather than resisted. The framework already treats openness as a six-dimensional vector partitioned by license-class thresholds (figure 3), and recognising the gradient sharpens the thesis, because it lets a program locate the precise dimension and threshold at which a given affordance switches on.

The third objection is *feasibility*: open deployment seems to demand infrastructure many programs lack. But entry sits on consumer hardware at zero marginal cost, and the further resilience of local, offline operation to connectivity disruption is argued in the tier analysis of section 5.3, whose account need not be restated here; the decisive T0-to-T1 transition requires only a personal device.

The fourth objection is *actualisation*: access is not learning, and possessing an open model guarantees no competency. This too is conceded by design, and is in fact why the framework is built as it is. Because affordances are relational, access is a necessary but insufficient condition, and the matrix and the governance scaffold exist precisely to actualise it – converting a possessed artifact into enacted practice through designed roles and assessed products.

Two scope conditions bound these claims – they concern informatics teachers specifically, not all educators, and preparation programs, not school procurement at large – and within that scope the framework refines four established models rather than displacing them. It moves the technological-knowledge locus of TPACK from consuming a service to stewarding an artifact, recasting the boundary between technological and content knowledge for a discipline whose content includes the tool [5, 42]. It adds to SAMR an orthogonal control/agency axis, so that a substitution-level task on an open stack can build more teacher competency than a redefinition-level task on a closed service – an extension consonant with the critique that SAMR under-theorises context and process [21, 50]. It operationalises the *create* level of the UNESCO AI competency framework as artifact-level creation, which is structurally out of reach on a closed stack [41]. And it supplies the deployment substrate that the professional and learner competences of DigCompEdu presuppose [52].

The framework is offered as testable. Five falsifiable propositions follow, each stated so that a well-designed study could refute it.

P1. Pre-service informatics teachers who complete an open-stack module (tier T1 or higher) score higher on mechanism-explanation tasks and on UNESCO create-level indicators than peers trained on prompt engineering alone.

P2. The breadth of matrix-role coverage a program provides predicts its pre-service teachers' self-efficacy for teaching *about* AI.

P3. Course-owned guardrails reduce policy-violating completions without depressing legitimate conceptual help-seeking, relative to generic vendor filters.

P4. Deployment tier predicts the breadth of affordance actualisation observed in program audits: T0 programs cluster at the consume and inspect affordances, whereas T1-and-above programs reach adapt and govern.

P5. Reproducibility practices – pinned weights and fixed decoding – transfer to students' scientific habits when they investigate other stochastic systems.

An illustrative design, not an implemented case: a four-week open-stack module in a bachelor pre-service informatics-teacher course. The module is a design sketch, offered to make the framework concrete rather than to report a trial; no cohort has yet run it. Across four weeks it follows the consume → inspect → adapt → govern band, and its spine is the four flagship cells of table 3. *Week one (consume, then inspect).* Students first use a small open model as ordinary consumers, then turn to critique: they read its model and data cards against its observed behaviour, and run a hallucination-audit lab that traces particular failures to mechanism – tokenization, sampling, gaps in training coverage – rather than to folk theories about the machine (A1). *Week two (adapt).* Working from those internals, each student adapts the base model, by system prompt or a lightweight fine-tune, into a syllabus-scoped Socratic tutor that questions and hints but refuses to hand over full solutions, then runs regression probes that compare the adapted behaviour against the base to surface any pedagogical side-effects of the change (A2). *Week three (deploy and pin).* Students deploy the tutor quantized and offline on lab hardware, so that authentic learner data never leaves the room, and verify that the deployment emits no telemetry (A3); they then pin its checkpoint hash and decoding parameters – temperature, top-p, seed – so that every classmate meets identical behaviour and any flagged interaction can be re-run exactly (A4). *Week four (govern and share).* Students draft, implement, and audit the class usage policy on a week of anonymized logs, adjudicating AI-graded submissions under human-override rules and documenting each contested case (A5), and close by choosing a compatible open license and publishing the resulting course configuration under it, so the next cohort can reuse and revise their work (A6).

This vignette is the concrete instantiation of the very open-stack module whose learning effect proposition P1 is framed to test.

7. Limitations

Five limitations bound the claims advanced here, and stating them plainly is part of the framework's own discipline of grading confidence. First, this is a conceptual and framework-building contribution that reports no effectiveness data of its own: it synthesizes existing evidence and reasons from it, but it does not measure whether an open-stack module produces the competencies the matrix specifies, and so it offers a design instrument awaiting trial rather than a validated intervention. Second, its evidence base is a structured, not an exhaustive, review. The primary corpus originated in a single database, title-and-abstract screening was performed by a single reviewer (the sole author), no protocol was registered, no risk-of-bias grading was applied, and no effect estimates were pooled – departures from full systematic-review procedure that follow from the chosen review type [18, 60] and that a targeted supplementary stream mitigates but does not eliminate. Third, the model landscape moves faster than any literature can track, so specific systems named today may be superseded before print; the framework mitigates this by pitching its claims at the level of affordances and license classes rather than particular models, invoking model families only as exemplars, so that the affordance vector should outlast the releases that currently populate it. Fourth, the coding and synthesis were interpretive and single-authored: the affordance vector, the four-role scheme, and the cell-by-cell grading reflect one reasoned reading of the corpus, and a second coder could partition or grade differently. Fifth, and most consequentially, the framework is unvalidated. No cell in the openness–agency matrix (section 5.2) reaches claim-grade E – peer-reviewed evidence for these practices conducted on open-weight models does not yet exist, because every E-eligible study in the corpus was run on a proprietary, hosted system. This is not a contradiction of the paper's argument but a corroboration of the very gap it identifies, and the validation pathway named in section 8 is designed to close it.

8. Conclusions

This paper set out to supply the analytic lens that the procurement framing of model adoption withholds from informatics teacher preparation, and its four research questions can now be answered in turn. RQ1 – which properties of free and open-weight models function as pedagogical affordances – is answered by the affordance vector, six standards-anchored affordances that recast openness from a price or a badge into a graded, teachable property (section 5.1). RQ2 – how those affordances map onto the competencies and enacted roles of future teachers – is answered by the openness–agency matrix, whose twenty-four cells name concrete, gradeable practices across the Designer, Facilitator, Monitor, and Evaluator roles (section 5.2). RQ3 – under which deployment tiers the affordances become actualizable – is answered by the four-tier ladder that locates the decisive affordances at its modest low end (section 5.3). And RQ4 – which governance conditions must accompany deployment and how they themselves become teachable – is answered by the four-axis scaffold that turns integrity, privacy, and licensing into a student-authored artifact (section 5.4).

The through-line of these answers is a single maxim: model selection in teacher education is a curriculum decision, not a procurement decision. The paper's four contributions follow that line. **C1** re-conceptualizes openness as a vector of six pedagogical affordances for informatics teacher preparation; **C2** arranges those affordances against four enacted teacher roles as an openness–agency matrix of concrete, gradeable practices; **C3** maps the evidence across two review streams and names the gap the matrix confronts; and **C4** supplies the deployment tiers and governance scaffold that make the framework operable in real programs.

Because no matrix cell yet rests on direct evidence, the framework's next obligations are empirical, and two studies define the validation pathway. The first is a Delphi study that puts the matrix and its claim grades to a panel of teacher educators, seeking expert consensus on the practices, their role assignments, and their priority before any cohort is taught. The second is a pilot module study that instantiates the four-week open-stack design of section 6 and tests propositions P1 through P3 – the learning effect of an open-stack module, the coverage-to-self-efficacy link, and the effect of course-owned guardrails against generic vendor filters – converting the empty E column into evidence one cell at a time.

Seen whole, the argument closes a lineage. The freedoms to run, study, modify, and share, first named for software [13] and carried into teacher education by a sustained program of free-software didactics [62–65], extend without a break to open-weight language models. Free software and open weights are not two topics but one continuous research program into a single question – what a technology must permit for a teacher to relate to it as an author rather than a consumer – and this framework is that program's next instalment.

Funding

This research received no external funding.

Author contributions

Conceptualization, methodology, formal analysis, investigation, writing – original draft, and writing – review and editing: Vladyslav Ye. Velychko. The author has read and agreed to the published version of the manuscript.

Conflicts of interest

The author declares no conflict of interest.

Data availability statement

The screening and coding log of the structured review is available from the author on reasonable request.

Acknowledgments

The author thanks the editorial team of the CTE Workshop Proceedings.

Ethics approval

Not applicable: the study did not involve human participants or animals.

Declaration on Generative AI

In preparing this work, the author used generative AI tools (Anthropic Claude) for literature scoping, structural drafting and language editing. All sources were independently verified, all AI-assisted text was reviewed and revised, and the author takes full responsibility for the content of this publication.

References

[1] Apache Software Foundation, 2004. Apache license, version 2.0. Available from: https://www.apache.org/licenses/LICENSE-2.0.

[2] Arksey, H. and O'Malley, L., 2005. Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology*, 8(1), pp.19–32. Available from: https://doi.org/10.1080/1364557032000119616.

[3] Biderman, S., Schoelkopf, H., Anthony, Q.G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M.A., Purohit, S., Prashanth, U.S., Raff, E., Skowron, A., Sutawika, L. and van der Wal, O., 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato and J. Scarlett, eds. *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*. PMLR, *Proceedings of Machine Learning Research*, vol. 202, pp.2397–2430. Available from: https://proceedings.mlr.press/v202/biderman23a.html.

[4] Caspersen, M., Cassell, L., Davies, G., Pears, A., Seidman, S. and Topi, H., 2008. What is masters level education in informatics? *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: Association for Computing Machinery, ITiCSE '08, p.341. Available from: https://doi.org/10.1145/1384271.1384383.

[5] Celik, I., 2023. Towards Intelligent-TPACK: An empirical study on teachers' professional knowledge to ethically integrate artificial intelligence (AI)-based tools into education. *Computers in Human Behavior*, 138, p.107468. Available from: https://doi.org/10.1016/j.chb.2022.107468.

[6] Cipriano, E., Ferrato, A., Limongelli, C., Schicchi, D. and Taibi, D., 2025. Leveraging Large Language Models to Assist Teachers in Code Grading. In: A.I. Cristea, E. Walker, Y. Lu, O.C. Santos and S. Isotani, eds. *Artificial Intelligence in Education: 26th International Conference, AIED 2025, Palermo, Italy, July 22–26, 2025, Proceedings, Part IV*. Cham: Springer Nature Switzerland, *Lecture Notes in Computer Science*, vol. 15880, pp.204–217. Available from: https://doi.org/10.1007/978-3-031-98459-4_15.

[7] Dagienė, V., 2011. Informatics Education for New Millennium Learners. In: I. Kalaš and R.T. Mittermeir, eds. *Informatics in Schools. Contributing to 21st Century Education*. Berlin, Heidelberg: Springer Berlin Heidelberg, *Lecture Notes in Computer Science*, vol. 7013, pp.9–20. Available from: https://doi.org/10.1007/978-3-642-24722-4_2.

[8] Denny, P., Becker, B.A., Craig, M., Wilson, G. and Banaszkiewicz, P., 2019. Research This! Questions that Computing Educators Most Want Computing Education Researchers to Answer. *Proceedings of the 2019 ACM Conference on International Computing Education Research*. New York, NY, USA: Association for Computing Machinery, ICER '19, p.259–267. Available from: https://doi.org/10.1145/3291279.3339402.

[9] Dillenbourg, P., 2013. Design for classroom orchestration. *Computers & Education*, 69, pp.485–492. Available from: https://doi.org/10.1016/j.compedu.2013.04.013.

[10] European Commission: European Education and Culture Executive Agency, 2023. *AI report – By the European Digital Education Hub's Squad on artificial intelligence in education*. Brussels: Publications Office of the European Union. Available from: https://doi.org/10.2797/828281.

[11] European Parliament and Council of the European Union, 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). *Official Journal of the European Union*, L 119, pp.1–88. Available from: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[12] European Parliament and Council of the European Union, 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance). *Official Journal of the European Union*, L 2024/1689. Available from: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

[13] Free Software Foundation, 2026. The Free Software Definition. Available from: https://www.gnu.org/philosophy/free-sw.html.

[14] Gainutdinova, T.U. and Shirokova, O.A., 2016. Features of Professional Teachers Training of Informatics in a Programming Course. In: R. Valeeva, ed. *Proceedings of the 2nd International Forum on Teacher Education (IFTE 2016)*. Future Academy, *European Proceedings of Social and Behavioural Sciences*, vol. 12, pp.30–37. Available from: https://doi.org/10.15405/epsbs.2016.07.6.

[15] Gibson, J.J., 2014. *The Ecological Approach to Visual Perception*. Classic ed. New York: Psychology Press. Available from: https://doi.org/10.4324/9781315740218.

[16] Gong, L., Chen, J. and Wu, F., 2025. Is ChatGPT a Competent Teacher? Systematic Evaluation of Large Language Models on the Competency Model. *IEEE Transactions on Learning Technologies*, 18, pp.530–541. Available from: https://doi.org/10.1109/TLT.2025.3564177.

[17] Google, 2026. Gemma Terms of Use. Available from: https://ai.google.dev/gemma/terms.

[18] Grant, M.J. and Booth, A., 2009. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, 26(2), pp.91–108. Available from: https://doi.org/10.1111/j.1471-1842.2009.00848.x.

[19] Groeneveld, D., Beltagy, I., Walsh, E., Bhagia, A., Kinney, R., Tafjord, O., Jha, A., Ivison, H., Magnusson, I., Wang, Y., Arora, S., Atkinson, D., Authur, R., Chandu, K., Cohan, A., Dumas, J., Elazar, Y., Gu, Y., Hessel, J., Khot, T., Merrill, W., Morrison, J., Muennighoff, N., Naik, A.,

Nam, C., Peters, M., Pyatkin, V., Ravichander, A., Schwenk, D., Shah, S., Smith, W., Strubell, E., Subramani, N., Wortsman, M., Dasigi, P., Lambert, N., Richardson, K., Zettlemoyer, L., Dodge, J., Lo, K., Soldaini, L., Smith, N. and Hajishirzi, H., 2024. OLMo: Accelerating the Science of Language Models. In: L.W. Ku, A. Martins and V. Srikumar, eds. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, pp.15789–15809. Available from: https://doi.org/10.18653/v1/2024.acl-long.841.

[20] Hadjerrouit, S., 2009. Teaching and Learning School Informatics: A Concept-Based Pedagogical Approach. *Informatics in Education*, 8(2), pp.227–250. Available from: https://doi.org/10.15388/infedu.2009.14.

[21] Hamilton, E.R., Rosenberg, J.M. and Akcaoglu, M., 2016. The Substitution Augmentation Modification Redefinition (SAMR) Model: a Critical Review and Suggestions for its Use. *TechTrends*, 60(5), pp.433–441. Available from: https://doi.org/10.1007/s11528-016-0091-y.

[22] Hammond, M., 2010. What is an affordance and can it help us understand the use of ICT in education? *Education and Information Technologies*, 15(3), pp.205–217. Available from: https://doi.org/10.1007/s10639-009-9106-z.

[23] Hubwieser, P., 2006. Functions, Objects and States: Teaching Informatics in Secondary Schools. In: R.T. Mittermeir, ed. *Informatics Education – The Bridge between Using and Understanding Computers*. Berlin, Heidelberg: Springer, *Lecture Notes in Computer Science*, vol. 4226, pp.104–116. Available from: https://doi.org/10.1007/11915355_10.

[24] Hughes, J. and McGettrick, A., 2003. Directions and Challenges in Informatics Education. In: L. Cassel and R.A. Reis, eds. *Informatics Curricula and Teaching Methods: IFIP TC3 / WG3.2 Conference on Informatics Curricula, Teaching Methods and Best Practice (ICTEM 2002) July 10–12, 2002, Florianópolis, SC, Brazil*. Boston, MA: Springer US, *IFIP — The International Federation for Information Processing*, vol. 117, pp.115–123. Available from: https://doi.org/10.1007/978-0-387-35619-8_12.

[25] Ishii, N. and Matsuda, T., 1999. Integrating Informatics Education: A training program for novice teachers. In: G. Cumming, T. Okamoto and L. Gomez, eds. *Advanced Research in Computers and Communications in Education: New Human Abilities for the Networked Society : Proceedings of ICCE '99, 7th International Conference on Computers in Education, Chiba, Japan, Volume 2*. Amsterdam: IOS Press, *Frontiers in Artificial Intelligence and Applications*, vol. 55, pp.19–26.

[26] Jacucci, G., Baskin, A., D'Andrea, V. and Guida, G., 1998. Problem-Based Informatics Education for Engineers: an Object-Oriented Perspective. *The Knowledge Revolution: The Impact of Technology on Learning*. IACEE, pp.118–121. 7th World Conference on Continuing Engineering Education: the Knowledge Revolution - the Impact of New Technologies on Learning, Turin, Italy, May 10-13, 1998.

[27] Jošt, G., Taneski, V. and Karakatič, S., 2024. The Impact of Large Language Models on Programming Education and Student Learning Outcomes. *Applied Sciences*, 14(10), p.4115. Available from: https://doi.org/10.3390/app14104115.

[28] Kirschner, P.A., 2002. Can we support CSCL? Educational, social and technological affordances for learning. In: P.A. Kirschner, ed. *Three Worlds of CSCL: Can We Support CSCL?* Heerlen: Open Universiteit Nederland, pp.7–47. Available from: https://research.ou.nl/ws/portalfiles/portal/24123339/Three%20worlds%20of%20CSCL%20Can%20we%20support%20CSCL.pdf.

[29] Laurillard, D., 2012. *Teaching as a Design Science: Building Pedagogical Patterns for Learning and Technology*. New York: Routledge. Available from: https://doi.org/10.4324/9780203125083.

[30] Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A.S., Yvon, F., Gallé, M., Tow, J., Rush, A.M., Biderman, S., Webson, A., Ammanamanchi, P.S., Wang, T., Sagot, B., Muennighoff, N., Villanova del Moral, A., Ruwase, O., Bawden, R., Bekman, S., McMillan-Major, A., Beltagy, I., Nguyen, H., Saulnier, L., Tan, S., Ortiz Suarez, P., Sanh, V., Laurençon, H., Jernite, Y., Launay, J., Mitchell, M., Raffel, C., Gokaslan, A., Simhi, A., Soroa, A., Aji, A.F., Alfassy, A., Rogers, A., Kreisberg Nitzav, A., Xu, C., Mou, C., Emezue, C., Klamm, C.,

Leong, C., van Strien, D., Adelani, D.I., Radev, D., González Ponferrada, E., Levkovizh, E., Kim, E., Bar Natan, E., De Toni, F., Dupont, G., Kruszewski, G., Pistilli, G., Elsahar, H., Benyamina, H., Tran, H., Yu, I., Abdulmumin, I., Johnson, I., Gonzalez-Dios, I., de la Rosa, J., Chim, J., Dodge, J., Zhu, J., Chang, J., Frohberg, J., Tobing, J., Bhattacharjee, J., Almubarak, K., Chen, K., Lo, K., Von Werra, L., Weber, L., Phan, L., Ben allal, L., Tanguy, L., Dey, M., Romero Muñoz, M., Masoud, M., Grandury, M., Šaško, M., Huang, M., Coavoux, M., Singh, M., Jiang, M.T.J., Vu, M.C., Jauhar, M.A., Ghaleb, M., Subramani, N., Kassner, N., Khamis, N., Nguyen, O., Espejel, O., de Gibert, O., Villegas, P., Henderson, P., Colombo, P., Amuok, P., Lhoest, Q., Harliman, R., Bommasani, R., López, R.L., Ribeiro, R., Osei, S., Pyysalo, S., Nagel, S., Bose, S., Hassan Muhammad, S., Sharma, S., Longpre, S., Nikpoor, S., Silberberg, S., Pai, S., Zink, S., Timponi Torrent, T., Schick, T., Thrush, T., Danchev, V., Nikoulina, V., Laippala, V., Lepercq, V., Prabhu, V., Alyafeai, Z., Talat, Z., Raja, A., Heinzerling, B., Si, C., Taşar, D.E., Salesky, E., Mielke, S.J., Lee, W.Y., Sharma, A., Santilli, A., Chaffin, A., Stiegler, A., Datta, D., Szczechla, E., Chhablani, G., Wang, H., Pandey, H., Strobelt, H., Fries, J.A., Rozen, J., Gao, L., Sutawika, L., Saiful Bari, M., Al-shaibani, M.S., Manica, M., Nayak, N., Teehan, R., Albanie, S., Shen, S., Ben-David, S., Bach, S.H., Kim, T., Bers, T., Fevry, T., Neeraj, T., Thakker, U., Raunak, V., Tang, X., Yong, Z.X., Sun, Z., Brody, S., Uri, Y., Tojarieh, H., Roberts, A., Chung, H.W., Tae, J., Phang, J., Press, O., Li, C., Narayanan, D., Bourfoune, H., Casper, J., Rasley, J., Ryabinin, M., Mishra, M., Zhang, M., Shoeybi, M., Peyrounette, M., Patry, N., Tazi, N., Sanseviero, O., von Platen, P., Cornette, P., François Lavallée, P., Lacroix, R., Rajbhandari, S., Gandhi, S., Smith, S., Requena, S., Patil, S., Dettmers, T., Baruwa, A., Singh, A., Cheveleva, A., Ligozat, A.L., Subramonian, A., Névéol, A., Lovering, C., Garrette, D., Tunuguntla, D., Reiter, E., Taktasheva, E., Voloshina, E., Bogdanov, E., Winata, G.I., Schoelkopf, H., Kalo, J.C., Novikova, J., Zosa Forde, J., Clive, J., Kasai, J., Kawamura, K., Hazan, L., Carpuat, M., Clinciu, M., Kim, N., Cheng, N., Serikov, O., Antverg, O., van der Wal, O., Zhang, R., Zhang, R., Gehrmann, S., Mirkin, S., Pais, S., Shavrina, T., Scialom, T., Yun, T., Limisiewicz, T., Rieser, V., Protasov, V., Mikhailov, V., Pruksachatkun, Y., Belinkov, Y., Bamberger, Z., Kasner, Z., Rueda, A., Pestana, A., Feizpour, A., Khan, A., Faranak, A., Santos, A., Hevia, A., Unldreaj, A., Aghagol, A., Abdollahi, A., Tammour, A., HajiHosseini, A., Behroozi, B., Ajibade, B., Saxena, B., Muñoz Ferrandis, C., McDuff, D., Contractor, D., Lansky, D., David, D., Kiela, D., Nguyen, D.A., Tan, E., Baylor, E., Ozoani, E., Mirza, F., Ononiwu, F., Rezanejad, H., Jones, H., Bhattacharya, I., Solaiman, I., Sedenko, I., Nejadgholi, I., Passmore, J., Seltzer, J., Bonis Sanz, J., Dutra, L., Samagaio, M., Elbadri, M., Mieskes, M., Gerchick, M., Akinlolu, M., McKenna, M., Qiu, M., Ghauri, M., Burynok, M., Abrar, N., Rajani, N., Elkott, N., Fahmy, N., Samuel, O., An, R., Kromann, R., Hao, R., Alizadeh, S., Shubber, S., Wang, S., Roy, S., Viguier, S., Le, T., Oyebade, T., Le, T., Yang, Y., Nguyen, Z., Kashyap, A.R., Palasciano, A., Callahan, A., Shukla, A., Miranda-Escalada, A., Singh, A., Beilharz, B., Wang, B., Brito, C., Zhou, C., Jain, C., Xu, C., Fourrier, C., León Periñán, D., Molano, D., Yu, D., Manjavacas, E., Barth, F., Fuhrimann, F., Altay, G., Bayrak, G., Burns, G., Vrabec, H.U., Bello, I., Dash, I., Kang, J., Giorgi, J., Golde, J., Posada, J.D., Sivaraman, K.R., Bulchandani, L., Liu, L., Shinzato, L., Hahn de Bykhovetz, M., Takeuchi, M., Pàmies, M., Castillo, M.A., Nezhurina, M., Sänger, M., Samwald, M., Cullan, M., Weinberg, M., De Wolf, M., Mihaljcic, M., Liu, M., Freidank, M., Kang, M., Seelam, N., Dahlberg, N., Broad, N.M., Muellner, N., Fung, P., Haller, P., Chandrasekhar, R., Eisenberg, R., Martin, R., Canalli, R., Su, R., Su, R., Cahyawijaya, S., Garda, S., Deshmukh, S.S., Mishra, S., Kiblawi, S., Ott, S., Sang-aroonsiri, S., Kumar, S., Schweter, S., Bharati, S., Laud, T., Gigant, T., Kainuma, T., Kusa, W., Labrak, Y., Bajaj, Y.S., Venkatraman, Y., Xu, Y., Xu, Y., Xu, Y., Tan, Z., Xie, Z., Ye, Z., Bras, M., Belkada, Y. and Wolf, T., 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. Available from: https://doi.org/10.48550/arXiv.2211.05100.

[31] Liffiton, M., Sheese, B.E., Savelka, J. and Denny, P., 2024. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. New York, NY, USA: Association for Computing Machinery, Koli Calling '23. Available from: https://doi.org/10.1145/3631802.3631830.

[32] Lin, M.P.C., Chang, D., Hall, S. and Jhajj, G., 2024. Preliminary Systematic Review of Open-Source Large Language Models in Education. In: A. Sifaleras and F. Lin, eds. *Generative Intelligence and Intelligent Tutoring Systems*. Cham: Springer Nature Switzerland, *Lecture Notes in Computer Science*, vol. 14798, pp.68–77. Available from: https://doi.org/10.1007/978-3-031-63028-6_6.

[33] Linck, B., 2013. Competence descriptions for informatics education—using the example of logic programming. *Education and Information Technologies*, 18(2), pp.193–204. Available from: https://doi.org/10.1007/s10639-012-9237-5.

[34] Long, D. and Magerko, B., 2020. What is AI Literacy? Competencies and Design Considerations. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, CHI '20, p.1–16. Available from: https://doi.org/10.1145/3313831.3376727.

[35] Lortie, D.C., 2002. *Schoolteacher: A Sociological Study*. 2nd ed. Chicago and London: University of Chicago Press.

[36] Luik, P., Saadjärv, R. and Lehesaar, V., 2024. Teaching Informatics in Lower-Secondary Schools: Views of Informatics Teachers. *2024 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. Available from: https://doi.org/10.1109/EDUCON60312.2024.10578832.

[37] Lunenberg, M., Korthagen, F. and Swennen, A., 2007. The teacher educator as a role model. *Teaching and Teacher Education*, 23(5), pp.586–601. Available from: https://doi.org/10.1016/j.tate.2006.11.001.

[38] Mannekote, A., Davies, A., Pinto, J.D.D., Zhang, S., Olds, D., Schroeder, N.L.L., Lehman, B., Zapata-Rivera, D. and Zhai, C., 2024. Large language models for whole-learner support: opportunities and challenges. *Frontiers in Artificial Intelligence*, 7, p.1460364. Available from: https://doi.org/10.3389/frai.2024.1460364.

[39] Markel, J.M., Opferman, S.G., Landay, J.A. and Piech, C., 2023. GPTeach: Interactive TA Training with GPT-based Students. *Proceedings of the Tenth ACM Conference on Learning @ Scale*. New York, NY, USA: Association for Computing Machinery, L@S '23, p.226–236. Available from: https://doi.org/10.1145/3573051.3593393.

[40] Meta Platforms, Inc., 2025. Llama 4 Community License Agreement. Available from: https://developer.meta.com/ai/llama4/license/.

[41] Miao, F. and Cukurova, M., 2024. *AI competency framework for teachers*. Paris: UNESCO. Available from: https://doi.org/10.54675/ZJTE2084.

[42] Mishra, P. and Koehler, M.J., 2006. Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record*, 108(6), pp.1017–1054. Available from: https://doi.org/10.1111/j.1467-9620.2006.00684.x.

[43] Mittermeir, R.T., Bischof, E. and Hodnigg, K., 2010. Showing Core-Concepts of Informatics to Kids and Their Teachers. In: J. Hromkovič, R. Královič and J. Vahrenhold, eds. *Teaching Fundamentals Concepts of Informatics*. Berlin, Heidelberg: Springer, *Lecture Notes in Computer Science*, vol. 5941, pp.143–154. Available from: https://doi.org/10.1007/978-3-642-11376-5_14.

[44] Nikolova, I. and Nikolov, R., 1993. Teacher Training in Informatics: Analyzing The Problems. In: D.C. Johnson and B. Samways, eds. *Informatics and Changes in Learning, Proceedings of the IFIP TC3/WG3.1/WG3.5 Open Conference on Informatics and Changes in Learning, Gmunden, Austria, 7-11 June, 1993*. North-Holland, *IFIP Transactions*, vol. A-34, pp.253–260.

[45] Norman, D., 2013. *The Design of Everyday Things*. Revised and Expanded ed. New York: Basic Books.

[46] Open Source Initiative, 1988. The MIT License. Available from: https://opensource.org/license/mit.

[47] Open Source Initiative, 2024. The Open Source AI Definition – 1.0. Available from: https://opensource.org/ai/open-source-ai-definition.

[48] Parriaux, G., Pellet, J.P. and Komis, V., 2025. A Glimpse Into Primary and Secondary Teachers' Knowledge to Teach Informatics. In: Z. Pluhár and B. Gaál, eds. *Informatics in Schools. Innovative*

Approaches to Computer Science Teaching and Learning. Cham: Springer Nature Switzerland, *Lecture Notes in Computer Science*, vol. 15228, pp.125–138. Available from: https://doi.org/10.1007/978-3-031-73474-8_10.

[49] Prasad, S., Greenman, B., Nelson, T. and Krishnamurthi, S., 2023. Generating Programs Trivially: Student Use of Large Language Models. *Proceedings of the ACM Conference on Global Computing Education Vol 1*. New York, NY, USA: Association for Computing Machinery, CompEd 2023, p.126–132. Available from: https://doi.org/10.1145/3576882.3617921.

[50] Puentedura, R.R., 2006. Transformation, Technology, and Education. Available from: http://hippasus.com/resources/tte/.

[51] Raihan, N., Siddiq, M.L., Santos, J.C.S. and Zampieri, M., 2025. Large Language Models in Computer Science Education: A Systematic Literature Review. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. New York, NY, USA: Association for Computing Machinery, SIGCSETS 2025, p.938–944. Available from: https://doi.org/10.1145/3641554.3701863.

[52] Redecker, C., 2017. *European framework for the digital competence of educators – DigCompEdu*. Luxembourg: Publications Office of the European Union. Available from: https://doi.org/10.2760/159770.

[53] Responsible AI Licenses (RAIL), 2022. Responsible AI License (RAIL). Available from: https://www.licenses.ai/.

[54] Semerikov, S.O., Vakaliuk, T.A., Mintii, I.S. and Didkivska, S.O., 2023. Challenges facing distance learning during martial law: results of a survey of Ukrainian students. *Educational Technology Quarterly*, 2023(4), p.401–421. Available from: https://doi.org/10.55056/etq.637.

[55] Sheese, B., Liffiton, M., Savelka, J. and Denny, P., 2024. Patterns of Student Help-Seeking When Using a Large Language Model-Powered Programming Assistant. *Proceedings of the 26th Australasian Computing Education Conference*. New York, NY, USA: Association for Computing Machinery, ACE '24, p.49–57. Available from: https://doi.org/10.1145/3636243.3636249.

[56] Shi, Y., Yu, K., Dong, Y. and Chen, F., 2026. Large language models in education: a systematic review of empirical applications, benefits, and challenges. *Computers and Education: Artificial Intelligence*, 10, p.100529. Available from: https://doi.org/10.1016/j.caeai.2025.100529.

[57] Shlenova, M., Konoplenko, N., Yuryeva, K., Korneiko, Y. and Hlukhovska, M., 2023. Comparative analysis of the distance learning implementation in Ukrainian system of higher education during the COVID-19 pandemic and martial law. *Interactive learning environments*, 32(9), pp.4968–4977. Available from: https://doi.org/10.1080/10494820.2023.2207196.

[58] Standl, B., Bentz, A., Ulbrich, M., Vielsack, A. and Wagner, I., 2020. Design- and Evaluation-Concept for Teaching and Learning Laboratories in Informatics Teacher Education. In: K. Kori and M. Laanpere, eds. *Informatics in Schools. Engaging Learners in Computational Thinking*. Cham: Springer International Publishing, *Lecture Notes in Computer Science*, vol. 12518, pp.133–145. Available from: https://doi.org/10.1007/978-3-030-63212-0_11.

[59] Taylor, A., Vassar, A., Renzella, J. and Pearce, H., 2024. dcc –help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations with Large Language Models. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. New York, NY, USA: Association for Computing Machinery, SIGCSE 2024, p.1314–1320. Available from: https://doi.org/10.1145/3626252.3630822.

[60] Tricco, A.C., Lillie, E., Zarin, W., O'Brien, K.K., Colquhoun, H., Levac, D., Moher, D., Peters, M.D.J., Horsley, T., Weeks, L., Hempel, S., Akl, E.A., Chang, C., McGowan, J., Stewart, L., Hartling, L., Aldcroft, A., Wilson, M.G., Garritty, C., Lewin, S., Godfrey, C.M., Macdonald, M.T., Langlois, E.V., Soares-Weiser, K., Moriarty, J., Clifford, T., Tunçalp, Ö. and Straus, S.E., 2018. PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Annals of Internal Medicine*, 169(7), pp.467–473. Available from: https://doi.org/10.7326/M18-0850.

[61] Turcsányi-Szabó, M., 1997. Present role of informatics teachers in view of applications. In: D. Passey and B. Samways, eds. *Information Technology: Supporting change through teacher education*. Boston, MA: Springer US, IFIP Advances in Information and Communication Technology,

pp.62–68. Available from: https://doi.org/10.1007/978-0-387-35081-3_7.

[62] Velychko, V., Fedorenko, E. and Serdiuk, O., 2022. Theoretical Bases of Application of Free Software in Preparation of Pre-service Teachers of Mathematics, Physics and Computer Science. *Proceedings of the 1st Symposium on Advances in Educational Technology - Volume 1: AET*. INSTICC, SciTePress, pp.261–271. Available from: https://doi.org/10.5220/0010923600003364.

[63] Velychko, V.E., 2017. Organizational-Pedagogical Conditions of Free and Open Source Software Using in the Preparation of Teachers of Mathematics, Physics and Computer Science. *Zhytomyr Ivan Franko State University Journal. Pedagogical Sciences*, 1(87), pp.43–48. Available from: https://doi.org/10.35433/pedagogy.1(87).2017.43-48.

[64] Velychko, V.Y., Fedorenko, E., Kaidan, N. and Kaidan, V., 2023. The Use of Cloud Technologies in the Process of Professional Training of Future Mathematics Teachers. *Proceedings of the 2nd Myroslav I. Zhaldak Symposium on Advances in Educational Technology - AET*. INSTICC, SciTePress, pp.378–389. Available from: https://doi.org/10.5220/0012064700003431.

[65] Velychko, V.Y., Fedorenko, E.H. and Kassim, D.A., 2018. Conceptual Bases of Use of Free Software in the Professional Training of Pre-Service Teacher of Mathematics, Physics and Computer Science. In: A.E. Kiv and V.N. Soloviev, eds. *Proceedings of the 1st International Workshop on Augmented Reality in Education, Kryvyi Rih, Ukraine, October 2, 2018*. CEUR-WS.org, *CEUR Workshop Proceedings*, vol. 2257, pp.93–102. Available from: https://doi.org/10.55056/ceur-ws.org/Vol-2257/paper11.pdf.

[66] Verkhovna Rada of Ukraine, 2010. Law of Ukraine “On Personal Data Protection” No. 2297-VI. Available from: https://zakon.rada.gov.ua/laws/show/2297-17.

[67] Wang, S., Xu, T., Li, H., Zhang, C., Liang, J., Tang, J., Yu, P.S. and Wen, Q., 2025. Large Language Models for Education: *A survey and outlook*. *IEEE Signal Processing Magazine*, 42(6), pp.51–63. Available from: https://doi.org/10.1109/MSP.2025.3594309.

[68] Webb, M.E., Bell, T., Davis, N., Katz, Y.J., Fluck, A., Sysło, M.M., Kalaš, I., Cox, M., Angeli, C., Malyn-Smith, J., Brinda, T., Micheuz, P. and Brodnik, A., 2018. Tensions in specifying computing curricula for K-12: Towards a principled approach for objectives. *it – Information Technology*, 60(2), pp.59–68. Available from: https://doi.org/10.1515/itit-2017-0017.

[69] Zdravkova, K., Dalipi, F., Ahlgren, F., Ilijoski, B. and Olsson, T., 2024. Unveiling the Impact of Large Language Models on Student Learning: A Comprehensive Case Study. *2024 IEEE Global Engineering Education Conference (EDUCON)*. pp.1–8. Available from: https://doi.org/10.1109/EDUCON60312.2024.10578855.