



# 생성형 AI 기반 스캐폴딩 학습환경이 파이썬 학습자의 자기효능감과 실재감에 미치는 영향

## Effects of a Generative AI-Based Scaffolding Environment on Python Learners' Self-Efficacy and Presence

강원태<sup>†</sup> · 이경희<sup>††</sup>  
Won Tae Kang<sup>†</sup> · KyungHee Lee<sup>††</sup>

### 요약

본 연구는 생성형 AI 기반 스캐폴딩 학습환경이 대학 파이썬 수업에서 학습자의 자기효능감과 실재감에 미치는 영향을 분석하였다. 4년제 대학 1학년 57명을 대상으로 단일 집단 사전-사후 검사를 실시하였으며, 수업은 최대지원-부분지원-점진적 독립-완전 독립의 4단계 스캐폴딩 모델로 설계하였다. 분석 결과, 자기학습 효능감(M diff=0.77)과 인지적 실재감(M diff=0.50)이 가장 크게 향상되었고 사회적 실재감(M diff=0.35)은 상대적으로 작은 변화를 보였다. 또한, 생성형 AI 경험이 있는 집단만이 모든 변인에서 유의미한 향상을 보인 반면, 경험이 없는 집단은 자기학습 효능감에서 가장 큰 향상을 나타냈다. 이러한 결과는 생성형 AI 기반 교수 설계에서 인지적 유용성 강화와 초심자를 위한 적응적 스캐폴딩 전략이 필요함을 시사한다.

**주제어** 파이썬, 생성형 AI, 자기효능감, 실재감, 스캐폴딩

### ABSTRACT

This study examined the effects of a generative AI-based scaffolding learning environment on university students' self-efficacy and sense of presence in an introductory Python course. A one-group pretest-posttest design was conducted with 57 first-year students at a four-year university. The course was structured using a four-stage scaffolding model consisting of Full Support, Partial Support, Gradual Fading, and Complete Independence. The results indicated that self-regulated learning efficacy (M diff = 0.77) and cognitive presence (M diff = 0.50) showed the greatest improvements, whereas social presence (M diff = 0.35) demonstrated relatively smaller gains. In addition, only students with prior experience using generative AI exhibited significant improvements across all variables, while those without such experience showed the greatest improvement in self-regulated learning efficacy. These findings suggest the need to strengthen cognitive affordances and incorporate adaptive scaffolding strategies—particularly for novice learners—in the design of generative AI-supported instruction.

**Keywords** Python, Gen AI, Self-Efficacy, Presence, Scaffolding

†정회원      경북대학교 인공지능전공 강사 호서대  
††중신회원   호서대학교 혁신융합학부 조교수  
논문투고      2025년 10월 20일  
심사완료      2025년 11월 28일  
게재확정      2025년 12월 07일  
발행일자      2025년 12월 31일

## 1. 서론

생성형 AI의 등장은 고등교육 전반에 걸쳐 교수-학습 패러다임을 급격하게 전환시키며 교육 기술 분야에서도 기존 논의의 범주를 넘어서는 새로운 연구 영역을 형성하고 있다 [1]. 대규모 언어 모델(LLM)은 학습자와 실시간으로 상호작용하며 개인화된 피드백을 제공할 수 있어 전통적인 정적 콘텐츠 기반 학습을 넘어 학습 경험의 질적 향상을 가져올 잠재력을 지닌다. 특히 생성형 AI는 전통적인 교실 환경에서 교수자가 모든 프로그래밍 교육과 같이 높은 인지적 부담이 요구되는 영역에서는 생성형 AI가 학습자에게 즉각적인 도움을 제공함으로써 교수자 개인의 제약을 보완할 수 있다는 점에서 주목을 받고 있다[2].

이러한 생성형 AI 활용 가능성은 스캐폴딩(Scaffolding)의 관점에서 특히 의미가 크다. 스캐폴딩은 Vygotsky(1978)의 근접발달영역(ZPD) 이론에 근거한 교수 전략으로 학습자가 독립적으로 수행하기 어려운 과제를 적절한 도움을 통해 달성하도록 지원하는 임시적, 적응적 지원 체계를 의미한다[3, 4]. 그러나 전통적인 스캐폴딩은 교수의 직접적 개입에 의존하기 때문에 대규모 강의, 온라인 수업, 실습 중심 수업 등의 환경에서는 개별 학습자의 수준을 실시간으로 파악하여 맞춤형 피드백을 제공하는 데 구조적 한계가 존재해 왔다. 생성형 AI는 이러한 한계를 극복하여 학습자의 코드, 질문, 오류 상태를 즉각 분석하고 개인 수준에 맞는 설명과 단계적 해결 방안을 제공함으로써 스캐폴딩의 자동화, 확장화 가능성을 제시한다. 그럼에도 불구하고 생성형 AI 기반 스캐폴딩이 학습자의 심리적, 인지적 측면에 미치는 영향을 다각도로 분석한 실증 연구는 여전히 제한적이다. 기존 연구들은 AI 도구의 활용 만족도나 학습 편의성을 중심으로 효과를 평가하는 경향이 있으며[5, 6], 스캐폴딩의 어떠한 요소가 학습자의 심리적, 인지적 경험 변화에 기여하는지에 대한 설명력은 충분하지 않다. 특히 프로그래밍 교육에서 중요한 변인인 자기효능감(Self-efficacy)과 실재감(Presence)은 학습 동기와 몰입, 고차원적 사고 수행 등 학습 성과와 밀접하게 연관되어 있음에도 불구하고 생성형 AI 기반 스캐폴딩이 이러한 변인에 미치는 영향은 체계적으로 검증되지 않았다.

자기효능감은 특정 학습 과제를 성공적으로 수행할 수 있다는 학습자의 신념을 의미하며 학습 동기와 성취를 예측하는 핵심 요인으로 알려져 있다[7]. 생성형 AI는 즉각적인 피드백과 단계적 지원을 제공해 학습자의 성공 경험 축적을 돕고, 이는 곧 자기효능감 향상으로 이어질 가능성이 크다. 실재감은 학습자가 상호작용적 학습 환경에서 경험하는 몰입감과 상호작용의 질을 포괄하는 중요한 학습 경험 변수로 교수적, 사회적, 인지적 요소가 상호작용하여 형성된다[8]. AI가 제공하는 스캐폴딩의 질과 적절성은 이 세 영역 전반에서 학습자의 실재감을 변화시킬 수 있다. 그러나 기존 연구에서는 AI 기반 스캐폴딩의 효과를 단일 차원에서 측정하거나, 학습 만족도 수준의 표면적 지표에 의존한 경우가 많아

[5, 6], 어떤 하위 요인이 어떻게 변화하는지, 특정 학습자 특성(AI 사용 경험, 프로그래밍 경험)에 따라 효과가 어떻게 달라지는지에 대한 체계적 분석이 부족했다. 이는 고등교육 맥락에서 생성형 AI 기반 스캐폴딩 설계를 위한 실질적 가이드라인을 제시하는 데 한계를 초래한다.

따라서 본 연구는 이러한 학문적 간극을 해소하고자 생성형 AI 기반 스캐폴딩의 효과를 하위 요인 단위로 구조화하여 측정하고, 그 변화 양상을 실증적으로 분석하는 것을 목적으로 한다. 이를 위해 4년제 대학 1학년 학생을 대상으로 생성형 AI 기반 스캐폴딩이 적용된 파이썬 프로그래밍 수업의 사전-사후 변화를 분석하였으며, 다음과 같은 연구 문제를 도출하였다.

첫째, 생성형 AI 기반 스캐폴딩 수업 참여는 학습자의 자기효능감(자기학습 효능감, 기술 효능감, 수업참여 효능감)에 유의미한 변화를 미치는가? 둘째, 생성형 AI 기반 스캐폴딩 수업 참여는 학습자의 실재감(인지적 실재감, 사회적 실재감, 교수 실재감)에 유의미한 변화를 미치는가? 셋째, 학습자의 생성형 AI 사용 경험과 파이썬 사용 경험은 자기효능감 및 실재감 변화에 차이를 발생시키는가?

본 연구는 이러한 분석을 통해 생성형 AI 기반 스캐폴딩 교육의 효과를 실증적으로 규명하고, 고등 교육 맥락에서 AI 기반 교수 설계 및 학습 환경 개발을 위한 기초 자료와 학문적 시사점을 제공하고자 한다.

## 2. 이론적 배경

### 2.1 생성형 인공지능과 교육적 활용

인공지능(AI)은 인간의 지각, 학습, 추론 능력을 모방하는 시스템으로 시작하여 최근에는 텍스트, 이미지, 코드 등 새로운 콘텐츠를 창조할 수 있는 생성형 AI(Generative AI)로 진화하며 고등교육의 패러다임을 전환하고 있다[9-12]. 유럽연합(EU)의 정의나 관련 기술 연구들에 따르면, 생성형 AI는 대규모 언어 모델(LLM)을 기반으로 사용자 입력에 대해 자연스럽고 맥락에 맞는 결과물을 산출한다는 점에서 기존의 규칙 기반 AI와 구별된다. 대표적으로 ChatGPT는 맥락 이해를 바탕으로 인간과 유사한 대화를 수행하며 GitHub Copilot과 같은 도구는 방대한 코드 데이터베이스를 학습하여 개발자의 의도를 파악하고 코드를 자동 생성함으로써 생산성을 혁신하고 있다[13-15].

이러한 생성형 AI의 등장은 교육 현장, 특히 프로그래밍 교육에서 새로운 가능성을 시사한다. 선행연구들을 종합해 보면, 생성형 AI는 학습자와 실시간으로 상호작용하며 개인화된 피드백을 제공함으로써 학습자의 이해도와 성과를 높이는 데 긍정적인 영향을 미치는 것으로 보고된다[16]. 대학 교육에서 AI 기반 조교 시스템은 학습 시간을 단축시키고, 개인화된 피드백은 학습자의 태도와 만족도를 향상시키는 주요 요인으로 작용한다.

그러나 생성형 AI를 단순히 자동화 도구로만 바라보는

것은 교육적 잠재력을 축소하는 것이다. 본 연구는 생성형 AI가 단순한 정보 제공을 넘어, 학습자가 직면한 문제 상황을 함께 해결하고 고차원적 사고를 촉진하는 '협업 파트너'로서 기능할 수 있다는 점에 주목한다. 특히 프로그래밍 입문 과정에서 AI가 제공하는 즉각적인 예러 진단과 코드 설명은 초심자가 겪는 인지적 부하를 낮추고, 자기 주도적 학습 환경을 구축하는 핵심 기제가 될 수 있다.

## 2.2 자기효능감의 개념 및 교육적 의미

자기효능감(Self-Efficacy)은 Bandura(1977)가 제안한 개념으로 특정한 목표를 달성하기 위해 필요한 행동을 조직하고 실행할 수 있다는 자신의 능력에 대한 믿음을 의미한다[7]. 이는 단순한 기능적 보유 능력이 아니라 다양한 상황에서 그 능력을 발휘할 수 있다는 자신감 기제로 학습자의 동기 부여와 학업 성취를 예측하는 가장 강력한 심리적 요인 중 하나이다[17, 18]. 정보 기술의 발달과 함께 이러한 개념은 컴퓨터나 정보기술을 효과적으로 사용할 수 있다는 믿음인 '정보기술 자기효능감(ITSE)'으로 구체화되어 논의되고 있다[19].

프로그래밍 교육 맥락에서 자기효능감은 학습 지속성을 결정짓는 중요한 변인이다. 선행연구들에 따르면 프로그래밍 학습 초기에 겪는 잦은 오류와 실패 경험은 학습자의 효능감을 저하시키는 주된 원인이 되며, 반대로 작은 프로젝트의 완수나 문제 해결의 성공 경험은 효능감을 높이는 핵심 기제로 작용한다. 이러한 관점에서 최근 도입된 생성형 AI 도구들은 학습자에게 새로운 차원의 성공 경험을 제공할 수 있다.

기존 연구들은 멘토의 피드백이 효능감에 미치는 긍정적 영향을 강조했다면, 본 연구에서는 생성형 AI가 그 멘토의 역할을 수행할 수 있음에 주목한다. 생성형 AI는 코드 자동 완성을 넘어 실시간으로 오류를 수정해주고, 문제 해결의 실마리를 제공함으로써 학습자가 혼자서는 해결하기 어려운 과제를 완수하게 돕는다. 이러한 'AI 기반의 성공 경험 축적'은 학습자의 심리적 부담을 경감시키고, 결과적으로 파이썬 프로그래밍에 대한 자기효능감 향상으로 이어질 것이라는 가설을 설정할 수 있다.

## 2.3 학습 실재감의 개념 및 교육적 중요성

학습 실재감은 학습자가 학습 환경 내에서 느끼는 몰입감과 존재감을 의미하며, 크게 인지적, 사회적, 정서적 존재감으로 구성된다[22]. 이러한 실재감은 학습자가 수업에 주도적으로 참여하고 동료 및 교수자와 상호작용하며 소속감을 느끼게 하는 핵심 기제로 작용한다. [23, 24]. 선행연구들을 종합해 볼 때, 높은 학습 실재감은 학습 목표 설정과 자기조절 학습 능력을 강화하고 결과적으로 학업 성취도 향상과 밀접한 관련이 있음을 알 수 있다.

최근 생성형 AI의 도입은 이러한 학습 실재감 논의를 새로운 차원으로 확장하고 있다. 여러 연구에 따르면 생성형

AI는 학습자 개인의 스타일과 진도를 분석하여 맞춤형 자료를 제공하고 즉각적인 피드백을 산출함으로써 학습자의 흥미와 몰입도를 유의미하게 높이는 것으로 보고된다[24]. 특히 자기 결정 이론과 성취 목표 이론의 관점에서 볼 때, AI가 제공하는 자율적인 학습 경로와 적절한 난이도의 과제 제안은 학습자가 도전을 지속하게 만드는 중요한 동기 부여 요소로 작용한다[25]. 하지만 기존 연구들은 AI 도구의 전반적인 만족도나 효능감을 확인하는 데 집중된 경향이 있다[26-28]. 이에 본 연구는 단순히 AI가 실재감을 높인다는 사실을 넘어, 파이썬 프로그래밍이라는 구체적인 맥락에서 AI의 어떤 스캐폴딩 기능이 인지적, 사회적 실재감의 세부 요인에 기여하는지를 규명하고자 한다. 이는 높은 자기효능감이 학습 실재감으로 이어지고, 이것이 다시 학업 성취로 연결된다는 선행연구의 흐름을 생성형 AI 환경에서 재확인하는 과정이 될 것이다.

## 2.4 파이썬과 생성형 AI 활용 스캐폴딩 전략

파이썬은 간결하고 직관적인 문법 구조 덕분에 데이터 분석 및 인공지능 분야에서 널리 활용되며 초보자에게 적합한 언어로 평가받는다[29-31]. 그러나 실제 학습 현장에서는 논리적 사고의 복잡성으로 인해 여전히 학습 장벽이 존재하며, 이를 극복하기 위해 프로젝트 기반 학습이나 협업 학습과 같은 다양한 교수 전략이 시도되어 왔다. 이러한 맥락에서 본 연구는 Vygotsky(1978)의 사회문화적 학습 이론에 기반한 '스캐폴딩(Scaffolding)' 개념에 주목한다. 스캐폴딩은 학습자가 근접 발달 영역(ZPD) 내에서 유능한 타인의 도움을 받아 잠재적 역량을 발휘하도록 돕는 임시적 지원 체계를 의미한다[32].

전통적인 스캐폴딩이 교수자나 동료에 의존했다면, 최근의 생성형 AI는 이 역할을 확장하고 있다[5, 6]. 기존의 규칙 기반 시스템과 달리, 생성형 AI는 학습자의 질문 의도와 오류 유형을 파악하여 실시간으로 맞춤형 비계를 제공할 수 있는 역량을 갖추고 있기 때문이다. 즉, AI는 단순한 정답 제공자가 아니라 학습자가 스스로 지식을 구성하도록 돕는 '인지적 파트너'로서 기능할 수 있다. 이에 본 연구는 생성형 AI 활용 맞춤형 스캐폴딩(GAPS) 전략을 제안한다. 이 전략의 핵심은 학습자의 숙련도가 높아짐에 따라 AI의 지원을 점진적으로 줄여가는 '페이딩(Fading)' 기법을 적용하는 것이다. 이는 선행연구에서 강조한 바와 같이, 학습자가 AI에 의존하지 않고 독립적인 문제 해결 능력을 내재화하여 자기 주도적 학습자로 성장하도록 돕는 데 목적이 있다.

## 3. 연구 방법

### 3.1 연구 설계

본 연구는 생성형 AI 기반 스캐폴딩을 적용한 파이썬교양 필수 수업 참여가 대학생의 자기효능감 및 실재감에 미치는 영향을 분석하기 위해 단일 집단 사전-사후 검사 설계를

를 채택했다. 이는 실험 처치(생성형 AI 기반 수업)의 효과를 확인하기 위해, 동일한 연구 대상 집단에 대하여 수업 전과 수업 후 종속 변인의 변화를 측정하고 비교하는 양적 연구 설계 방법이다. 연구의 독립 변인은 '생성형 AI 활용 교양 필수 수업 참여'이며, 종속 변인은 '자기효능감'과 '실재감'이다. 또한, 학습자의 초기 특성인 '생성형 AI 사용 경험 유무'와 '파이썬 경험 유무'를 하위 요인으로 설정하여 수업 효과의 집단 간 차이를 심층적으로 탐색했다.

### 3.2 연구 대상 및 표집 방법

연구 대상은 국내 경북지역 4년제 공과 대학교 1학년 학생으로 구성되었으며, 2025학년도 1학기에 개설된 '디지털 문해력' 교양 필수 과목을 수강했다. 해당 교과목을 수강하는 학생들을 대상으로 연구 참여 의사를 확인하여 자발적으로 동의한 참여자를 표집하는 편의 표집 방식을 사용했다.

수업 시작 직후 연구의 목적을 설명하고 동의한 참여자를 대상으로 자기효능감과 학습 실재감에 대한 사전 설문을 실시했다. 또한, 학습자의 초기 특성 파악을 위해 파이썬 및 생성형 AI 사용 경험 유무를 함께 측정했다. 연구 참여자들은 15주간의 정규 학기 수업에서 문제 해결, 코드 작성 보조,

학습 내용 요약, 질문 응답 등의 활동에 생성형 AI를 교수자의 단계적 가이드에 따라 활용하며 교과목을 이수했다. 수업이 모두 종료된 직후 동일한 문항으로 구성된 사후 설문을 실시하여 수업 참여 후 종속 변인의 변화 수준을 측정했다.

연구 참여자는 총 80명(2개 분반)이었으나, 사전-사후 설문에 모두 응답한 참여자를 대응 쌍으로 구성한 결과 57명(N=57)의 자료만을 최종 분석에 사용했다.

### 3.3 수업 설계 및 생성형 AI 활용 방식

본 수업은 스캐폴딩 이론을 기반으로 생성형 AI의 지원 수준을 단계적으로 조정하는 4단계 모델로 설계되었다 [30]. Table 1에 따르면 1-4주차는 최대 지원 단계(Full Scaffolding)에 해당한다. 이 기간 교수자는 생성형 AI 활용을 시범 보이고 학습자의 관심을 유도하며, 복잡한 문법을 단순화하는 전략으로 프로그래밍 입문 장벽을 최소화한다. 학습자는 이러한 지원 아래에서 생성형 AI의 코드 예시와 설명을 활용해 코드를 생성·분석하며 기초 문법과 프로그래밍 사고를 형성한다. 5-7주차는 부분적 지원 축소 단계(Partial Fading)로, 이 시기에는 생성형 AI가 직접적인 코드 생성을 제공하지 않고 학습자가 작성한 논리를 검증

Table 1. Weekly Plan for Generative AI-Based Scaffolding in Python Course

Week	Topic	Learning Content	Scaffolding Goal	Generative AI Application and Role
1	Orientation & Environment Setup	Course introduction, Python installation and environment configuration, Introduction to generative AI usage	[Modeling] Minimize barriers to accessing new environments and tools	[Maximum Support] Instructor demonstrates coding practices using gen AI, and students receive step-by-step guidance for environment setup
2	Computational Thinking & Algorithms	Understanding algorithm concepts, flowchart creation, learning problem-solving procedures	[Recruiting] Generate interest and motivation for fundamental programming concepts (algorithms)	[Maximum Support] Students ask questions about ambiguous aspects of algorithm definitions and flowchart creation to clarify concepts
3	Basic Syntax I	Practice with variables, data types, and operators	[Simplification] Help understand complex syntax structures easily and reduce anxiety about errors	[Maximum Support] Generate simple code, then analyze and modify the generated code to learn syntax
4	Basic Syntax II	Practice with input/output and built-in functions	[Direction] Support repetitive syntax error resolution to maintain focus on learning objectives (function usage)	[Maximum Support] Input error codes that occur during practice assignments and request explanations of solutions and causes
5	Control Statements	Practice with conditional statements (if, match)	[Fading] Guide focus on conditional logic design instead of simple syntax generation	[Partial Reduction] Present student-created logical structures (conditions) and request validation of logic and outcome predictions
6	Loop Statements	Practice with for and while loops	[Frustration Control] Support rapid resolution of difficult errors such as infinite loops in iteration	[Partial Reduction] For complex loop errors, request debugging directions and checkpoints (not direct solutions) to attempt independent resolution
7	Data Types I	Practice with lists, tuples, and strings	[Collaboration] Explore methods for manipulating complex data structures in consultation with tutor	[Partial Reduction] Inquire about the most efficient Python built-in functions for specific results and integrate them into own code
8	Midterm Exam	Mid-term evaluation (coding and concept understanding)	[Evaluation] Assess foundational knowledge learned independently without scaffolding	[No Use]
9	Data Types II	Practice with sets and dictionaries	[Collaborative Problem-Solving] Design complex structures for actual project application together with tutor	[Progressive Independence] Receive assistance only for overall project structure (framework) design, while implementing specific details (sub-functions) independently
10	Functions	Practice with function definition, calling, and return values	[Conceptualization] Understand conceptually how to effectively separate and reuse functions	[Progressive Independence] Input self-written code and request feedback on "improvements in function definition and modularization aspects"
11	Modules	Practice with standard modules and user-defined modules	[Independent Tool Use] Expand problem-solving capabilities by utilizing external resources	[Progressive Independence] Inquire about usage examples and limitations of required standard modules, using them like official documentation

Week	Topic	Learning Content	Scaffolding Goal	Generative AI Application and Role
12	Classes & Objects	Object-oriented concepts, class design and implementation	[Mastery] Learn abstract concepts like object-orientation through concrete examples	[Progressive Independence] Request various real-life example codes for specific concepts (e.g., inheritance, polymorphism) to deepen conceptual understanding
13	File I/O	Text file read/write, exception handling	[Problem Analysis & Design] Independently predict and resolve complex scenarios such as I/O error handling	[Progressive Independence] Discuss possible exception scenarios, but write specific exception handling implementation independently
14	Comprehensive Project	Design and implementation of simple application program	[Full Fading] Complete final deliverable and self-verify without generative AI assistance	[Minimal/No Support] Use prohibited in principle; when stuck, resolve through textbooks, official documentation, or peer discussion instead of asking questions
15	Final Exam & Reflection	Final evaluation and reflection on learning experience	[Evaluation] Assess final learning outcomes with all scaffolding withdrawn	[No Use]

하거나 디버깅의 방향성만을 제시하여 스스로 문제를 해결하도록 유도한다. 9-13주차는 점진적 독립 단계(Gradual Fading)에 해당하며, 생성형 AI는 프로젝트의 전체 구조 설계나 개념적 조건 등 최소한의 지원만을 제공하고 구체적인 구현은 학습자가 독립적으로 수행한다. 14주차는 완전 독립 단계(Complete Fading)로, 생성형 AI 사용을 원칙적으로 제한하여 학습자가 자신의 역량만으로 종합 프로젝트를 완성하도록 한다. 또한 8주차 중간고사와 15주차 기말고사는 생성형 AI의 도움 없이 학습자의 독립적 수행 역량을 평가하는 시점으로 설정했다.

기존 생성형 AI 활용 프로그래밍 교육 연구들은 대체로 AI를 오류 수정이나 코드 생성과 같은 기능적 지원을 제공하는 상시적 보조 도구로 활용하는 데 그쳤다. 이러한 접근은 학기 전반에 걸쳐 일정한 수준의 AI 지원을 제공함으로써 학습자의 AI 의존도를 높일 위험을 내포하고 있었다. 반면, 본 연구는 스캐폴딩의 동적 철회 구조에 기반하여, 초기의 '최대 지원(Modeling)' 단계에서 출발해 학기 말에는 AI 사용을 의도적으로 제한하는 '완전 독립(Independence)' 단계로 이행하는 체계적 로드맵을 적용했다는 점에서 기존 연구와 구별된다.

이러한 단계적 스캐폴딩 철회(Fading) 과정은 학습자가 초기의 의존적 학습자로부터 점차 자기주도적 문제 해결자로 성장하도록 설계되었으며, 각 단계에서 생성형 AI의 역할 역시 지식 제공자에서 협력자, 최종적으로는 참고 자원으로 변화한다. 이를 통해 학습자는 생성형 AI를 효과적으로 활용함과 동시에, AI 없이도 독립적으로 파이썬 프로그래밍 문제를 해결할 수 있는 역량을 함양하게 된다.

### 3.4 연구 도구

연구 도구는 장지수(2024)의 생성형 AI 챗봇을 활용한 온라인 프로그래밍 학습 환경에서 학습자의 실재감, 자기효능감, 프로그래밍 지식수준 간의 관계 분석 연구[28]에서 개발한 도구를 본 연구의 환경과 목적에 맞게 수정·보완하여 사용하였다. 구체적으로 기존 문항에서 '챗봇'으로 명시된 주어를 '생성형 AI'로 변경하고, '프로그래밍 학습'이라는 포괄적 맥락을 '파이썬 프로그래밍 수업 및 실습 과정'으로 구체화함으로써 생태학적 타당도를 높였다. 수정된 문항은 교육학 전문가 3인과 컴퓨터교육 전문가 2인의 검토를 거쳐

안면 타당도를 확보한 후 연구에 적용되었다.

연구 도구의 내적 일관성을 확인하기 위해 Cronbach's  $\alpha$  계수를 산출한 결과, Table 2와 같이 모든 하위 요인 값이 최소 0.842에서 최대 0.964로 매우 양호하거나 우수한 수준의 신뢰도를 확보한 것으로 나타났다. 이는 본 연구에 사용된 척도들이 학습자의 변인 특성을 일관성 있게 측정했음을 의미한다

Table 2. Reliability of Research Instruments (Cronbach's Alpha)

Scale	Sub-Factors	Q.	Pre $\alpha$	Post $\alpha$
Self-Efficacy	Self-Learning Efficacy	3	0.902	0.923
	Classroom Engagement Efficacy	3	0.883	0.915
	Technical Efficacy	2	0.927	0.924
Presence	Social Presence	6	0.842	0.901
	Cognitive Presence	6	0.879	0.964
	Teaching Presence	4	0.88	0.902

하위 요인 가운데 가장 높은 신뢰도를 보인 것은 사후 검사의 실재감 중 인지적 실재감 요인으로,  $\alpha=0.964$ 로 확인되었다. 이는 수업 후 학습자들이 생성형 AI가 제공하는 지적 유용성을 매우 일관되게 인식하고 응답했음을 시사한다. 반면, 가장 낮은 신뢰도는 사전 검사에서의 실재감 중 사회적 실재감 요인으로  $\alpha=0.842$ 를 기록하였다. 이 값은 학술 연구에 활용하기 충분히 안정적인 수준이지만, 해당 척도(대화감, 소외감 해소 등)를 구성하는 항목들이 학습자에게 상대적으로 다양한 정서적/사회적 경험으로 해석될 가능성을 내포한다. 이와 같이 모든 척도가 전반적으로 안정적인 신뢰도를 확보하였으므로 본 연구의 통계 분석 결과를 해석하는데 있어 도구의 신뢰성은 문제가 없는 것으로 판단된다.

### 3.5 자료 분석

수집된 자료는 통계 프로그램 SPSS를 이용하여 분석하였다. 모든 통계적 유의성 검증은 유의 수준  $\alpha=.05$ 를 기준으로 하였다. 연구 대상의 인구통계학적 특성(성별, 연령)을 파악하기 위해 빈도 분석 및 백분율(%)을 실시하였다. 측정 도구의 내적 일관성을 검증하기 위해 Cronbach's  $\alpha$  계수를 산출하였다.

자기효능감 및 실재감의 수업 전후 변화를 검증하기 위해 대응표본 t-검정을 실시했다. 이는 동일 집단의 두 시점(사전-사후) 평균 차이를 검증하는 데 사용되었다. 각 변인의 평균(M), 표준편차(SD), 사전-사후 점수 간 상관계수(r), t 값, 그리고 유의확률(p)을 제시하였다. '생성형 AI 사용 경험 유무'와 '파이썬 사용 경험 유무' 하위 요인에 따라 학습 효과의 차이가 있는지 심층적으로 분석하기 위해 각 그룹(경험 있음/경험 없음)을 대상으로 개별적인 대응표본 t-검정을 실시하여 효과의 차이를 비교 분석하였다.

## 4. 연구 결과

### 4.1 생성형 AI 기반 스캐폴딩이 자기효능감에 미친 영향

본 연구는 생성형 AI 기반 스캐폴딩 학습 환경이 파이썬 학습자의 자기효능감에 미치는 영향을 분석하기 위해 사전-사후 대응표본 t-검정을 실시하였다. 총 57명의 자료를 분석에 활용하였으며, 대응표본 t-검정의 자유도(df)는 N-1 기준에 따라 56으로 산출되었다. 분석 결과 학습자들은 수업 참여 전후에 자기효능감에서 유의미한 변화를 보였다.

**Table 3.** Effects of Gen AI-Based Scaffolding on Self-Efficacy Factors

Sub-Factors	Q	Pre M(SD)	Post M(SD)	M Diff.	t	p	Cohen's d
Self-Learning Efficacy	3	3.02 (0.92)	3.79 (0.81)	0.77	-4.52	<.001	0.60
Classroom Engagement Efficacy	3	3.29 (0.91)	3.95 (0.80)	0.67	-4.01	<.001	0.53
Technical Efficacy	2	3.41 (0.89)	4.08 (0.69)	0.67	-4.43	<.001	0.59
Total	8	3.26 (0.82)	3.92 (0.72)	0.66	-4.63	<.001	0.61

전체 참여자의 자기효능감 총점 평균은 Table 3과 같이 사전 3.26(SD=0.82)에서 사후 3.92(SD=0.72)로 약 0.66점 향상된 것으로 나타났으며, 이러한 변화는 통계적으로 매우 유의하였다( $t(56)=-4.63, p<.001$ ). 또한, 효과크기(Cohen's d)는 0.61로 산출되어, 생성형 AI 기반 스캐폴딩이 학습자의 자기효능감 향상에 교육적으로 의미 있는 중간 수준(Medium Effect Size)의 효과를 미친 것으로 확인하였다. 이는 생성형 AI 기반 학습 환경이 학습자의 학습 목표 달성 기대와 복잡한 과제 수행에 대한 긍정적인 인식을 강화하는데 효과적이었음을 시사한다.

하위 요인별 분석 결과, 자기학습 효능감, 기술 효능감, 수업참여 효능감 모두에서 평균 차이(M diff)가 0.67 이상으로 나타나 통계적으로 유의미한 향상이 확인되었다( $p<0.001$ ). 세 요인 중 가장 큰 향상 폭을 보인 것은 자기학습 효능감(M diff=0.77)으로 학습자가 스스로 어려운 내용을 이해하고 기본 개념을 습득할 수 있다는 자신감을 반영하는 요인이다. 이러한 결과는 생성형 AI가 개별 학습 수준에

맞춘 상세하고 반복적인 설명을 제공함으로써 스캐폴딩 역할을 수행했기 때문으로 해석된다. 즉, AI 지원 스캐폴딩이 학습자의 인지적 불확실성을 완화하고 자기주도적 학습 능력에 대한 확신을 가장 크게 높였음을 의미한다.

기술 효능감과 수업참여 효능감 역시 각각 0.67의 평균 차이를 보이며 유의미하게 향상되었다. 이는 생성형 AI가 파이썬 코딩 기술 습득과 복잡한 학습 환경 적응에 대한 자신감(기술 효능감), 성취 기대와 과제 수행에 대한 긍정적 인식(수업참여 효능감)을 동시에 증진시켰음을 보여준다. 이러한 효과는 AI가 적절한 난이도의 피드백과 단계적 도움을 제공함으로써 학습자의 불안감을 완화하고 성공적인 학습 경험을 축적하게 한 결과로 볼 수 있다.

사후 측정에서 전체 자기효능감 평균 점수가 4.0에 근접한 것은 대다수 학습자가 수업에서 제시된 내용을 이해하고 실습 과제를 해결할 수 있다는 높은 자신감을 얻게 되었음을 의미한다. 이는 AI 기반의 맞춤형 학습 지원이 학습자의 불안감을 줄이고 단계적 스캐폴딩을 통해 긍정적 학습 성공 경험을 강화한 결과로 해석된다.

사전-사후 자기효능감 총점 간의 상관계수는  $r=0.03$ 로 매우 낮게 나타났다. 이는 초기 자기효능감 수준이 수업 후 최종 효능감에 거의 영향을 미치지 않았다는 것을 의미하며, 수업 전 자신감이 낮았던 학습자도 AI 기반 스캐폴딩의 도움을 통해 효능감을 크게 향상시킬 수 있었음을 보여준다. 이러한 결과는 생성형 AI를 활용한 교육 설계가 학습자의 능력 향상뿐 아니라 자기효능감과 같은 심리적 요인 증진에도 중요한 교육적 가치를 제공함을 시사한다. 따라서 향후 교육 프로그램 설계에서는 AI 기반 스캐폴딩 전략을 체계적으로 활용하여 학습자의 효능감을 강화하는 방안이 필요할 것이다.

### 4.2 생성형 AI 기반 스캐폴딩이 실재감에 미친 영향

본 연구는 생성형 AI를 활용한 스캐폴딩 학습 환경에서 학습자가 경험하는 실재감의 변화를 탐색하기 위해 사전-사후 대응표본 t-검정을 실시하였다. 실재감 척도는 학습 환경에 대한 인식 변화를 보다 명확히 파악하기 위해 사회적 실재감, 인지적 실재감, 교수 실재감 세 가지 하위 요인으로 구분하여 분석했다.

**Table 4.** Effects of Gen AI-Based Scaffolding on Presence Factors

Sub-Factors	Q.	Pre M(SD)	Post M(SD)	M Diff.	t	p	Cohen's d
Social Presence	6	3.48 (0.62)	3.83 (0.71)	0.35	-2.82	.007	0.37
Cognitive Presence	6	3.63 (0.57)	4.13 (0.67)	0.5	-3.93	<.001	0.52
Teaching Presence	4	3.61 (0.63)	4.01 (0.65)	0.4	-3.23	.002	0.43
Total	16	3.59 (0.51)	3.99 (0.64)	0.4	-3.66	<.001	0.49

분석 결과 Table 4와 같이 실재감 전체 총점의 사전 평균은 3.59(SD=0.51), 사후 평균은 3.99(SD=0.64)로 나타났다. 수업 후 실재감 점수는 통계적으로 유의하게 증가하였다( $t(56)=-3.66, p<.001$ ). 추가적으로 산출한 효과크기(Cohen's d)는 0.49로 생성형 AI 기반 스캐폴딩이 학습 실재감 증진에 교육적으로 의미 있는 중간 수준의 효과를 지녔음을 확인하였다. 이는 학습자들이 수업 이전부터 일정 수준의 실재감을 기대하고 있었으며, 수업 참여 후 생성형 AI 기반 스캐폴딩 경험을 통해 그 인식이 더욱 강화되었음을 시사한다.

하위 요인별 분석에서도 세 요인 모두  $p\leq.007$  수준에서 통계적으로 유의미한 향상이 나타났다. 가장 큰 향상 폭을 보인 요인은 인지적 실재감(M diff=0.50)으로 사후 평균은 4.13으로 확인되었다. 인지적 실재감은 AI가 제공하는 정보의 유용성, 고차원적 사고 지원, 지식 구성의 촉진에 대한 학습자의 인식을 반영하는 요인으로 생성형 AI가 단순한 도구를 넘어 지적 활동을 안내하는 학습 파트너로 기능했음을 보여준다.

교수 실재감(M diff=0.40)도 유의미한 향상을 보였다( $p=.002$ ). 이는 생성형 AI가 명확하고 일관된 교수적 피드백을 제공함으로써 학습자가 설계적, 전달적 측면의 교수적 지원을 실질적으로 체감했음을 의미한다. 즉, AI가 교수자의 일부 역할을 보완하거나 대체하는 교수적 스캐폴딩으로 작용했음을 확인할 수 있다.

반면, 사회적 실재감(M diff=0.35)은 가장 낮은 향상 폭을 보였으나 통계적으로 유의미한 증가를 나타냈다( $p=.007$ ). 이는 생성형 AI와의 상호작용이 학습자에게 정서적 안정감, 몰입감, 대화감을 제공했음을 의미하지만, 학습자들이 여전히 AI를 사회적 상호작용의 대상이라기보다는 학습 효율을 높이는 도구로 인식하고 있음을 시사한다.

사전-사후 실재감 총점 간 상관계수는  $r=-0.01$ 로 유의미한 관련성이 없었다. 이는 초기 실재감 수준이 낮았던 학습자도 수업 과정에서 제공된 AI 기반 스캐폴딩을 통해 높은 실재감을 경험할 수 있었음을 의미한다. 종합적으로, 생성형 AI는 학습자의 인지적 참여, 교수적 명료성, 정서적 몰입을 동시에 촉진하여 매개 요소로 기능하며 학습 환경의 질적 향상에 기여한다. 따라서 향후 AI 기반 학습 시스템 설계에서는 인지적 유용성과 교수적 명료성을 유지하되 사회적 실재감 요소를 강화하는 방향으로 스캐폴딩 전략을 고도화할 필요가 있다.

### 4.3 하위 요인별 분석

본 분석은 생성형 AI 사용 경험과 파이썬 사용 경험이 초기 학습자 특성으로서 생성형 AI 활용 수업의 효과를 결정하거나 강화하는 선행 요인으로 작용하는지를 검증하는 데 초점을 두었다.

#### 4.3.1 파이썬 사용 경험에 따른 요인별 분석

분석 결과 파이썬 경험이 없는 집단(N=48)은 Table 5와 같이 자기효능감의 세 요인 모두( $p\leq.001$ )와 실재감의 인지적 실재감 요인( $p=.002$ )에서 유의미한 향상을 보였다. 이는 생성형 AI가 프로그래밍 배경지식이 부족한 학습자의 학습 격차를 보완하며, 자기효능감과 실재감을 동시에 증진시키는 역할을 수행했음을 시사한다.

추가적으로 각 집단의 변화가 가지는 교육적 함의를 검증하기 위해 효과크기(Cohen's d)를 산출하였다. 분석 결과, 파이썬 경험이 있는 집단(N=9)은 표본 수가 적어 일부 요인이 통계적 유의성 기준에 근접하거나 경계선에 머물렀지만 효과크기 측면에서는 교육적으로 중요한 변화가 확인되었다. 특히, 자기학습 효능감( $d=1.44$ )과 교수 실재감( $d=1.15$ )은 1.0을 상회하는 큰 효과크기(Large Effect Size)를 보여, 프로그래밍 기초를 가진 학습자들이 생성형 AI의 피드백을 자신의 지식과 통합하여 높은 수준의 효능감 향상을 경험했음을 시사한다. 또한, 기술 효능감( $p=.051, d=0.76$ )과 사회적 실재감( $p=.038, d=0.82$ ) 역시 중간에서 큰 효과크기를 보여, AI 기반 스캐폴딩이 숙련된 학습자에게도 심도 있는 학습 경험을 제공했음을 확인할 수 있었다.

반면, 파이썬 경험이 없는 집단(N=48)은 모든 하위 요인에서 약 0.5 수준의 중간크기를 나타냈다. 이는 생성형 AI가 프로그래밍 배경지식이 부족한 초심자에게도 안정적이고 일관된 학습 효과를 제공하는 스캐폴딩으로 기능했음을 뒷받침한다.

Table 5. Comparative Analysis of Factors by Prior Python Experience

Scale	Sub-Factors	Exp. Status	Pre M(SD)	Post M(SD)	M Diff.	t	p	Cohen's d
Self-Efficacy	Self-Learning Efficacy	Exp.	2.59 (0.49)	3.96 (0.92)	1.37	-4.33	.003	1.44
		No Exp.	3.10 (0.96)	3.76 (0.79)	0.66	-3.45	<.001	0.50
	Classroom Engagement Efficacy	Exp.	3.22 (0.69)	3.96 (0.87)	0.74	-1.99	.081	0.66
		No Exp.	3.30 (0.95)	3.95 (0.79)	0.65	-3.51	<.001	0.51
	Technical Efficacy	Exp.	3.28 (0.83)	4.22 (0.83)	0.94	-2.29	.051	0.76
		No Exp.	3.44 (0.90)	4.05 (0.67)	0.61	-3.8	<.001	0.55
Presence	Cognitive Presence	Exp.	3.30 (0.40)	4.09 (0.67)	0.79	-2.48	.038	0.83
		No Exp.	3.69 (0.58)	4.14 (0.68)	0.45	-3.21	.002	0.46
	Teaching Presence	Exp.	3.08 (0.25)	4.14 (0.75)	1.06	-3.45	.009	1.15
		No Exp.	3.71 (0.63)	3.99 (0.63)	0.28	-2.14	.038	0.31
	Social Presence	Exp.	3.09 (0.32)	3.93 (0.79)	0.84	-2.47	.038	0.82
		No Exp.	3.55 (0.64)	3.82 (0.70)	0.27	-1.99	.053	0.29

### 4.3.2 생성형 AI 사용 경험에 따른 요인별 분석

생성형 AI 사용 경험이 있는 집단(N=47)은 Table 6과 같이 모든 척도의 하위 요인에서 통계적으로 유의미한 향상(p<.01)을 보였다. 효과크기(Cohen's d)는 0.40~0.57의 범위로 나타나, 모든 영역에서 중간 수준의 고른 교육적 효과가 확인되었다.

반면, 경험이 없는 집단(N=10)은 대부분의 요인에서 유의미한 차이가 나타나지 않았으나, 자기학습 효능감에서만 유일하게 유의한 향상(p=.032)이 확인되었다. 특히 이 요인의 효과크기는 d=0.80으로 산출되어, 전체 하위 요인 가운데 가장 큰 효과를 보였다. 이는 평균 차이(M diff=1.13)가 가장 컸던 결과와 일치하며, 생성형 AI 경험이 없는 학습자들에게 자기학습 효능감 증진이 상대적으로 크게 나타났음을 시사한다.

종합하면, 생성형 AI 사용 경험이 있는 학습자들에게는 전반적인 학습 효율을 향상시키는 실질적인 도구로 기능한 반면, 기술 사용 경험이 부족한 초심자에게는 '스스로 할 수 있다'는 자기 확신을 강화하는데 결정적인 역할을 수행했음을 알 수 있다.

Table 6. Comparative Analysis of Factors by Prior Gen AI Experience

Scale	Sub-Factors	Exp. Status	Pre M(SD)	Post M(SD)	M Diff.	t	p	Cohen's d
Self-Efficacy	Self-Learning Efficacy	Exp.	3.07 (0.85)	3.76 (0.85)	0.69	-3.77	<.001	0.55
		No Exp.	2.80 (1.22)	3.93 (0.58)	1.13	-2.53	.032	0.80
	Classroom Engagement Efficacy	Exp.	3.27 (0.83)	3.94 (0.85)	0.67	-3.84	<.001	0.56
		No Exp.	3.37 (1.24)	4.00 (0.50)	0.63	-1.29	.229	0.41
	Technical Efficacy	Exp.	3.39 (0.90)	4.04 (0.74)	0.65	-3.88	<.001	0.57
		No Exp.	3.50 (0.88)	4.25 (0.42)	0.75	-2.09	.067	0.66
Presence	Cognitive Presence	Exp.	3.61 (0.60)	4.13 (0.72)	0.52	-3.53	<.001	0.51
		No Exp.	3.68 (0.45)	4.10 (0.43)	0.42	-1.78	.11	0.56
	Teaching Presence	Exp.	3.61 (0.64)	4.04 (0.67)	0.43	-3.11	.003	0.45
		No Exp.	3.65 (0.58)	3.90 (0.53)	0.25	-0.9	.393	0.23
	Social Presence	Exp.	3.44 (0.63)	3.83 (0.77)	0.39	-2.72	.009	0.40
		No Exp.	3.67 (0.55)	3.83 (0.30)	0.16	-0.74	.48	0.23

## 5. 결론

본 연구는 생성형 AI 기반 스캐폴딩 전략이 대학 파이썬 교양 필수 수업에서 학습자의 자기효능감과 실재감에 미치는 영향을

규명하고자, 국내 4년제 대학 1학년 학생 57명을 대상으로 사전-사후 대응표본 t-검정과 하위 요인별 심층 분석을 실시하였다. 분석 결과, 모든 하위 요인에서 수업 참여 후 통계적으로 유의미한 향상이 나타나 생성형 AI 기반 교육이 학습자의 심리적, 인지적 경험에 전반적으로 긍정적 영향을 미쳤음을 입증한다.

자기효능감의 세 요인(자기학습, 기술, 수업참여 효능감) 모두 p<.001 수준에서 향상되었으며, 특히 자기학습 효능감(M diff=0.77)이 가장 큰 개선 폭을 보였다. 이는 생성형 AI의 맞춤형 지원이 복잡한 개념이나 어려운 과제를 스스로 해결할 수 있다는 학습자의 확신을 가장 크게 높였음을 시사한다. 실재감 또한 인지적, 교수적, 사회적 실재감 모두에서 유의미한 향상이 나타났으며, 이 중 인지적 실재감(M diff=0.50)이 가장 큰 변화 폭을 보였다. 이는 학습자들이 AI를 지식 구성과 고차원적 사고를 촉진하는 유용한 도구로 인식하게 되었음을 의미한다.

다만, 하위 요인 간 향상 정도에는 차이가 존재하였다. 사회적 실재감(M diff=0.35)은 가장 작은 향상 폭을 보여, AI의 교육적 효과가 정서적 교감이나 사회적 만족보다는 인지적, 교수적 효율성 향상에 집중되어 있음을 보여주었다. 또한 초기 경험 유무에 따른 분석에서 생성형 AI 사용 경험이 있는 집단만이 모든 변인에서 유의미한 향상을 보여 초기 경험의 중요성을 확인하였다. 반면, 경험이 없는 집단은 자기학습 효능감에서 가장 큰 향상(M diff=1.13)을 기록하여 AI가 초심자의 자신감을 고취하는 보완적인 역할을 수행했음을 입증하였다.

본 연구는 생성형 AI를 활용한 교육이 기존 교수-학습의 한계를 넘어설 수 있다는 세 가지 교육적 시사점을 제공한다.

첫째, 인지적 자율성 강화이다. 자기학습 효능감과 인지적 실재감에서 가장 큰 향상 폭이 나타난 것은 생성형 AI가 학습을 수동적 지식 수용자에서 능동적 지식 구성자로 전환시키는 핵심 매개체로 기능했음을 보여준다. 생성형 AI는 학습자의 현재 이해 수준에 맞춰 단계적 지원을 제공함으로써 점진적 독립 학습을 촉진하였다. 둘째, 기술적 진입 장벽 완화이다. 파이썬 경험이 부족한 학습자도 모든 자기효능감 요인에서 유의미한 향상을 보였다는 점은 생성형 AI가 새로운 기술 학습에서 느끼는 부담과 불안을 완화하는 교육적 보조체로서 효과적임을 의미한다. 셋째, 경험 격차 관리의 필요성이다. 생성형 AI 경험 유무에 따라 학습 효과가 차별적으로 나타났다는 점은 AI 기반 수업 설계 시 기술 적응 단계를 반드시 포함해야 함을 시사한다. 단순히 도구를 제공하는 것만으로는 충분하지 않으며, 초기 경험이 부족한 학생을 위한 체계적 오리엔테이션과 동반될 때 수업 효과를 보편적으로 확산할 수 있다.

이러한 분석 결과를 바탕으로 생성형 AI를 교육에 통합하기 위한 실천적 제언은 다음과 같다. 첫째, AI 활용 가이드라인의 체계적 마련이 필요하다. 특히 자기학습 효능감을 극대화하기 위해 AI를 단순한 정답 제공 도구가 아니라 문

제 분석과 해결 과정의 협력자로 활용하도록 하는 명확한 교수 지침을 구축해야 한다. 둘째, 맞춤형 적응 교육 프로그램의 도입이 요구된다. 생성형 AI 사용 경험이 부족한 학생들을 위해 학기 초 2~3주간 AI 윤리, 프롬프트 엔지니어링, 효과적인 상호작용 전략 등을 포함하는 AI 리터러시 부트캠프를 운영하여 초기 진입 장벽을 완화할 필요가 있다. 셋째, 사회적 실재감 강화를 위한 상호작용 설계 개선이 필요하다. AI 응답에 대해 동료 및 교수자와 토론하도록 유도하거나 AI 피드백 스타일을 정서적 교감을 높이도록 조정함으로써 사회적 실재감의 취약성을 보완할 수 있다. 이를 통해 AI는 정보 제공자를 넘어 학습자의 발달 수준에 따라 피드백의 강도와 형태를 조절하는 스캐폴딩 도구로 기능할 수 있으며, 이는 학습자의 지속적 참여와 심층적 이해를 촉진한다.

본 연구는 단일 집단 설계로 진행된 만큼 개인적 경험이나 시간 경과가 결과에 미칠 수 있는 변수를 완전히 통제할 수 없었다는 한계를 지닌다. 이를 보완하기 위해 생성형 AI 경험과 파이썬 선수학습 경험을 조절 변인으로 설정하여 집단 간 차이를 심층 분석하였으며 이를 통해 초기 특성에 따른 교육 효과의 차별성을 규명하고 연구 타당성을 강화하고자 하였다. 향후 연구에서는 질적 면담을 병행한 혼합 연구 설계를 적용하여 생성형 AI 기반 스캐폴딩의 작동 기제를 보다 심층적으로 탐색할 것을 제안한다.

## 참고문헌

- [1] Jin, Y., Yan, L., Echeverria, V., Gašević, D., & Martinez-Maldonado, R. (2024). Generative AI in higher education: A global perspective of institutional adoption policies and guidelines. *Computers and Education: Artificial Intelligence*, 8, 100348. <https://doi.org/10.1016/j.caeai.2024.100348>
- [2] Goswami, D., & Soupez, J-B. R. G. (2024). Generative AI in engineering education. Research Gate. [https://www.researchgate.net/publication/381606695\\_Generative\\_AI\\_in\\_Engineering\\_Education](https://www.researchgate.net/publication/381606695_Generative_AI_in_Engineering_Education)
- [3] Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.
- [4] Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2), 89-100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>
- [5] Koh, Y., Park, J., & Lee, K. (2023). Development and Application of Scaffolding Based Humanoid Robot Education Program for Students with Disabilities. *Korea Association of Information Education*, 27(3), 361-374.
- [6] Lee, H. (2025). Exploratory Study on Scaffolding Strategies for Autonomous Generative AI Design Education: A Case of Learning by Doing-Based PBL Design Thinking Education. *Archives of Design Research*, 38(1), 199-220. <https://doi.org/10.15187/adr.2025.02.38.1.199>
- [7] Bandura, A. (1997). *Self-efficacy: The exercise of control*. W. H. Freeman and Company.
- [8] Al Hakeem, A. M., & Al-Ali, M. K. J. (2025). The Impact of Using Artificial Intelligence Chatbots on Student Academic Performance. *American Journal of Business Practice*, 2(9), 52-65.
- [9] Park, J. (2025). Teaching Programming with AI Tools - A Case Study for Non-Technical People. *Journal of Practical Engineering Education*, 17(1), 31-40. <https://doi.org/10.14702/JPEE.2025.031>
- [10] Kang, E. (2024). Design of ChatGPT-based Software Liberal Arts Curriculum for Non-Computer Majors. *Journal of Digital Contents Society*, 25(2), 421-429. <https://doi.org/10.9728/dcs.2024.25.2.421>
- [11] Hwang, H., & Lee, J. (2025). Designing a Programming Course Model Using AI-Based Pair Programming Techniques. *The Journal of Korean Association of Computer Education*, 28(2), 13-22. <https://doi.org/10.32431/kace.2025.28.2.002>
- [12] Lee, G. (2023). University students' change of awareness and self-directed learning competencies after experience using and applying Chat GPT. *Journal of Teaching & Learning research*, 16(3), 71-94. <https://doi.org/10.23122/kactl.2023.16.3.003>
- [13] Fakhri, M. M., Ahmar, A. S., Isma, A., Rosidah, & Fadhilatunisa, D. (2024). Exploring Generative AI Tools Frequency: Impacts on Attitude, Satisfaction, and Competency in Achieving Higher Education Learning Goals. *EduLine: Journal of Education and Learning Innovation*, 4(1). <https://doi.org/10.35877/454RI.eduline2592>
- [14] Korea Copyright Commission. (2024). [Issue Report] 2023-19 European Union (EU) Artificial Intelligence Act (AI Act) and copyright law(Copyright Trends).
- [15] Naik, I., Naik, D., & Naik, N. (2024). Chat Generative Pre-Trained Transformer (ChatGPT): Comprehending its Operational Structure, AI Techniques, Working, Features and Limitations. In 2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG) IEEE. <https://doi.org/10.1109/ICTBIG59752.2023.10456201>
- [16] Min, J., & Kim, M. (2025). A Qualitative Study on Generative AI-Based Teaching and Learning - Focusing on the experience of teachers in school -. *The Journal of Korean Association of Computer Education*, 28(8), 25-40. <https://doi.org/10.32431/kace.2025.28.8.003>
- [17] Martocchio, J. J. (1994). Effects of conceptions of ability on anxiety, self-efficacy, and learning in training. *Journal of Applied Psychology*, 79(6), 819-825. <https://doi.org/10.1037/0021-9010.79.6.819>
- [18] Eden, D., & Aviram, A. (1993). Self-efficacy training to speed reemployment: Helping people to help themselves. *Journal of Applied Psychology*, 78(3), 352-360. <https://doi.org/10.1037/0021-9010.78.3.352>
- [19] Han, J., & Kim, S. (2006). Discussion on the Research to Examine Relations Among Learner's Self-Direction, Attitudes and Self-Efficacy toward Technology in Web-based Learning. *Journal of Korean Association for*

*Educational Information and Media*, 12(4), 29-50.

- [20] Kim, M., & Park, S. (2010). Determinants and Outcomes of Self-efficacy. *GRI REVIEW*, 12(2), 5-34. <https://doi.org/10.23286/gri.2010.12.2.001>
- [21] Lee, J. (2024). The Impact of a Python Project Class Using Generative AI as an Individualized Learning Feedback Tool on High School Student's Self-Efficacy and Computational Thinking: With a Focus on Using ChatGPT. Master's Thesis, Korea National University of Education.
- [22] Kang, D., Kim, J., & Chong, H. (2011). The Structural Relationship among affective characteristics, Learning presence, Learning flow, Learning satisfaction in Distance Education. *Journal of Korean Association for Educational Information and Media*, 17(1), 33-152.
- [23] Jung, H., Yoon, N., Yoon, S., & Lee, J. (2011). Effects of Academic Self-efficacy and Learning Presence on Learning Achievement in Elementary Social Studies Class. *Research in Social Studies Education*, 18(1), 49-63.
- [24] Hahn, J. (2022). Direct and Indirect Relationships between Academic Self-efficacy, Teaching Presence, Behavior Regulation, and Perceived Achievement in a University Flipped Learning Environment. *The Journal of Learner-Centered Curriculum and Instruction*, 22(23), 587-600. <https://doi.org/10.22251/jlcci.2022.22.23.587>
- [25] Joo, M. (2025). A Study on the use of AI Chatbot feedback in argumentative writing education and learner acceptance patterns - Focusing on the development of GPTs-based customized feedback Chatbots and analysis of educational effects. *The Korean Journal of Literacy Research*, 16(4), 209-254. <https://doi.org/10.37736/KJLR.2025.08.16.4.07>
- [26] Lim, S. (2020). AI Chatbot Users' Satisfaction and Intention for Continued Use : Moderating Effects of Chatbot Type and Motivations. *The Journal of the Korea Contents Association*, 20(10), 630-640. <https://doi.org/10.5392/JKCA.2020.20.10.630>
- [27] Jung, Y. (2025). Analysis of the Effectiveness and Utilization Patterns of Mathematics AI Courseware According to Elementary School Students' Achievement Goal Orientation. Seoul National Education University.
- [28] Jang, J. (2024). An Analysis of the Relationship among Learner's Sense of Presence, Self-efficacy, and Programming Knowledge Level in an Online Programming Learning Environment Utilizing a Generative AI Chatbot. Master's Thesis, Hanyang University.
- [29] Kang, H., Lee, J., & Kim, H. (2020). A Study on Computer Programming Education Model based on Python. *Journal of Digital Contents Society*, 21(4), 693-700.
- [30] Kim, J., Chang, H., Kim, Y., & Kim, S. (2024). Study on Ways to Improve Teaching Efficiency through Redesign Python Courses in University. *JOURNAL OF PLATFORM TECHNOLOGY*, 12(6), 105-115.
- [31] Hong, H. (2021). Development of a Basic Liberal Arts Curriculum Related to Coding to Strengthen Core Competencies -Focusing on Python programming basics. *Korean Journal of General Education*, 15(4), 187-203.
- [32] Yang, D. (2024). Scaffolding Learning Strategy-Based Net-Type Competitive Activity According to the Class Explore Learners' Educational Experiences. *Korean Journal of Convergence Science*, 13(10), 171-192.



강원태

- 2007년 대구가톨릭대학교 전자계산교육(교육학 석사)
- 2022년 대구가톨릭대학교 컴퓨터소프트웨어학(박사수료)
- 2024년~현재 경북대학교 전자공학부 인공지능전공 시간강사

✚ 관심분야 : 프로그래밍, 교육과정평가, 이러닝, AI  
 ✉ wt\_kang@daum.net



이경희

- 2006년 계명대학교 컴퓨터교육전공(교육학석사)
- 2020년 제주대학교 컴퓨터교육전공(교육학박사)
- 2020년~현재 호서대학교 혁신융합학부 조교수

✚ 관심분야 : SWAI교육, 교육과정, 교수학습 이론 및 전략, 교사교육  
 ✉ dreamer@hoseo.edu