Enabling Postgraduate Projects in Computing Education through Synthetic Research Data Generation

Paul Denny
University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Kaitlin Riegel
University of Auckland
Auckland, New Zealand
kaitlin.riegel@auckland.ac.nz

Andrew Luxton-Reilly
University of Auckland
Auckland, New Zealand
andrew@cs.auckland.ac.nz

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

James Prather
Abilene Christian University
Abilene, TX, USA
james.prather@acu.edu

Stephen MacNeil
Temple University
Philadelphia, PA, USA
stephen.macneil@temple.edu

Abstract

Research projects in postgraduate computing education courses often rely on existing datasets due to the practical and ethical constraints of collecting authentic learner data within a single semester. However, finding high-quality data sets can be a considerable challenge for educators. Large language models (LLMs) offer a promising way to address this challenge by enabling the on-demand generation of synthetic data that closely resembles real learner data. This allows research students to practice key elements of the research process, including data cleaning, analysis, and reporting. This paper reports on an initiative in which 44 postgraduate computing education students across 15 groups used LLMs to generate datasets for their research projects. The resulting datasets were highly varied, including buggy code in multiple languages, code with stylistic variations, UML diagrams, and natural language programming prompts. Students valued the efficiency and scalability of this approach but, as expected, raised concerns about authenticity. Several groups also over-relied on AI for analysis, which was not intended. We discuss lessons learned and highlight the potential of synthetic data to enable authentic, scalable, and accessible research experiences in computing education.

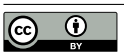
CCS Concepts

• **Social and professional topics** → **Computing education**.

Keywords

synthetic data, computing education, generative AI, large language models, LLMs, postgraduate projects, synthetic data generation

ACM Reference Format:
Paul Denny, Kaitlin Riegel, Andrew Luxton-Reilly, Juho Leinonen, James Prather, and Stephen MacNeil. 2026. Enabling Postgraduate Projects in Computing Education through Synthetic Research Data Generation. In *28th Australasian Computing Education Conference (ACE 2026), February 09–13, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3786228.3786241



ACE 2026, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2352-0/26/02
https://doi.org/10.1145/3786228.3786241

1 Introduction

Postgraduate courses in computer science often include a component of independent research, where students undertake a project requiring the formulation of research questions, the collection and analysis of data, and the written presentation of findings. In the context of computing education projects, relevant data would usually focus on various aspects of learners – their code, behaviours, or reflections – and this makes data gathering particularly challenging. Designing and conducting studies involving human participants is rarely feasible within the short timeframe of a taught course (typically one semester or term), especially given challenges such as navigating complex ethics approval processes and gaining access to sufficiently large numbers of participants.

To overcome these constraints, instructors may look to existing public datasets that their students can analyse as part of their project work. Two examples in the computing education literature are Blackbox [3] and FalconCode [6], both of which offer large-scale collections of student programming data. These resources have enabled numerous studies, but they also present several practical challenges when used in teaching contexts. Firstly, large authentic datasets can be difficult for newcomers to explore. Indeed, this was the motivation that led Brown and Kölling to later release Blackbox Mini as a simplified, more navigable subset of the full dataset [4]. Secondly, the kinds of studies that students can explore are limited by the public datasets available. For instance, both the FalconCode and Blackbox datasets contain code snapshots, supporting a narrow range of project types, which focus on code analysis and are tied to specific programming languages.

A third and significant challenge is the persistence of high-quality datasets. The FalconCode project represents a fascinating case study. The FalconCode dataset provides over five semesters of student code samples from the United States Air Force Academy's introductory programming course [6]. The authors released it to address the scarcity of datasets containing student code for computing education research, and to "lower the barrier to entry for CS education research." Although it was first published in the 2023 SIGCSE Technical Symposium proceedings, at the time of writing, the original public link to FalconCode is inactive. This illustrates the fragile nature of such valuable resources.

More recently, large language models (LLMs) appear to offer an exciting possibility: the on-demand generation of synthetic datasets. This idea began to gain momentum in 2023, when Hämäläinen et al.

demonstrated that LLMs could generate synthetic user research data that was both plausible and difficult for humans to distinguish from real responses [12]. Additional research across multiple domains has shown that LLM-generated data can approximate authentic distributions while avoiding the logistical and ethical barriers associated with human data collection [19, 29, 34]. Within computing education specifically, very recent work has focused on whether LLMs can generate student-like programming errors, with studies by MacNeil et al. [24] and Leinonen et al. [18] showing that synthetic buggy code can replicate key features of authentic student submissions. These promising early results suggest that synthetic data could be extended beyond debugging contexts to support a wider variety of research questions.

Building on these developments, we saw an opportunity to explore how synthetic data might support postgraduate research projects in computing education. If LLMs can generate realistic learner-like data on demand, then students could undertake research projects aligned with their interests that would otherwise be infeasible within a single semester. In this pedagogical research-training context, synthetic data is used to support key aspects of the research process – such as research design, data cleaning, analysis, and reporting – but not to substitute for authentic learner data in drawing empirical conclusions about real learners. This approach raises intriguing pedagogical and psychological questions. Would students feel equally motivated to work with data they knew to be synthetic, or would the fact it is artificial reduce their interest? Similar debates are currently playing out in the entertainment industry around the use of AI-generated media [26, 30]. We hypothesised that perceptions might depend on how realistic the data appeared. We were also interested in students' views on the ethical implications of using AI-generated data, including issues of transparency and bias, and whether engaging directly with data generation tools might deepen their understanding of generative AI as a research methodology – a valuable outcome in a computing education research course.

We report our experience integrating synthetic research data generation into a postgraduate computing education course. Our aim is to examine whether and how synthetic data generation can act as a pedagogical scaffold for postgraduate research. Forty-four students (15 groups) were provided with OpenAI API keys and guided in prompt engineering to generate datasets tailored to their project topics. We analysed both the student reports (to identify the types of projects made possible by synthetic data) and student survey responses (to capture perceptions of advantages, limitations, and learning experiences). We make the following contributions:

(1) We provide the first account of how synthetic data generation can be embedded into a taught postgraduate research course, documenting both the range of projects students pursued and the kinds of datasets they created.
(2) We report student perceptions of working with synthetic data, highlighting both enthusiasm for its efficiency and scalability, and concerns around authenticity, bias, and over-reliance on AI.
(3) We provide a rich reflection on the experience for the instructors in the course, outlining several key lessons for other instructors interested in using this approach in the future.

2 Related work

2.1 Synthetic Data in Computing Education

There is growing recognition of the potential for large language models (LLMs) to supplement traditional pedagogical approaches in computing courses [11, 27, 28]. Denny et al. discuss how generative AI is transforming computing education, identifying challenges but also opportunities for new pedagogies that teach students to prompt, specify, and critically evaluate AI-generated code [8].

While most of the attention has been focused on how LLMs can enhance undergraduate teaching and learning [13, 23], one under-explored area is their potential to generate synthetic datasets to support computing education research students. This is an important area to explore – many empirical studies make use of large-scale datasets from learners but few of these datasets are accessible for other researchers to use. In fact, in a review of educational studies involving programming process data, Ihantola et al. report that open datasets are "exceedingly rare" [15]. Synthetic data may offer an approach for addressing this key challenge, especially for meeting the teaching needs of computing education research courses.

A key question is the extent to which synthetic data can mimic authentic data. Recent work has shown that LLMs can be prompted to generate code containing realistic bugs. MacNeil et al. [24] conducted one of the first comparative studies of this kind, examining whether LLMs could replicate the distribution of errors made by computing students. Their results indicated that while unguided models tend to produce unrealistic error patterns, carefully designed prompts that incorporate information about common bugs and their frequencies can guide models to generate more plausible synthetic data. Building on this work, Leinonen et al. [18] compared synthetic buggy code submissions directly against real student submissions across multiple programming languages. Their findings showed that with carefully engineered prompts, LLMs can generate incorrect solutions whose test case failure distributions resemble those of authentic student data.

Together, these studies highlight the promise of using LLMs to generate realistic student-like data. They also motivate further research into whether synthetic data can be extended beyond code-level errors to support broader forms of educational research, such as analysing student behaviour or feedback. Understanding the limitations of synthetic data, including how postgraduate researchers view the use of synthetic data for their projects, is also essential.

2.2 Synthetic Data Beyond Computing Education

The use of synthetic data has been more widely explored in other domains, motivated by issues of data scarcity and privacy. In Human-Computer Interaction, for instance, Hämäläinen et al. [12] examined GPT-3's ability to generate open-ended questionnaire responses and found that, while synthetic responses were often perceived as highly human-like, they also suffered from errors and low diversity. This highlights both the opportunities of synthetic data for ideation and piloting, and the risks of relying on it without validation.

In learning analytics, researchers have investigated synthetic data as a means of enabling collaboration and benchmarking while

respecting student privacy. For example, Dorodchi et al. [9] demonstrated that synthetic data generators can produce student record datasets that preserve key distributions and support machine learning tasks with minimal loss of accuracy. Similarly, Zhan et al. [34] conducted a comparative study of synthetic data generators applied to real student datasets, assessing statistical similarity, learning-analytics model performance, and privacy preservation, and found that several generators produced synthetic data closely aligned with real learning outcomes. Complementary work by Liu et al. [20] showed that integrating differential privacy into synthetic data generation frameworks can further strengthen protections for students, albeit often at the expense of reduced data utility.

Beyond the education domain, a broader literature on LLM-generated synthetic data has highlighted important design trade-offs. Studies in natural language processing have found that synthetic data can enhance performance in objective classification tasks such as spam detection, but is less effective for highly subjective tasks such as humour or sarcasm detection [19]. In specialised domains such as healthcare, synthetic data generated from LLMs has been shown to substantially improve downstream model performance for tasks like named entity recognition, while also mitigating privacy risks [29]. More generally, recent surveys emphasise two central requirements for effective synthetic data – faithfulness to real distributions and sufficient diversity – and propose multi-step frameworks for generation, curation, and evaluation [22].

While the use of synthetic data continues to gain traction in a variety of fields, its application within computing education remains relatively new.

3 Methods

3.1 Course Context

Our exploration of this idea took place in a postgraduate research course in computing education taught at the University of Auckland (approved by the University of Auckland Human Participants Ethics Committee #25279). The course ran for 12 weeks in Semester 1, 2025 (starting 3 March), and was assessed primarily through a group-based research project supported by individual presentations and peer review activities. A total of 60 students were enrolled in the course, and they worked on their projects in groups of three (20 groups total). In this article, we focus on the 15 groups (with 44 students; one group of two) who used synthetic datasets – we exclude 5 groups that worked on projects with real datasets. The overall aim of the course was to provide an authentic research experience, requiring students to motivate an area of study, generate or collect data, analyse the results, and communicate their findings in the form of a research paper and oral presentation.

3.2 Group Formation and Topic Selection

Students were invited to self-organise into groups of three. Once a group was formed, they could select a project topic from a set of provided areas. The areas were fairly broad in order to provide flexibility for groups to interpret and formulate the research questions they wanted to explore. The five high-level areas that were associated with synthetic data generation were:

- Programming style feedback
- Programming error detection and fixing with LLMs
- Automatically solving image-based questions
- Variation in automatically generated code structures
- Robustness of student-written prompts to LLMs

Example 1: *The transition from MATLAB to C, for me has been quite easy with only a few problems to do with int and double (setting up the variables). So far the code itself is in a similar level of difficulty compared to MATLAB with only a few extra steps that are needing to be taken. However, I find myself enjoying coding in C more at this current point in time.*

Example 2: *At first, it felt a bit intimidating because the syntax in C looks less user- friendly compared to MATLAB. For example, in C, we need to declare the type of each variable, while in MATLAB, we don't have to think about what type it is - MATLAB just figures it out. To be honest, remembering all the curly braces and semicolons in C has been a pain, especially after getting used to the cleaner MATLAB scripts.*

Figure 1: Examples shown to students to illustrate synthetic generation of open-response data in the style of a short personal reflective account of the transition between learning two programming languages. Example 1 is an authentic student response from a survey, and Example 2 is synthetic.

3.3 Project Structure and Assessment Design

The research project was structured as a sequence of staged deliverables designed to scaffold the groups through the research process. Each group completed five written submissions (four staged submissions and a final written research report that combined all of the stages), as well as two oral presentations (one early presentation that focused on the motivation for the work, and a final presentation delivered at the end of the course). We now provide a brief summary of the four staged submissions and the final report, including their respective deadlines.

1. Research Article Introduction (due 23 March). The first written task required groups to draft an introductory section of their paper. The goal of this deliverable was to motivate the chosen topic and articulate its significance within the context of computing education research. Each introduction concluded with one to three clearly defined research questions.

2. Review of Related Work (due 13 April). The second submission focused on situating the project within existing research. Students conducted a literature review and were expected to show evidence of synthesis, pulling together related work and organising the findings in a way that highlights similarities and identifies gaps.

3. Explanation of Data Source (due 4 May). The third deliverable described the method of data collection, which for most groups involved synthetic data generation. One class session, delivered on 11 April, introduced the practical aspects of synthetic data generation, including use of API keys, performance and cost comparisons of available models, and the importance of prompt design and refinement. Further details are described in Section 3.4. This session also illustrated several examples of synthetic data, including the example shown in Figure 1, which compared an authentic student response to a survey question with one generated by an LLM. For this deliverable, no analysis of the data was expected, only an account of how the data was generated and prepared for later analysis.

4. Method, Analysis, and Results (due 18 May). The final staged submission required students to describe the methods they used to analyse their data (from stage 3), report the key findings, and present results through appropriate use of figures and tables. Groups were expected to show how their analyses addressed their stated research questions.

5. Camera-ready final report (due 13 June). The final report, deliverable 5, combined all of the previous sections, and included new sections on discussion, limitations, conclusions and future work. Groups were expected to submit this deliverable as though it were the 'camera-ready' version of a research article adhering to ACM's "Proceedings of the ACM Conference" style (i.e. 'sigconf').

3.4 Synthetic Data Generation

To support the data generation process, every group was issued a unique OpenAI API key and asked to keep their keys private and to monitor usage carefully. The API keys were issued from an OpenAI account for the University of Auckland, and had access to all available models. During the introductory session on 11 April, a comparison of model costs was shown to students to highlight the extreme differences (for example, at the time, the standard rate for GPT-4.5 was US$150 per million output tokens, whereas for GPT-4o mini it was US$0.6 per million output tokens). The code shown in Listing 1, which makes a single request and then displays token usage, was shown to students as a starting point. Each group was also provided another code template, which made several API calls, including to display all models available to the API key and to repeat a series of API calls a fixed number of times. Groups were asked to keep track of their usage, and estimate costs, with the aim of keeping them below USD$50 for the semester.

Listing 1: Example provided to students illustrating basic use of the OpenAI API in Python

```
from openai import OpenAI
client = OpenAI(api_key="ab-cdef-ghi ... xyz")

completion = client.chat.completions.create(
 model="gpt-4o",
 messages=[
  {
   "role": "user",
   "content": "Please respond to the following question
        like a student..."
  }
 ]
)

print(completion.choices[0].message.content)
tokens_used = completion.usage.total_tokens
print(f"Tokens used in this request: {tokens_used}")
```

Although groups had access to all models available at the time, two model tiers were suggested: GPT-4o for higher-quality generations, and GPT-4o-mini for larger-scale, lower-cost outputs. Instructional material presented examples of how prompts could be crafted to elicit responses resembling student reflections, explanations, or programming-related tasks. For instance, students were shown how altering the phrasing of a prompt could produce more diverse outputs. One trivial example involved appending the phrase *"Please generate five varied responses to this question"* to a single API call which was much more effective at generating varied responses than making five separate API requests. Groups were also reminded that prompt engineering was very important and takes time, and that spending effort refining prompts was central to achieving useful results.

Students were also shown examples from and asked to read Leinonen et al. [18], which illustrated the generation of code containing various bugs. In particular, that paper shows examples of how modifications to a prompt can influence a model to generate more realistic outputs (i.e., providing test cases, or typical bug distributions, to aid the model in generating likely bugs).

3.4.1 Expectations and Reporting. While the topics varied across groups, the expectation was that the generated datasets would resemble authentic educational data, such as survey-style responses (including free-text responses to reflective prompts) and programming submissions (e.g., buggy code, solutions to programming tasks). Each group was required to describe their synthetic data generation process in their report (deliverable 3), including the prompts used, the rationale for the chosen approach, and any steps taken to clean the generated data.

3.4.2 Usage and Costs. Throughout the period of the project, the institution account that issued the API keys was eligible for a number of daily free tokens, with fixed limits, for different models. Specifically, at the time, 1 million tokens were available per day across gpt-4.5-preview, gpt-4.1, gpt-4o, o1, and o3. In addition, 10 million tokens per day were available for gpt-4.1-mini, gpt-4.1-nano, gpt-4o-mini, o1-mini, o3-mini, o4-mini, and codex-mini-latest. Only usage beyond these limits and any usage of other models, were billed at the standard rates. The total billed cost over the period was USD$270.

3.5 End-of-Course Survey

Student perceptions of working with synthetic data were collected through convenience sampling at the end of the semester, where all students were invited to complete an anonymous survey. The survey included five Likert items, on a five-point scale from *Strongly Disagree* (1) to *Strongly Agree* (5), with statements about authenticity, preference for real versus synthetic data, opportunities afforded by synthetic data, awareness of ethical issues, and insights gained into generative AI (see Table 1 for the questions). The survey also included the following three open-response questions inviting students to reflect on their experiences, as well as identify perceived advantages and limitations in generating suitable synthetic data for their project:

- **Item 1:** Please describe your personal experience and reflections using synthetic data for your project this year.
- **Item 2:** What advantages do you believe synthetic data generation offers for research projects in computing education?
- **Item 3:** What limitations do you believe synthetic data imposes on research projects in computing education?

A total of 38 students responded to the questionnaire, 30 of whom had worked with synthetic data in their project and were included in the analyses. Of these, five were missing at least one response to a Likert item. There was insufficient data to impute. As Little's MCAR test was non-significant, the descriptive statistics and t-tests are presented individually with the missing cases excluded. As a

methodological note on the use of a *t*-test, *t*-tests between Likert items exhibit similar power when contrasted with comparable non-parametric tests [7].

An experienced researcher engaged with the student response data and used a codebook approach to inductive thematic analysis [2, 5]. They became familiar with the data and made initial codes across the three items, which were then refined and revised after coding all responses to each prompt (responses could be assigned multiple codes). These codes and the initial themes were reviewed by another experienced researcher, who agreed with the existing codes and recommended a further four be included. Themes were finalised and grouped under broader headings for presenting the results.

3.6 Analysis of Student Work

In addition to the survey, we analysed the written reports submitted by each group, focusing on how synthetic data was generated, described, and used in their projects. Two researchers discussed relevant aspects to analyse, which was followed by one researcher collecting the relevant information from student reports. These two sources of evidence – the survey and the reports – provide insight into both the methodological practices of students when using LLMs for data generation, and their views on the authenticity and value of this approach in postgraduate research.

4 Results

4.1 Student Perceptions of Synthetic Data

We analysed both open-response reflections and Likert-scale survey data to investigate students' perceptions of working with synthetic data in the course.

Descriptive statistics of responses to the Likert items are presented in Table 1. It also includes confidence intervals of *t*-tests for average response deviation from *neutral* and corresponding effect sizes. We found that students, overall, did not clearly indicate whether synthetic data was more or less similar to real human data and that they did not have a preference for working with real human data or synthetic data. To more closely look at this finding we investigated the relationship between the two responses and found, logically, the more similar students found synthetic data to real human data, the less they expressed a preference for working with real human data (r = -.45, p = .02).

There was no evidence that students believed synthetic data allowed them greater exploration of research questions (p = .18). Interestingly, there was a reasonably prevalent degree of reported recognition regarding ethical issues around the different forms of data generation. Finally, there was a strong consensus that using synthetic data provided insight into the role of GenAI in researching.

The emergent themes from analysing the open responses are grouped under four broader headings: (1) Reception, (2) Purposes, Use, and Value, (3) Study Efficiency and Practicality, and (4) Research Integrity. The themes are discussed in the following subsections and their distributions across items appear in Figure 2.

4.1.1 Reception. Students discussed their perspectives on synthetic data collection (Figure 2→'Reception') largely in response to

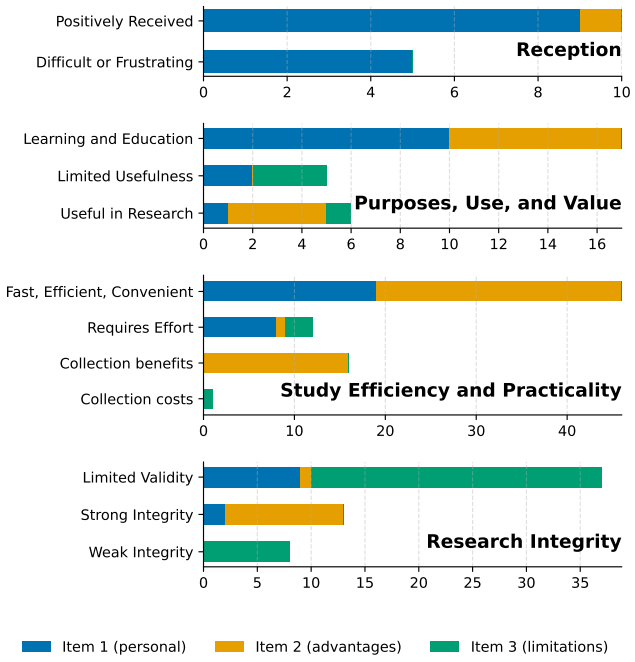


Figure 2: Distribution of emergent themes across the three open-response items. The themes are grouped under four headings: (1) Reception, (2) Purposes, Use, and Value, (3) Study Efficiency and Practicality, and (4) Research Integrity.

Item 1. *Positively Received* responses included describing the process as *'rewarding'*, *'worthwhile'*, *'interesting'*, and *'exciting'*. Though, some students noted it was difficult or involved *"some frustrations with prompting"* (*Difficult or Frustrating*). However, these responses were not always framed negatively, with students viewing the challenge as part of the process. Together, this suggests a relatively positive reception.

4.1.2 Purposes, Use, and Value. There was some conflict in the responses around the use and value of synthetic data. Students indicated it had *Limited Usefulness*.

> *We started to wonder: is it meaningful to analyze fake data to draw real-world conclusions? It felt like our work was pointless because the data itself was "wrong".*

However, others found it *Useful in Research*. In particular, when responding to **Items 2 and 3**, students indicated the usefulness of synthetic data collection, specifically in early phases of research.

> *... for performing early studies or giving researchers early confirmation if a study is worth gathering expensive real data for, I think it is a valuable tool.*

Collectively, this suggests students might not be inclined to view synthetic data as valuable beyond pilot testing.

However, as evident in Figure 2→'Purposes, Use, and Value', many students identified the important role it can play in *Learning and Education*, for example, through "*supporting targeted educational research and tool development*". Specifically, students noted, while responding to **Item 1**, its value in deepening understandings

Table 1: Descriptive statistics of responses to the Likert items and t-tests reporting confidence intervals of the deviation from neutral with effect sizes.

	N	Mean (SD)	Skew (SE)	Kurtosis (SE)	*t*	95% CI	*p*	*d*
1. The synthetic data we generated was very similar to real student/human data	27	3.41 (0.97)	-0.40 (0.45)	-0.23 (0.87)	2.18	0.02, 0.79	0.08	0.42
2. I would prefer to have worked with real student/human data rather than generating synthetic data	28	3.50 (1.11)	0.09 (0.44)	-1.30 (0.86)	2.39	0.07, 0.93	0.07	0.45
3. The synthetic data we generated allowed us to explore research questions that would not have been viable with real student/human data in this course	27	3.26 (0.98)	0.22 (0.45)	-0.92 (0.87)	1.37	-0.13, 0.65	0.18	0.26
4. I was aware of ethical issues related to human data generation and synthetic data generation	27	3.78 (1.05)	-1.03 (0.45)	0.77 (0.87)	3.85	0.36, 1.19	<0.01*	0.74
5. Working with synthetic data provided insight into the use of Generative AI tools for research purposes	27	4.07 (1.11)	-1.63 (0.45)	2.78 (0.87)	5.04	0.64, 1.51	<0.001*	0.97

Note. SD = standard deviation, SE = standard error, CI = confidence interval; Holm-Bonferroni adjusted, two-sided *p* values are reported with $\alpha = .05$.

of prompt engineering ($n = 5$), the research topic ($n = 4$), and OpenAI API and keys ($n = 3$). Students' responses to **Item 2** reinforce these reflections where they explicitly raise its suitability for educational purposes.

4.1.3 Study Efficiency and Practicality. Figure 2→'Study Efficiency and Practicality' outlines the four themes encompassed by this heading. Students reflected extensively on synthetic data being *Fast, Efficient, Convenient.*

> *Synthetic data allows researchers to quickly build structured datasets without needing access to real student work.*

In contrast, a number of students highlighted that the process *Requires Effort*, in particular, attending to the significant effort in prompt engineering. Students in **Item 1** outlined the explorative, precise, and iterative requirements of the process, describing that *"it required a lot of careful tuning and iteration"*.

There were comments on the need to carefully understand the scope of the research and relationship with the data generation, as well as the need to spend time cleaning, supplementing, and validating data.

> *So even though you can generate a lot of data, you still need to spend time cleaning it and maybe manually supplementing it to make it more realistic.*

However, perceived effort was not necessarily discussed negatively.

Several students identified the *Collection Benefits* when using synthetic data, specifically, (1) avoiding privacy and ethical considerations ($n = 9$), (2) cost effectiveness ($n = 3$), and (3) database size ($n = 4$). Only one student noted the *Collection Costs* with respect to resourcing. Taken together, the data could suggest that the efficiency benefits of synthetic data were generally seen to outweigh its drawbacks, especially in time-constrained research contexts where navigating ethics approvals, recruiting participants, and handling sensitive human data can be challenging.

4.1.4 Research Integrity. Connected largely to students' views of the validity and reliability of synthetic data, Figure 2→'Research Integrity' presents themes related to their perceived integrity of research involving its usage. It is clear that students view the data as having *Limited Validity.*

> *A major concern was whether this data truly reflected real-world distributions – since we controlled the entire generation process, it sometimes felt like being both the player and the referee, raising questions about objectivity and generalizability.*

In favour of the research having *Strong Integrity*, two students responded to **Item 1** indicating they found synthetic data realistic, while eleven discussed in **Item 2** the benefits of data controllability or reproducibility.

> *Synthetic data generation offers scalability and control ... while enabling reproducible experiments.*

Finally, a few students considered the research to have *Weak Integrity*. Students referenced the fact that different models have different outputs ($n = 3$) and the difference in data complexity with "real world" data ($n = 5$).

> *LLMs are imperfect and often too uniform, at least compared to the enormously chaotic machinations of a human mind.*

Overall, these reflections reveal a clear awareness of the validity limitations of synthetic data, but also an appreciation of its scalability and reproducibility.

4.2 Project Types and Synthetic Datasets

A summary of each group's synthetic dataset is provided in Table 2. Across the 15 groups, the projects demonstrated substantial diversity in both the design and scope of the synthetic datasets. First, many examined LLM capabilities in programming tasks, including code generation, debugging, and explanation (Groups A, C, E, F, G, I, K, Q), often testing how factors such as prompt style or temperature (which controls diversity of LLM outputs) affected correctness or pedagogical quality. Second, several projects investigated ways to evaluate or enhance LLM feedback (Groups B, H, N), for example by incorporating static-analysis tools (Group B and N). Third, projects examined synthetic prompts (Groups M and S), for example how manipulating the background profiles of simulated students affected the prompts generated by those "students". Finally, two projects explored more visual data (Groups D, J), specifically UML diagrams (Group D) and weighted graphs (Group J).

Dataset sizes ranged from 100 to 40000 synthetic outputs. However, only one team (Group H) provided a formal rationale for dataset size through a statistical power analysis. Other groups typically determined the amount of data pragmatically, mostly based on perceived sufficiency. In several cases, students appeared to equate

Group	Goal/task	Dataset size	Model(s) used	Artefact(s) created	Validation	Human-in-the-loop
A	Assess LLM code robustness across prompt styles and temperatures	1.5k prompt–code pairs	Unclear	Python code outputs for 25 CS1 tasks	Yes	Yes
B	Study impact of static analysis and filtering on LLM feedback quality	3.5k code snippets	GPT-4o, GPT-4o-mini	Defective code + model-generated feedback	Yes	Yes
C	Evaluate LLM debugging accuracy across languages	2.8k buggy samples + fixes	GPT-4o-mini	Buggy code + fixes/explanations (Python, Java, C, JavaScript)	Yes	Yes
D	Assess GPT-4o-mini's ability to interpret UML diagrams	971 UML diagrams	GPT-4o-mini	UML diagrams with textual descriptions	Yes	No
E	Study how prompt strategies affect code structure and complexity	40k code snippets	GPT-4o-mini (code solutions), GPT-4.1 (programming problems)	Python solutions to 10 intro problems	Yes	No
F	Develop multi-agent RAG framework for error correction	11k erroneous code solutions (varied n reported)	GPT-4.1-nano	Erroneous code solutions to MBPP benchmark	Yes	No
G	Examine prompt effectiveness for code generation	1.2k prompt–code pairs	GPT-4o-mini	Synthetic student prompts	Yes	Yes
H	Analyze LLM feedback on variable naming	100 code + 200 feedback samples	GPT-4o-mini	Code samples with varied variable naming + LLM feedback	Yes	No
I	Assess LLM explanations of code with semantic errors	2.5k code samples + 2.5k explanations + 500 evaluations of explanations	GPT-4o-mini	Erroneous code examples with fixes, explanations, and simulated student ratings	No	No
J	Evaluate LLM solving of Dijkstra's shortest path problem	900 weighted graphs	Randomized script	Weighted directed graphs with images, Q–A, and metadata	Yes	No
K	Study variation and correctness in LLM code completion	1.8k LeetCode solutions	GPT-4o-mini	Code for easy LeetCode programming problems	No	No
M	Examine prompt robustness vs. student performance	450 synthetic prompts	GPT-4o-mini	Synthetic prompts with background profiles and code-generation responses	Yes	No
N	Compare GPT and static analysis feedback on code style	1.5k code solutions	GPT-4o	Beginner Python code with style issues	No	No
Q	Measure diversity of LLM-generated code	100 code solutions	ChatGPT, Claude, Gemini, LLaMA, DeepSeek	LeetCode solutions from 5 LLMs	No	No
S	Analyze how student level affects prompt quality	5k prompt–response pairs	GPT-4o-mini	Prompts by 5 education levels + Fibonacci code	No	No

Table 2: Descriptions of the datasets students produced. Goal/task describes the high-level research objective of the group. Dataset size outlines the number of outputs produced. Model(s) used describes which LLMs were used in data generation. Artefact(s) created highlights the types of artefacts included in the synthetic dataset. Validation signals whether LLM outputs were validated in any way (either automatically or manually). Human-in-the-loop shows if humans were included in the data generation process, e.g. by having them validate outputs.

larger datasets with higher quality, even when this was not warranted by the research question or task complexity. Groups N and S, for instance, each generated thousands of samples but did so for only single, relatively simple programming problems, yielding a high volume of data but with limited diversity or analytical value. This tendency to "go big" contrasts sharply with Group H, which conducted a statistical power analysis to justify a much smaller dataset of 100 outputs.

Although students were instructed to use LLMs for dataset generation only (for deliverable 3, which was independent of the methods for data analysis in deliverable 4; see Section 3.3), a striking finding was that many groups extended automation to the analysis stage. Several teams built end-to-end pipelines where the LLM not only generated but also validated and analyzed the data with minimal human oversight. For example, Group I created a large-scale simulated environment of 2500 student-like code submissions containing semantic errors and built an automated pipeline that generated, explained, and statistically evaluated each case using GPT-4o-mini. Their system simulated student ratings across six pedagogical dimensions, computed correlations between linguistic metrics and helpfulness, and produced visual analyses of LLM performance – entirely without human raters. Similarly, Group E used GPT-4o-mini to create 40000 code solutions to introductory problems and then filtered and analyzed them through additional LLM-driven scripts to quantify structural diversity. The minority (4/15) of groups that included a 'human-in-the-loop' in the synthetic data generation mainly used humans to verify the quality of the generated data. For three groups, Groups A, B, and C, who used a human-in-the-loop approach, a small subset of the generated data was validated by humans. Group G manually reviewed all generated data. However, while few groups had a human-in-the-loop, most (10/15) groups included some validation of the generated data, but this was often automatic. For example, some groups checked that the produced code compiles (e.g., Groups C and D), while others validated that unit tests fail when generating buggy code (e.g., Group F) or pass when generating code that is supposed to be correct (e.g., Groups E and M).

Model selection was relatively uniform: almost all groups employed GPT-4o-mini or GPT-4o. This is unsurprising since these models were specifically mentioned in the lectures, and students

were provided API keys for OpenAI model use. Generation methods ranged from simple parameterized prompting to multi-stage workflows combining generation, filtering, and evaluation steps.

Overall, the projects illustrate both the opportunities and methodological challenges of using generative AI for data creation. Students demonstrated considerable creativity in constructing automated pipelines and leveraging LLMs beyond their original task. The widespread tendency to automate the entire workflow from generation to analysis suggests that students increasingly perceive LLMs not merely as content generators but as comprehensive research assistants capable of conducting end-to-end studies.

5 Reflections and Lessons Learned

5.1 Project and Data Design

5.1.1 Project Topics. When designing the project specification, we listed five fairly broad areas from which students could select their topic (see Section 3.2). This allowed us to prepare some supporting material around each area, such as a small set of relevant papers that groups could use as a starting point for their literature reviews. We also selected areas where there had been some recent interest in the literature. In hindsight however, this may have overly limited the types of data students generated and hence, the research questions they could explore. Given the flexibility of synthetic data generation, in future iterations of the course we will provide greater autonomy to students around topic selection. Some amount of scaffolding is still needed – even with the prescribed topic areas in our current study, some groups found navigating this required significant effort (e.g., *"the freedom to explore many directions often led to a lack of focus and an overly broad initial scope"*).

5.1.2 Controlling Data Design. Although in the current work all of the data was generated by the student groups, there may be value in exploring instructor-generated synthetic data as well. In previous iterations of the course, we invested significant effort in sourcing high-quality datasets that students could select for their group projects. In contrast, it may be less effort for us to create a set of prompt-based 'generators' that could be used to produce novel datasets. These could be designed to intentionally include known structures, patterns, or biases for students to uncover. A benefit of this approach is that it could lead to more consistent data quality across groups, making assessment more equitable. Nevertheless, designing and generating their own datasets enables students to develop a range of practical research and data-handling skills.

5.2 Research Practice and Skill Development

5.2.1 Developing Technical Skills and Deepening Learning. We found that students noted and appreciated learning relevant skills related to interacting with LLM APIs. Specifically, we found students reported deepening their understanding of prompt engineering, of how to work with OpenAI APIs and keys, and of their research topic through the process of data generation (e.g., *"using this method allows us to better understand and write prompts, learn how to use large language models more effectively, and call APIs"*). This demonstrates the suitability of using this process in education from a student perspective. Moreover, managing API keys, monitoring token usage, and exploring various prompt refinement and engineering strategies (e.g., zero shot, single-shot, multi-shot, chain of thought, prompt templating, role-based prompts) are all relevant skills in modern research and industry practice.

5.2.2 API Costs. While students noted the comparative cost benefits of this approach to research, the cost was not insignificant. The billed cost for running these projects was USD$270, and were it not for the daily allowance of free tokens on some models this cost would be considerably higher. We were fortunate that the institutional account we used to generate the API keys had previously been gifted sufficient credits from OpenAI to cover the bills.

Without financial support, it would be important to take deliberate steps to mitigate costs. One option would be to configure API keys to only permit usage of the most cost-efficient models. Another option would be to explore the use of open source small language models (SLMs), which can run on local machines with appropriate hardware support. A growing body of work suggests that SLMs can approach and even surpass the performance of large proprietary models in educationally relevant tasks such as answering student questions [21, 32], generating feedback on code [16, 25], and analysing student reviews of courses and teaching staff [14].

5.2.3 Exposure to the Research Lifecycle. Perhaps the most important pedagogical benefit of the approach is that it enables students to experience the entire process of conducting research within a single semester. One commonly-used model (from the LMA Research Data Management Working Group) outlines the following phases of the research lifecycle[1]: plan and design, collect and create, analyse and collaborate, evaluate and archive, share and disseminate, and access and reuse. By using synthetic data, students can engage with every phase of the lifecycle with the exception of *access and reuse*, which relates to published work. In other words, synthetic-data projects allow students to *prototype* a full research study from inception to dissemination (with the caveat that the data collection phase is replaced by a data-generation phase). This provides students with training in the research lifecycle when it might otherwise be infeasible, readying them to engage with real data collection – when time and resources allow – in their future work.

5.3 Over-Reliance on LLMs for Data Analysis

One lesson we learned is that it is important to set clear boundaries to help students distinguish between when generative AI should be used and when human evaluation is necessary. Our intention was that students would use the API for the data generation aspect (deliverable 3) of the project only, and then employ other techniques (either custom scripts for large scale analysis or manual thematic coding of open response data) for the data analysis task (deliverable 4). In practice, not only did many projects use the API to perform the analysis of the generated data, but few projects utilised any human oversight of the data that was generated. For example, Group S, who studied how students' educational level (primary school, middle school, senior high school, undergraduate, postgraduate) affects their prompts, constructed rubrics to evaluate the prompts. However, instead of using the rubrics themselves for evaluation, they had LLM-simulated 'students' rate prompts using the rubrics.

[1] https://researchsupport.harvard.edu/research-lifecycle

As another example, Group D examined how multimodal models describe UML diagrams. They used ChatGPT-4o-mini to first generate a set of UML task descriptions, and then from those descriptions, corresponding UML diagrams (by generating PlantUML source code). They then asked the same model to produce textual descriptions from each of those diagrams. Rather than manually assessing the accuracy of these descriptions, the group compared the two model-generated texts using an LLM, which rated their similarity, accuracy, and relevance. Because the same family of models produced and evaluated the data, any shared systematic bias in description quality would be reinforced rather than revealed, and at no point did the group conduct human validity checks.

In the end, only four groups manually validated model outputs. Promisingly, survey responses from these groups illustrated the importance of manual validation (e.g. from a member of Group C: *"Make sure to leave enough time to clean and validate the data, because a lot of the generated samples will have issues"*). As a result of this lesson, we will make expectations around manual validation and analysis more explicit in future iterations of the course.

5.4 Limitations Related to Authenticity, Ethics, Engagement, and Bias

5.4.1 Reduced Engagement and Authenticity. Though the thematic analysis indicated working with synthetic data was, generally, positively received (if difficult), some students felt unenthusiastic about analysing artificially generated data. We found evidence that the perceived preference for working with real human data was negatively related to how authentic students perceived synthetic data to be. This can present a potential engagement and motivation problem for students who do not perceive their work to be meaningful, in particular, beyond learning or pilot testing (e.g. *"Sometimes it just feels pointless"*).

For postgraduate researchers, who often see themselves as emerging scholars, the absence of a real human connection to the data may reduce the sense of doing authentic and meaningful work. Recent work by Farangi et al. has shown that researchers have mixed emotions when using generative AI for research, but most acutely feel a loss of creativity, which seems to align with our results [10].

5.4.2 Loss of Human Research Opportunities. Although not a central focus of this work, it is worth noting some secondary implications of synthetic data use for human research participation. Small-scale educational studies often provide students with opportunities to act as research participants, gaining valuable experience and contributing to a sense of community. In fact, two of our student groups (not analysed as part of this synthetic data generation study) used an online tool to collect 'real' data from their peers in the course. Such projects highlight the collaborative and community-oriented aspects of educational research which are diminished when synthetic data is used instead.

Participation in human studies is sometimes accompanied by a small compensation. The total cost of running the synthetic-data projects in our course was a little over USD$250. Assuming a typical reimbursement rate of around USD$25 per person, this equates to roughly ten human participants, enough to support a modest authentic study. Thus synthetic data generation, while convenient and cost effective, redistributes resources away from direct human engagement. While students reported an awareness of ethical issues in different forms of data collection, these did not appear among their concerns so, perhaps, should be discussed in future.

5.4.3 Bias and Homogenisation. Concerns around reliability and bias using synthetic data emerged as a common theme among student groups, with the suggestion it "*could introduce biases depending on how the model is trained and may fail to reflect the misconceptions that cause real student mistakes*". Another group felt: *"relying too much on synthetic data may introduce biases or oversimplify patterns"*. Prior work has shown that LLMs can encode biases originating from their training data and from reinforcement learning from human feedback [17]. Empirical evaluations of LLMs acting as educational 'teachers' further demonstrate variation in how models tailor content across demographic groups [31]. Understanding how such biases actually impact synthetic educational data and how they may shape student interpretations remains an important direction for future research.

Another risk is the generation of large volumes of data with limited diversity, leading to homogenisation. We found that some groups generated very large datasets, but for very narrow tasks (such as Groups N and S, who generated thousands of solutions to a single very simple programming task). This resulted in a large quantity of data, but with limited diversity and analytical value (as one student said: *"it didn't output varied synthetic code - all code looked the same, or some of the code were too short"*). We see a related risk that relying on synthetic data could narrow the diversity of ideas explored by students – exactly the opposite of what we hope to achieve. Recent work has shown that generative AI can make users more efficient yet less diverse in what they produce, as models tend to homogenise ideas across individuals [1, 12, 33]. This underscores the need to help students remain critical and creative when working with AI.

6 Conclusion

In this paper, we reported our experience using synthetic data as a pedagogical scaffold for postgraduate research projects in computing education. This approach helps address the challenge that educators face in providing access to a variety of high-quality datasets within a limited timeframe. We investigated students' views on synthetic data generation, and we analysed their written research reports, revealing substantial variety in the datasets produced. Students valued the efficiency and scalability of synthetic data generation and appreciated learning to use AI as a research tool. However, concerns about authenticity sometimes reduced engagement, and some groups relied too heavily on LLMs for both data generation and analysis, limiting opportunities for human validation and evaluation. We outlined several lessons that will guide future iterations of this approach and see considerable potential for synthetic data to broaden access to authentic and scalable research experiences in computing education.

Acknowledgments

This work was supported by the Research Council of Finland grant #356114.

References

[1] Dhruv Agarwal, Mor Naaman, and Aditya Vashistha. 2025. AI Suggestions Homogenize Writing Toward Western Styles and Diminish Cultural Nuances. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1117, 21 pages. doi:10.1145/3706598.3713564

[2] Virginia Braun and Victoria Clarke. 2022. Conceptual and design thinking for thematic analysis. *Qualitative psychology* 9, 1 (2022), 3.

[3] Neil C. C. Brown, Amjad Altadmri, Sue Sentance, and Michael Kölling. 2018. Blackbox, Five Years On: An Evaluation of a Large-scale Programming Data Collection Project. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Espoo, Finland) *(ICER '18)*. Association for Computing Machinery, New York, NY, USA, 196–204. doi:10.1145/3230977.3230991

[4] Neil C. C. Brown and Michael Kölling. 2020. Blackbox Mini - Getting Started With Blackbox Data Analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 1387. doi:10.1145/3328778.3367006

[5] Victoria Clarke and Virginia Braun. 2014. Thematic analysis. In *Encyclopedia of critical psychology*. Springer, 1947–1952.

[6] Adrian de Freitas, Joel Coffman, Michelle de Freitas, Justin Wilson, and Troy Weingart. 2023. FalconCode: A Multiyear Dataset of Python Code Samples from an Introductory Computer Science Course. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 938–944. doi:10.1145/3545945.3569822

[7] Joost FC de Winter and Dimitra Dodou. 2010. Five-point likert items: t test versus Mann-Whitney-Wilcoxon (Addendum added October 2012). *Practical Assessment, Research, and Evaluation* 15, 1 (2010).

[8] Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing Education in the Era of Generative AI. *Commun. ACM* 67, 2 (Jan. 2024), 56–67. doi:10.1145/3624720

[9] Mohsen Dorodchi, Erfan Al-Hossami, Aileen Benedict, and Elise Demeter. 2019. Using Synthetic Data Generators to Promote Open Science in Higher Education Learning Analytics. In *2019 IEEE International Conference on Big Data (Big Data)*. 4672–4675. doi:10.1109/BigData47090.2019.9006475

[10] Mohamad Reza Farangi, Hassan Nejadghanbar, and Guangwei Hu. 2025. Use of generative AI in research: ethical considerations and emotional experiences. *Ethics & Behavior* 35, 7 (2025), 527–543.

[11] Diana Franklin, Paul Denny, David A. Gonzalez-Maldonado, and Minh Tran. 2025. *Generative AI in Computer Science Education: Challenges and Opportunities*. Cambridge University Press.

[12] Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 433, 19 pages. doi:10.1145/3544548.3580688

[13] Mohammed Hassan, Yuxuan Chen, Paul Denny, and Craig Zilles. 2025. On Teaching Novices Computational Thinking by Utilizing Large Language Models Within Assessments. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 471–477. doi:10.1145/3641554.3701906

[14] Yan Cathy Hua, Paul Denny, Jörg Wicker, and Katerina Taškova. 2025. Data-Efficient Adaptation and a Novel Evaluation Method for Aspect-based Sentiment Analysis. arXiv:2511.03034 [cs.CL] https://arxiv.org/abs/2511.03034

[15] Petri Ihantola, Arto Vihavainen, Alireza Ahadi, Matthew Butler, Jürgen Börstler, Stephen H. Edwards, Essi Isohanni, Ari Korhonen, Andrew Petersen, Kelly Rivers, Miguel Ángel Rubio, Judy Sheard, Bronius Skupas, Jaime Spacco, Claudia Szabo, and Daniel Toll. 2015. Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies. In *Proceedings of the 2015 ITiCSE on Working Group Reports* (Vilnius, Lithuania) *(ITICSE-WGR '15)*. Association for Computing Machinery, New York, NY, USA, 41–63. doi:10.1145/2858796.2858798

[16] Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024. Open Source Language Models Can Provide Feedback: Evaluating LLMs' Ability to Help Students Using GPT-4-As-A-Judge. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 52–58. doi:10.1145/3649217.3653612

[17] Jinsook Lee, Yann Hicke, Renzhe Yu, Christopher Brooks, and René F. Kizilcec. 2024. The life cycle of large language models in education: A framework for understanding sources of bias. *British Journal of Educational Technology* 55, 5 (2024), 1982–2002. doi:10.1111/bjet.13505

[18] Juho Leinonen, Paul Denny, Olli Kiljunen, Stephen MacNeil, Sami Sarsa, and Arto Hellas. 2025. LLM-itation is the Sincerest Form of Data: Generating Synthetic Buggy Code Submissions for Computing Education. In *Proceedings of the 27th Australasian Computing Education Conference (ACE '25)*. Association for Computing Machinery, New York, NY, USA, 56–63. doi:10.1145/3716640.3716647

[19] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations. arXiv:2310.07849 [cs.CL] https://arxiv.org/abs/2310.07849

[20] Qinyi Liu, Ronas Shakya, Jelena Jovanovic, Mohammad Khalil, and Javier Hoz-Ruiz. 2025. Ensuring privacy through synthetic data generation in education. *British journal of educational technology* 56, 3 (2025), 1053–1073.

[21] Suqing Liu, Zezhu Yu, Feiran Huang, Yousef Bulbulia, Andreas Bergen, and Michael Liut. 2024. Can Small Language Models With Retrieval-Augmented Generation Replace Large Language Models When Learning Computer Science?. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 388–393. doi:10.1145/3649217.3653554

[22] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey. arXiv:2406.15126 [cs.CL] https://arxiv.org/abs/2406.15126

[23] Wenhan Lyu, Yimeng Wang, Tingting (Rachel) Chung, Yifan Sun, and Yixuan Zhang. 2024. Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale* (Atlanta, GA, USA) *(L@S '24)*. Association for Computing Machinery, New York, NY, USA, 63–74. doi:10.1145/3657604.3662036

[24] Stephen MacNeil, Magdalena Rogalska, Juho Leinonen, Paul Denny, Arto Hellas, and Xandria Crosland. 2024. Synthetic Students: A Comparative Study of Bug Distribution Between Large Language Models and Computing Students. In *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1* (Virtual Event, NC, USA) *(SIGCSE Virtual 2024)*. Association for Computing Machinery, New York, NY, USA, 137–143. doi:10.1145/3649165.3690100

[25] Victor-Alexandru Padurean, Tung Phung, Nachiket Kotalwar, Michael Liut, Juho Leinonen, Paul Denny, and Adish Singla. 2025. Humanizing Automated Programming Feedback: Fine-Tuning Generative Models with Student-Written Feedback. In *Proceedings of the 18th International Conference on Educational Data Mining*, Caitlin Mills, Giora Alexandron, Davide Taibi, Giosuè Lo Bosco, and Luc Paquette (Eds.). International Educational Data Mining Society, Palermo, Italy, 434–441. doi:10.5281/zenodo.15870290

[26] Jeongeun Park, Changhoon Oh, and Ha Young Kim. 2024. AI vs. human-generated content and accounts on Instagram: User preferences, evaluations, and ethical considerations. *Technology in Society* 79 (2024), 102705. doi:10.1016/j.techsoc.2024.102705

[27] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 108–159. doi:10.1145/3623762.3633499

[28] James Prather, Juho Leinonen, Natalie Kiesler, Jamie Gorson Benario, Sam Lau, Stephen MacNeil, Narges Norouzi, Simone Opel, Vee Pettit, Leo Porter, Brent N. Reeves, Jaromir Savelka, IV Smith, David H., Sven Strickroth, and Daniel Zingaro. 2025. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE 2024)*. ACM, New York, NY, USA, 300–338. doi:10.1145/3689187.3709614

[29] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does Synthetic Data Generation of LLMs Help Clinical Text Mining? arXiv:2303.04360 [cs.CL] https://arxiv.org/abs/2303.04360

[30] Yiluo Wei and Gareth Tyson. 2024. Understanding the Impact of AI-Generated Content on Social Media: The Pixiv Case. In *Proc. of the 32nd ACM International Conference on Multimedia (MM '24)*. ACM, New York, NY, USA, 6813–6822. doi:10.1145/3664647.3680631

[31] Iain Weissburg, Sathvika Anand, Sharon Levy, and Haewon Jeong. 2025. LLMs are Biased Teachers: Evaluating LLM Bias in Personalized Education. arXiv:2410.14012 [cs.CL] https://arxiv.org/abs/2410.14012

[32] Zezhu Yu, Suqing Liu, Paul Denny, Andreas Bergen, and Michael Liut. 2025. Integrating Small Language Models with Retrieval-Augmented Generation in Computing Education: Key Takeaways, Setup, and Practical Insights. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh, PA, USA) *(SIGCSETS 2025)*. Association for Computing Machinery, New York, NY, USA, 1302–1308. doi:10.1145/3641554.3701844

[33] Cynthia Zastudil, Christine Holyfield, Christine Kapp, Kate Hamilton, Kriti Baru, Liam Newsam, June A. Smith, and Stephen MacNeil. 2025. Helping or Homogenizing? GenAI as a Design Partner to Pre-Service SLPs for Just-in-Time Programming of AAC. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '25)*. Association for Computing Machinery, New York, NY, USA, Article 47, 18 pages. doi:10.1145/3663547.3746384

[34] Chen Zhan, Oscar Blessed Deho, Xuwei Zhang, Srecko Joksimovic, and Maarten de Laat. 2023. Synthetic data generator for student data serving learning analytics: A comparative study. *Learning Letters* 1 (2023), 5–5.