# Generative AI Empowering the Reform and Practice of Diversified Teaching Evaluation in the Python Programming Course

**Yu Fan***

*Haikou University of Economics, Haikou, 571127, China*
**Corresponding author: fy_808@126.com*

***Abstract:*** *Technical characteristics of generative AI (large-scale pre-trained models), including natural language understanding and generation, code semantic modeling, contextual reasoning and multi-turn interaction, match well with the functions of teaching evaluation tasks like generating tasks, collecting evidence, interpreting performances, giving feedbacks and adjusting. To solve some problems existed in evaluating the Python Programming course, e.g., limitation of summative measurement, fragmentation of process evidence, delay of feedback adjustment, lack of interpretability of judgment, we build a logical framework for generative AI-empowered diversified teaching evaluation by mapping the technical characteristic-evaluation function, course competency objective-evaluation dimension and empowerment mechanism-structural elements. Construct an evaluation system from three aspects: multi-dimensional evaluation indicators oriented towards progressive programming competency; Process evaluation data generation and semantic analysis; Human-AI collaboration intelligent feedback and dynamic adjustment. Explore on how to reconfigure roles and allocate tasks when embedding generative AI within the evaluation process, how to intelligently interpret evaluation results and transform it into teaching decisions, quality assurance and boundary regulation. Our research provides an interpretable, traceable and adjustable support path for the evaluation of the Python Programming course.*

***Keywords:*** *Generative AI; Python Programming; diversified teaching evaluation; human-AI collaboration; evaluation system*

## Introduction

The Python Programming course emphasizes comprehensive skills such as code implementation, debugging and testing, algorithm design, and project organization. The learning performances are scattered across different sources (code text, runtime logs, interactive question-and-answer records, and reflective texts). One summative test cannot fully reflect the formation process of programming competency. Semantic analysis, task generation, feedback draft generation, and process modeling by Generative AI provide new cognitive collaborators and evidence generators for diversified teaching evaluation. Reforming evaluation should address problems including a single evaluation subject, weak process evidence, delayed feedback adjustment, and lack of interpretability of judgments; exploring the structural relationship among Generative AI and course objectives, evaluation dimensions, and teaching decisions. Significance of this study lies in establishing a logical framework, system architecture, and operational mechanism of Generative AI-empowered diversified evaluation on the Python course, clarifying roles' allocation, evidence governance, and boundary regulation in human-AI collaboration, providing theoretical foundation and methodology references for improvement of evaluation validity, reliability, and the efficiency of teaching transformation. Necessary due to the complexity of programming competency evaluation, risk of Generative AI application, and an urgent need of a diversified evaluation system with interpretable, traceable, and adjustable mechanisms.

## 1. Logical Framework for Generative AI-Empowered Diversified Teaching Evaluation in the Python Programming Course

### *1.1 Alignment Between the Technical Characteristics of Generative AI and Teaching Evaluation Functions*

The generative AI built upon large-scale pre-trained models has technical characteristics in natural language understanding and generation, code semantic modeling, contextual reasoning, multi-turn interaction, and multimodal representation that correspond to the functions of teaching evaluation including generating tasks, collecting evidence, interpreting performances, providing feedbacks, and giving adjustment recommendations. The functions of teaching evaluations transcend summative measurements by diagnosing learning states, identifying cognitive biases, tracking programming processes, and personalizing feedbacks. Generative AI generates different tasks, test cases, error explanation, and refactoring suggestion; performs semantic analysis for code texts, runtime logs, natural language reflection, and interaction record during Python learning; and connects diagnostic evaluation, formative evaluation, and summative evaluation continuously.

Technical characteristics do not align with evaluation functions by default – problems like model hallucination, lack of interpretability, data bias, and verification of generated content can impact judgment on evaluation reliability. Teachers set evaluation goals and valid boundaries; Model performs semantic analysis, evidence organization, and generate feedback drafts; Learners perform self-evaluations and peer evaluations (a human-AI collaboration). Technical capabilities of Generative AI becomes evaluation functions that constrain the scope of technical application — this is the first layer foundation of our logical framework.

### *1.2 Mapping Between Competency Objectives and Evaluation Dimensions in the Python Programming Course*

Competency objectives in Python Programming course are: understanding of syntax and semantics; use of control flow; choice of data structures; function and modular design; exception handling; file operations; object-oriented programming; using common libraries; designing algorithms; debugging and testing; code conventions; documenting expressions, etc.; Evaluation dimensions should be: knowledge understanding, code correctness, algorithm efficiency, structural clarity, debugging strategies, testing awareness, convention habits, problem solving, transfer and innovation, reflection and collaboration. No correspondence, but there is a mapping with observable evidence. The code texts reflect syntax and structure, the runtime results reflect correctness and efficiency, the debugging records reflect strategies and metacognition, while the reflective texts reflect awareness of transfer and collaboration.

Mapping: Uses Generative AI for syntactic/semantic parsing of code, identifies themes in natural language reflections, correlates runtime logs and test results, models trajectories of interaction processes. Changes from individual scores to evidence chains; from point-of-view judgments to process explanation. Designing a mapping needs to maintain dimensions as interpretable, traceable, and reviewable – without replacing real programming performance through generated text nor equating model output with competency judgment. Offers objective on course level for diversified evaluation.

### *1.3 Generative Empowerment Mechanisms and Structural Elements of Diversified Teaching Evaluation*

Diversified teaching evaluation refers to diversification in evaluation subject, evaluation task, evaluation evidence, evaluation method, and feedback approach. Empowerment mechanisms of Generative AI: Evaluation task generation mechanism, Process evidence semanticization mechanism, Feedback generation and adjustment mechanism, Human-AI collaborative judgment mechanism, and Evaluation result interpretation mechanism. The task generation mechanism expands evaluation context and difficulty; Semanticization mechanism transforms code, logs, dialogues, and reflections into analyzable evidence; feedback mechanism provides targeted and timely information; human-AI collaboration mechanism retains the judgment rights for teachers and learners; interpretation mechanism ensures that evaluation results are understandable and traceable.

Objective layer: The competency orientation of Python course; Task layer: Diversified evaluation activities are designed; Evidence layer: Multi-source learning performance is collected; Judgment layer:

Human-AI collaboration evaluation is conducted; Feedback layer: Improvement information is produced; Adjustment layer: Teaching decisions and learning adjustments can be made; Regulation layer: Problems such as data privacy, verification of generated content and academic integrity are dealt with. By embedding Generative AI in it, evaluation evolves from static measurement to dynamic interpretation, from single point judgment by teachers to human-AI collaborative judgment, from notifying results to adjusting processes. Generative AI becomes a cognitive collaborator and evidence generator inside the evaluation ecosystem rather than just an automatic scorer. It provides a logical structure which acts as the foundation for further system construction and operational optimization.

## 2. System Construction of Generative AI-Supported Diversified Teaching Evaluation for the Python Programming Course

### *2.1 Design of Multi-Dimensional Evaluation Indicators Oriented Toward Progressive Programming Competency*

The process of designing indicators for progressive programming competency requires identification of observable, traceable, and reviewable competency constructs based on the objectives of the Python Programming course. Language foundations, control flow, data structures, function abstraction, exception handling, file and module operations, object-oriented programming, applying commonly used libraries, algorithm design, debugging and testing, code conventions and documentation expression forms the basic facets in the competency structure. Conceptual understanding, code correctness, algorithm efficiency, structural clarity, debugging strategies, testing awareness, convention habits, problem solving, transfer design, and reflection and collaboration are the evaluation dimensions. Designing indicator is not only about listing out knowledge points but it corresponds with a matrix mapping competency progression levels and evaluation dimensions such that recognition and reproduction, understanding and explanation, transfer and synthesis, and design and innovation makes up a continuous gradient.

Operational level: Each indicator has a set of performance standard, evidence source, judgment rule and weight relationship. By using generative AI for creating indicator description, task example, draft scoring rubric and variant evaluation context, teachers will calibrate them based upon course objective and discipline logic. Weights are dynamic depending on learning stage, type of task and state of competency progression. There could be multiple evaluation subject such as teacher, peer, learner himself/herself or intelligent systems; distinguish the evidence authority and interpretive responsibility between different subjects when judging an indicator. Validity of this indicator system depends on corresponding competency construct and Python programming performance while reliability is dependent on reviewable scoring rules and cross-verification of multi-source evidence. Intervention from Generative AI extends the semantic space of indicators and coverage of tasks but we must avoid interference due to model bias, generation hallucination and indicator drift to evaluation constructs.

### *2.2 Generative AI-Driven Process Evaluation Data Generation and Semantic Analysis*

The Process evaluation data comes from different sources (code editing trajectories, runtime logs, exception stack information, test case results, version differences, interactive question-and-answer records, annotation documents, reflective texts). Through generative AI which generates various tasks, test cases, error explanations, code summaries, and semantic tags, we can convert the previously fixed questions in evaluation tasks into contextualized, hierarchical, and adjustable tasks. The data generation needs to establish the mapping for course competency indicators without detaching generated contents with evaluation goals. After multi-source data collection, it goes through the semantic analysis phase: structural representation of code texts by abstract syntax trees, control flow, data flow, variable naming, complexity, style features; semantic interpretation of natural language reflections by topic extraction, cognitive level identification, and confusion localization; form process portraits by time modeling and association analysis of learning behavior sequences[1].

Importance of semantic analysis: Organize scattered learning traces into an evident chain to trace evaluation judgments to specific programming performance and cognitive processes. In this process, Generative AI carries out a semantic bridging task which converts implicit information in code, logs, dialogues and reflections into evidence tags corresponding to indicators; Results can not be directly concluded as competencies, they require cross-validation, confidence assessment and teacher review. Authorization, de-identification, minimized collection and access boundaries must be considered if we

consider data governance. Traceability and appealability of evaluation results are the basic condition for operating the system. The process evaluation shifts from result verification to process interpretation, from single point measurement to continuous tracking, from teacher's experience based judgment to human-AI collaborative evidence reasoning.

### *2.3 Design of Human-AI Collaborative Intelligent Evaluation Feedback and Dynamic Adjustment*

Human-AI collaborative intelligent evaluation feedback consists of three levels: task level, strategy level, and metacognitive level. Task-level feedback covers code errors, syntax and semantics, test pass status, and functional implementation degree; Strategy-level feedback covers debugging paths, algorithm selection, structural organization, and test design; Metacognitive-level feedback covers learning plans, depth of reflection, awareness of transfer, and collaboration methods. Generative AI produces draft feedbacks for error explanation, refactoring suggestion, test case, and learning resource which is reviewed, revised, supplemented by teachers according to their teaching objective, learner state, and course norm. Learner participates in evaluation judgment through self-evaluation, peer evaluation, and response to feedback forming a feedback structure of multi-subject negotiation.

Dynamic adjustment: Task difficulty, scaffolding density, prompt intensity, collaborative grouping, resource pushing and teaching pace are adjusted according to evaluation evidence. The semantic interpretation happens when receiving evaluation data into the feedback generation stage; Feedback results affect teaching decisions and learning adjustments; Adjusted performance will be reevaluated in the evaluation cycle as a closed-loop evaluation-feedback-adjustment-re-evaluation process. In this closed loop, Generative AI plays the role of state identification, suggestion generation, and resource matching while teachers play value judgments, boundary regulation, and ultimate decision making responsibilities. Quality assurance should focus on the interpretability, reviewability, and appealability of feedback and boundary regulation, not replacing with model output of teacher judgment, prevent feedback information to generate cognitive load, and prevent out-of-bounds use of evaluation data. Design of human-AI collaboration feedback and adjustment enables the diversified evaluation system to have structure capacity for continuous operation and dynamic optimization providing an interpretable, traceable, and adjustable support path for the evaluation of the Python Programming course[2].

## 3. Operational Mechanisms and Optimization Strategies of Generative AI-Driven Diversified Teaching Evaluation in the Python Programming Course

### *3.1 Role Reconfiguration and Task Allocation When Generative AI Is Embedded in the Evaluation Process*

With the integration of Generative AI into evaluation, teachers transform themselves into evaluation designers, evidence interpreters, and judgment arbitrators that calibrate course objectives, approve evaluation indicators, design task context, review model output, and make final judgments. Generative AI performs computable work like evaluation task generation, code semantic parsing, process data annotation, error pattern recognition, feedback draft generation, and resource matching. Learners change role from evaluation object to participate in evaluation by engaging in self-evaluation, peer evaluation, feedback response, and supplementing evidence. Peer evaluation offers social references and collaboration. The reconfiguration of roles changes the evaluation subject from single center structure to multi-subject network where the model acts as cognitive collaborator and evidence generator without assuming value judgments and final adjudication.

Separate computable tasks from value judgment tasks: Model can do semantic feature extraction of large-scale texts/code, similarity comparison, anomaly detection, test case generation, feedback draft generation. Teachers can do objective appropriateness, educational significance, academic integrity, evaluation fairness, high-stakes judgments. Task allocation should have an authority-responsibility matching mechanism; Marking the confidence level, evidence sources, and applicability boundaries of model output, keep traces on teacher review record and learner appeal pathway. Clearer task boundary in evaluation process leads to more stable operations of human-AI collaboration and more traceable judgment quality of diversified evaluation.

### *3.2 Intelligent Interpretation of Evaluation Results and Transformation into Python Teaching Decisions*

Intelligent interpretation of evaluation results refers to providing explainable, traceable, and reviewable explanation from model output, semantic tag, indicator score, and process portrait. With generative AI connecting code quality feature, debugging path, test coverage, conceptual understanding, depth of reflection, collaboration performance with competency indicators of the Python course leads to diagnostic interpretations towards learning state and teaching state. The explanatory information includes evidence source, judgment base, confidence level, and improvement direction, without a single score hiding the formation process of programming competency. Intelligent interpretation makes evaluation data become useful knowledge for teaching: help teachers detect grammar difficulty, weakness in algorithm design, lack of debugging strategy, and project organization problem[3].

The translation of evaluation results to teaching decisions includes linking pathways between evaluation interpretation and teaching actions. Upon reaching a decision level, evaluation results can affect adjustments in task difficulty, arrangements of teaching pace, densities of scaffolding, groupings for collaboration, pushing resources, individualized guidance or iterations of course content. Generative AI produces decision recommendations, teaching resources, and intervention plans based on evaluation evidence that are selected by teachers with revisions considering course objectives, learner states, and classroom conditions. The effect of decision transformation will be followed-up through subsequent evaluation evidence as cyclic processes of evaluation, interpretation, decision-making, adjustment, and re-evaluation. Transforming Python teaching decisions from experiences driven to evidence supported yet under teachers' control of teaching directions and value judgments.

### *3.3 Quality Assurance and Boundary Regulation for Generative AI Evaluation Applications*

Quality assurance should include such as validity, reliability, fairness, interpretability, traceability, and security. Validity refers to consistency between evaluation indicators and the construct of Python programming competency; Reliability refers to the reviewability of judgment rules and cross-verification of multi-source evidence; Fairness refers to representational bias towards different learners in model output; Interpretability is whether the conclusion can be traced back to a particular piece of evidence; Traceability is retention of model version, prompt record, review opinion, and revision trace; Security includes data authorization, de-identification, minimize collection, and access control. The quality assurance for Generative AI evaluation applications extends not only to model performance indicators but also to evidence governance and judgment responsibility through the whole evaluation process[4].

Boundary Regulation: The function boundaries and responsibility of Generative AI on evaluation application should be clear; model hallucination, data bias, semantic misjudgment and over-generation of feedback can affect evaluation conclusion. Model output will never be the final result in high-stakes evaluation, nor replacing the teacher's value judgment or learner's subjective expression. It shall not collect data irrelevant to evaluation goal. Generated content needs human review, multi-source verification and confidence assessment. Evaluation results need appeal and correction mechanisms. Optimization strategy includes dynamically calibrating evaluation indicators, multi-source cross-validation, human-AI negotiated scoring, feedback quality assessment, audit model update, improve evaluation literacy from multiple subjects. With quality assurance and boundary regulation, Generative AI driven diversified evaluation has an interpretable, traceable and adjustable structure that provides a continuously optimized support mechanism for the evaluation of the Python Programming course.

## Conclusion

This paper focuses on Generative AI-empowered diversified teaching evaluation in the Python Programming course, building it's structure through three levels: logical framework, system construction, and operational optimization. Clarify that technical characteristics match evaluation functions, competency objectives map to evaluation dimensions, and empowerment mechanisms match structural elements. Formulate an evaluation system from perspectives such as indicator design, semantic analysis of process data, and human-AI collaboration feedback and dynamic adjustment, discuss role reconfiguration, task allocation, result interpretation, transformation into teaching decisions, quality assurance, and boundary regulation. In our future work, we will further deepen the

human-AI collaborative judgment mechanism, improve the verification of generated content and multi-source cross-verification capability, strengthen the evaluation data governance and appeal pathway, extend the support function of Generative AI in programming process modeling, personalized feedback, and course iteration, continuously optimize the dynamic calibration of evaluation indicators and feedback quality assessment, make diversified teaching evaluation operate under a structure that is interpretable, traceable, and adjustable, provide support for continuous improvement in the evaluation of the Python Programming course.

**Fund Projects**

Project supported by the Education Department of Hainan Province, (project number: Hnjg2025ZC-97)

**References**

*[1] Xiong Yu, et al. "Generative Educational Evaluation: Innovation and Practice of a Human-AI Collaborative Evaluation Paradigm Driven by Hybrid Augmented Intelligence." Journal of Distance Education 44.3 (2026): 72-82.*
*[2] Jiang Huifang, and Zeng Wenjie. "Generative Artificial Intelligence Promotes the Transformation of Educational Evaluation." Chinese Journal of Education. 8 (2025): 41-48.*
*[3] Yang Yue, et al. "Teaching Reform of the 'Python Programming' Course Based on 'One Model, Three Levels, and Multiple Evaluations'." Wireless Internet Technology 22.8 (2025): 123-128.*
*[4] Lu Guoqing, Yang Qin, and He Xiangchun. "The Value, Challenges, and Pathways of Generative Artificial Intelligence Empowering Formative Evaluation in Higher Education." e-Education Research 45.11 (2024): 84-91.*