

AIMC 2024 (09/09 - 11/09)

Designing an AI-creativity music course

Matthew Yee-King Mark d'Inverno Andrea Fiorucci

Published on: Aug 29, 2024

URL: <https://aimc2024.pubpub.org/pub/eqrzla34>

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Designing an AI-creativity music course

This article describes and evaluates an online generative AI course. The course is based on three AI models combined into a pop song-generating system. A fine-tuned GPT-2 model writes lyrics, Music-VAE composes musical scores and instrumentation and DiffSinger synthesises a singing voice. We explain the decisions made in designing the course, which is based on Piagetian constructivist ‘learning-by-doing’. We present details of the five-week course design with learning objectives, technical concepts, and creative and technical activities. We explain how we overcame technical challenges to build a complete pop song generator system consisting of Python scripts, pre-trained models, and Javascript code, all of which runs in a dockerised Linux container via a web-based IDE. A quantitative analysis of student activity provides evidence of engagement and a benchmark for future improvements. A qualitative analysis of a workshop with experts validated the overall course design, and it suggested the need for a stronger creative brief and much clearer and detailed ethical and legal content.

Author Keywords

gen-AI, AI-music, pedagogy

1. Introduction

Although some of AI’s ethical, legal and cultural limitations are increasingly being recognised, we still find ourselves in a boom time for artificial intelligence (AI). As we write, the whole educational system is working out how to deal with chatGPT¹. In addition, there is sustained high-profile coverage in the media of the latest advances in applications of AI to creative or artistic content generation. For example, we hear of OpenAI’s jukebox creating lost Frank Sinatra songs², visual artists suing Stability AI³ and AI finishing Beethoven’s unfinished symphony⁴. Along with these remarkable achievements, creative-domain AI technology is rapidly becoming more accessible, and this is already transforming a range of practices in the creative industries⁵.

We see fantastic opportunities for this technology from our perspective as experienced musicians who use AI in their creative practice and educators who have taught AI to diverse groups of students on-campus and online for several decades. Through our experiences, we see the potential to design and deliver new courses of all kinds, such as MOOCs, online, on-campus, and hybrid degrees, that use state-of-the-art AI technology to allow students to develop new creative and technical practices.

Given the multi-disciplinary nature of using AI in creative domains, it is necessary to consider course designs carefully and evaluate and iterate on those designs while taking diverse stakeholder perspectives (from research and industry) into consideration. We use the phrase ‘AI-creativity’ to refer to AI systems operating in creative domains and practices. We introduce it in order to be as inclusive as possible, encompassing views where AI can be independently creative, where AI is seen as a creative collaborator and or those that see AI as one more tool for creative expression.

Here we report findings from the development, delivery, and preliminary evaluation of a new AI-creativity online course and make the following contributions:

1. A detailed description of a new online AI-creativity course and explanation of decisions made throughout its design process;
2. A re-usable method for the quantitative analysis of student learning activity within the course and results for 238 students who have taken the course; and
3. An accompanying qualitative approach enabling the analysis of workshops involving pedagogical and professional experts to ensure our learning outcomes are the right ones, the course design is appropriate, and the learning outcomes are being met.

With these contributions, we provide a framework within which future course designers and practitioners of AI-creativity can design, implement and evaluate their materials and the student experience. Through this work, future learners will be better prepared with the creative and technical skills to effectively exploit the transformative potentials of AI in our rapidly changing creative and related industries.

2. Background

2.1 Challenges and opportunities for teaching AI in creative contexts

Technical subjects like AI are hard to teach, especially when considering a broader student body of non-science students and end-users with varied backgrounds and different mindsets [1]. Educators commonly employ deductive methods to teach AI, which one might characterise as “studying a large body of pre-existing technical knowledge and learning how to apply it deductively to constrained problems designed to test this knowledge” [2]. For example, this approach is apparent in the defacto textbook for teaching AI [3].

However, we are not sure deductive pedagogy is the best approach. We do not think it prepares students for the complexity of using AI in the real world and, indeed, the transformative effect it will have within ‘Creative Industries 4.0’[4]. So, instead of a deductive approach, we design our courses based on an inductive, constructivist approach inspired by [4][5] and, more recently, [6]. These educators advocate an exploratory approach which engages with real-world complexity at an earlier stage of learning.

This inductive, learning-by-doing approach is commonly found in the growing number of ‘creative computing’ and STEAM courses [4] and is becoming more popular in engineering generally [5]. The approach has been found to have positive impacts on student outcomes, such as self-efficacy and to encourage more effective learning behaviour [6][2].

Teaching AI through the lens of creative activity is a natural continuation of the STEAM approach. There is a growing number of courses and even degree programmes that embed this approach in AI education. For example, the MSc Data Science and AI for the Creative Industries at University of the Arts London from

Grierson and others⁶, Computational Media Art courses at Hong Kong University of Science and Technology led by Papatheodorou⁷, Computational Creativity courses at Queen Mary, London from Colton and others⁸, Goldsmiths' Machine Learning for Musicians and Artists online course by Fiebrink⁹, the AI for Media, Art and Design course from Hämäläinen and Guckelsberger at Aalto University, Finland¹⁰, and the Creativity and AI specialisation on Coursera from Parsons¹¹

[7] list opportunities for teaching AI within creative contexts, such as considering systems holistically and developing and understanding one's own creative processes. Authenticity is a related and important aspect of STEAM pedagogy. In a systematic review of authenticity in design-based engineering education (a relative of STEAM), [8] states that "The common theme of all the different authenticity definitions is their relation to real-world experiences". [9] evaluated their music-oriented programming environment EarSketch and found that its electronic music production environment's authenticity was a factor impacting students' desire to continue learning.

If we wish for students to engage in a deep, sustained, and authentic way with the range of AI systems that are garnering the high-profile media attention with which we are all familiar, we need to handle several pedagogical and technical challenges. The systems are challenging to describe and understand - for example, GPT3 has 175 billion parameters and is based on decades of development in natural language processing and deep neural networks. The systems are resource-intensive to develop and train - OpenAI's jukebox required hundreds of GPUs and many weeks to train [10]. Even if the student has the hardware and resources to run a given system, the datasets and code are not always openly described or available to investigate.

In the course that we have designed, we have explicitly set out to address some of these pedagogical and technical challenges and hope that the reader will stay with us later in the paper to find out more.

2.2 Evaluating AI courses

In 2019, Fiebrink considered a range of approaches to teaching ML to non-technical creative practitioners, noting at the time that "Little published research examines how to teach ML effectively to any group." This suggests that the published literature lacks robust evaluation [11].

Evaluating of AI teaching appears to have improved since 2019. In their 2023 review of 49 studies discussing the teaching of AI to students, Ng et al. identified the following evaluation methods: exploratory qualitative studies, mixed methods and qualitative and quantitative evaluation. The data sources used by the studies were observational data, assessment tests, questionnaire surveys, and teacher perception interviews [12]. Most of the studies employed observation and questionnaires, which might be considered weaker forms of evaluation since they do not follow a model wherein pre- and post-intervention metrics are taken.

We found several further studies that offer some interesting ideas but have limited evaluation. For example [7], discuss opportunities and methods for teaching AI in creative contexts, but they do not report an evaluation of

their suggested methods. [13] consider what every child needs to know about AI, but they do not really consider how to teach it to them or how to evaluate the results. Sanusi notes the potential of inquiry-based learning for teaching ML [14] but does not explain how to evaluate it. In 2020, [15] found several published instructional units for teaching ML to children, and they provided an analysis of the approaches involved but did not provide any evaluation metrics. In 2021, [16] evaluated a range of different tools which provide students with access to generative image models in a ‘synthetic art’ course, focusing on the systems’ creative potential.

In summary, educators and researchers are building a body of practice and research around STEAM pedagogy and teaching AI in creative contexts. Our work continues in this vein and provides a natural extension of what has been developed to date. The originality of our work is in the technical ambition, pedagogical specificity, and the development of a mixed-methods evaluation framework.

3. Course design

Table 1: Overview of the pop-song generator course

Week title	Learning objectives	Model	Data	Technical concepts	Creative activity
Introduction to generative systems	<ol style="list-style-type: none"> 1. Describe the plan and key steps for the AI and creativity case study 2. Implement a generative system that can learn a model of a text document and use it to generate more text 3. Using examples from the literature, explain what a generative system is 	Markov model	text	Auto-regression, statistical models	Different inputs, different order

Generating lyrics with GPT-2	<ol style="list-style-type: none"> 1. Describe how self-attention allows for a combination of contextual and sequential data in transformer networks 2. Instantiate a pre-trained language generating pipeline using GPT-2 and huggingface 3. Explain how the process of fine-tuning works and why it is necessary to fine-tune pre-trained neural network models 	GPT-2	text	Self attention, fine-tuning	Prompt engineering, fine-tuning with different datasets
Music composition with MusicVAE	<ol style="list-style-type: none"> 1. Explain what a variational auto-encoder is and give examples of applications for VAEs 2. Describe the concept of a latent space and explain why they are important when exploring the capabilities of pre-trained models 3. Load and use a pre-trained model to generate multi-track MIDI files 	Music-VAE	MIDI and audio	Autoencoders, latent space	Exploring latent space via latent vectors

Singing voice synthesis with Diffsinger	1. Give examples of historical speech synthesis systems and techniques 2. Describe the complexity of current generation speech synthesis models 3. Using examples from the literature, discuss different aspects of the concept of creativity	Diffsinger	text and audio	Feature processing and model orchestration	Realism and different input patterns
Putting it all together	1. Using examples from the literature, discuss different aspects of the concept of creativity 2. Differentiate between a Skinnerian view on creativity and a Dewian view 3. Put together a complete system which can generate lyrics, music and singing, then mix them into an audio file	All	All	Theory of Ai and creativity, Linking all systems together	Working with pop song fragments

In this section, we will give an overview of the course design, which is summarised in Table 1. Before we do so, and for the purposes of significance and reproducibility, we are keen to be explicit about the list of requirements and constraints for the course which informed its design. They are as follows:

1. It contains AI-creativity content allowing for authentic, engaging creative and technical activities.
2. It covers multiple AI models whose outputs need to be combined into a cohesive and engaging final item.
3. Students should not require special hardware or be required to spend excessive time on complex setup work which is not linked to course learning objectives.
4. Implement best-practice learning design for the Coursera platform and the pedagogical practices developed with Coursera for our online computer science degree.

5. It would be organised into five weeks' content with a total study time of about 25 hours.
6. It fits with a set of three other AI case studies in the 20-week, 100-hour undergraduate AI course.
7. Suitable for final year CS undergraduates, but readily adaptable for future wider audiences.
8. It can be readily refined and extended to include more extensive case studies and a more explicit investigation into issues such as ethics, appropriation, and copyright.

We considered a range of creative activity domains, such as images, sound, and text. We eventually settled on the concept of a pop song generator. Generating lyrics, music, and a singing voice for a short song provides a combination of different techniques and models with a clear overall concept. We also have expertise in the music domain, which helped us with technical and creative development.

We present the material as five weeks of content delivered as one part of a 20-week online undergraduate course in artificial intelligence, part of our online computer science degree. So the initial audience was undergraduate (UG) CS students. We plan to expand this audience by making the course available for any learner in a standalone MOOC format on Coursera. This content is the final part of an AI course comprised of several AI case studies, including automated scientific discovery, game-playing AIs, and evolving robot morphologies.

Existing best practices on the Coursera platform and our online CS degree (launched in 2019 and now with 5000 active students) have also influenced the course design [17]. This practice suggests optimal learning design features such as alternating between videos within a 5-15-minute length range and short formative quizzes reviewing the video content. The course should also contain structured activities that have clear outputs connected to learning objectives and that can ideally be completed within the platform.

3.1 Developing the pop song generator prototype

We decided upon three major components for the pop song generator system: a language model to generate lyrics, a symbolic music model for the musical score, and a voice synthesis model to sing.

GPT-2 lyric generation

“I want to go dancing with your eyes closed. I want you to dance with my arms crossed. I want you to dance with my heartbeat.” - GPT-2 fine-tuned with Eurovision data

The first component used huggingface's GPT-2 implementation for language generation [18]. We selected huggingface as it provides a mature API and a repository of pre-trained models. With this setup, we can generate language in three lines of Python. We also provide students with a huggingface GPT-2 model that we fine-tuned in advance with a dataset of Eurovision Song Contest lyrics. Since students would need special hardware to fine-tune, we showed them the process of preparing the dataset and fine-tuning the model in a video. We made it possible for students to follow our process if the necessary hardware was available to them.

In addition to fine-tuning and the basic idea of autoregression, GPT -2 allowed us to discuss transformers and how attention is used to blend context into word embeddings.

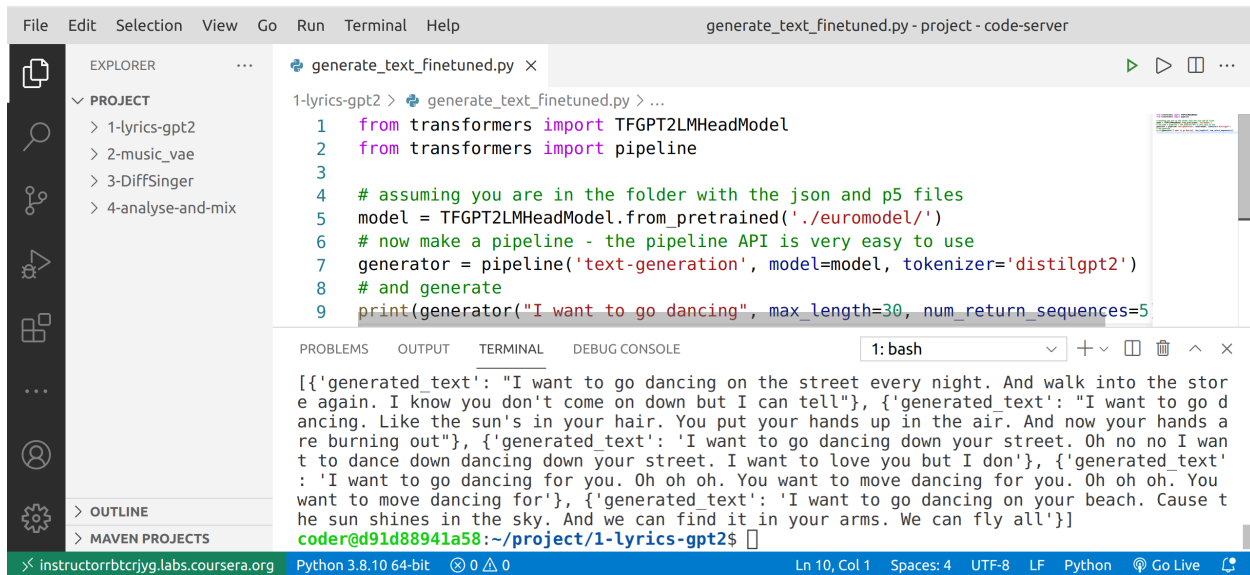
Music-VAE MIDI generation

The second component used Google Magenta's Music-VAE for musical arrangement generation [19]. We chose this as the researchers provide a range of online examples built in the Javascript language, which ran straight away in the web browser and a Python implementation. In the end, we could not operationalise the Python implementation, so we provided students with a minimal tensorflow-js version. This version ran on the command line in the Coursera Labs environment using node.js. We provide more details about the lab environment below. We used the Music-VAE system as a vehicle for the concepts of variational auto-encoders and latent spaces. In particular, we demonstrated how students could creatively explore the latent space by permuting latent vectors.

Singing voice synthesis with Diffsinger

The most technically challenging element was the singing system. During the course development, we realised that converting text and musical scores to singing is a very active, though niche area of research. We used a slightly modified version of an open-source implementation by Keon Lee [20] of the Diffsinger text to singing synthesis system [21]. Diffsinger uses several models in combination to generate and process a series of different features, so we used this to teach students about model orchestration and feature processing. Similarly to the other two systems, the singing system allows students to experiment creatively at several levels. They could pass in the different musical note and lyric sequences or dig into the code and adjust the lower-level control parameters for the synthesis system, such as pitch and amplitude modulation.

3.2 Running the pop song generator on Coursera



```

File Edit Selection View Go Run Terminal Help generate_text_finetuned.py - project - code-server
EXPLORER PROJECT
  > 1-lyrics-gpt2
  > 2-music_vae
  > 3-DiffSinger
  > 4-analyse-and-mix
generate_text_finetuned.py x
1-lyrics-gpt2 > generate_text_finetuned.py > ...
1 from transformers import TFGPT2LMHeadModel
2 from transformers import pipeline
3
4 # assuming you are in the folder with the json and p5 files
5 model = TFGPT2LMHeadModel.from_pretrained('./euromodel/')
6 # now make a pipeline - the pipeline API is very easy to use
7 generator = pipeline('text-generation', model=model, tokenizer='distilgpt2')
8 # and generate
9 print(generator("I want to go dancing", max_length=30, num_return_sequences=5))

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
[{'generated text': "I want to go dancing on the street every night. And walk into the store again. I know you don't come on down but I can tell"}, {'generated text': "I want to go dancing. Like the sun's in your hair. You put your hands up in the air. And now your hands are burning out"}, {'generated text': 'I want to go dancing down your street. Oh no no I want to dance down dancing down your street. I want to love you but I don'}, {'generated text': 'I want to go dancing for you. Oh oh oh. You want to move dancing for you. Oh oh oh. You want to move dancing for'}, {'generated text': 'I want to go dancing on your beach. Cause the sun shines in the sky. And we can find it in your arms. We can fly all'}]
coder@d91d88941a58:~/project/1-Lyrics-gpt2$

```

Figure 1: The Coursera Labs IDE running in a web browser. We set this up with all three systems ready to execute. GPT-2 is running in the terminal at the bottom right.

Coursera provides a docker-based system called Labs where it is possible to run Jupyter notebooks or any arbitrary web application. The student can access their own instance of the application, effectively their own private server, using their web browser. We were keen to enable all three AI systems to run together so that students could easily move from watching a video to running a state-of-the-art model via their web browser. Figure 1 shows a screenshot of our Coursera environment running a GPT-2 model.

The default setup for Coursera Labs has limited power, with 4GB of RAM, two CPU cores, and no GPU acceleration. For security reasons, the environment has minimal internet access. Therefore, we ensured all models could run in inference mode (as opposed to training mode) in a reasonable time without accessing blocked parts of the internet. The final setup consisted of a dockerised Linux environment that provided a browser-based IDE (Visual Studio Code), embedded terminal, and all required software and models that were installed and ready to run.

We appreciate that setting up a local Python environment, learning how to install packages, working around clashing versions and incompatibilities, and so forth, are important parts of becoming a machine learning engineer. However, as educators, we must decide what we wish to teach in a given context. For this context, our learning objectives were oriented more toward understanding and using the systems than installing them.

We found that the GPT-2 models were fast enough in inference (generative) mode, typically generating output in a few seconds once the models were loaded. Music-VAE ran in node.js from the Lab terminal, taking around 30 seconds to generate a MIDI file. DiffSinger ran in about a minute, as our customised version needed to

render twice to achieve the correct length of audio for the song. The full implementation of our pop song generator is openly available in a GitHub repository¹².

3.3 Producing the teaching materials

Once the pop-song software worked on Coursera, we produced the actual teaching materials, including wide-ranging content such as videos, multiple-choice quizzes (MCQs), workshops in the Coursera Labs environment, peer reviews, and asynchronous discussion activities. Videos were produced using a one-person-operated video studio developed during the pandemic. The video studio is equipped with hardware and custom software, making it possible to create a live video edit with multiple camera shots and a green screen.

Producing the lab worksheets was a significant part of the development work. The worksheets would take students through the processes they had seen in the videos. They would do this inside the virtual lab environment. Open-ended challenge exercises encouraged students to go beyond the content in the videos. In particular, we encourage the students to explore the creative aspects of the AI-creativity systems in each course section. For example, they experimented with prompts and the fine-tuned version of the GPT-2 model, permuted latent vectors with the Music-VAE model, and explored low and high-level control parameters in DiffSinger. Table 1 summarises some of the creative and technical concepts covered each week. Discussion prompts allow for the lightweight sharing of system outputs and small search and report activities. Peer reviews allow for more in-depth checking of technical and creative progress, graded by students against simple rubrics. Ultimately, we assessed students with a written exam wherein students answered short and long-form theoretical and technical questions.

4. Analysis

We have described our ‘minimum viable product’ course design, but is the course design successful? We have employed several methods to evaluate the course design against our goals. In this paper, we will present two tranches of our ongoing evaluation: a quantitative analysis looking at ‘time-on-task’ and a qualitative evaluation based on a thematic analysis of a workshop we conducted with AI and creative industry experts.

4.1 Quantitative analysis of student activity

238 students have now taken the course from our UG CS programme on Coursera. The Coursera platform generates extensive data exports allowing for detailed analysis of student activity. For this paper, we have focused our quantitative analysis on the duration of student access to the five parts of the course compared to the degree cohort as a whole. We chose duration as it is a well-established proxy metric for student engagement - Wong et al. state that "Time-on-task has long been recognized to be a significant variable that is correlated with learner engagement as well as a predictor of learners’ achievement" [22].

Method

We extracted log files containing timestamps for every student's access to the course items, which we refer to as 'hits'. We can then organise the hits into chronological sequences for each student on each course and compute the intervals. We can then sum the intervals to find the total time spent on a course. We exclude intervals greater than 4 hours as we assume those indicate students ending their study session. We also measured the time spent in lab activities in the pop song AI content and labs in the entire BSc degree. We organised the times into six groups: BSc CS as a whole (5842 students) and the five weeks and labs of the AI course (238 students). Thus our research questions for this analysis are:

1. How long do students spend per week in the pop song generator course compared to the BSc CS as a whole?
2. How long do students spend in the lab activities in the pop song generator course compared to the BSc CS as a whole?

Results

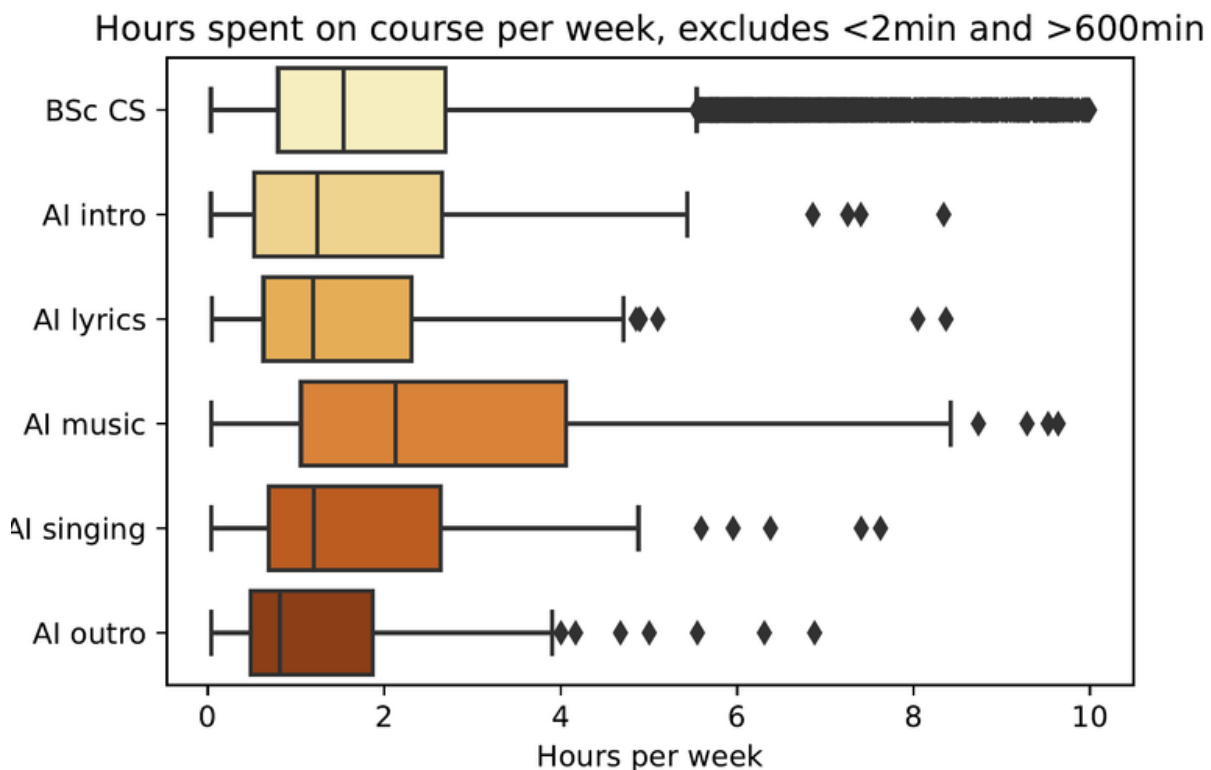


Figure 2: Range of times spent per week on the course and the overall degree for comparison. The main bars show the mean and deviation, and the points to the right show outliers. The variation between BSc and AI is significant, except for week four which is similar to the BSc.

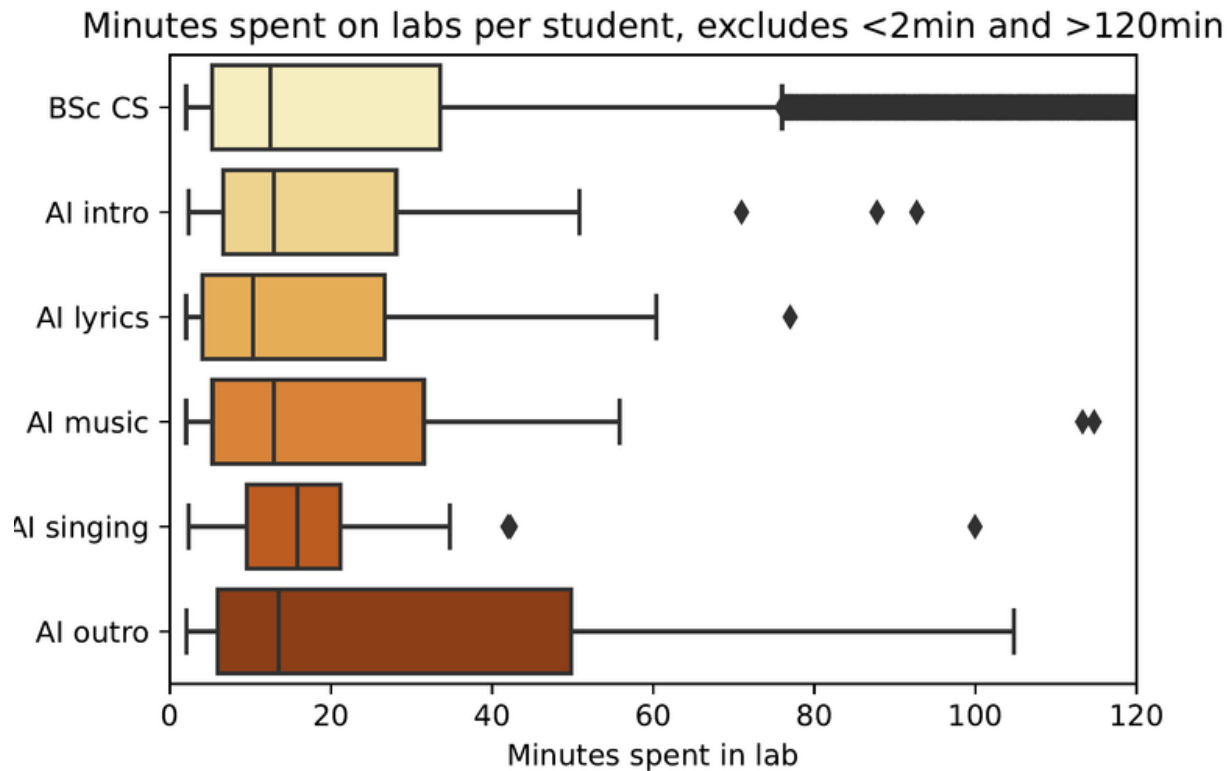


Figure 3: Range of lab times spent in the degree as a whole and each week of the AI-creativity course.

Figure 2 shows the range of times in hours spent by students in the BSc CS and each week of the AI pop song content. The boxes contain 50% of the data, and the lines in the boxes show the median. The whiskers show where 1.5 x the interquartile range falls, which for normal distributions contains 99% of the dataset. Therefore anything outside the whiskers is considered an outlier. Students spend between 0.5 and 6 hours studying per week per course on the BSc CS and a similar amount on the AI course, except for the MusicVAE week, where students appear to spend considerably more time. We ran Wilcoxon rank-sum tests between the six distributions to check if the variation was significant. We chose this non-parametric test as we do not want to assume a normal distribution of the values. We found that the distribution of time spent on the AI content was significantly different from the BSc CS except for week 4 (Diffsinger). Generally, time spent in AI weeks appeared slightly lower than BSc CS, but week 3 (MusicVAE) was higher.

Figure 3 shows the range of times in minutes spent on the labs in the AI course and in the BSc CS as a whole. BSc students generally spend 5-35 minutes on labs, but many outliers are probably indicative of the wide variety of labs in the degree. The Wilcoxon rank-sum test did not find any significant differences between the observed time spent on the AI labs and the labs across the BSc CS. The larger-looking time band for AI week 5 is not significantly different from the other weeks.

Discussion of quantitative data

Students spend less time per week on the AI course than on the degree, except for week 3, where they spent significantly more time. This metric only considers time spent on the platform indicated by course item hits, so it does not take into account activity outside of the platform and ignores downloading and watching videos later.

Possible reasons for the lower time spent per week vs the whole BSc are that we placed the content at the end of the course, and it was credit-assessed with an exam as opposed to coursework that was more directly linked to the activities. We have now moved the content earlier in the course, and the planned MOOC version will have a different assessment model since it will not have paid markers, only peer reviews. Interestingly, the time spent on the MusicVAE content is significantly higher than in other weeks. This week, there is a more structured lab worksheet with multiple scripts that provide a range of experiments for students to engage with. Also, they can generate actual music they can hear this week. Concerning the time spent in labs, some students spent considerable amounts of time in the labs, whereas others spent much less. Whilst the results here are only partially gratifying, there is much high-quality, high-rated content on the degree to compete with. These metrics certainly give us a benchmark against which to evaluate changes we make to the course.

4.2 Qualitative evaluation: Expert workshop

In this section, we describe a qualitative analysis based on a workshop we ran with a curated panel of experts. The main goal of the workshop was to find out how we can improve the existing AI course. Specifically, we aimed to address the following research questions:

1. How can we improve the existing AI course for creative practices?
2. What are the current limitations of the course?
3. How can we improve the course for a diverse Coursera audience?

Method

We organised a three-hour workshop in November 2022 with a multidisciplinary group of people, including a music composer, distance learning expert, user evaluation expert, and AI-creativity educators and practitioners. We take note of [23], who talked about the importance of considering different stakeholders in assessing the interaction between AI software and creative practice. We started the workshop with a presentation consisting of an overview of the online BSc CS, a brief description of the AI and creativity field, and finally, the high-level aims of the workshop. In the second part of the workshop, we presented our course design and opened up the floor to an immediate response from the workshop panel. The workshop was fully recorded, together with a transcript.

To analyse the text transcript of the workshop, we employed Braun and Clarke's Reflexive Thematic Analysis (RTA)[24]. Other qualitative methodologies exist, such as Grounded Theory[25] and Interpretative

Phenomenological Analysis[26], but we chose RTA due to its emphasis on sense-making, understanding, and giving someone a voice and its clear protocol. We carefully executed the six-step protocol of RTA (familiarizing yourself with the data, generating initial codes, generating initial themes, reviewing themes, defining and naming themes, and producing the report) using a semi-inductive approach¹³.

Results and analysis

After carrying out the steps above, we identified five themes, as shown in Figure 4. For each theme, we give a title, some sub-themes, some exemplary quotes and, concerning our main research goal, suggestions for future improvements of the course

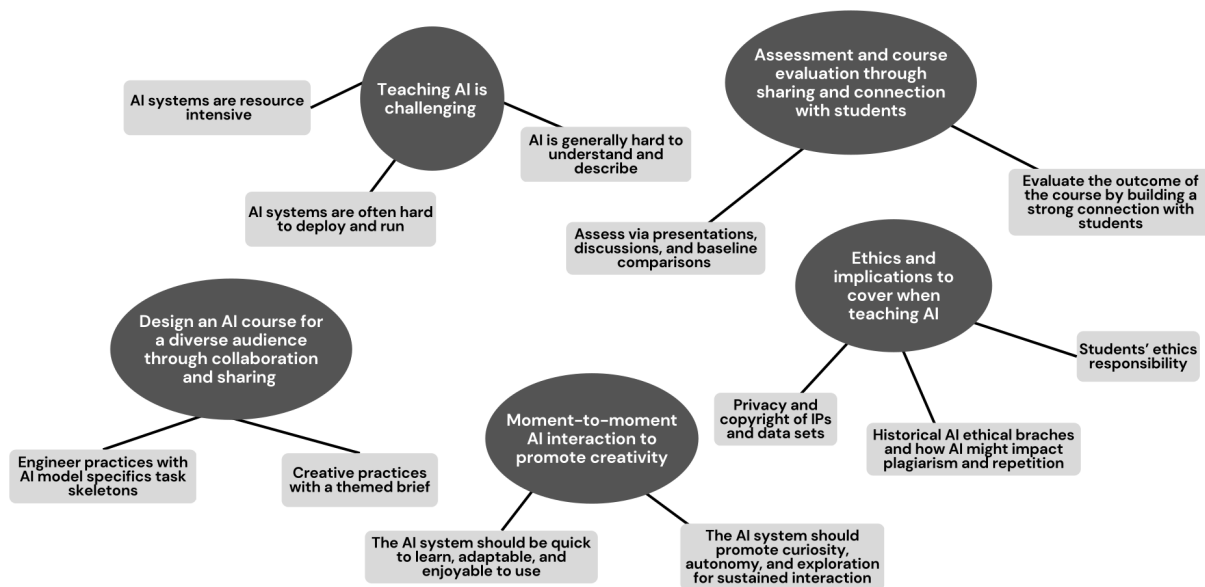


Figure 4: The final generated five themes and sub-themes from the Thematic Analysis process.

Teaching AI is challenging: The process of teaching AI can be really challenging. AI systems can be hard to describe and understand due to their complex structure and parameters (“GPT-3 has 175 billion parameters and uses all kinds of heavily iterated technology”). Even if we take a valid pedagogical approach to deliver instructional material about AI, these systems are generally resource-intensive and require a lot of skills and time to deploy and run (“You wouldn’t be able to load the model into any hardware that any institution has in the UK, because it requires too much memory.”). Suggestion: Whilst we have already taken this into account in the course design with the simpler GPT-2 model, we can consider linking to GPT-3 via a cloud service.

Moment-to-moment AI interaction to promote creativity: A great element when it comes to creativity is exploration (“The creative concept is this idea of exploring this space”). AI systems should allow the user some

autonomy and clearly communicate the possible interactions at any given moment (“Because they’ve chosen the task, they are motivated and they want to solve those problems.”). These systems should be easy to learn, adaptable, and enjoyable to use in order to promote curiosity and sustained interaction (“About a sustained interaction that actually the AI tool is sufficiently good that you actually want to carry on using.”). Suggestion: consider modifying the exercise worksheets to allow deeper student exploration of each of the components of the AI pop song generator.

Design an AI course for a diverse audience through collaboration and sharing: We should take into consideration the idea of designing the course for a diverse audience. The industry may be happy with the way engineers approach AI, but there seems to be a lack of individuals who can use AI in a creative way (“There aren’t enough people that have a kind of artistic sensitivity and understand the creative process and understand what it is to really explore and use AI to explore the creative content and production space.”). The course should retain its technical content, but at the same time, it should provide a themed creative brief for the less technical and more creative individuals (“Almost like an artistic or creative motivation for wanting you to show off what they can do. Yeah, this is the theme that I chose.”). The diverse audience can come together via collaborative tasks and sharing the obtained results (“So there’s something presumably here about students understanding the different ways that they approach it in each other.”). Suggestion: restructure the course so that individuals can approach it either creatively or in a more engineer-type way. We are planning to introduce role-playing within the course to tackle assignments from different perspectives.

Assessment and course evaluation through sharing and connection with students: The course should provide a valid method to evaluate both the technical and the creative domains of AI. The produced work can be compared to a baseline project (“Maybe compared to baseline? So what’s the simplest and stupidest thing that you can produce with this software”). Students should be encouraged to share their work via presentations and forum discussions (“You have to present your work or you have to put in a forum.”). In terms of the course evaluation, the course leader can invite students’ cohorts back to learn about their learning experiences (“Could ask for students or participants to kind of volunteer to stay in touch with you.”). Suggestion: provide a valid method to assess both the technical and creative work. We can assess the produced work against a baseline project and give students plenty of opportunities to share their work and get feedback from other students.

Ethics and implications to cover when teaching AI: When it comes to AI, ethics is a really important topic to consider, especially in a course when we are potentially introducing new people to this field. The course should cover the implications of using AI for creative practices, from IP and dataset copyrights to awareness of ethics breaches in the history of creativity (“Specific creative industries IP issues, right? Because um, it’s certainly something which impacts a lot of people in creative industries”). Part of the responsibility of ethics should be in the interest of students (“We’re teaching a course like this, and it is their responsibility to revise at least some ethics in tandem with this”). Our response: introduce another week of content covering the ethics of using AI with concrete examples and possible implications.

5. Concluding remarks and future work

In the introduction, we enumerated three intended contributions. First, in outlining a new AI-Creativity course (now in its third run), we have provided a pedagogical, technical, and evaluation platform for other educators to use. Second, we have designed and applied a benchmarking evaluation method for the quantitative analysis of student engagement. Third, we demonstrated an orthogonal qualitative approach, which engaged stakeholders in designing the course, and we have shown how it has revealed important themes along which we need to reflect, design, and test our AI-creativity materials.

In future work, we will develop the course into a standalone MOOC with additional material relating to ethical and legal aspects of AI-creativity. We will adapt our learning activities for more sustained student engagement through deeper exploratory activities motivated by using creative briefs. We are excited by the MOOC format, as the opportunities will be far greater for iterated experimentation with different styles of assessment, pre- and post-testing of knowledge, and other factors such as evaluating the quality of musical output through peer review.

We are excited that the emergence of accessible AI-creativity tools and the field's cross-disciplinary nature can, with carefully designed learning materials, provide opportunities to include AI-creativity in wider areas of university and even school curricula.

Ethics statement

The data used in this analysis was gathered in a workshop and from the Coursera platform. All human participants consented to the data being gathered. The GitHub and Google Scholar data was gathered using low frequency HTTP requests in compliance with the constraints imposed by the platforms.

Footnotes

1. "it means the end of homework" - Elon Musk 2023 [↵](#)
2. <https://www.theguardian.com/music/2020/nov/09/deepfake-pop-music-artificial-intelligence-ai-frank-sinatra> [↵](#)
3. <https://www.cbsnews.com/news/ai-stable-diffusion-stability-ai-lawsuit-artists-sue-image-generators/> [↵](#)
4. <https://www.scientificamerican.com/podcast/episode/beethovens-unfinished-10th-symphony-brought-to-life-by-artificial-intelligence/> [↵](#)
5. how-ai-is-transforming-the-creative-industries?, The Economist, April, 2021 [↵](#)
6. <https://www.arts.ac.uk/subjects/creative-computing> [↵](#)

7. <https://cma.hkust-gz.edu.cn/about-cma/> ↵
8. <https://www.qmul.ac.uk/undergraduate/> ↵
9. <https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists/info> ↵
10. <https://github.com/PerttuHamalainen/MediaAI> ↵
11. <https://www.coursera.org/specializations/creativity-ai> ↵
12. anonymised ↵
13. see the supplementary material for a detailed description of the process ↵

References

1. Chimalakonda, S., & Nori, K. V. (2013). What makes it hard to teach software engineering to end users? *2013 26th International Conference on Software Engineering Education and Training (CSEET)*, 324–328. ↵
2. Yee-King, M., Grierson, M., & d’Inverno, M. (2017). STEAM WORKS: Student coders experiment more and experimenters gain higher grades. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 359–366. IEEE. ↵
3. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach, global edition 4th*. Pearson. ↵
4. Catterall, L. G. (2017). A brief history of STEM and STEAM from an inadvertent insider. *The STEAM Journal*, 3(1), 5. ↵
5. Marques, M. R., Quispe, A., & Ochoa, S. F. (2014). A systematic mapping study on practical approaches to teaching software engineering. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 1–8. ↵
6. Magerko, B., Freeman, J., Mcklin, T., Reilly, M., Livingston, E., Mccoid, S., & Crews-Brown, A. (2016). Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education (TOCE)*, 16(4), 1–25. ↵
7. Ackerman, M., Goel, A., Johnson, C., Jordanous, A., León, C., Perez y Perez, R., ... Ventura, D. (2017). Teaching computational creativity. *International Conference on Computational Creativity. ICC3*. ↵
8. Wang, J., Dyehouse, M., Weber, N. R., & Strobel, J. (2012). Conceptualizing authenticity in engineering education: A systematic literature review. *2012 ASEE Annual Conference & Exposition*, 25–340. ↵
9. Wanzer, D. L., McKlin, T., Freeman, J., Magerko, B., & Lee, T. (2020). Promoting intentions to persist in computing: An examination of six years of the EarSketch program. *Computer Science Education*, 30(4),

394–419. [↵](#)

10. Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., & Sutskever, I. (2020). Jukebox: A generative model for music. *arXiv Preprint arXiv:2005.00341*. [↵](#)
11. Fiebrink, R. (2019). Machine learning education for artists, musicians, and other creative practitioners. *ACM Transactions on Computing Education (TOCE)*, 19(4), 1–32. [↵](#)
12. Ng, D. T. K., Lee, M., Tan, R. J. Y., Hu, X., Downie, J. S., & Chu, S. K. W. (2022). A review of AI teaching and learning from 2000 to 2020. *Education and Information Technologies*, 28(7), 8445–8501. <https://doi.org/10.1007/s10639-022-11491-w> [↵](#)
13. Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for k-12: What should every child know about AI? *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 9795–9799. [↵](#)
14. Sanusi, I. T., & Oyelere, S. S. (2020). Pedagogies of machine learning in k-12 context. *2020 IEEE Frontiers in Education Conference (FIE)*, 1–8. IEEE. [↵](#)
15. Marques, L. S., Gresse von Wangenheim, C., & Hauck, J. C. (2020). Teaching machine learning in school: A systematic mapping of the state of the art. *Informatics in Education*, 19(2), 283–321. [↵](#)
16. Friedmana, D., & Pollaka, D. (2021). Image co-creation by non-programmers and generative adversarial networks. *Joint Proceedings of the ACM IUI 2021 Workshops*. [↵](#)
17. Hickey, A., Urban, A., & Karsten, E. (2020). *Drivers of quality in online learning*. [↵](#)
18. Wolf, T. (2019). Huggingface’s transformers: State-of-the-art natural language processing. *arXiv Preprint arXiv:1910.03771*. [↵](#)
19. Roberts, A. (2018). *MusicVAE: Creating a palette for musical scores with machine learning, march 2018*. [↵](#)
20. Lee, K. (2021). *DiffSinger*. [↵](#)
21. Liu, J. (2021). DiffSinger: Diffusion acoustic model for singing voice synthesis. *arXiv E-Prints*, arXiv-2105. [↵](#)
22. Wong, A., & Chong, S. (2018). Modelling adult learners’ online engagement behaviour: Proxy measures and its application. *Journal of Computers in Education*, 5(4), 463–479. [↵](#)
23. Colton, S., Pease, A., Corneli, J., Cook, M., Hepworth, R., & Ventura, D. (2015). Stakeholder groups in computational creativity research and practice. *Computational Creativity Research: Towards Creative*

Machines, 3–36. [↵](#)

24. Braun, V., & Clark, V. (2019). Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4), 589–597. [↵](#)