Received: 19 Jul. 2026 | Revised: 20 Jul. 2026 | Accepted: 23 Jul. 2026 | Published online: 26 Oct. 2026
RESEARCH ARTICLE

Contemporary Education and Teaching Research
2026,Vol. 7(10)330-347
DOI: 10.61360/BoniCETR262020491003

From Perceived Attributes to Reported Use: Generative AI Adoption for Programming-Related Learning among Taiwanese Undergraduates in Information-Related Programs

Tien-Chi Huang[1], Pei-Shan Wu[1], Yi-Wen Huang[1], Hsin-Ping Tseng[2] & Yu-Hung Chiang[3,*]

[1]Department of Information Management, National Taichung University of Science and Technology

[2]Doctoral Program of Intelligent Engineering, National Taichung University of Science and Technology

[3]Department of Asia-Pacific Industrial and Business Management, National University of Kaohsiung

Abstract: Generative AI is increasingly used in programming-related learning, but favorable perceptions do not necessarily correspond to reported use or better learning. Drawing on Diffusion of Innovations theory and psychological boundary conditions, this cross-sectional survey examined associations among five perceived innovation attributes, usage intention, and self-reported generative AI-assisted programming activity among 1,034 Taiwanese undergraduates in information-related programs who reported prior generative AI experience. The reported-use construct referred to three programming-specific activities during the preceding four weeks and did not assess code quality, conceptual understanding, transfer, or independent programming ability. Partial least squares structural equation modeling was complemented with Explainable Boosting Machine (EBM) regression. Trialability, relative advantage, compatibility, and observability were positively associated with intention; low complexity was not significant in the linear model. Intention was positively associated with reported use, and this association was stronger at higher self-directed learning levels and weaker at higher cognitive load. EBM described model-specific nonlinear predictive patterns. Because the measures were cross-sectional and self-reported, the results characterize adoption behavior rather than learning effects.

Keywords: generative artificial intelligence, programming-related learning, diffusion of innovations, self-directed learning, cognitive load

1. Introduction

Generative artificial intelligence (generative AI) has rapidly entered higher education and is especially consequential for programming-related learning. Tools such as ChatGPT, GitHub Copilot, and Claude can generate code, diagnose errors, explain programming logic, and respond to follow-up questions (Samala et al., 2025). These affordances may support code comprehension and iterative problem solving, but assisted task completion is not equivalent to learning. Generated materials can also be inaccurate: Sarsa et al. (2022) found that only 67.2% of evaluated line-level code explanations were correct. In a laboratory study of novice programmers, generative AI compounded metacognitive difficulties and produced an illusion of competence among some struggling learners (Prather et al., 2024). Students likewise report overreliance and difficulty verifying generated code (Alpizar-Chacon & Keuning, 2025; Klingbeil et al., 2024). Because learning to program requires learners to interpret, test, and revise code rather than merely obtain an executable answer, adoption behavior must be distinguished from learning outcomes.

Research on educational adoption of generative AI has mainly used the Technology Acceptance Model and the Unified Theory of Acceptance and Use of Technology (Strzelecki & ElArabawy, 2024; Chiu, 2025). These frameworks explain attitudes and intentions through constructs such as perceived usefulness and ease of use. Rogers' (2003) Diffusion of Innovations (DOI) theory provides a complementary account through five perceived attributes: relative advantage, complexity, compatibility, trialability, and observability. Early studies suggest that these attributes help explain generative AI adoption in education (Alamri, 2025; Raman et al., 2024), but adoption evidence concerns

Corresponding Author: Yu-Hung Chiang
Department of Asia-Pacific Industrial and Business Management, National University of Kaohsiung

©The Author(s) 2026. Published by BONI FUTURE DIGITAL PUBLISHING CO.,LIMITED. This is an open access article under the CC BY License(https://creativecommons.org/licenses/by/4.0/)

perceptions, intentions, or use and should not be treated as evidence of learning.

Evidence on learning is more heterogeneous and should be separated by domain. In a preregistered field experiment in secondary mathematics, unguarded GPT-4 improved assisted-practice performance but reduced subsequent unassisted performance, whereas a tutor with learning-oriented guardrails largely removed this deficit (Bastani et al., 2025). That study identifies a general educational risk rather than a programming-specific effect. Computing-education findings are mixed: a controlled study found that an AI code generator improved novice code-authoring performance without a significant reduction in manual code-modification performance or one-week retention (Kazemitabaar et al., 2023), whereas Prather et al. (2024) documented metacognitive harms among some struggling novices. A classroom deployment of a guarded programming assistant further showed that tool design can constrain direct-answer behavior, although inaccurate responses and overreliance still occurred and learning outcomes were not established (Kazemitabaar et al., 2024). This mixed literature supports a bounded study of adoption rather than an assumption that more use produces better learning.

Adoption research also often stops at usage intention. Intention is a proximal correlate of use, but its translation can vary with personal and contextual conditions (Granić, 2025; Li, 2023). The present cross-sectional questionnaire cannot demonstrate a temporal intention–behavior gap. Instead, it examines whether contemporaneous intention is associated with a self-reported-use construct covering AI assistance with code writing or debugging, activation or integration of AI within an editor, and completion of programming tasks with AI assistance during the preceding four weeks. This construct does not represent platform logs, sustained behavior, or learning quality.

Self-directed learning (SDL) ability and cognitive load (CL) offer two psychological boundary conditions for the intention–reported-use association. Useful AI interaction requires learners to define goals, supply context, monitor responses, and integrate output into their own work (Lai, 2024). Students with stronger SDL may therefore be better able to enact an intention through strategic prompting, testing, and revision. Conversely, coordinating course concepts, prompts, output evaluation, and code integration can exceed limited working-memory capacity (Sweller, 1988; Huang et al., 2026), particularly for novices. Considering SDL and CL together tests whether the intention–reported-use association varies with reported regulatory and cognitive conditions.

A third methodological limitation is the dominance of regression and structural equation modeling, which estimate average linear relationships but may obscure saturation and other nonlinear patterns (Shahzad et al., 2025). Such patterns are plausible in programming education because additional perceived ease or visible peer success may not contribute uniformly across the full response range. This study therefore combines partial least squares structural equation modeling (PLS-SEM) with an Explainable Boosting Machine (EBM), an interpretable generalized additive model that estimates nonlinear main-effect shape functions (Lou et al., 2013; Nori et al., 2019). PLS-SEM tests the proposed theoretical paths, whereas EBM describes how predictors relate to behavior across their observed ranges.

Accordingly, this study develops a psychologically informed DOI model of reported generative AI adoption for programming-related learning among Taiwanese undergraduates in information-related programs with prior generative AI experience. It tests associations from five innovation attributes to usage intention, from intention to reported use, and the moderation of that association by SDL and CL. The complementary PLS-SEM and EBM analyses distinguish prespecified linear structural paths from model-specific nonlinear predictive patterns. The study does not evaluate instructional strategies or learning outcomes; its educational implications are

hypothesis-generating. The study addresses five questions:

RQ1. To what extent are the five DOI attributes associated with usage intention among Taiwanese undergraduates in information-related programs who report prior generative AI experience?

RQ2. Is usage intention associated with self-reported generative AI-assisted programming activity during the preceding four weeks?

RQ3. Does SDL moderate the association between usage intention and reported programming-related use?

RQ4. Does CL moderate the association between usage intention and reported programming-related use?

RQ5. What complementary nonlinear information does EBM provide about predictors of reported programming-related use?

2. Theoretical Framework and Hypothesis Development

DOI theory explains adoption through users' perceptions of an innovation's attributes (Rogers, 2003). In the proposed model, relative advantage, low complexity, compatibility, trialability, and observability predict intention to use generative AI for programming-related learning. Each represents a distinct appraisal of expected benefit, effort, fit, opportunity for experimentation, or visible results. Intention then predicts the reported-use construct, while SDL and CL are modeled as moderators. The framework separates intention formation from contemporaneous reported use; it does not model code correctness, conceptual understanding, transfer, or unaided ability. The following sections develop eight hypotheses.

2.1. Relative advantage and usage intention

Relative advantage is the degree to which an innovation is perceived as better than the practice it replaces (Rogers, 2003). In programming education, conventional resources such as documentation, worked examples, and instructor feedback remain important, but they may not provide immediate, context-sensitive responses during individual coding activity. Generative AI can offer rapid error diagnosis, code explanations, and iterative feedback tailored to the learner's current problem. Dialogue with large language models can also support critical thinking and problem solving by moving students from passive retrieval toward active knowledge construction (Samala et al., 2025). When learners believe that AI can help them understand a difficult concept, locate an error, or develop a solution more efficiently than available alternatives, they form stronger positive outcome expectations. Relative advantage thus links the perceived instructional utility of AI to motivation to adopt it.

H1: The perceived relative advantage of generative AI in programming education is positively associated with students' usage intention.

2.2. Low complexity and usage intention

Complexity concerns how difficult an innovation is to understand and use (Rogers, 2003). Following Moore and Benbasat (1991), the present study operationalizes the construct as low complexity, using positively worded items so that higher scores represent greater perceived ease of learning, mastery, and operation. Natural-language interaction may lower an initial operational barrier (Luera et al., 2024), but interface ease does not ensure accurate explanations or code (Sarsa et al., 2022). Learners must still specify problems, evaluate responses, and integrate suggestions into an existing program. When those activities appear manageable, intention may be stronger; when they appear demanding, intention may weaken.

H2: Perceived low complexity (i.e., perceived ease of use) of generative AI in programming education is positively associated with students' usage intention.

2.3. Compatibility and usage intention

Compatibility is the extent to which an innovation is consistent with an adopter's values, prior experience, and needs (Rogers, 2003). Computing students already learn through documentation searches, code examples, integrated development

environments, iterative testing, and peer discussion. Generative AI may be perceived as compatible when it complements these routines rather than requiring learners to replace them. Alignment with existing practices reduces psychological resistance and the regulatory cost of adoption because students can incorporate AI into familiar debugging and concept-learning workflows without extensively reorganizing how they study. Compatibility should therefore preserve attentional and self-regulatory resources and make adoption appear more feasible and instructionally relevant.

H3: Perceived compatibility of generative AI with students' existing programming-learning practices is positively associated with their usage intention.

2.4. Trialability and usage intention

Trialability refers to the extent to which an innovation can be tested on a limited basis before full commitment (Rogers, 2003). Trial opportunities reduce uncertainty and perceived risk during early adoption. In programming education, trialability entails more than accessing a platform. Students can submit a known problem, compare an AI explanation with course materials, test generated code, revise prompts, and evaluate whether the tool improves their understanding. These low-stakes cycles provide direct evidence of usefulness and limitations, enabling learners to calibrate trust before relying on AI in more complex assignments. Instructionally designed trials are especially important because they allow verification and reflection without making successful task completion dependent on unexamined output. When learners can experiment, observe consequences, and withdraw at little cost, the psychological threshold for forming a positive intention is lower.

H4: Trialability of generative AI in programming education is positively associated with students' usage intention.

2.5. Observability and usage intention

Observability is the degree to which the results of an innovation are visible to others (Rogers, 2003). Programming outcomes are comparatively concrete: code can be inspected, executed, and tested, and debugging progress can be demonstrated. When students see peers use generative AI to explain an error, develop code architecture, or improve a solution, the visible result reduces uncertainty about what the tool can accomplish. Observability also activates vicarious learning (Bandura, 1986). Successful peer demonstrations can strengthen outcome expectations and modeling-based self-efficacy, particularly when instructors make both the generated response and the verification process visible. Thus, observable learning processes and outcomes provide social evidence that can support intention formation.

H5: Observability of generative AI outcomes in programming education is positively associated with students' usage intention.

2.6. Usage intention and reported use

Behavioral intention is the learner's stated commitment to perform a behavior and is commonly treated as a proximal predictor of technology use (Venkatesh et al., 2003). The present reported-use construct covers three self-reported programming-specific activities during the preceding four weeks. A stronger intention is expected to be associated with more reported use, although contemporaneous self-reports cannot establish temporal enactment or sustained behavior.

H6: Usage intention is positively associated with self-reported generative AI-assisted programming activity during the preceding four weeks.

2.7. The moderating role of self-directed learning ability

SDL ability is the capacity to set learning goals, select strategies, monitor progress, and adapt action in response to feedback (Zimmerman, 2002). These processes are central to productive AI-assisted programming. Students must identify what they do not understand, formulate a useful prompt, evaluate the response against code behavior and course knowledge, and decide whether to revise the prompt or their solution. Lai (2024) argued that deeper chatbot engagement can support goal-directed

prompting and reflective knowledge construction. Across a 21-country sample of 1,073 postgraduates, self-regulated learning strategies also predicted progressively deeper forms of AI use, from technical support to text development and transformation (Jin et al., 2025). This evidence suggests that regulatory capacity determines not simply whether a learner accesses AI, but whether an intention becomes sustained and strategic activity.

Students with strong SDL can plan where AI fits within a programming task, monitor whether it advances understanding, and correct ineffective interaction. Those with weaker SDL may intend to use the tool but abandon it after poor outputs or accept answers without the verification needed for continued learning. SDL is therefore modeled as a positive boundary condition: it supplies the metacognitive resources required to carry an existing intention through the prompting, testing, and integration cycle. The conversational interface may reduce some planning demands, but it does not eliminate the need to regulate learning.

H7: SDL positively moderates the association between usage intention and reported programming-related use, such that the association is stronger at higher SDL levels.

2.8. The moderating role of cognitive load

Cognitive load theory holds that working memory has limited capacity and that performance deteriorates when concurrent demands exceed that capacity (Sweller, 1988; Sweller et al., 2019). AI-assisted programming can combine several demanding activities: understanding the programming concept, maintaining the state of the code, formulating prompts, interpreting responses, checking correctness, and integrating suggestions. Huang et al. (2026) found that intrinsic cognitive load reduced university students' ChatGPT adoption intentions. Stadler et al. (2024) further showed that large-language-model information seeking can lower subjective cognitive load while producing shallower scientific reasoning, indicating that perceived ease and productive cognitive engagement are not equivalent.

An AI response may appear to simplify a task while shifting verification and integration demands onto the learner. When these demands compete with resources needed for code comprehension, even students with strong intentions may disengage, rely superficially on output, or avoid the tool. CL is therefore modeled as a negative moderator of the intention–reported-use association.

H8: CL negatively moderates the association between usage intention and reported programming-related use, such that the association is weaker at higher CL levels.

Together, H1–H5 specify the DOI antecedents of usage intention, H6 links intention to reported use, and H7–H8 specify regulatory and cognitive moderation. Figure 1 presents the research model.

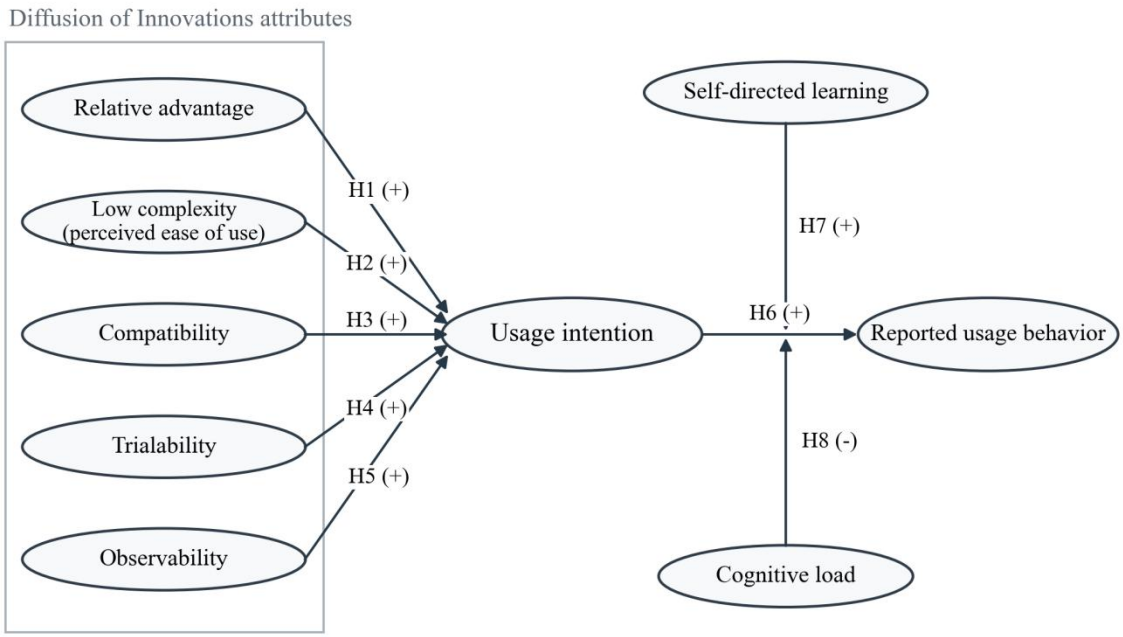

Figure 1. *Proposed research model*

3. Methodology

3.1. Research design and instrument

This study used a cross-sectional online survey to examine perceptions, intention, and reported generative AI-assisted programming activity. The questionnaire comprised demographic and background questions, a descriptive checklist of generative AI tools and broad use scenarios during the preceding three months, and 33 Likert items representing nine constructs: relative advantage, low complexity, compatibility, trialability, observability, usage intention, SDL, CL, and reported use. Construct items used a five-point scale from 1 (strongly disagree) to 5 (strongly agree).

The items were adapted to generative AI-assisted programming while retaining the meanings of their source constructs. Relative advantage and compatibility were adapted from Alamri (2025); low complexity, trialability, observability, and usage intention from Raman et al. (2024); CL from Chen et al. (2025); and reported use from Haque and Hundhausen (2025). The three reported-use items referred specifically to the preceding four weeks: using generative AI to assist with writing or debugging code, activating or integrating AI within an IDE or editor, and completing programming tasks with AI assistance. Higher low-complexity scores represented greater perceived ease. Supplementary Table S1 reports the English wording and sources. Participants completed a Chinese questionnaire; the English wording is supplied for reporting and is not a separately validated English-language scale.

A pilot study with 53 valid responses examined item clarity and internal consistency before the main survey. Across the nine constructs, composite reliability (CR) ranged from .797 to .868. Cronbach's alpha was .629 for observability and .660 for the reported-use construct, slightly below .70, whereas the corresponding CR values exceeded .70 (Nunnally, 1978). On the basis of the reported pilot statistics, the instrument was retained for the main study. These pilot findings were treated as preliminary evidence; reliability and validity were reassessed using the main sample.

3.2. Participants, ethics, and data collection

Eligible retained records were from consenting undergraduates in Taiwan who selected an information-related program category and answered yes to a screening question asking whether they had ever used any generative AI tool. Program categories

were information engineering, information management, information science, information communication/digital media, electrical/computer engineering, and other IT-related programs. Current enrollment in a programming course was not required or recorded. Programming background therefore varied; 11 respondents (1.06%) reported zero programming credits completed or in progress. The analytic population is consequently undergraduates in information-related programs with prior generative AI experience, not all computing students or students currently enrolled in programming courses. The study was approved by the Human Research Ethics Committee of National Cheng Kung University on October 6, 2025. Participation was voluntary, and informed consent was obtained before questionnaire completion.

Data were collected from November 25 to December 6, 2025, using a self-administered online questionnaire hosted on SurveyCake and non-probability convenience sampling. The survey link was broadly disseminated nationwide, participation was voluntary, and recruitment was not targeted to a single institution. Institution names, department names, and recruitment-source identifiers were not collected. The questionnaire recorded six broad major categories, reported in Table 1, but these categories do not identify individual departments. Consequently, the numbers of institutions and departments represented, institution-specific shares, and institutional clustering cannot be assessed. In total, 1,087 responses were received. Response-quality screening used two criteria: implausibly short completion times and abnormal near-uniform response patterns, indicated by selecting the same response option for almost all construct items. Fifty-three responses meeting at least one of these criteria were excluded, leaving 1,034 valid responses for analysis. The available records do not preserve the number excluded under each criterion, the completion-time cutoff, or the exact operational threshold for a near-uniform response pattern. All retained platform hash values were unique, but the hash-generation rule and any platform controls used to prevent repeat participation were not documented; unique hashes therefore do not establish person-level uniqueness.

3.3. Data analysis

Analysis proceeded in three stages. First, frequencies and percentages described the retained sample and the separate three-month checklist of generative AI tools and broad use scenarios. Second, the measurement and structural models were estimated in SmartPLS 4 using partial least squares structural equation modeling (PLS-SEM; Hair et al., 2021; Ringle et al., 2022). The nine constructs were modeled reflectively. Reliability was examined with Cronbach's alpha and composite reliability; convergent validity with standardized loadings and average variance extracted; and discriminant validity with the Fornell–Larcker criterion and heterotrait–monotrait ratio. Harman's single-factor result and full-collinearity variance inflation factors were reported as diagnostic checks for common-method bias (Kock, 2015). Structural-path significance was evaluated with 5,000 bootstrap resamples. SDL and CL were tested as moderators of the intention–reported-use association.

The structural analysis reported standardized path coefficients, bootstrap p values, hypothesis decisions, and coefficients of determination (R^2) for usage intention and reported use. Moderation coefficients were interpreted as conditional associations within the fitted cross-sectional model.

Third, an Explainable Boosting Machine (EBM) regression supplemented the prespecified PLS-SEM analysis. Reported use was the target, and the five DOI attributes, SDL, CL, and usage intention were predictors. Five-fold cross-validation evaluated root mean squared error (RMSE). Main-effect importance scores and univariate shape functions described model-specific predictive contributions and functional form; no EBM interactions are reported. Because PLS-SEM predicted intention from DOI attributes whereas EBM predicted reported use conditional on all included features, the two analyses

address different outcomes and are interpreted separately.

4. Results

4.1. Respondent characteristics and Generative AI experience

Table 1 summarizes the 1,034 retained respondents, all of whom consented, identified as undergraduates in an information-related program category, and answered yes to prior generative AI use. Most were aged 20–23 years (92.55%), gender was nearly balanced, and juniors formed the largest academic-year group. Information Management was the most common program category, the 3–4-credit category was the most common programming background, and 50.00% rated their programming proficiency as average. Eleven respondents (1.06%) reported zero programming credits completed or in progress.

Table 1. *Demographic characteristics of respondents (N = 1,034)*

Characteristic	Category	n	%
Age	19 or younger	77	7.45
	20–21	413	39.94
	22–23	544	52.61
Gender	Male	519	50.19
	Female	515	49.81
Academic year	Freshman	92	8.90
	Sophomore	219	21.18
	Junior	395	38.20
	Senior or extended-year	328	31.72
Major	Information Management	365	35.30
	Information Science	220	21.28
	Information Engineering	178	17.21
	Information Communication / Digital Media	131	12.67
	Electrical / Computer Engineering	74	7.16
	Other IT-related	66	6.38
Programming credits completed	0	11	1.06
	1–2	261	25.24
	3–4	459	44.39
	5–6	199	19.25
	7 or more	104	10.06
Self-rated programming proficiency (reported categories)	1 (very low)	35	3.38
	2 (below average)	184	17.79
	3 (average)	517	50.00
	4 (above average)	298	28.82

Table 2 reports a descriptive checklist of tools and broad scenarios selected for the preceding three months. This checklist was separate from the programming-specific four-week reported-use construct. ChatGPT was selected most frequently (72.73%), followed by Gemini (47.78%) and DeepSeek (21.37%). Reading documentation or summarizing material was the most frequently

selected scenario (45.36%), but that activity is not necessarily a programming activity. The checklist was not an eligibility test and should not be interpreted as evidence that every reported generative AI use was programming-related. Multiple selections were permitted, so percentages do not sum to 100%.

Table 2. *Descriptive generative AI tools and broad use scenarios (N = 1,034; multiple responses permitted)*

Type	Response	n	%
Tool	ChatGPT	752	72.73
	Google Gemini	494	47.78
	DeepSeek	221	21.37
	GitHub Copilot	175	16.92
	Anthropic Claude	156	15.09
	xAI Grok	88	8.51
	Cursor IDE	47	4.55
	Codeium	31	3.00
	Other	4	0.39
Scenario	Reading documentation / summarization	469	45.36
	Debugging / error explanation	376	36.36
	Refactoring or adjusting code	364	35.20
	Generating example code	363	35.11
	Writing tests / generating test data	307	29.69
	Problem solving / concept clarification	289	27.95
	Translating / reading English technical documents	242	23.40
	Requesting practice problems / solutions	158	15.28

Note. Multiple responses were permitted; percentages therefore do not sum to 100%.

4.2. Measurement model

The reported measurement-model diagnostics are summarized in Table 3. Cronbach's alpha ranged from .640 to .810. The minimum value of .640 is below the conventional .70 guideline; consequently, the evidence does not support a claim that every construct exceeded .70 on alpha. CR ranged from .807 to .873, with all reported values above .70. Considered together, the ranges indicate acceptable composite reliability but some weakness in alpha-based internal consistency for at least one construct.

AVE values ranged from .529 to .697, and standardized indicator loadings ranged from .675 to .931. Thus, the reported AVE values exceeded .50 and the loadings exceeded .60, supporting convergent validity under the stated criteria. For the Fornell–Larcker assessment, square roots of AVE ranged from .727 to .835 and were reported to exceed the corresponding interconstruct correlations. All HTMT ratios were also reported to fall below the recommended criterion.

Two diagnostics addressed the possibility of common-method bias. The first unrotated factor in Harman's test explained 34.27% of the variance, below the stated 50% comparison value. Full-collinearity VIFs ranged from 1.68 to 2.64, below the stated 3.3 comparison value. These diagnostics did not indicate severe common-method variance under the reported rules of thumb, but they do not eliminate the possibility of bias inherent in a single-source, self-report survey.

Table 3. *Summary of reported measurement-model evidence*

Assessment	Reported result	Criterion	Interpretation
Cronbach's alpha	.640–.810	.70	Minimum below criterion; not all constructs met .70
Composite reliability	.807–.873	.70	All reported values above criterion
Average variance extracted	.529–.697	.50	All reported values above criterion
Standardized loadings	.675–.931	.60	All reported loadings above criterion
Square root of AVE	.727–.835	> interconstruct correlations	Fornell–Larcker criterion reported as met
HTMT	All reported below criterion	Recommended criterion	Criterion reported as met
Harman first-factor variance	34.27%	50%	Below stated comparison value
Full-collinearity VIF	1.68–2.64	3.3	All reported values below criterion

Note. The table reports the retained range-level measurement diagnostics and their prespecified comparison criteria.

4.3. Structural model and hypothesis tests

The PLS-SEM structural model was evaluated using 5,000 bootstrap resamples. The five DOI attributes jointly accounted for 67.7% of the variance in usage intention (R^2 = .677), and the equation for reported use accounted for 34.5% (R^2 = .345). Figure 2 presents the structural estimates.

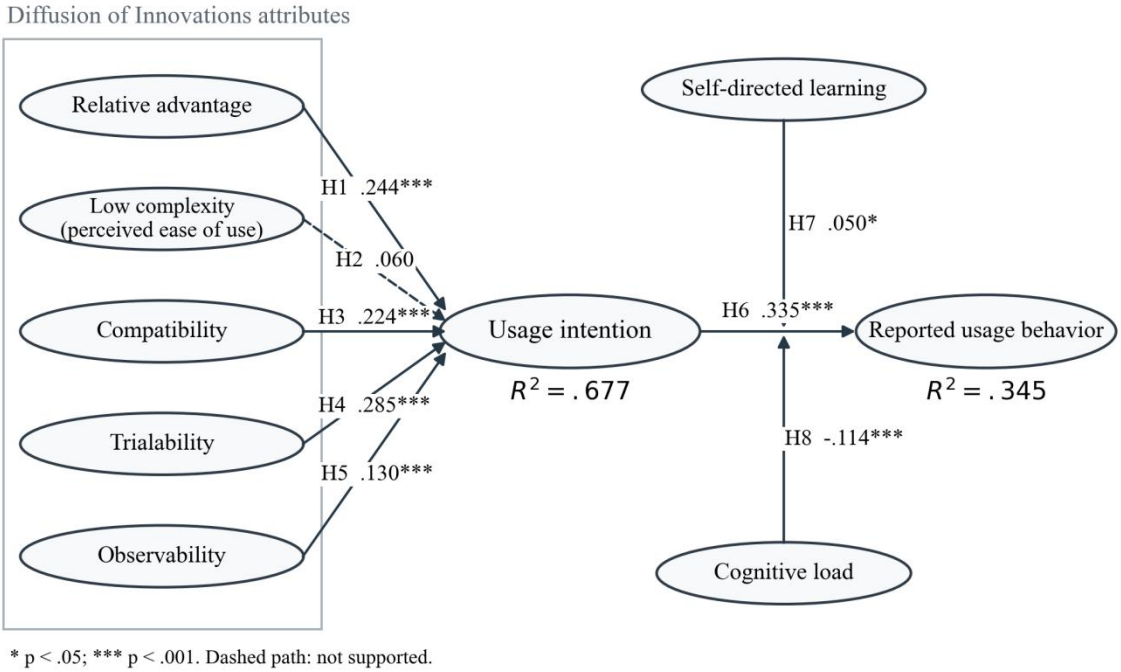

Figure 2. *Research model path-analysis results*

Four of the five hypothesized DOI paths to usage intention were statistically significant and positive. Trialability had the largest coefficient (β = .285, p < .001), followed by relative advantage (β = .244, p < .001), compatibility (β = .224, p < .001), and observability (β = .130, p < .001). These findings supported H1, H3, H4, and H5. The path from perceived low complexity—that is, greater perceived

ease of learning and operation—to usage intention was positive but not statistically significant (β = .060, p = .091); H2 was therefore not supported. Because low complexity was positively coded, this result should not be described as evidence that greater "complexity" failed to deter use.

Usage intention was positively associated with reported use (β = .335, p < .001), supporting H6. SDL positively moderated this association (β = .050, p = .038), supporting H7, whereas CL negatively moderated it (β = −.114, p < .001), supporting H8. Table 4 reports the hypothesis decisions.

Table 4. *Summary of hypothesis-testing results*

Hypothesis	Path	β	p	Decision
H1	Relative advantage → usage intention	.244	< .001	Supported
H2	Low complexity (perceived ease) → usage intention	.060	.091	Not supported
H3	Compatibility → usage intention	.224	< .001	Supported
H4	Trialability → usage intention	.285	< .001	Supported
H5	Observability → usage intention	.130	< .001	Supported
H6	Usage intention → reported use	.335	< .001	Supported
H7	SDL × intention → reported use	.050	.038	Supported
H8	CL × intention → reported use	−.114	< .001	Supported

Note. Arrows denote direct paths; multiplication signs denote moderation terms.

4.4. EBM regression results

The EBM regression predicted reported use from the eight specified features using five-fold cross-validation. Training RMSE ranged from .102 to .128 (M = .115, SD ≈ .008), and test RMSE ranged from .110 to .136 (M = .121, SD ≈ .009). These values describe performance across the reported folds; no external test set or comparison model was used.

Figure 3A reports model-specific main-effect importance on the training data. Low complexity had the highest score (.1085), followed by compatibility (.0854), observability (.0726), CL (.0628), relative advantage (.0623), trialability (.0418), usage intention (.0363), and SDL (.0120). Importance is predictive within this fitted model and does not indicate direction, causality, or statistical significance.

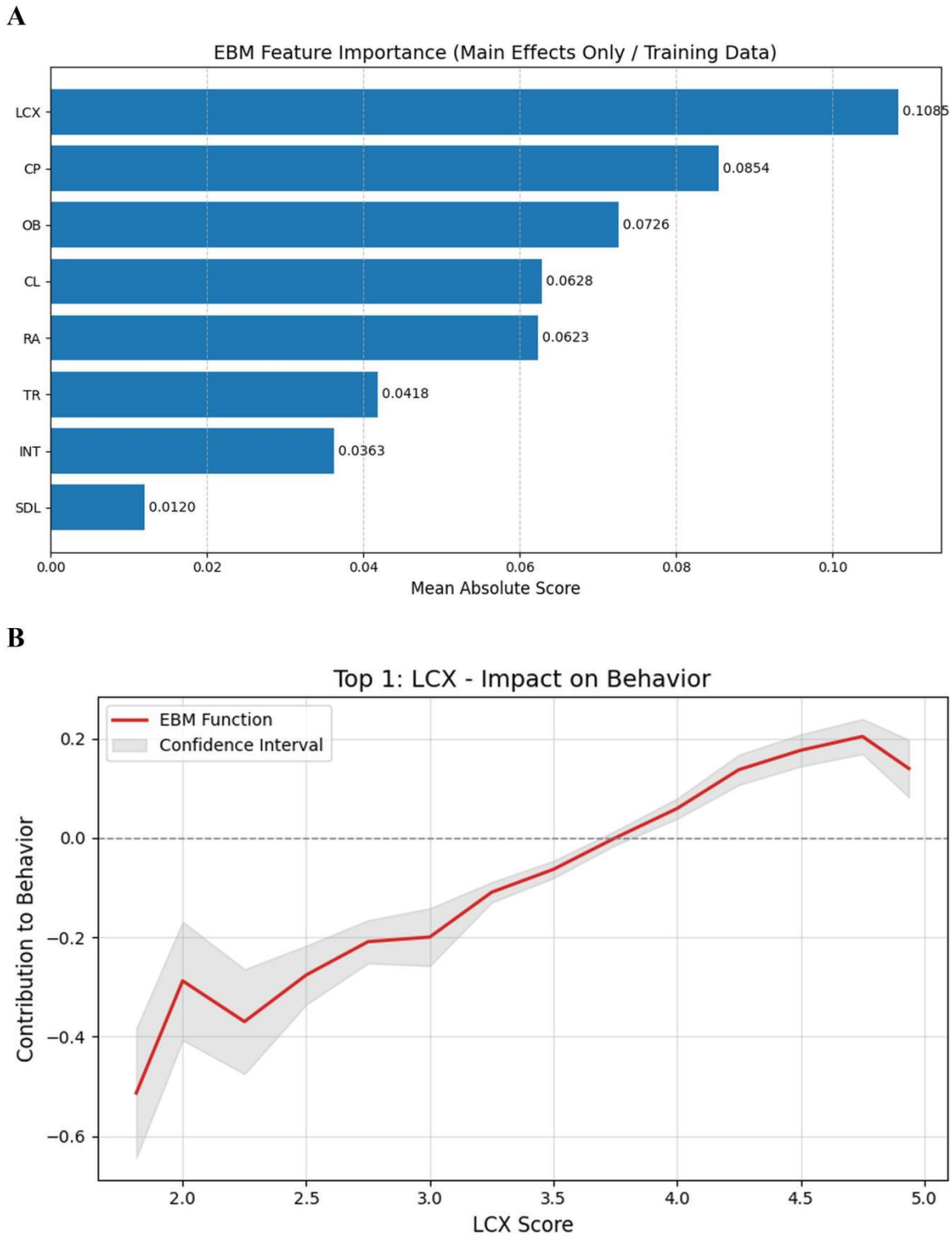

Figure 3. *EBM feature importance and low-complexity contribution function*

Note. (A) Main-effect feature importance on the training data. (B) Baseline-relative EBM contribution function for low complexity with its fitted confidence interval. The vertical axis is a model contribution, not a causal effect.

The low-complexity shape function (Figure 3B) increased across most of the observed range. Contributions rose from below the fitted baseline at low scores to above it at higher scores, with a small

decline at the extreme upper end.

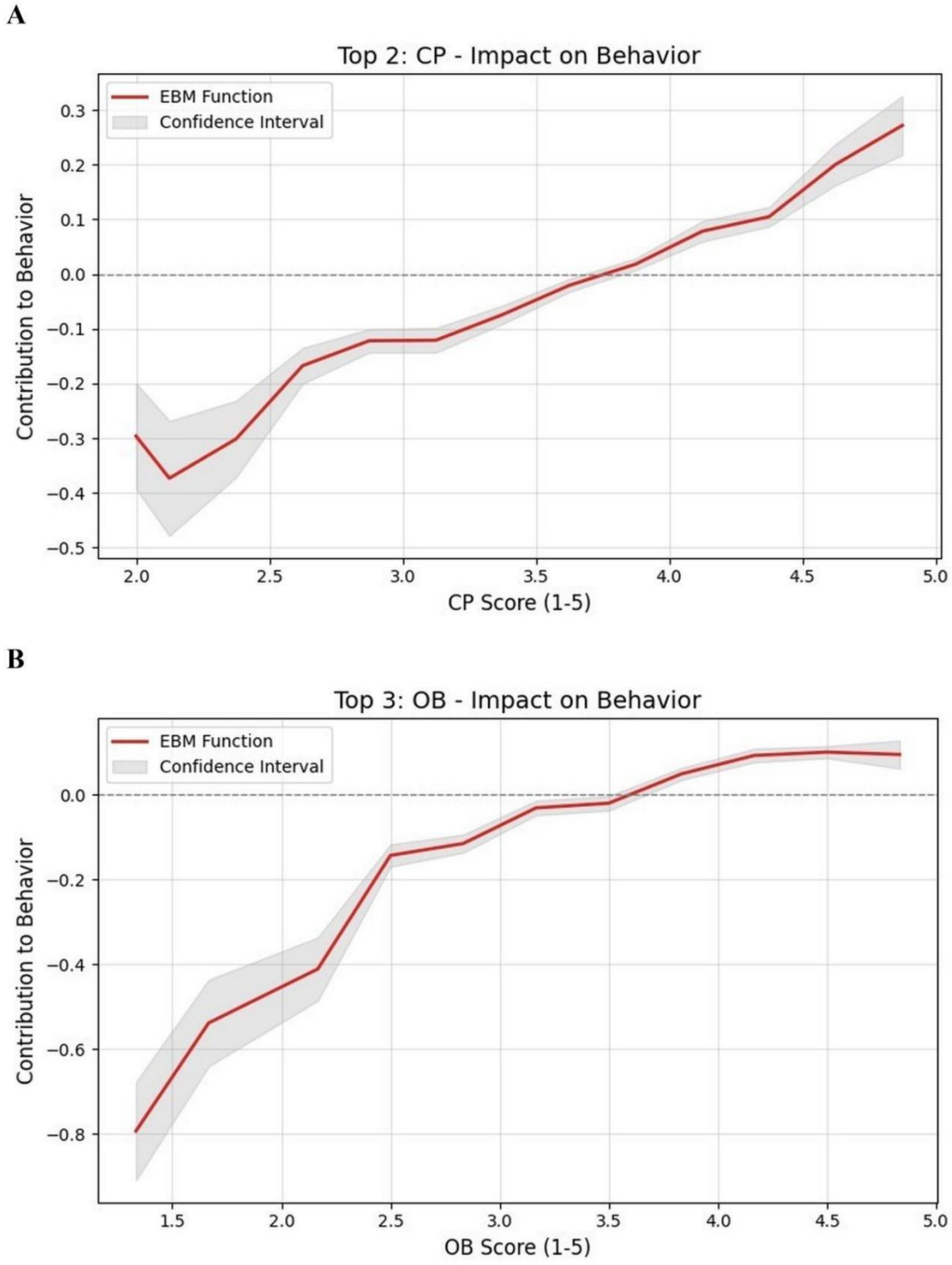

Figure 4. *EBM contribution functions for compatibility and observability*

Note. (A) Baseline-relative contribution function for compatibility. (B) Baseline-relative contribution function for observability. Confidence intervals are shown; the vertical axes represent model contributions, not causal effects.

Compatibility showed a similar upward pattern (Figure 4A), while observability rose through the lower and middle ranges and then leveled off (Figure 4B). These are baseline-relative contribution patterns, not experimentally verified thresholds.

The PLS-SEM and EBM results answer different questions: PLS-SEM estimated a linear path from low complexity to intention, whereas EBM estimated conditional nonlinear contributions to reported use. The EBM patterns neither retest H1–H8 nor validate causal mechanisms.

5. Discussion

This study examined associations among five DOI attributes, usage intention, and contemporaneous reported programming-related use, together with moderation by SDL and CL. Interpretation is limited to Taiwanese undergraduates in information-related programs with prior generative AI experience. The cross-sectional self-reports characterize adoption, not code quality or learning.

5.1. DOI attributes and usage intention (RQ1)

The five DOI attributes jointly explained 67.7% of the variance in usage intention. Trialability had the largest standardized coefficient (β = .285), followed by relative advantage (β = .244), compatibility (β = .224), and observability (β = .130); low complexity was not significant (β = .060, p = .091). The supported paths are consistent with DOI theory and educational adoption studies (Alamri, 2025; Raman et al., 2024), but their ordering is descriptive within this model.

Because DOI attributes concern perceived adoption conditions, these associations can motivate future tests of bounded trials and visible verification, but they do not establish that any instructional routine improves learning. The nonsignificant low-complexity path also cautions against equating a natural-language interface with easy or accurate programming use.

Programming-specific evidence supports that caution: generated explanations can contain errors (Sarsa et al., 2022), and some novice learners have difficulty evaluating suggestions (Prather et al., 2024). Future studies should therefore distinguish interface ease from the technical and metacognitive demands of evaluating output.

5.2. Usage intention and reported use (RQ2)

Usage intention was positively associated with reported use (β = .335, p < .001), and the model explained 34.5% of its variance. Because intention and use were measured in the same questionnaire, this result does not demonstrate a temporal intention–behavior gap. The outcome captured reported frequency or intensity of three programming-related activities, not logs, code quality, conceptual understanding, transfer, or independent programming ability. Stronger intention or reported use therefore cannot be interpreted as better learning.

The wider literature reinforces this distinction: AI assistance can improve performance during supported activity without guaranteeing subsequent unaided performance (Bastani et al., 2025), and computing-education results vary by learner and task conditions (Kazemitabaar et al., 2023; Prather et al., 2024). Verification routines are appropriate candidates for experimental study, not conclusions from the present survey.

5.3. The moderating role of SDL (RQ3)

The SDL-by-intention interaction was positive but small (β = .050, p = .038): the intention–reported-use association was somewhat stronger at higher SDL levels within the fitted model. Replication is needed, and the result does not establish that SDL training would increase effective use or improve learning.

The direction is consistent with goal setting, monitoring, strategy adjustment, and reflection (Lai, 2024; Jin et al., 2025). However, those studies and the present moderation result do not test specific programming scaffolds. Instructional applications should therefore be evaluated directly.

5.4. The moderating role of CL (RQ4)

The CL-by-intention interaction was negative (β = −.114, p < .001): the positive intention–reported-use association was weaker at higher reported CL. Coordinating programming concepts, prompts,

response interpretation, and verification may impose competing demands (Huang et al., 2026; Sweller, 1988; Sweller et al., 2019). The interaction does not identify a particular load component or establish an effect on learning quality.

Task segmentation and verification supports may warrant experimental evaluation, but no such practice was examined here. The SDL and CL coefficients were not formally compared and should not be converted into an instructional priority ranking.

5.5. Complementary information from the EBM (RQ5)

The EBM predicted reported use from the DOI attributes, SDL, CL, and intention. Low complexity, compatibility, and observability had the highest main-effect importance scores, and their fitted contribution functions were nonlinear over observed values.

These importance scores and centered contribution functions are model-specific predictive descriptions; they do not establish causal influence, thresholds, developmental stages, or instructional priorities. The EBM target also differed from the PLS-SEM intention target.

Accordingly, the EBM results are best treated as hypothesis-generating. Prospective studies using a common outcome and time frame are needed before the observed shapes can inform intervention design.

6. Conclusions

Among 1,034 Taiwanese undergraduates in information-related programs with prior generative AI experience, trialability ($\beta = .285$), relative advantage ($\beta = .244$), compatibility ($\beta = .224$), and observability ($\beta = .130$) were positively associated with usage intention; low complexity was not significant ($\beta = .060$, $p = .091$). Intention was positively associated with reported programming-related use ($\beta = .335$), and that association varied positively with SDL ($\beta = .050$) and negatively with CL ($\beta = -.114$). EBM described complementary nonlinear predictive patterns for reported use.

The findings characterize adoption perceptions, intention, and reported use; they do not show that generative AI improved code, conceptual understanding, transfer, or independent programming ability. Instructional proposals should therefore remain separate hypotheses for direct evaluation.

Several limitations constrain interpretation. Cross-sectional, simultaneous self-reports preclude causal and temporal conclusions. The convenience sample was limited to undergraduates in information-related programs who reported prior generative AI experience; current programming-course enrollment was not established, and 1.06% reported zero programming credits completed or in progress. Institution identifiers and recruitment-source indicators were not retained, so institution counts, dominant shares, and clustering could not be assessed. Although 1,087 responses were received and 53 responses were excluded using completion-time and near-uniform-response screening, the available records did not preserve exclusion counts by criterion, completion durations, the completion-time cutoff, the exact near-uniformity threshold, or documented duplicate-prevention settings. The reported-use scale covered three recent programming-specific activities but not logs, learning quality, or unaided performance. Common-source bias, unitary CL measurement, adapted-scale limitations, and model-specific EBM estimates provide further constraints.

Future studies should preregister eligibility and screening rules, preserve a complete recruitment flow, report institution-level sample shares, and model clustering when identifiers are available. Longitudinal platform or IDE logs should be paired with code quality, conceptual understanding, transfer, and unaided performance. Experiments can then test whether bounded assistance or verification supports improve those outcomes.

Declarations

Ethics approval: The study was approved by the Human Research Ethics Committee of National Cheng Kung University on October 6, 2025.

Consent to participate: Participation was voluntary, and informed consent was obtained from all participants before questionnaire completion.

Funding: National Science and Technology Council (NSTC), Taiwan

Grant No. 114-2813-C-025-032-H

Author contributions:

Tien-Chi Huang: Conceptualization, Methodology, Writing – original draft, Supervision, Funding acquisition.

Pei-Shan Wu: Investigation, Formal analysis, Writing – original draft, Supervision, Funding acquisition.

Yi-Wen Huang: Investigation, Formal analysis.

Hsin-Ping Tseng: Validation, Project administration.

Yu-Hung Chiang: Writing – review & editing, Project administration.

Conflicts of interest: The authors declare no conflicts of interest.

Generative AI editing disclosure: OpenAI Codex was used to support English-language editing, document formatting, and reference consistency checking during preparation of this submission. The authors reviewed the output and retain responsibility for the final manuscript.

References

Alamri, J. M. (2025). Antecedents of generative artificial intelligence technology adoption: Extended innovation of diffusion model with cultural dimensions and risks perceptions. Journal of Ecohumanism, 4(1), 1718–1738. https://doi.org/10.62754/joe.v4i1.5992

Alpizar-Chacon, I., & Keuning, H. (2025). Student's use of generative AI as a support tool in an advanced web development course. In Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 312–318). Association for Computing Machinery. https://doi.org/10.1145/3724363.3729106

Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakcı, Ö., & Mariman, R. (2025). Generative AI without guardrails can harm learning: Evidence from high school mathematics. Proceedings of the National Academy of Sciences, 122(26), e2422633122. https://doi.org/10.1073/pnas.2422633122

Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Prentice-Hall.

Chen, J., Mokmin, N. A. M., & Qi, S. (2025). Generative AI-powered arts-based learning in middle school history: Impact on achievement, motivation, and cognitive load. The Journal of Educational Research, 118(6), 688–700. https://doi.org/10.1080/00220671.2025.2510395

Chiu, M.-L. (2025). Exploring user awareness and perceived usefulness of generative AI in higher education: The moderating role of trust. Education and Information Technologies, 30(15), 21317–21351. https://doi.org/10.1007/s10639-025-13612-7

Granić, A. (2025). Emerging drivers of adoption of generative AI technology in education: A review. Applied Sciences, 15(13), Article 6968. https://doi.org/10.3390/app15136968

Hair, J. F., Jr., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Danks, N. P., & Ray, S. (2021). Partial least squares structural equation modeling (PLS-SEM) using R: A workbook. Springer. https://doi.org/10.1007/978-3-030-80519-7

Haque, S., & Hundhausen, C. (2025). Generative AI access, usage, and perceptions: An empirical comparison of computing students in the United States and Bangladesh. In Proceedings of the 2025 ACM Conference on International Computing Education Research V. 1 (pp. 109–124). Association for Computing Machinery. https://doi.org/10.1145/3702652.3744231

Huang, T. C., Chen, K. Y., Huang, Y. M., & Chiang, Y. H. (2026). Cognitive and behavioral determinants of ChatGPT adoption in education: An integrated theoretical approach. Interactive Learning Environments, 34(5), 3020–3040. https://doi.org/10.1080/10494820.2025.2559918

Jin, F., Lin, C.-H., & Lai, C. (2025). Modeling AI-assisted writing: How self-regulated learning

influences writing outcomes. Computers in Human Behavior, 165, Article 108538. https://doi.org/10.1016/j.chb.2024.108538

Kazemitabaar, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). Studying the effect of AI code generators on supporting novice learners in introductory programming. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (Article 455, pp. 1–23). Association for Computing Machinery. https://doi.org/10.1145/3544548.3580919

Kazemitabaar, M., Ye, R., Wang, X., Henley, A. Z., Denny, P., Craig, M., & Grossman, T. (2024). CodeAid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Article 650, pp. 1–20). Association for Computing Machinery. https://doi.org/10.1145/3613904.3642773

Klingbeil, A., Grützner, C., & Schreck, P. (2024). Trust and reliance on AI—An experimental study on the extent and costs of overreliance on AI. Computers in Human Behavior, 160, Article 108352. https://doi.org/10.1016/j.chb.2024.108352

Kock, N. (2015). Common method bias in PLS-SEM: A full collinearity assessment approach. International Journal of e-Collaboration, 11(4), 1–10. https://doi.org/10.4018/ijec.2015100101

Lai, J. W. (2024). Adapting self-regulated learning in an age of generative artificial intelligence chatbots. Future Internet, 16(6), Article 218. https://doi.org/10.3390/fi16060218

Li, K. (2023). Determinants of college students' actual use of AI-based systems: An extension of the technology acceptance model. Sustainability, 15(6), Article 5221. https://doi.org/10.3390/su15065221

Lou, Y., Caruana, R., Gehrke, J., & Hooker, G. (2013). Accurate intelligible models with pairwise interactions. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 623–631). Association for Computing Machinery. https://doi.org/10.1145/2487575.2487579

Luera, R., Rossi, R. A., Siu, A., Dernoncourt, F., Yu, T., Kim, S., Zhang, R., Chen, X., Salehy, H., Zhao, J., Basu, S., Mathur, P., & Lipka, N. (2024). Survey of user interface design and interaction techniques in generative AI applications [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2410.22370

Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. Information Systems Research, 2(3), 192–222. https://doi.org/10.1287/isre.2.3.192

Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). InterpretML: A unified framework for machine learning interpretability [Preprint]. arXiv. https://doi.org/10.48550/arXiv.1909.09223

Nunnally, J. C. (1978). Psychometric theory (2nd ed.). McGraw-Hill.

Prather, J., Reeves, B. N., Leinonen, J., MacNeil, S., Randrianasolo, A. S., Becker, B. A., Kimmel, B., Wright, J., & Briggs, B. (2024). The widening gap: The benefits and harms of generative AI for novice programmers. In Proceedings of the 2024 ACM Conference on International Computing Education Research—Volume 1 (pp. 469–486). Association for Computing Machinery. https://doi.org/10.1145/3632620.3671116

Raman, R., Mandal, S., Das, P., Kaur, T., Sanjanasri, J. P., & Nedungadi, P. (2024). Exploring university students' adoption of ChatGPT using the diffusion of innovation theory and sentiment analysis with gender dimension. Human Behavior and Emerging Technologies, 2024(1), Article 3085910. https://doi.org/10.1155/2024/3085910

Ringle, C. M., Wende, S., & Becker, J.-M. (2022). SmartPLS 4 [Computer software]. SmartPLS. https://www.smartpls.com

Rogers, E. M. (2003). Diffusion of innovations (5th ed.). Free Press.

Samala, A. D., Rawas, S., Wang, T., Reed, J. M., Kim, J., Howard, N. J., & Ertz, M. (2025). Unveiling the landscape of generative artificial intelligence in education: A comprehensive taxonomy of applications, challenges, and future prospects. Education and Information Technologies, 30(3), 3239–3278. https://doi.org/10.1007/s10639-024-12936-0

Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic generation of programming exercises and code explanations using large language models. In Proceedings of the 2022 ACM Conference on International Computing Education Research—Volume 1 (pp. 27–43). Association for Computing Machinery. https://doi.org/10.1145/3501385.3543957

Shahzad, M. F., Xu, S., An, X., Asif, M., & Javed, I. (2025). Do generative AI technologies play a double-edged sword role in education? Findings from hybrid approach using PLS-SEM and fsQCA. Education and Information Technologies, 30(14), 19647–19676. https://doi.org/10.1007/s10639-025-13528-2

Stadler, M., Bannert, M., & Sailer, M. (2024). Cognitive ease at a cost: LLMs reduce mental effort but compromise depth in student scientific inquiry. Computers in Human Behavior, 160, Article 108386. https://doi.org/10.1016/j.chb.2024.108386

Strzelecki, A., & ElArabawy, S. (2024). Investigation of the moderation effect of gender and study level on the acceptance and use of generative AI by higher education students: Comparative evidence from Poland and Egypt. British Journal of Educational Technology, 55(3), 1209–1230. https://doi.org/10.1111/bjet.13425

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257–285. https://doi.org/10.1016/0364-0213(88)90023-7

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

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS Quarterly, 27(3), 425–478. https://doi.org/10.2307/30036540

Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. Theory Into Practice, 41(2), 64–70. https://doi.org/10.1207/s15430421tip4102_2

How to Cite: Huang, T.-C., Wu, P.-S., Huang, Y.-W., Tseng, H.-P., & Chiang, Y.-H. (2026). From Perceived Attributes to Reported Use: Generative AI Adoption for Programming-Related Learning among Taiwanese Undergraduates in Information-Related Programs. *Contemporary Education and Teaching Research, 07*(10), 330−347. https://doi.org/10.61360/BoniCETR262020491003