UDC 004.8
DOI: 10.15587/1729-4061.2026.356830

DEVELOPMENT OF MULTI-AGENT GENERATIVE PIPELINES FRAMEWORK FOR LEARNING PLAN GENERATION WITH DETERMINISTIC CONSTRAINT VERIFICATION

Mohammad Fadly Syahputra
Corresponding Author
PhD*
E-mail: nca.fadly@usu.ac.id
ORCID: https://orcid.org/0000-0002-5683-6910
Opim Salim Sitompul
PhD*
ORCID: https://orcid.org/0000-0001-6069-1841
Fahmi
PhD
Department of Electrical Engineering***
ORCID ID: https://orcid.org/0000-0002-6760-4824
Maya Silvi Lydia
PhD**
ORCID: https://orcid.org/0009-0006-5779-5678
Pauzi Ibrahim Nainggolan
PhD**
ORCID: https://orcid.org/0000-0002-7493-6413
Rendra Mahardika
Master*
ORCID: https://orcid.org/0009-0006-7239-0521
Riza Sulaiman
PhD
Department of Institute of Visual Informatics
Universiti Kebangsaan Malaysia
UKM Bangi, Selangor, Malaysia, 43600
ORCID: https://orcid.org/0000-0002-5862-5279
*Department of Information Technology***
Department of Computer Science*
***Universitas Sumatera Utara
Dr. T. Mansur str., 9, Padang Bulan, North Sumatera, Indonesia, 20155

Large language models (LLMs) are increasingly used to generate structured learning plans aligned with outcome-based education (OBE). The object of the study is a multi-agent workflow for generating a structured OBE learning-plan package with a final deterministic verification stage. The problem addressed is the low reliability of LLM-generated outputs, which frequently violate schema rules, numeric constraints, and cross-artifact consistency requirements. To solve this problem, a multi-agent generative pipeline is proposed, decomposing the task into six specialized agents followed by deterministic constraint verification applied to the final artifact bundle. Structural reliability is measured using completeness and compliance, while cross-artifact coherence is evaluated through redundancy, spacing, phase progression, and assessment fit. The evaluation involves 12 courses with 10 repeated runs per course (120 runs per variant) across four different LLMs to assess cross-model robustness. The results show that the multi-agent pipeline achieves completeness of 0.9682–1.00 and compliance of 0.9376–0.9698, significantly outperforming the single-agent configuration (completeness 0.5926–0.6580; compliance 0.4698–0.4853). These improvements are explained by task decomposition, which reduces structural failure propagation, and deterministic verification, which rejects invalid outputs and preserves referential integrity. Ablation analysis indicates that the Course Character agent exerts the highest impact on overall performance. The proposed framework can be applied in higher education curriculum planning under OBE conditions, using minimal course metadata and producing machine-verifiable structured artifacts

Keywords: multi-agent, generative AI, structured generation, constraint validation, ablation analysis

Received 17.01.2026
Received in revised form 10.02.2026
Accepted date 26.03.2026
Published date 30.04.2026

How to Cite: *Syahputra, M. F., Sitompul, O. S., Fahmi, F., Lydia, M. S., Nainggolan, P. I., Mahardika, R., Sulaiman, R. (2026). Development of multi-agent generative pipelines framework for learning plan generation with deterministic constraint verification. Eastern-European Journal of Enterprise Technologies, 2 (2 (140)), 17–31. https://doi.org/10.15587/1729-4061.2026.356830*

1. Introduction

Outcome-based education (OBE) treats a learning plan as a structured academic document rather than a narrative description. In this setting, coherence between course-level outcomes and broader curriculum expectations is essential [1]. Assessment design in higher education also depends on a clear relation between intended outcomes and the evidence used to evaluate them [2]. Therefore, preparing a learning plan is not only a writing task, but also a task of maintaining alignment across related academic elements.

In current higher education practice, this task remains difficult. Intended competencies must be translated into course outcomes, weekly learning activities, and assessment evidence that remains consistent with the original instructional intent. Recent review studies show that the use of ChatGPT and related generative systems in higher education has expanded across diverse academic applications [3], while empirical case studies show that GenAI is already being implemented in university teaching and learning contexts [4]. This makes the reliability of structured educational generation a relevant research issue, especially because educational use of large language models still involves important challenges and limitations [5].

The relevance of the topic is also visible in recent work on instructional planning. One study examined how GenAI tools can support the design and implementation of inquiry-based lesson plans [6]. Another study showed that GenAI can be used to co-construct adaptive lesson plans, while also stressing that the quality of such planning depends on teachers' AI-specific pedagogical knowledge and prompt design [7]. These findings indicate that generative support for lesson planning is no longer hypothetical, which strengthens the need for research on more reliable structured generation.

Practical relevance is further supported by adoption-related evidence. Research on beginning and first-year language teachers reported uneven readiness for the use of generative AI in professional work, which points to a real implementation issue rather than a purely technical one [8]. Another study found that behavioral intention to use GenAI tools for teaching and learning is shaped by factors such as self-efficacy, perceived usefulness, attitude, and subjective norm, and it also emphasized the value of teacher development programs and policy support [9]. Therefore, research on reliable generation methods for structured OBE learning-plan construction is relevant.

2. Literature review and problem statement

The paper [10] examines dynamic course content integration (DCCI), which links an LLM with Canvas LMS by retrieving course materials and injecting them into the Ask ME assistant through contextual prompting. The reported pilot results are positive, with an average satisfaction score of 4.652/5 and 78.06% of respondents stating that Canvas integration reduced platform switching. Still, the issue of structural reliability in educational generation remains open, because control is achieved mainly through retrieved context and prompt design rather than through formal checks of required fields and inter-element links. This limitation is understandable, since the system was built to provide context-aware assistance inside an LMS rather than to generate machine-verifiable educational artifacts.

The paper [11] presents a theory-grounded multi-agent prototype (CoPL) informed by cultural-historical activity theory for personalized learning implementation. The system assigns distinct roles to multiple agents, engages pre-service teachers in dynamic dialogue, and includes breakpoints where human feedback can guide later steps. Yet the problem of structural correctness remains unresolved, because the study does not specify deterministic verification for field completeness, identifier consistency, or link validity across generated components. This gap likely follows from the main role of the tool as a reflective support environment for inclusive planning rather than as a rule-governed generator of formal educational artifacts.

The paper [12] presents EduMAS, an LLM-powered multi-agent framework that combines specialized agents with graph-based knowledge navigation and coordinated reasoning. It is shown that win rates range 48–68% (complex concept integration), 56–62% (cross-disciplinary understanding), and 58–68% (theory-to-application translation), with reported stability including $\sigma = 3.9\%$ for theory-to-application translation. But unresolved issues remained for formal learning-plan construction, because the framework is oriented to support and explanation, not to the production of a complete curriculum bundle governed by explicit structural rules. The reason for this may be the additional coordination cost of multi-agent operation and the design priority given to adaptive support quality rather than deterministic validation of interdependent artifacts. A way to overcome these difficulties can be to combine multi-agent orchestration with explicit constraint checks applied to the final artifact set.

The paper [13] presents an empirical study of LLM-based analysis of written lesson plans against a human expert standard. The results show moderate agreement for explicit instructional features, with $\alpha = 0.689$ and 73.8% exact agreement, while weaker reliability appears on high-inferential criteria that require deeper pedagogical interpretation. However, unresolved issues remain in preventive structural control, because the study evaluates completed lesson plans after they are produced rather than validating formal relations during generation. This may stem from the fact that inferential pedagogical judgment is harder to stabilize than explicit structural checking. One way to address this difficulty is to introduce generation-stage validation of mandatory fields and trace links before inconsistencies propagate across related artifacts.

The paper [14] presents a review of debiasing and dehallucination methods for large language models. It is shown that the surveyed literature spans pre-processing, in-processing, intra-processing without training, and post-processing during inference, and that many mitigation methods target trustworthiness of generated content rather than formal structural control. But there were unresolved issues related to formal correctness in structured educational planning, because semantic trust mitigation does not itself verify mandatory fields, identifier consistency, or the correctness of links between dependent components. The reason for this may be a fundamental mismatch between content-level trust and structure-level validity, which motivates constraint-based enforcement in addition to trust mitigation.

The paper [15] presents SLOT (structured LLM output transformer), a post-processing approach for producing schema-conformant structured outputs from LLM generations. It is shown that the framework evaluates performance through schema accuracy and content similarity, and that the fine-tuned Mistral-7B model with constrained decoding achieved 99.5% schema accuracy and 94.0% content similarity, outperforming Claude-3.5-Sonnet by 25 and 20 percentage points on the respective metrics. But there were unresolved issues related to multi-artifact structural correctness, because local schema validity for a single output does not ensure that mandatory fields remain complete and that cross-component links remain correct across outcomes, sequencing, and assessment mapping. The reason for this may be that schema enforcement is local to one output instance, whereas curriculum bundles require additional checks across several dependent artifacts.

The paper [16] presents grammar-constrained decoding (GCD) to improve syntactic correctness and accuracy for structured parsing without finetuning. It is shown that

GCD consistently improves syntactic correctness and often improves downstream accuracy, with especially clear benefits for smaller models and low-shot settings. For example, in zero-shot evaluation, gemma2-2b improved from 0.02 to 0.21 on FOLIO and from 0.00 to 0.15 on GSM-symbolic under constrained decoding. But there were unresolved issues related to domain-specific structural compliance, because syntactic well-formedness alone does not guarantee mandatory fields, valid dependencies, or correct relations among curriculum elements. The reason for this may be that grammars control output form, whereas structured OBE planning requires semantic and relational validation across multiple interconnected components, which motivates deterministic constraint verification beyond decoding constraints.

All this suggests that it is advisable to conduct a study on an end-to-end generative approach for OBE planning that addresses the unresolved reliability and structural-control gap in existing work. There is still no generation approach that combines more stable control than prompt structuring alone with formalized validation of mandatory fields and cross-element connections for a complete learning-plan bundle under explicit curriculum constraints.

3. The aim and objectives of the study

The aim of the study is to develop a multi-agent generative pipeline capable of producing structured learning plans aligned with outcome-based education (OBE) requirements under deterministic constraint verification. The proposed pipeline ensures that the generated learning-plan artifacts undergo explicit rule-based validation within the multi-agent architecture. Experimentally, the pipeline is required to demonstrate superior structural validity and coherence compared with a single-agent configuration, and to provide measurable evidence of performance gains through controlled agent-contribution analysis.

This will allow ensuring schema validity, preserving cross-artifact referential integrity among generated components, and establishing an engineering reference pipeline for the reproducible deployment of multi-agent generative in structured OBE-based curriculum planning.

To achieve this aim, the study formulates the following objectives:

– to develop multi-agent generative pipelines framework for learning plan generation with deterministic constraint verification;

– to measure the degree to which generated artifacts satisfy schema validity and cross-artifact coherence under explicitly defined deterministic constraints and verification rules;

– to compare the measured structural validity and coherence of the proposed multi-agent pipeline with those obtained under a single-agent configuration;

– to identify the contribution of each agent through controlled ablation analysis (A0–A6) and structured degradation mapping of generated artifacts under the shared constraint-verification protocol.

4. Materials and methods

4. 1. The object and hypothesis of the study

The object of the study is a multi-agent workflow for generating a structured OBE learning-plan package with a final deterministic verification stage. The package contains six connected artifacts: course character classification, Bloom taxonomy targeting, course learning outcomes (CLO), learning objectives (LO), a 14-week weekly plan, and assessment weights.

In this workflow, the artifacts are handled as one connected package rather than as separate text outputs. For that reason, the method focuses on two aspects. The first is field correctness, which concerns the presence and form of required elements. The second is consistency between dependent artifacts, including CLO–LO relations, placement of LOs in weekly planning, and coverage of outcomes by assessment components.

The main hypothesis is that the multi-agent workflow will produce OBE-aligned learning plans with higher completeness and compliance than a single-agent baseline when both settings use the same course input.

This expectation follows from the structure of the method. Pedagogical work is divided into narrower responsibilities, and the assembled package is screened before acceptance. The expected advantage concerns structural quality of the generated plan. It does not imply a direct claim about classroom effectiveness or student learning outcomes.

The study rests on several assumptions. First, the OBE requirements used in the selected institutional setting can be expressed as explicit checking rules for outcome hierarchy, Bloom-level consistency, week allocation, and assessment distribution.

Second, course metadata, short course descriptions, and institutional context provide enough information to construct the required package. Third, each repetition uses unchanged source input, so output variation can be linked to the generation process rather than to changes in the source material.

Fourth, the comparison uses four open-weight language models: Gemma2-9b, DeepSeek-R1-8b, Llama3.1-8b, and Qwen3-8b. They are used to observe whether the workflow remains stable under one common evaluation setting.

Several simplifications were adopted to keep the procedure manageable. The semester horizon was fixed at 14 weeks. Assessment categories were paired with one reference Bloom level for checking purposes: quiz with C1, midterm with C2, tasks with C3, final exam with C4, case study with C5, and project with C6.

When no formal course description is available, a short surrogate description may be inferred from metadata. The study also applies one institutional rule specification during evaluation and uses inference-only deployment without task-specific fine-tuning. These decisions support reproducibility, but they also confine interpretation to the present procedural setting.

4. 2. Data

The system uses a small set of institutional and course-level inputs to define the generation context and keep the outputs consistent with program-level intent and course scope [1]. Rather than relying on dataset-driven training, pedagogical knowledge is encoded as structured agent instructions and correctness is enforced through deterministic checks, including schema validation and explicit domain-rule verification. The inputs comprise:

a) institutional and program context (e.g., degree level and program orientation);

b) course metadata (e.g., title, credits, and course-type indicators);

c) an optional course description.

When a formal course description is missing, the pipeline does not leave the context empty. Instead, the Bloom taxonomy agent produces a short proxy description from institutional and program cues, which is then used only to support downstream outcome generation.

The evaluation covers 12 courses, with 10 repeated runs for each course under the same input conditions, resulting in 120 attempts per variant. Repeated runs were included because the generator is stochastic and can produce additional attempts in response to verifier feedback. Reporting mean ± SD across these runs provides a more reliable view of robustness than a single-run result. The course inputs remained fixed throughout all repetitions. All courses in this workload included short descriptions, but the optional-description pathway was kept to reflect real deployment conditions in which such descriptions may be unavailable.

Given the minimal inputs, the multi-agent pipeline produces a full learning plan through staged decisions. It first classifies course nature (theoretical, practical, or mixed) to shape learning strategy and assessment style [17]. Next, it sets cognitive targets by mapping course characteristics to primary and supporting Bloom levels. It then generates CLOs and LOs while preserving the intended cognitive hierarchy, and distributes LOs across the semester schedule in a way that supports scaffolding and cognitive-load considerations [18].

During generation, the output is checked against a set of rules to maintain structural correctness and pedagogical consistency. Schema-level checks verify required fields, data types, and allowable ranges, such as Bloom codes in {C1, ..., C6} and valid week indices [12]. Additional rule checks examine arithmetic and hierarchy constraints, including assessment weights summing to 100%, LO Bloom levels not exceeding those of their parent CLOs, and limits set by degree level and course character. Sequence-related checks are also applied to avoid cognitive backtracking and to keep learning activities and assessments consistent with course character. When a violation is detected, the verifier initiates another attempt focused on the affected part. Cases that still fail after repeated checks are marked for manual review.

4. 3. Outcome-based specification and alignment constraints

Building on the problem framing in the Introduction, this paper adopts OBE as a specification layer that defines the target learning-plan artifacts and the verifiable alignment requirements that the proposed method must satisfy. The emphasis is not on pedagogical effectiveness claims, but on producing an auditable, structured learning-plan bundle in which intended outcomes, weekly instructional intentions, and assessment evidence are explicitly linked and can be checked deterministically. This stance is consistent with recent work that operationalizes constructive alignment and OBE practice through explicit artifact structure and tool-supported alignment workflows, including LLM- and NLP-assisted alignment tooling in higher-education settings [19, 20].

In this work, CLOs are treated as course-level outcomes that formalize a course's intended competence boundary and serve as an anchor for outcome coherence and curriculum-level traceability. In OBE implementations, CLO statements are commonly positioned as the course-level units that must remain systematically coherent with higher-level outcomes and support explicit curriculum-level traceability. Accordingly, each CLO in the generated plan is stated in action-oriented form and tagged with an indicative cognitive demand level using Bloom's taxonomy, where Bloom tagging functions as a controlled signal for coherence across the plan [21]. To operationalize CLOs into schedulable and assessable units, each CLO is refined into Learning Objectives (LOs), defined as objective statements that:

a) carry explicit action verbs;

b) declare a primary parent CLO;

c) inherit a Bloom tag consistent with those verbs; this treatment aligns with evidence that Bloom-level categorization of outcome statements can be approached as a structured classification task under explicit labeling rules.

Instruction is modeled as a fixed 14-week horizon. Each week contains a topic and weekly objective statements. These objectives are not treated as free-form prose. Each week must reference one or more LO identifiers, so outcome linkage remains auditable at the weekly level. In implementation terms, weekly units expose LO tags, enabling deterministic checks for LO coverage across weeks and for traceability completeness. This choice follows alignment tooling that prioritizes inspectable mappings over narrative interpretation [19].

Operationalize constructive alignment as three required mappings such as CLO to LO (decomposition), LO to week (instructional anchoring), and LO to assessment (evidence allocation) [20]. Assessments are represented as weighted components (e.g., project, case study, quiz, tasks, midterm, final), and each component explicitly references the LOs it measures to avoid unassessed outcomes. Bloom taxonomy is used as a progression signal at the phase level (early-middle-late weeks), allowing reinforcement while retaining a checkable progression pattern [21]. Together, these mappings form the minimum constraint set used for verification and for isolating module-level errors [22].

These OBE-derived requirements determine agent responsibilities in the proposed method. The pipeline must infer evidence-linked CLOs from course descriptions, generate operational LOs with parent links and Bloom tags, allocate assessment evidence with explicit LO coverage, and construct a weekly schedule, which objectives are anchored to LOs within feasibility bounds. Accordingly, agent outputs are specified in terms of trace-link production rather than narrative quality. Evaluation is therefore grounded in verification checks and metrics computed from the structured artifacts and trace links, not in adoption or learning-outcome studies.

4. 4. Implementation

4. 4. 1. LLM backbone and pipeline architecture

Four LLMs are used in this study, namely Gemma [23], DeepSeek [24], Llama [25], and Qwen3 [26], strictly for inference; task-specific fine-tuning and dataset-based training are not performed. Agent behavior is specified through instruction prompts, and correctness is enforced by a deterministic constraint verifier. The proposed architecture breaks outcome-based learning-plan generation into six specialized agents, each responsible for a distinct pedagogical subtask. This decomposition supports modular fault isolation and stage-wise verification, so violations can be corrected early rather than propagating across the plan [11]. The generator uses faculty-, program-, and course-level metadata as grounding context, which restricts the output space to remain consistent with program outcomes and course scope. Execution follows explicit dependencies: Bloom classification establishes cognitive constraints, course-character classification stabilizes downstream activity and assessment choices,

CLO/LO agents build the outcome hierarchy, and schedule and assessment-weight modules consume CLO/LO outputs. Outputs are released as a structured JSON bundle and must pass JSON-schema validation and domain-rule checks before being returned.

4. 4. 2. Course character taxonomy agent

The Course character taxonomy agent takes the course description and the inferred Bloom levels as input, and assigns one primary and one supporting course-character code from eight categories: theoretical (TEO), conceptual (KON), analytical (ANA), procedural (PRO), applied (APL), computational (KOM), practical (PRA), and design/engineering (DES).

This classification guides downstream design decisions – especially the expected depth of coverage, the types of learning activities, and the assessment formats – so that they remain consistent with the intended competency profile and support assessment validity [17]. Concretely, the agent combines keyword cues in the course description with Bloom-level compatibility rules to select the two-character profile. The resulting codes are then used as explicit constraints by subsequent agents to keep weekly topics, learning activities, and assessment methods aligned.

4. 4. 3. Bloom taxonomy agent

The Bloom taxonomy agent maps inputs to cognitive levels using Bloom's Revised taxonomy, receiving faculty, program, course name, and an optional course description; when the description is missing, the agent generates a concise inferred description, and in all cases it outputs primary/supporting Bloom codes (C1–C6) and educational-level constraints.

By establishing cognitive classification early, the generator prevents downstream agents from producing activities and assessments misaligned with learning objectives [21]. The agent classifies Indonesian action verbs to Bloom levels (C1: remember, list/identify; C2: explain, summarize; C3: apply, calculate; C4: analyze, differentiate; C5: evaluate, judge; C6: design, construct) and enforces course-type and educational level constraints (Diploma max C4, Bachelor max C5, Graduate max C6).

4. 4. 4. Course learning outcome (CLO) agent

The course learning outcome (CLO) Agent generates a small set of course learning outcomes (2–4 CLOs) that are consistent with the course character and the targeted Bloom levels. It takes as input the course description (provided or inferred), Bloom codes, course-character codes, and the educational level, and returns a structured CLO list with identifiers, outcome statements, and the required mappings. Within this framework, CLOs function as the main alignment anchor, because they make the links between intended outcomes, learning activities, and assessments explicit under constructive-alignment principles [1].

The agent selects operational action verbs that match the target Bloom level and then checks each CLO against degree-level caps (Diploma $\leq$ C4, Bachelor $\leq$ C5, Graduate $\leq$ C6). Each CLO is expressed in the format "Mahasiswa mampu [action verb] [object]" and is limited to 20 words to preserve measurability and reduce ambiguity.

4. 4. 5. Learning objective (LO) agent

The learning objective agent decomposes each CLO into 2–3 operational LOs. It takes the course description and the CLO list as input and produces a hierarchical LO list with parent links that preserve Bloom-level constraints. LOs preserve traceability between macro-level outcomes and operational indicators [27].

Identifiers are assigned in the form LO-[parent_index] [sub_index], and a key constraint is enforced: an LO's Bloom level must not exceed its parent CLO's Bloom level. This hierarchy supports precise mapping to weekly activities while maintaining cognitive progression and scaffolded sequencing [18].

4. 4. 6. Weekly schedule agent

The weekly schedule agent distributes LOs across a 14-week semester schedule following cognitive progression principles, receiving course description, CLO list, and LO list as input and producing a schedule with week numbers, assigned LO codes, learning topics, and learning objectives. The temporal distribution follows scaffolding principles and cognitive load management to optimize learning progression [18].

The agent groups LOs by Bloom code, sorts from lowest to highest cognitive level, and assigns them according to pedagogical progression: weeks 1–3 prioritize C1–C2 (foundational knowledge), weeks 4–9 prioritize C3–C4 (application and analysis), and weeks 10–14 prioritize C5–C6 (evaluation and creation). Learning objectives are copied directly from LO statements to maintain alignment, and LO repetition across weeks supports spaced practice.

4. 4. 7. Assessment weight agent

The assessment weight agent assigns weights to six assessment components using the course-character codes and the generated CLO/LO set as inputs. It outputs a weight vector, which total is constrained to 100%. The weighting is derived from the intended outcomes and their cognitive targets, so that assessment emphasis follows the course's dominant Bloom profile and remains consistent with constructive-alignment requirements [19].

A fixed mapping is defined between assessment types and Bloom levels (Quiz: C1, Midterm: C2, Tasks: C3, Final exam: C4, Case study: C5, Project: C6). The agent first computes the dominant Bloom level from the CLO/LO distribution, then assigns the largest share (25–35%) to the corresponding assessment type. The remaining weight is distributed across the other components proportionally, subject to the 100% normalization constraint. The scheme aligns assessment weighting with the targeted cognitive levels and can help limit overweighting of components that are not aligned with those targets [17].

4. 5. Constraint verifier

4. 5. 1. Verifier scope and role

The Constraint verifier will execute on the final artifact bundle and does not generate content. It consumes the multi-agent output and applies deterministic checks, namely completeness, compliance, redundancy, spacing, phase progression, and assessment fit score. This constraint verifier separation is standard in reliability-oriented structured output pipelines, where schema validation and rule-based verification are prerequisites for machine-usable artifacts [15, 28].

4. 5. 2. Completeness

Completeness evaluates whether all required components are present in the generated artifact. It is used as a preliminary indicator of structured-output reliability in downstream

pipelines. In the multi-agent output, missing fields can be more damaging than imperfect phrasing because validation and analytics may fail when expected keys are absent. Completeness is computed by checking whether each required component is satisfied and then aggregating the results into a normalized score. This formulation follows structured output benchmarks and extraction studies that treat field coverage and missingness as core reliability [16, 29]

$$\text{Completeness} = \frac{\text{require component satisfied}}{\text{require component}}. \quad (1)$$

Component is considered satisfied when the required JSON evidence is present. Beyond verifying presence, this constraint is independent of output formatting, which keeps it robust to surface variation while still penalizing incomplete or unusable artifacts.

4. 5. 3. Compliance

Compliance measures whether the output satisfies the hard constraints required by the learning-plan schema, including format requirements, boundary conditions, and internal consistency rules. Constraint checks are standard in structured extraction and structured generation because they distinguish fully valid outputs from those that are only nearly valid, which supports iterative refinement during system development. Compliance formalizes rules such as restricting week indices to 1–14, requiring each week to include 1–3 LO codes and 1–3 CLO items, and ensuring that assessment weights sum to 100. Rule-based validity is widely treated as a primary quality gate in structured output and extraction evaluation [28, 30]

$$\text{Compliance} = 1 - \frac{\text{violated rules}}{\text{total rules}}. \quad (2)$$

Violations are computed deterministically from the JSON to enable reproducible scoring across runs and ablation settings. Compliance penalizes structural errors even when all required fields exist, so it remains sensitive to outputs that appear plausible but still fail under downstream use.

4. 5. 4. Redundancy

Redundancy represents repetitive behavior in weekly schedules by measuring how often weeks become identical under a content signature, without penalizing justified reuse of LO codes. The signature is computed deterministically from the week's assigned LO identifiers, with CLO identifiers included when available in an order-invariant form. Repetition and degeneration are well-known failure modes in neural generation, and repetition-oriented metrics are commonly used to identify low-diversity, template-driven outputs [31]. In learning-plan generation, redundancy is problematic when the same topics and objectives recur without pedagogical justification, because it reduces instructional coverage. A week-level signature is defined, and the proportion of weeks that duplicate at least one other week under this signature is computed

$$\text{Redudancy} = \frac{\text{identical weeks}}{14}. \quad (3)$$

The metric lies in [0, 1], where lower values indicate fewer identical weeks. This signature-based duplication analysis aligns with repetition diagnostics used in generation evaluation and repetition reduction research [32].

4. 5. 5. Spacing

Spacing measures whether repeated coverage of the same LO is distributed across weeks rather than clustered. Distributed practice and spaced learning are consistently associated with improved long-term retention, motivating evaluation measures that reflect temporal dispersion of repeated learning objectives [31]. In this context, spacing is computed from week-to-LO assignments, and it summarizes the average gap between successive occurrences of the same LO code. A higher spacing score indicates that repetitions are spread across time, which better aligns with evidence-based learning schedules

$$\text{Spacing} = \frac{1}{|C|}\sum_{\tilde{n}\in C}\left(\frac{1}{(k_{\tilde{n}}-1)}\sum_{i=1}^{k_{\tilde{n}}-1}\left(\varpi_{\tilde{n},i+1}-\varpi_{\tilde{n},i}\right)\right). \quad (4)$$

Here C is the set of LO codes that appear at least twice, and $\varpi_{c,i}$ are the sorted weeks in which code c appears. If no LO repeats, spacing can be reported as undefined or set to a maximum sentinel value, depending on the experimental protocol.

4. 5. 6. Phase progression

Phase progression evaluates whether cognitive difficulty progresses across coarse phases of the semester, rather than requiring strict week-by-week monotonicity. Bloom-level analyses have been used to study the cognitive distribution of assessments and to evaluate LLM performance across levels of cognitive complexity [21]. For curriculum design, phase-level progression is a practical compromise because real courses often revisit earlier skills while still increasing overall cognitive demand across mid-semester and end-of-semester phases. Each week is therefore mapped to an aggregate Bloom level based on its assigned LOs, and the phase sequence is then evaluated for non-decreasing progression

$$\text{Phase Progression} = \\ = \frac{\text{phase transitions}\left(\text{non}-\text{decreasing}\right)}{\text{phase transitions}}. \quad (5)$$

Weeks are partitioned into phases (e.g., 1–3, 4–6, 7–9, 10–14), and each phase is assigned an aggregate Bloom score (e.g., mean of weekly representatives). A phase transition is counted as satisfying progression if the next phase's score is ≥ the previous phase's score. This produces a normalized score in [0,1] that is robust to legitimate within-phase fluctuations. The choice is consistent with analytics perspectives that favor interpretable progression indicators for course design and intervention.

4. 5. 7. Assessment fit

Assessment fit score measures whether the assessment weight distribution matches the declared course characteristics, such as conceptual-versus-applied orientation. Constructive-alignment research frames assessment as an integral part of outcome- and activity-aligned course design, and shows that misalignment can harm validity and learning [33]. Recent higher-education technology work also reports that explicit outcome-activity-assessment mappings enable analytics-driven improvement and support gover-

nance practices [34, 35]. Assessment fit is operationalized as a rule-based similarity between the observed weight vector and a characteristic-dependent target pattern, enabling reproducible comparisons across ablations and course types

$$\text{Assesment Fit} = 1 - \frac{\sum_{j \in A} \alpha_j \left| \varpi_j - \tilde{\varpi}_j(\text{characteristics}) \right|}{\sum_{j \in A} \alpha_j}. \quad (6)$$

Here, ϖ_j is the generated weight for assessment type $j \in A$ (project, case, quiz, tasks, midterm, final) and $\tilde{\varpi}_j(\cdot)$ is the target (or recommended) weight given the course characteristics under a transparent rule set. The coefficients α_j allow emphasizing more diagnostic components (e.g., project/tasks for applied courses) while keeping the score in [0, 1]. This formulation operationalizes constructive alignment as a measurable distance between intended assessment design and generated assessment design, consistent with alignment-focused frameworks and rubrics.

4. 6. Experimental protocol

4. 6. 1. Evaluation setting and run protocol

Two experiments are conducted using identical constraint-verification protocols and metric definitions, allowing direct comparison across variants. The evaluation covers 12 courses with 10 repeated runs per course, producing 120 attempted runs per variant. A run is counted as successful if it produces a complete learning-plan JSON artifact that passes the verifier (i.e., satisfies schema constraints and domain rules). Unsuccessful attempts are retained only in the run-count summary and are not included in metric aggregation. This separation allows acceptance frequency and accepted-artifact quality to be reported as distinct experimental outcomes.

Six automated constraint-verifier metrics are computed directly from the generated JSON artifacts: completeness, compliance, redundancy, spacing, phase progression, and assessment fit. Higher values indicate better performance for completeness, compliance, phase progression, and assessment fit, whereas lower values indicate better performance for redundancy. Spacing is interpreted as a distributional measure, where larger average distances between repeated LO occurrences generally indicate more evenly distributed repetitions. For clearer interpretation and comparability across runs, redundancy and spacing are additionally normalized prior to analysis. The same metric definitions, normalization steps, and acceptance criteria are applied to every variant and every model. This keeps all structural comparisons under a shared scoring regime.

Recent empirical studies evaluate LLM driven planning pipelines using measurable outputs and reproducible experimental protocols [1, 2]. This protocol is extended by rerunning the proposed multi-agent framework on DeepSeek, Qwen, and Llama model families, alongside the Gemma-based reference setting already used in the study. Each model repeats the same ablation settings, the same single-agent baseline, and the same verifier-based acceptance criteria, enabling direct cross-model structural comparison. Robustness is interpreted as stability of the relative variant ordering, the dominant agent contributions, and the A0-versus-SA performance gap under different generative backbones. Agent influence is then computed per model from ablation deltas to identify modules with consistently strong or weak structural contributions.

4. 6. 2. Multi agent ablation study (A0–A6)

This experiment removes exactly one agent/module at a time from the full configuration (A0) to attribute metric degradations to specific functional responsibilities, with ablation settings summarized in Table 1. For every ablation setting, the constraint verifier and all non-removed agents are held fixed, so differences can be attributed to the removed module under the same evaluation criteria. The workload, the metric definitions, and the acceptance rule for successful runs are also kept unchanged across all ablation settings. When one agent is removed, no replacement module is introduced, so the observed degradation reflects the loss of that component under the same deterministic verification regime. Results are summarized as mean values computed over runs that satisfy the verification checks, while acceptance counts are reported separately to preserve visibility of verifier pass rates.

Table 1

Ablation mapping (A0—A6)

Variant (V)	Removed agent
A0	None (all six agents active)
A1	Agent-1: course character
A2	Agent-2: Bloom taxonomy
A3	Agent-3: CLO
A4	Agent-4: LO
A5	Agent-5: assessment weight
A6	Agent-6: weekly schedule

A controlled ablation experiment was conducted to isolate and quantify the structural contribution of each agent within the proposed multi-agent pipeline. The ablation protocol systematically deactivated one agent at a time while maintaining identical constraint-verification settings and evaluation metrics. Variant A1 corresponds to the removal of the course character agent, allowing the assessment of its impact on structural validity and cross-artifact coherence. The procedure was continued sequentially up to variant A6, where A2 disables the Bloom taxonomy agent, A3 excludes the CLO agent, A4 removes the LO agent, A5 deactivates the assessment weight agent, and A6 omits the weekly schedule agent. This controlled degradation design enables component-level performance attribution under a shared deterministic constraint-verification regime. This design separates component-level attribution from changes in workload, scoring rules, and acceptance criteria. As a result, differences between A0 and A1–A6 can be interpreted as structural contribution differences under matched experimental conditions.

4. 6. 3. Baseline comparison (A0 vs SA)

To isolate the impact of architectural decomposition, this study compares the full multi-agent configuration (A0) against a single-agent baseline (SA) that generates the entire learning-plan bundle in a single pass. Both settings receive the same institutional, program, and course inputs, and both are evaluated with the same constraint verifier and the same metric suite on the same workload of 12 courses with 10 repeated runs per course. Metric aggregation is restricted to runs that pass verification, while run-acceptance counts are considered separately as an additional indicator of structural reliability. Under this matched protocol, any observed differences between A0 and SA can be attributed to the generation architecture rather than to changes in inputs, scoring rules, or acceptance conditions.

5. Research results of the deterministic multi-agent generative pipeline

5. 1. Multi-agent generative pipelines framework for learning plan

5. 1. 1. Framework architecture and components

The proposed method is implemented as an auditable workflow that records validation outcomes in deterministic logs and produces machine-checkable outputs. It converts minimal course metadata, together with institutional or program context and an optional course description, into a structured learning-plan bundle with explicit trace links, as shown in Fig. 1. Intermediate results are stored as structured fields. Relations among artifacts, including CLO to LO, LO to weekly schedule, and LO to assessment, are emitted as identifiers that can be checked deterministically. The design uses specialized agents that pass intermediate outputs through explicit handoffs, which is consistent with prior multi-agent LLM workflows for controlled task execution [36, 37].

Fig. 1 illustrates the architecture of the Learning Plan Generator. The input includes minimal course metadata, institutional or program context, and an optional course description. If the description is unavailable, the system first infers a short substitute description. The output is a structured artifact bundle consisting of:

a) course character;
b) Bloom taxonomy classification result;
c) CLO set;
d) LO set;
e) weekly schedule; and
f) assessment blueprint.

The pipeline is implemented as specialized agents that operate in a mixed sequential–parallel topology: upstream agents create the outcome backbone (course characterization to Bloom tagging to CLO inference to LO formulation), while downstream agents generate weekly scheduling and assessment in parallel using the LO set as the shared constraint anchor. This topology aligns with multi-agent orchestration frameworks that formalize role-based cooperation and structured inter-agent communication for task completion.

5. 1. 2. Sequential agents

The sequential stage establishes the minimum structure needed for traceable learning-plan generation:

– Course character taxonomy agent derives a compact machine-readable profile of the course from the available description. The profile covers domain focus, competency orientation, and content scope. When no description is provided, the agent uses the inferred description together with course metadata. This profile supplies a common reference for later decisions, including the expected emphasis of conceptual and applied material;

– Bloom taxonomy agent assigns an indicative cognitive-demand profile to the course and to candidate outcome statements by enforcing coherence between the selected action verb and the Bloom level. The resulting labels are stored and reused by later agents. This reduces divergence when several components are generated;

– CLO agent infers course learning outcomes as action-oriented competency boundaries. Each CLO is emitted with a unique identifier and an associated Bloom tag. These outputs form the parent layer for traceability and support referential integrity across the bundle;

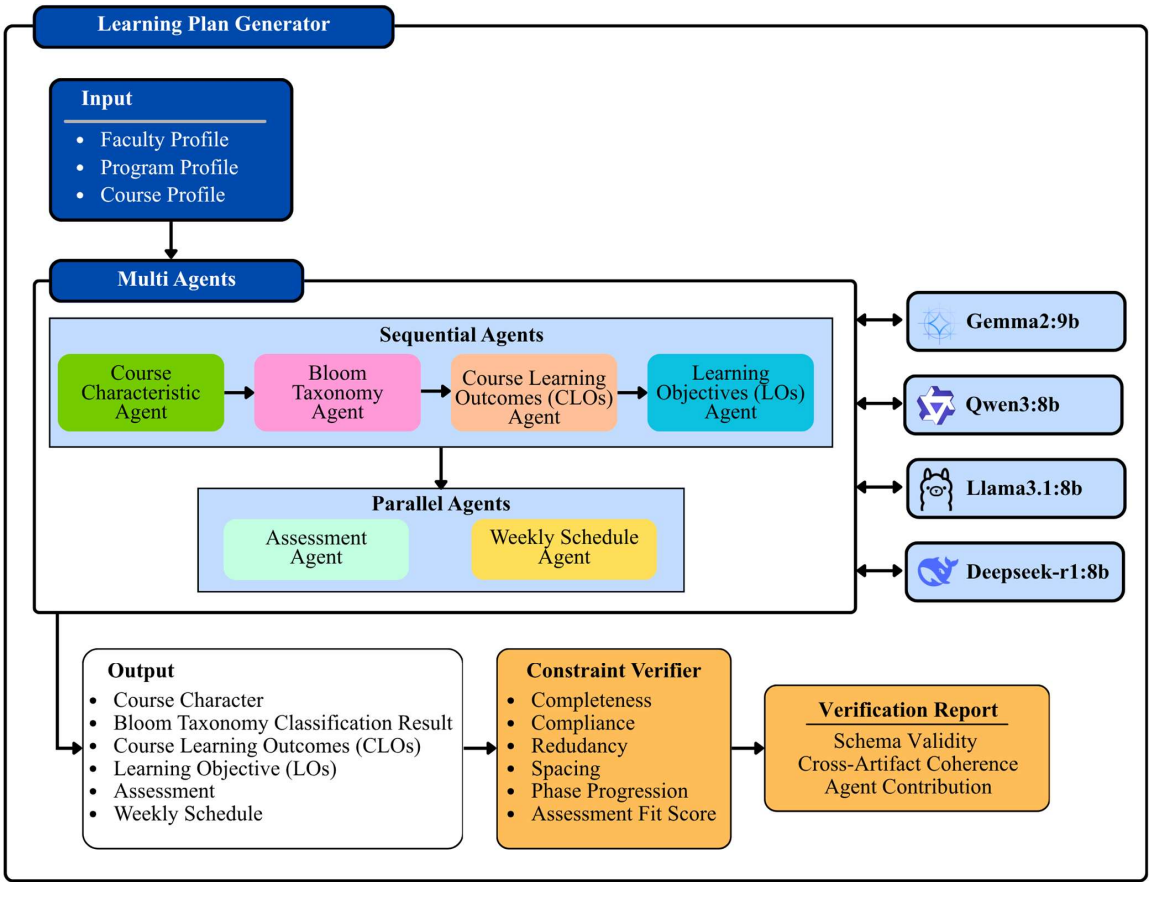

Fig. 1. Architecture of the learning plan generator

– LO agent refines each CLO into operational objectives that can be scheduled and assessed. Each LO declares:

a) a parent CLO id;

b) an action verb with Bloom-consistent tag;

c) a concise objective statement.

This agent also emits trace anchors (LO identifiers) that downstream generators must reference, ensuring that traceability is produced during generation rather than as post-hoc annotation.

5. 1. 3. Parallel agents

After LOs are produced, the method generates two artifacts in parallel, both conditioned on the LO set:

– Weekly schedule agent constructs a fixed-horizon schedule (e.g., 14 weeks) that includes a topic and weekly objective statements for each week. Importantly, each weekly objective must reference at least one LO identifier. In the architecture diagram, this is represented by the dotted trace link LO to Week anchoring, meaning that the weekly schedule is not treated as a free-form narrative, it is an outcome-anchored structure, which trace links are explicit and testable.

– Assessment weight agent generates a weighted assessment blueprint over assessment components (e.g., project-based learning, case study, quiz, tasks, midterm, final). Each component must explicitly reference the subset of LO identifiers it measures. In the diagram, the dotted trace link "LO to Assessment mapping" denotes this requirement: every LO must have evidence allocation, and every assessment component must declare its intended LO coverage.

5. 1. 4. Generation outcomes

The framework was executed to generate learning plans for 12 higher-education courses. Based on the experimental protocol described in Section 4. 6, the Qwen LLM generated learning plans for all trials while satisfying schema validity and maintaining coherent consistency across artifacts. Under the same controlled procedures, the remaining backbones produced slightly fewer complete learning plans, with Llama generating 119, Gemma 118, and DeepSeek 116 out of 120 trials.

Overall, the proposed framework yielded stable generation outcomes across the four evaluated LLMs. Output conformity is ensured through rule sets embedded within each agent and a deterministic verifier integrated into the generation workflow. The sequential process within the framework preserves rule compliance at each generation step, thereby maintaining alignment with OBE curriculum requirements. In addition, the parallel process supports workload distribution by allocating tasks and schedules across the instructional weeks, helping prevent excessive concentration of difficulty within specific weeks of the learning period.

5. 2. Schema validity and cross-artifact coherence

Fig. 2 reports structural completeness for all variants and models, computed as the fraction of required components that are present score = passed/6. The baseline A0 shows consistently high completeness across the four models, ranging from 0.9682 to 1.00, with Qwen3 8B achieving 1.00. In contrast, the SA produces substantially lower completeness 0.59–0.66, indicating that, on average, roughly two required components are missing from the six-component output. The ablation variants fall into two empirical patterns. Removing A1, A2, or A6 produces a clear decrease in completeness, with scores concentrated around 0.80–0.83. By contrast, A3, A4, and A5 remain close to the full configuration, at about 0.97–1.00. The spread across backbones is small within each variant. This indicates that component presence is influenced more by the pipeline design under the shared verifier than by the choice of base LLM.

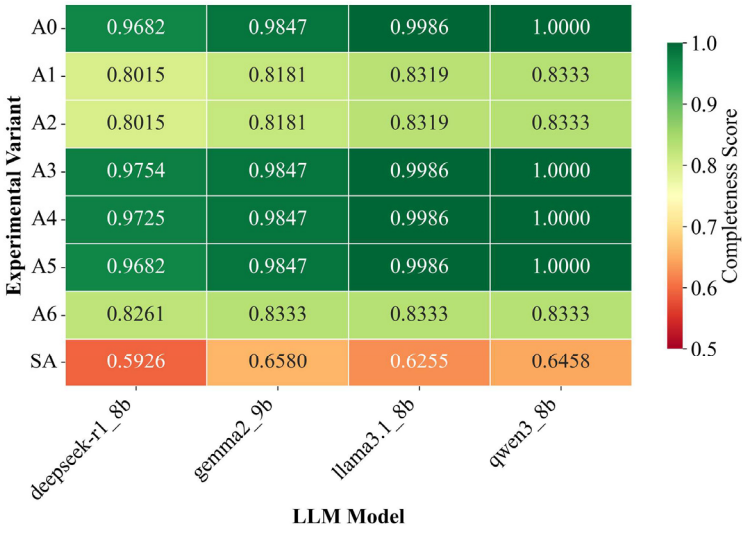

Fig. 2. Heatmap of structural completeness across all variants and models

Fig. 3 reports compliance, which summarizes satisfaction of schema rules and domain constraints, including cross-artifact checks. Unlike completeness, compliance does not only reflect whether the fields are present. It also reflects whether the generated fields and mappings satisfy the verifier after generation. A0 reaches high compliance, ranging from 0.9376 to 0.9698, with the highest value observed for Llama 3.1_8b. The ablation results show a more differentiated pattern than completeness. Removing the course character agent (A1) or the CLO generation agent (A3) produces the largest decline, reducing compliance to about 0.77–0.83. This suggests that course characterization and CLO construction play a major role in preserving downstream rule consistency. The remaining removals, namely A2, A4, and A6, stay in a moderate-to-high range of about 0.83–0.92. A5 does not lower compliance in this workload. In several model settings, it matches or slightly exceeds the A0 value. This indicates that the effect of that module on verifier satisfaction is not strictly monotonic. Compliance also varies more across models than completeness. This is consistent with the fact that rule satisfaction depends on both the architectural decomposition and the model's ability to produce constraint-consistent mappings.

Fig. 2, 3 show that the multi-agent design maintains high structural completeness while achieving substantially stronger compliance than the SA. The ablation patterns also indicate that some modules have larger effects on structural presence as reflected by completeness and others affect rule satisfaction and cross-artifact coherence as reflected by compliance, under a shared verifier and fixed workload.

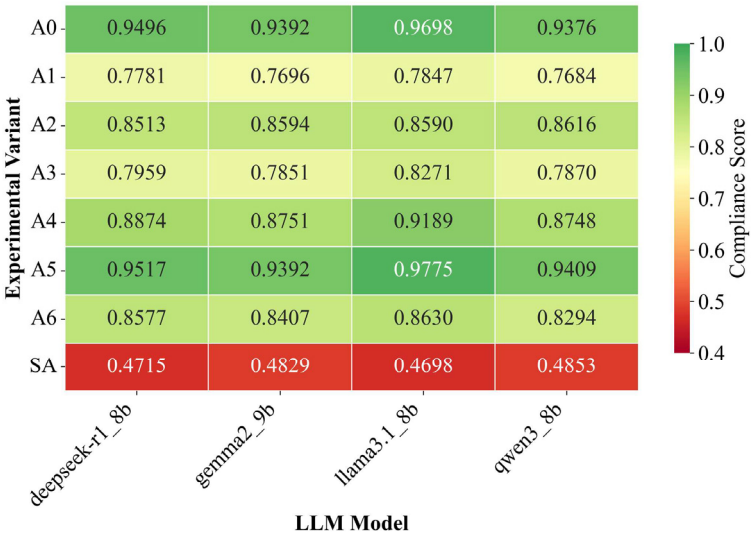

Fig. 3. Heatmap constraint compliance across all variants and model

By contrast, the differences in redundancy and spacing are weaker and less uniform. For redundancy, the preferred direction varies by model. A0 yields lower redundancy than SA for DeepSeek r1_8b and Qwen 3_8b, whereas SA is lower for Gemma 2_9b and Llama 3.1_8b. A similar pattern appears for spacing. SA attains higher spacing in three models, namely DeepSeek r1_8b, Gemma 2_9b, and Qwen 3_8b, while A0 exceeds SA only for Llama 3.1_8b, with a gap of about 0.16. Assessment fit is also mixed. A0 is higher for DeepSeek r1_8b, Llama 3.1_8b, and Qwen 3_8b, but SA is higher for Gemma 2_9b. Taken together, these results suggest that distributional properties of weekly allocation and assessment weighting remain more sensitive to model-specific behavior than the core structural measures captured by completeness, compliance, and phase progression.

5. 3. Baseline comparison

The performance difference between A0 and SA under matched inputs, verifier settings, and metric definitions is examined across four LLM backbones and six evaluation metrics, as presented in Fig. 4. The difference is defined as $\Delta = A0 - SA$, so positive values indicate an advantage for the multi-agent configuration. Table 2 reports the corresponding absolute scores for both settings. The clearest separation appears in the metrics that directly reflect structural validity. For completeness, A0 remains consistently high across all models, ranging from 0.9682 to 1.0000, whereas SA falls to 0.5926–0.6580. This corresponds to gains of approximately 0.33–0.37. Compliance shows an even larger gap. A0 reaches 0.9376–0.9698, while SA remains at 0.4698–0.4853, yielding gains of about 0.45–0.50. Phase progression also favors A0 in every model. The multi-agent configuration attains 1.0000 throughout, whereas SA ranges from 0.6788 to 0.8372, which corresponds to gains of about 0.16–0.32.

Table 2

Baseline model performance comparison between A0 and SA

Model	Variant	Completeness	Compliance	Redundancy	Spacing	Phase	Assessment fit
Deepseek r1_8b	A0	0.9682	0.9496	0.7803	0.5081	1	0.5696
	SA	0.5926	0.4715	0.8881	0.951	0.7436	0.5192
Gemma 2_9b	A0	0.9847	0.9392	0.9231	0.7118	1	0.3406
	SA	0.658	0.4829	0.9064	0.7433	0.6788	0.5436
Llama 3.1_8b	A0	0.9986	0.9698	0.9519	0.4871	1	0.4905
	SA	0.6255	0.4698	0.7886	0.3276	0.8372	0.4206
Qwen 3_8b	A0	1	0.9376	0.8063	0.5039	1	0.4898
	SA	0.6458	0.4853	0.9203	0.6227	0.7181	0.462

The baseline comparison shows that the multi-agent configuration provides consistent improvements on the metrics that directly reflect structural completeness, rule satisfaction, and progression constraints under the shared verifier. The consistency of this pattern across four backbones supports the interpretation that architectural decomposition and deterministic verification, rather than model choice alone, are the main sources of the observed gains in structured learning-plan generation.

5. 4. Agent contribution

Ablation results, averaged across four LLMs, indicate clear differences in how much each agent contributes to overall performance. Course character, CLO, and learning outcomes (LO) yield the largest positive mean drops when removed, which indicates that these modules provide the strongest benefit when present. By contrast, weekly schedule and assessment weight show small or negative mean drops, suggesting that their effects are weaker and depend more on model and context. This aggregated pattern is summarized in Fig. 5.

Agent-level sensitivity varies by model, and the ablation profiles make those differences explicit. For gemma2_9b, A3 produces the largest drop at 5.4%, positioning the CLO generation agent as the primary performance driver (Fig. 6). A1 and A4 follow at 6.2% each, while A5 shows a negative drop of 2%, suggesting limited contribution or even interference. A6 registers near zero impact, indicating the weekly scheduling agent adds little value for this architecture.

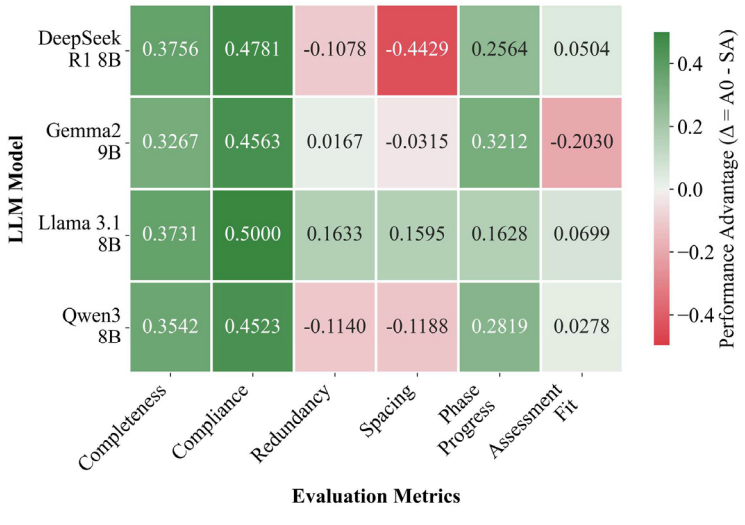

Fig. 4. Performance advantage ($\Delta = A0 - SA$) across models and metrics

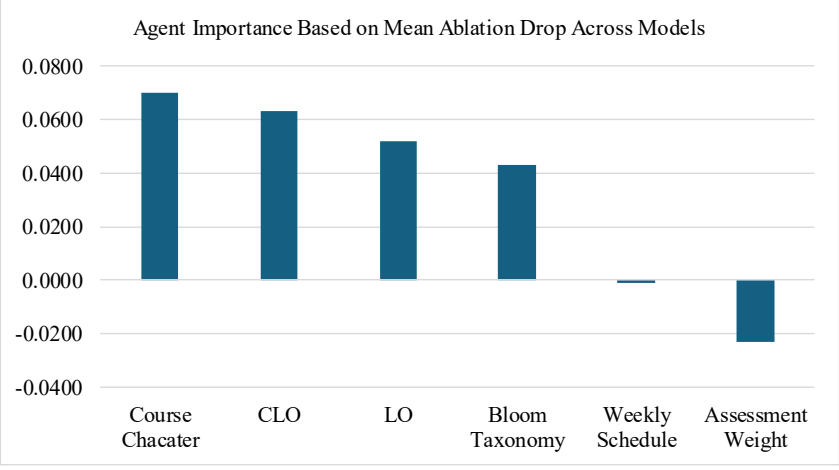

Fig. 5. Agent importance based on mean ablation drop across models

observed across all models (Fig. 8). A3 and A4 maintain moderate influence at 6.3% and 5.4%, but clearly operate in A1's shadow. Both A5 and A6 show negative drops of 4.9% and 2.4%, continuing the pattern where constraint-heavy agents prove counterproductive.

Qwen3_8b presents a more balanced profile, with A1 and A3 sharing primary responsibility at 6.3% and 6.0% respectively (Fig. 9). A4 contributes 5% as a tertiary factor. A6 shows a 3.5% negative drop, while A2 registers just 0.4%, the lowest impact observed, raising questions about Bloom taxonomy agent necessity.

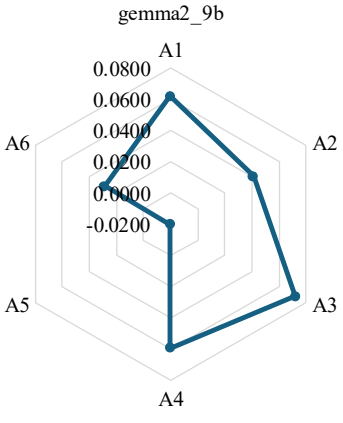

Fig. 6. Agent-level ablation impact for gemma2_9b

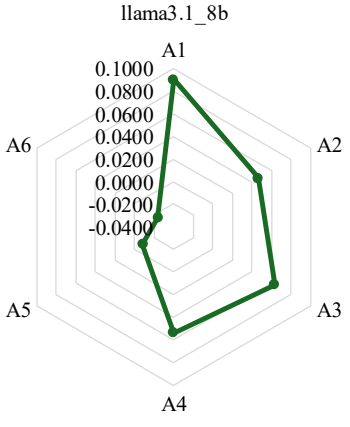

Fig. 8. Agent-level ablation impact for llama3.1_8b

Deepseek-R1_8b shows a different hierarchy, with A1 emerging as the strongest contributor at 6.6% degradation (Fig. 7). A3 and A4 provide secondary support at 5.4% and 5.2%, while A6 produces a substantial negative drop of 6.1%. Even more striking is A5's 7% negative effect, indicating that both assessment weight and scheduling agents actively degrade performance when included. This suggests DeepSeek benefits from fewer explicit constraints.

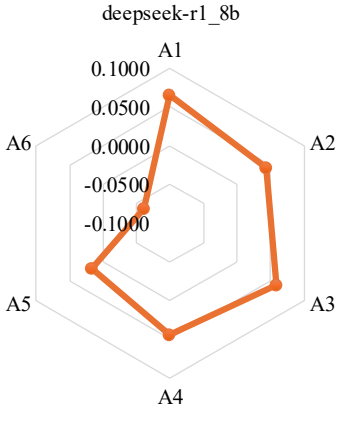

Fig. 7. Agent-level ablation impact for deepsek-R1_8b

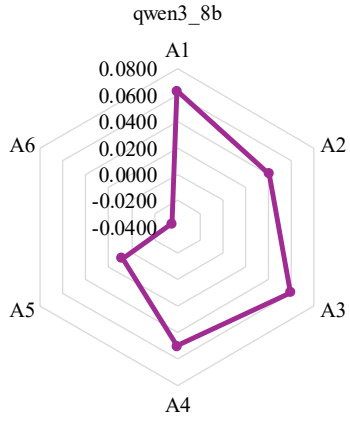

Fig. 9. Agent-level ablation impact for qwen3_8b

Llama3.1-8b demonstrates the most extreme sensitivity, with A1 dominating at 9%, the highest drop

Across all models, A1 consistently emerges as the most critical component, though sensitivity ranges from 6.2% to 9%. A3 serves as a reliable secondary contributor at 5.4–6.3%. The negative performance of A5 and A6 in three of four models indicates these agents impose constraints that conflict with inherent model capabilities, particularly in DeepSeek's architecture where combined negative effects exceed 13%.

The impact-stability analysis compares each agent's average ablation impact against the variability of that impact across models. Larger positive impacts indicate stronger dependence of performance on the agent, while lower

cross-model standard deviation indicates more consistent contributions. The global mean lines divide the space into four regions, separating high-impact agents from low-impact ones and stable agents from unstable ones. Agents in the high-impact, high-stability region are the most desirable because they combine strong contributions with robust behavior. This pattern is shown in Fig. 10.

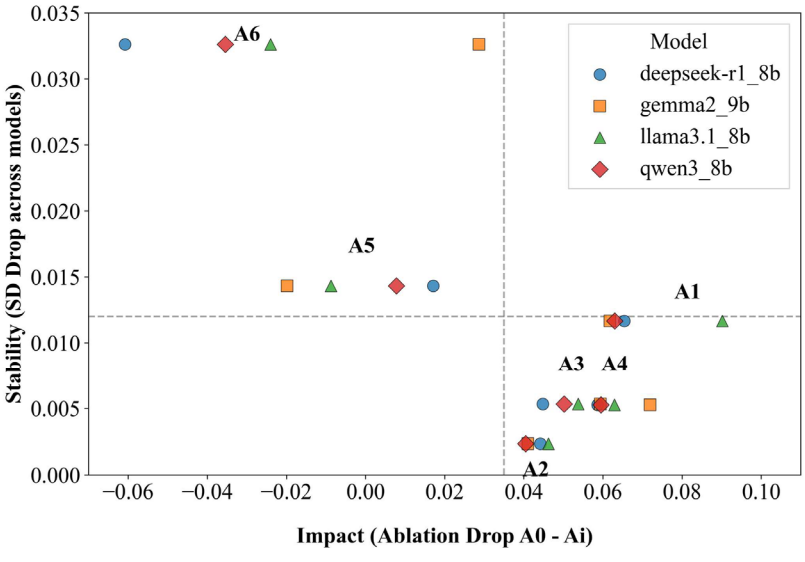

Fig. 10. Impact-stability analysis of ablation variants across models, showing agent impact across models

Across models, course character agent combines high impact with relatively stable behavior, making it the strongest overall component. CLO agent also shows a favorable balance between impact and stability, while LO agent delivers moderate impact with the highest stability. On the other hand, assessment weight agent remains low-impact and highly variable, which indicates a limited and unreliable contribution under this evaluation setting.

6. Discussion of results obtained in the study of the deterministic multi-agent generative pipeline

This study produced a deterministic multi-agent generative pipeline by combining task-specific agent decomposition with deterministic verification at the learning-plan level. The main finding is a stable structural advantage of the full multi-agent configuration over the single-agent baseline under the same workload and verifier. As defined in eq. (1) and eq. (2), completeness and compliance represent the presence of required components and the satisfaction of schema and OBE rules. Fig. 2–4, and Table 2 show that A0 maintains high values on these measures across all four evaluated LLM backends. Completeness for A0 ranges from 0.9682 to 1.00, whereas SA remains at 0.5926–0.6580. Compliance for A0 ranges from 0.9376 to 0.9698, whereas SA remains at 0.4698–0.4853. Eq. (5) and Table 2 also show that phase progression stays at 1.00 for A0 in all models, whereas SA remains at 0.6788–0.8372. These findings indicate that, under the tested setting, the proposed solution resolves the problematic part related to missing components, broken schema constraints, and invalid cross-artifact relations in structured OBE learning-plan generation. At the same time, the smaller and less uniform differences in redundancy and spacing indicate that distribution-oriented dimensions remain more sensitive to model behavior than structural validity measures.

The observed gains can be explained by the interaction between staged agent responsibilities and deterministic verification of the assembled bundle. The agents first generate the constituent sub-artifacts, and the final bundle is then checked by a deterministic verifier. When violations are detected, the pipeline can trigger targeted retries at the relevant stage. This design turns structural defects into explicit rule violations instead of allowing them to remain hidden in the final artifact. As a result, inconsistencies are less likely to propagate into dependent components such as CLO-LO relations, LO-to-week mapping, and assessment-weight allocation. In practical terms, the method can help prepare more consistent and machine-verifiable OBE learning-plan drafts for review. The expected effect is therefore not a direct claim about learning outcomes. It is a more reliable preparation process for curriculum artifacts that must satisfy explicit structural rules.

Compared with existing studies, the obtained result addresses a narrower but more formalized reliability target. CoPL uses multiple agents to support personalized learning implementation and reflective interaction with pre-service teachers [11]. EduMAS uses coordinated LLM-based agents to handle explanation-oriented educational assistance tasks [12]. Even so, neither study is designed for deterministic acceptance of a full curriculum bundle, which parts must remain structurally valid and explicitly linked under a shared rule set. SLOT improves schema conformity and preserves content in structured outputs [15]. Grammar-constrained decoding improves syntactic correctness and semantic accuracy in logical parsing tasks [16]. The present results suggest that, in OBE planning, local schema validity and syntactic well-formedness are still not enough when several curriculum artifacts must remain consistent with one another. The contribution of the present method lies in coupling task decomposition with bundle-level rule verification, rather than relying only on prompt design, local schema control, or constrained decoding.

One strength of the proposed method is that reliability is built from six specialized agents, explicit trace-link generation, and a hard-gated verifier applied to the final bundle. Another strength is that failures can still be traced to the component level, because the architecture preserves the source of each generated sub-artifact and the verifier checks the relations among them. A further strength is the presence of controlled ablation evidence, which makes it possible to see how reliability is formed inside the system instead of reading the final score as a black-box outcome. Fig. 5 and Fig. 10 show that the course character agent contributes the most across models. The CLO agent provides the next strongest contribution, while the LO agent shows moderate impact with the highest stability. In contrast, the assessment weight and weekly schedule agents have weaker and more model-sensitive effects. This pattern indicates that the modules do not contribute in the same way to structural reliability. It also shows why the method can address one part of the

broader problem, namely controllable structured generation, explicit traceability, and deterministic rejection of invalid OBE artifacts.

The study is bounded by the institutional rule set embedded in the deterministic verifier and by the experimental setting used in evaluation. Institutions differ in CLO-LO granularity, weekly progression policies, and assessment rules, so the current rule set should not be transferred unchanged to other curricular contexts. The experiments also used fixed model versions, prompt settings, decoding parameters, and 12 courses with fixed inputs across repeated runs. For that reason, the reported gains should be interpreted as evidence under the tested protocol rather than as a general claim across institutions, course collections, or model settings. A natural extension is to evaluate the pipeline under additional institutional rule sets and broader course workloads to examine transferability beyond the current configuration.

From an implementation perspective, strict verification and retry handling add orchestration overhead, and the effort needed to design and maintain detailed rule sets remains substantial. These costs can be reduced by repairing only the failing sub-structure, introducing earlier failure checks, caching stable intermediate outputs, and separating hard structural constraints from more flexible pedagogical preferences. The next stage is broader evaluation under additional institutional rule sets and larger course collections, supported by expert review. That extension will require formalizing heterogeneous policies, keeping repair overhead under control, and assembling sufficiently broad expert-reviewed datasets for external validation.

7. Conclusions

1. A multi-agent generative pipelines framework for learning plan was engineered. Rule-based checks are executed inside the generation workflow, so invalid artifacts are rejected before acceptance. This design reduces reliance on manual post-editing and makes the acceptance process traceable at the artifact level.

2. Under the shared deterministic rules, the full multi-agent configuration achieved near-perfect structural validity, with completeness 0.9682–1.0000 and compliance reaching up to 0.9698. The accepted outputs preserved referential integrity across interdependent components, including learning outcomes, weekly schedules, and assessment-weight allocations. Coherence dimensions driven by distributional patterns remained more sensitive to model behavior, which justifies reporting them as explicit metrics.

3. When evaluated under the same workload and constraint protocol, the multi-agent pipeline consistently outperformed the single-agent baseline. The baseline produced completeness 0.5926–0.6580 and compliance 0.4698–0.4853, whereas the proposed pipeline-maintained phase progression at 1.00. The performance gap aligns with reduced structural drift when responsibilities are decomposed and continuously checked by deterministic validation.

4. Controlled ablation (A0–A6) provided component-level evidence of how reliability is formed and where degradation emerges. The course character agent contributed the largest performance support, while CLO and LO agents delivered stable secondary contributions that strengthened traceability. In contrast, the assessment weight and weekly schedule components showed lower average impact and higher model-dependent variability, consistent with their distributional nature.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing

The study was carried out without financial support.

Data availability

Data generated or analyzed during this study are available from the corresponding author on reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies in creating the submitted work.

Authors' contributions

Mohammad Fadly Syahputra: Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing; **Opim Salim Sitompul**: Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision; **Fahmi**: Methodology, Validation, Writing – review & editing, Visualization, Supervision; **Maya Silvi Lydia**: Methodology, Validation, Investigation, Visualization, Supervision; **Pauzi Ibrahim Nainggolan**: Software, Resources, Data Curation, Writing – review & editing, Project administration; **Rendra Mahardika**: Software, Resources, Data Curation, Writing – original draft, Project administration; **Riza Sulaiman**: Validation, Investigation, Writing – review & editing, Visualization, Supervision.

References

1. Derouich, M. (2025). Ensuring outcome-based curriculum coherence through systematic CLO–PLO alignment and feedback loops. Discover Education, 4 (1). https://doi.org/10.1007/s44217-025-00915-7
2. Vlachopoulos, D., Makri, A. (2024). A systematic literature review on authentic assessment in higher education: Best practices for the development of 21st century skills, and policy considerations. Studies in Educational Evaluation, 83, 101425. https://doi.org/10.1016/j.stueduc.2024.101425
3. Baig, M. I., Yadegaridehkordi, E. (2024). ChatGPT in the higher education: A systematic literature review and research challenges. International Journal of Educational Research, 127, 102411. https://doi.org/10.1016/j.ijer.2024.102411

4. Belkina, M., Daniel, S., Nikolic, S., Haque, R., Lyden, S., Neal, P. et al. (2025). Implementing generative AI (GenAI) in higher education: A systematic review of case studies. Computers and Education: Artificial Intelligence, 8, 100407. https://doi.org/10.1016/j.caeai.2025.100407
5. Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F. et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274. https://doi.org/10.1016/j.lindif.2023.102274
6. Moundridou, M., Matzakos, N., Doukakis, S. (2024). Generative AI tools as educators' assistants: Designing and implementing inquiry-based lesson plans. Computers and Education: Artificial Intelligence, 7, 100277. https://doi.org/10.1016/j.caeai.2024.100277
7. Celik, I., Kontkanen, S., Laru, J., Dalyanci, A. A. (2026). Co-constructing adaptive lesson plans with GenAI: Pre-service teachers' Intelligent-TPACK and prompt engineering strategies. Computers & Education, 241, 105485. https://doi.org/10.1016/j.compedu.2025.105485
8. Moorhouse, B. L. (2024). Beginning and first-year language teachers' readiness for the generative AI age. Computers and Education: Artificial Intelligence, 6, 100201. https://doi.org/10.1016/j.caeai.2024.100201
9. Kong, S. C., Yang, Y., Hou, C. (2024). Examining teachers' behavioural intention of using generative artificial intelligence tools for teaching and learning based on the extended technology acceptance model. Computers and Education: Artificial Intelligence, 7, 100328. https://doi.org/10.1016/j.caeai.2024.100328
10. Mzwri, K., Turcsányi-Szabo, M. (2025). Bridging LMS and generative AI: dynamic course content integration (DCCI) for enhancing student satisfaction and engagement via the ask ME assistant. Journal of Computers in Education. https://doi.org/10.1007/s40692-025-00367-w
11. Zhang, L., Yao, Z., Hadizadeh Moghaddam, A. (2025). Designing GenAI Tools for Personalized Learning Implementation: Theoretical Analysis and Prototype of a Multi-Agent System. Journal of Teacher Education, 76 (3), 280–293. https://doi.org/10.1177/00224871251325109
12. Li, Q., Xie, Y., Chakravarty, S., Lee, D. (2024). EduMAS: A Novel LLM-Powered Multi-Agent Framework for Educational Support. 2024 IEEE International Conference on Big Data (BigData), 8309–8316. https://doi.org/10.1109/bigdata62323.2024.10826103
13. Hauk, D., Soujon, N. (2026). How reliable are large language models in analyzing the quality of written lesson plans? A mixed-methods study from a teacher internship program. Computers and Education: Artificial Intelligence, 10, 100538. https://doi.org/10.1016/j.caeai.2025.100538
14. Lin, Z., Guan, S., Zhang, W., Zhang, H., Li, Y., Zhang, H. (2024). Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models. Artificial Intelligence Review, 57 (9). https://doi.org/10.1007/s10462-024-10896-y
15. Shen, Z., Wang, D. Y.-B., Mishra, S. S., Xu, Z., Teng, Y., Ding, H. (2025). SLOT: Structuring the Output of Large Language Models. Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, 472–491. https://doi.org/10.18653/v1/2025.emnlp-industry.32
16. Raspanti, F., Ozcelebi, T., Holenderski, M. (2025). Grammar-Constrained Decoding Makes Large Language Models Better Logical Parsers. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), 485–499. https://doi.org/10.18653/v1/2025.acl-industry.34
17. Boud, D., Soler, R. (2015). Sustainable assessment revisited. Assessment & Evaluation in Higher Education, 41 (3), 400–413. https://doi.org/10.1080/02602938.2015.1018133
18. Sweller, J., van Merriënboer, J. J. G., Paas, F. (2019). Cognitive Architecture and Instructional Design: 20 Years Later. Educational Psychology Review, 31 (2), 261–292. https://doi.org/10.1007/s10648-019-09465-5
19. Pereira, E., Nsair, S., Pereira, L. R., Grant, K. (2024). Constructive alignment in a graduate-level project management course: an innovative framework using large language models. International Journal of Educational Technology in Higher Education, 21 (1). https://doi.org/10.1186/s41239-024-00457-2
20. Kilinc, C., Ranaweera, C., Ugon, J., Cain, A., Pierce, C. (2025). Leveraging NLP-based tools for constructive alignment. ASCILITE Publications, 157–166. https://doi.org/10.65106/apubs.2025.2636
21. Almatrafi, O., Johri, A. (2025). Leveraging generative AI for course learning outcome categorization using Bloom's taxonomy. Computers and Education: Artificial Intelligence, 8, 100404. https://doi.org/10.1016/j.caeai.2025.100404
22. Khan, H. F., Qayyum, S., Beenish, H., Khan, R. A., Iltaf, S., Faysal, L. R. (2025). Determining the alignment of assessment items with curriculum goals through document analysis by addressing identified item flaws. BMC Medical Education, 25 (1). https://doi.org/10.1186/s12909-025-06736-4
23. Gemma: Open Models Based on Gemini Research and Technology (2024). arXiv. Available at: https://arxiv.org/html/2403.08295v1
24. Gao, H., Hashim, H., Md Yunus, M. (2025). Assessing the reliability and relevance of DeepSeek in EFL writing evaluation: a generalizability theory approach. Language Testing in Asia, 15 (1). https://doi.org/10.1186/s40468-025-00369-6
25. Neyem, A., González, L. A., Mendoza, M., Alcocer, J. P. S., Centellas, L., Paredes, C. (2024). Toward an AI Knowledge Assistant for Context-Aware Learning Experiences in Software Capstone Project Development. IEEE Transactions on Learning Technologies, 17, 1599–1614. https://doi.org/10.1109/tlt.2024.3396735
26. Zhao, Q., Zhang, M. (2025). Elimination-based reasoning with LLM for multiple-choice educational question answering. Journal of King Saud University Computer and Information Sciences, 37 (7). https://doi.org/10.1007/s44443-025-00122-2

27. Stamov Roßnagel, C., Lo Baido, K., Fitzallen, N. (2021). Revisiting the relationship between constructive alignment and learning approaches: A perceived alignment perspective. PLOS ONE, 16 (8), e0253949. https://doi.org/10.1371/journal.pone.0253949
28. Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G. et al. (2024). Structured information extraction from scientific text with large language models. Nature Communications, 15 (1). https://doi.org/10.1038/s41467-024-45563-x
29. Balasubramanian, J. B., Adams, D., Roxanis, I., de Gonzalez, A. B., Coulson, P., Almeida, J. S., García-Closas, M. (2025). Leveraging large language models for structured information extraction from pathology reports. Journal of Pathology Informatics, 19, 100521. https://doi.org/10.1016/j.jpi.2025.100521
30. Hu, R., Yang, Y., Liu, S., Li, Z., Liu, J., Ding, X. et al. (2025). Large language model driven transferable key information extraction mechanism for nonstandardized tables. Scientific Reports, 15 (1). https://doi.org/10.1038/s41598-025-15627-z
31. Yuan, C., Huang, H., Cao, Y., Cao, Q. (2024). Screening through a broad pool: Towards better diversity for lexically constrained text generation. Information Processing & Management, 61 (2), 103602. https://doi.org/10.1016/j.ipm.2023.103602
32. Guo, Y., Shang, G., Clavel, C. (2025). Benchmarking Linguistic Diversity of Large Language Models. Transactions of the Association for Computational Linguistics, 13, 1507–1526. https://doi.org/10.1162/tacl.a.47
33. Tractenberg, R. E. (2021). The Assessment Evaluation Rubric: Promoting Learning and Learner-Centered Teaching through Assessment in Face-to-Face or Distanced Higher Education. Education Sciences, 11 (8), 441. https://doi.org/10.3390/educsci11080441
34. Xia, Q., Weng, X., Ouyang, F., Lin, T. J., Chiu, T. K. F. (2024). A scoping review on how generative artificial intelligence transforms assessment in higher education. International Journal of Educational Technology in Higher Education, 21 (1). https://doi.org/10.1186/s41239-024-00468-z
35. Oprea, S.-V., Bâra, A. (2025). Transforming Education With Large Language Models: Trends, Themes, and Untapped Potential. IEEE Access, 13, 87292–87312. https://doi.org/10.1109/access.2025.3570649
36. Li, G., Al Kader Hammoud, H. A., Itani, H., Khizbullin, D., Ghanem, B. (). CAMEL: Communicative Agents for “Mind” Exploration of Large Language Model Society. arXiv. https://doi.org/10.48550/arXiv.2303.17760
37. Park, J. S., O’Brien, J., Cai, C. J., Morris, M. R., Liang, P., Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 1–22. https://doi.org/10.1145/3586183.3606763