

Is ChatGPT better than me? Analyzing the applicability of Large Language Models to the syllabus of a university course

Luca Andrea Ludovico¹

University of Milan, Dept. of Computer Science, Laboratory of Music Informatics – Milan (Italy)

(submitted: 12/11/2024; accepted: 8/4/2025; published: 5/6/2025)

Abstract

This paper investigates and compares the use of three Large Language Models (LLMs), i.e., ChatGPT4, Google Gemini, and Claude 3.5, as decision support systems to plan the syllabus of a university course. The experiment was conducted in the context of the bachelor's degree in Music Informatics of the University of Milan. The course under the exam focuses on the MIDI protocol, a very technical subject in the field of sound and music computing. The responses provided by LLMs have been evaluated by the author, a domain expert who has been teaching the subject for more than 10 years. From here emerges the provocative question in the title: can an AI-based chatbot prove to be more effective than an experienced teacher in defining educational objectives, materials, and the lesson plan for a university course? The results of the experimentation show that all three LLMs enable the formulation of a fairly comprehensive syllabus, allowing for the structuring of a university course. Their responses present interesting points of convergence in terms of both structure and content, while also highlighting some specificities. At present, biases and limitations still exist that make chatbots excellent co-pilots but do not replace the role of the teacher. Finally, a generalization is proposed to outline potential benefits and risks in the applicability of LLMs to the planning of educational activities.

KEYWORDS: Large Language Models, Syllabus, University Course, Educational Activities.

DOI

<https://doi.org/10.20368/1971-8829/1136193>

CITE AS

Ludovico, L.A. (2025). Is ChatGPT better than me? Analyzing the applicability of Large Language Models to the syllabus of a university course. *Journal of e-Learning and Knowledge Society*, 21(1), 201-210.
<https://doi.org/10.20368/1971-8829/1136193>

1. Introduction

Language models (LMs) are computational models that can understand and generate human language. They have the transformative ability to predict the likelihood of word sequences or generate new text based on a given input (Chang et al., 2024). The most common types of LM are N-gram models, which estimate word probabilities based on the context (Brown et al., 1992). Large Language Models (LLMs) are sophisticated models with vast parameter counts and exceptional learning capabilities. At the heart of many LLMs, such as GPT-3 and GPT-4, is the self-attention mechanism

within the Transformer architecture, which serves as the essential component for language modeling. Transformers have impacted the field of Natural Language Processing (NLP) by efficiently handling sequential data, enabling parallel processing, and capturing long-range dependencies in text (Vaswani et al., 2017).

A noticeable feature of LLMs is *In-Context Learning* (ICL), where the models generate text based on a provided context or prompt, leading to more coherent and contextually appropriate responses, making them well-suited for conversational and interactive tasks (Brown et al., 2020). ICL refers to the ability of a language model to generate output based on given instructions or task demonstrations without requiring additional training or gradient updates. In other words, instead of needing extra training, LLMs can learn just by seeing examples and getting instructions.

Another aspect that plays a vital role in refining LLMs is *Reinforcement Learning from Human Feedback* (RLHF). This technique fine-tunes the model by using human-generated feedback as a reward, helping the model improve through iterative learning and

¹ corresponding author - email: luca.ludovico@unimi.it – address: via Giovanni Celoria 18, 20133, Milan (Italy)

enhancing its overall performance (Christiano et al., 2017).

Even if the foundational moment for LLMs can be traced back to the introduction of the Transformer architecture in 2017, the years 2021-2023 witnessed the proliferation of LLMs. Companies like *OpenAI*, *Google*, and *Meta* began to invest heavily in creating and deploying their LLMs. With models like *BERT*, *T5*, and more advanced versions of *GPT*, these companies made LLM technology more accessible through APIs, leading to widespread adoption in various applications. This period saw the rapid growth of AI startups and a competitive race to develop even larger and more capable models, such as *GPT-4*, *PaLM*, and *LLaMA*.

ChatGPT by *OpenAI* (chatgpt.com), *Google Gemini* (gemini.google.com), and *Claude* (claude.ai) by *Anthropic* are well-known examples of natural language processing tools driven by AI technology and trained to interact in a conversational way. In a dialogue with a human user, these chatbots are reportedly able to answer follow-up questions, admit their own mistakes, challenge incorrect premises, and reject inappropriate requests.

LLMs are proving to be highly versatile, finding practical applications across various specialized fields. For example, in healthcare, they assist with medical diagnoses, treatment plans, and research by summarizing medical records and literature. In education, LLMs personalize learning by helping students understand complex topics and even grading assignments. In law, they automate tasks such as legal research, contract drafting, and case law analysis, providing valuable support to legal professionals. Finance benefits from LLMs' ability to analyze markets, generate insights, and automate reporting. Scientific research is another area where LLMs streamline the process, aiding in literature reviews, data analysis, and research paper drafting. Beyond these fields, LLMs can enhance customer service with sophisticated chatbots, assist journalism with content creation and fact-checking, and improve human resources by automating recruitment and employee support processes. This wide range of use cases highlights their potential to address specialized domains through efficiency and innovation.

Moreover, by revolutionizing fields traditionally connected to creativity, LLMs are putting at risk the jobs dealing with textual information gathering, translation, original writing, etc. For this reason, the scientific community is divided between those who see enormous potential and those who focus on the dangers associated with indiscriminate use.

Moving from an agnostic point of view, this work aims to explore a specific scenario, namely the applicability and effectiveness of LLMs as decision support systems to plan the syllabus of a university course. In detail, AI-based chatbots will be asked to plan a 24-hour module on MIDI in the context of a bachelor's degree in Sound

and Music Computing. After a cross-comparison between the answers provided by different LLMs, we will present a subjective evaluation based on expert supervision. In fact, the paper's author has been teaching this technical subject for over 10 years and is a member of the MIDI Association.

The initial, and deliberately provocative, question in the title is whether LLMs, used as decision-support systems, are better than an experienced instructor at determining the syllabus of a university course. However, this concern raises more specific research questions:

- [RQ1] Can the suggestions made by LLMs help an instructor (or a student) without prior experience in planning an effective teaching/learning program on a well-focused subject?
- [RQ2] Are the suggestions made by LLMs valid from a didactic and pedagogical standpoint, in the eyes of an expert?
- [RQ3] Are the findings from this experiment generalizable to different educational contexts, e.g. other subjects or educational grades?

This paper extends a talk presented at the ATEE 2023 Annual Conference – “Teacher Education on the Move”, August 27-30, 2023, Budapest, Hungary.

2. Interactions with Large Language Models

All prompts were launched at the beginning of October 2024. Today's LLMs are designed to follow instructions and are trained on huge datasets. This large-scale training enables them to perform certain tasks in a “zero-shot” manner. In *zero-shot prompting*, the prompt given to the model contains no examples or demonstrations; instead, the prompt simply directs the model to perform a task without any additional guidance (Wei et al., 2022). We decided to adopt such an approach, even creating new accounts so as not to influence the answers based on previous conversations. Other approaches would be possible, such as *few-shot prompting* (a technique to enable in-context learning where we provide demonstrations in the prompt to steer the model to better performance), *chain-of-thought prompting* (where complex reasoning capabilities are enabled through intermediate reasoning steps), *meta prompting* (an advanced technique that focuses on the structural and syntactical aspects of tasks and problems rather than specific content details), etc. The methodology will be better clarified in Section 2.1.

The LLMs under examination were *ChatGPT-4* (see Section 2.3), *Google Gemini* (see Section 2.4), and *Claude 3.5 Sonnet* (see Section 2.5). The prompt, identical for all LLMs, is reported in Section 2.2.

Answers are copied and pasted to maintain the original layout of titles, paragraphs, and bulleted and numbered lists. Some adjustments were made, in particular

boldface, only to make the text more readable after adopting the journal template.

Please note that Sections 2.3 to 2.5 contain only and exclusively the responses provided by the LLMs; the author's comments will be reported in a later section.

2.1 Methodology

To evaluate the applicability of Large Language Models (LLMs) in designing a university syllabus, a structured experiment was conducted. The study focused on a 24-hour module on the MIDI protocol within the bachelor's degree in Music Informatics at the University of Milan.

The methodology involved the following steps:

1. *selection of LLMs*: three widely used LLMs (ChatGPT-4, Google Gemini, and Claude 3.5 Sonnet) were chosen based on their accessibility and advanced text-generation capabilities;
2. *prompt design*: a zero-shot prompting approach was adopted, providing each model with the same standardized prompt (see Section 2.2). This prompt was intentionally broad to assess the LLMs' ability to autonomously generate a structured response;
3. *data collection*: each model's syllabus proposal was extracted without modification, preserving its original structure, terminology, and level of detail. Minor formatting adjustments were made for readability.
4. *comparative analysis*: the generated syllabi were analyzed across three dimensions: 1. structural similarities (organization, weekly breakdown, and assessment methods), 2. content alignment (coverage of fundamental MIDI concepts, technical aspects, and advanced topics), 3. pedagogical suitability (practicality, coherence, and alignment with established educational standards).
5. *expert evaluation*: the syllabi were compared to the actual course syllabus, designed by the author, a domain expert with over 10 years of teaching experience, and a member of the MIDI Association. This qualitative assessment identified gaps, strengths, and cultural biases in the AI-generated proposals.

2.2 The Prompt

"In the context of a bachelor's degree program in Sound and Music Computing, I have to teach a 24-hour module focused on MIDI. Please prepare a teaching syllabus".

2.3 ChatGPT-4

Here's a proposed syllabus for a 24-hour module focused on MIDI (Musical Instrument Digital Interface) within a Bachelor's Degree in Sound and Music Computing. The course will cover both the

foundational and advanced aspects of MIDI technology, with practical and theoretical components.

Module Title: MIDI: Foundations and Applications

Module Hours: 24 hours (12 sessions, 2 hours each)

Pre-requisites: Basic knowledge of digital audio, basic music theory, and computer literacy.

Course Objectives:

- Understand the fundamental concepts and architecture of the MIDI protocol.
- Explore the technical aspects of MIDI message types, including Note, Control Change, Program Change, and System Messages.
- Develop proficiency in using MIDI hardware and software for sound synthesis, composition, and production.
- Learn how to integrate MIDI with DAWs (Digital Audio Workstations), sequencers, and other music software.
- Experiment with advanced MIDI techniques, such as automation, synchronization, and MIDI-based performance systems.
- Gain hands-on experience with MIDI controllers, synths, and virtual instruments.

Course Outline:

Week 1: Introduction to MIDI (2 hours)

- Overview of MIDI: History and evolution of MIDI.
- Basic Concepts: What is MIDI? The purpose and advantages of MIDI in music production.
- MIDI Devices and Components: Synthesizers, controllers, sequencers, and MIDI interfaces.
- Practical: Setting up a simple MIDI environment using a MIDI controller and software synth.

Week 2: MIDI Protocol and Message Structure (2 hours)

- MIDI Architecture: Explanation of MIDI data flow.
- Types of MIDI Messages: Note On/Off, Control Change (CC), Program Change, Pitch Bend, Aftertouch.
- Channel Messages vs. System Messages: Understanding the distinction.
- Practical: Analyzing MIDI data in a DAW (using a MIDI monitor).

Week 3: Working with MIDI in DAWs (2 hours)

- Integrating MIDI with DAWs: Logic Pro, Ableton Live, Cubase, FL Studio, etc.
- MIDI Tracks and Routing: How to set up MIDI tracks and route MIDI signals.
- Quantization and MIDI Editing: Basic manipulation of MIDI note data.
- Practical: Create a basic musical sequence using MIDI tracks.

Week 4: MIDI Controllers and Instruments (2 hours)

- MIDI Controllers: Keyboard controllers, pad controllers, wind controllers.

- Virtual Instruments and Plugins: Using VST/AU instruments.
- Practical: Configuring and using various MIDI controllers with virtual instruments in a DAW.

Week 5: Sound Design with MIDI (2 hours)

- MIDI and Sound Synthesis: Connecting MIDI with hardware and software synthesizers.
- Automation and Modulation: Using MIDI CCs for sound manipulation.
- Practical: Design a custom sound using MIDI and a software synthesizer.

Week 6: Advanced MIDI Techniques (2 hours)

- MIDI Automation: Automating parameters in real-time with MIDI CC messages.
- MIDI Mapping and Customization: Assigning hardware controls to software parameters.
- Practical: Create a project with extensive use of MIDI automation.

Week 7: MIDI Synchronization and Clock (2 hours)

- MIDI Clock and Timing: Synchronizing MIDI devices and DAWs.
- MIDI Time Code (MTC) and SMPTE: Understanding synchronization protocols.
- Practical: Sync multiple MIDI devices and software to a central clock.

Week 8: MIDI and Live Performance (2 hours)

- MIDI in Live Settings: Using MIDI for live performance control.
- MIDI Foot Controllers and Live Automation: Creating real-time controls for stage performances.
- Practical: Set up a basic live performance rig using MIDI controllers.

Week 9: Creative MIDI Applications (2 hours)

- MIDI for Generative Music: Algorithmic composition with MIDI.
- MIDI in Interactive Systems: Using MIDI in installations and multimedia.
- Practical: Experiment with MIDI for generative music and interactive sound design.

Week 10: MIDI 2.0 and the Future of MIDI (2 hours)

- Overview of MIDI 2.0: Key differences between MIDI 1.0 and MIDI 2.0.
- New Capabilities: Enhanced resolution, per-note control, backward compatibility.
- Practical: Explore a modern DAW setup with MIDI 2.0 compatible hardware/software.

Week 11: MIDI Troubleshooting and Workflow Optimization (2 hours)

- MIDI Troubleshooting: Common problems with MIDI devices and connections.
- Optimizing MIDI Workflows: Best practices for efficient MIDI use in studio and live settings.
- Practical: Solve a real-world MIDI troubleshooting scenario and optimize a MIDI-based workflow.

Week 12: Final Project Presentation (2 hours)

- Project Submission: Students present a project that showcases their use of MIDI (composition, performance setup, sound design, or interactive system).
- Feedback and Evaluation: Peer and instructor review of the projects.

Assessment:

- Class Participation and Practical Exercises: 30%
- Mid-Term Assignment: MIDI-based composition or sound design project (20%)
- Final Project: Full MIDI-based production or performance setup (50%)

Recommended Tools and Software:

- MIDI Controllers: Novation Launchkey, Akai MPK, Native Instruments Komplete Kontrol, etc.
- DAWs: Ableton Live, Logic Pro, Cubase, FL Studio.
- Software Synths: Serum, Massive, Arturia V Collection, Kontakt.
- MIDI Monitoring Tools: MIDI-OX, Protokol.

Suggested Readings:

- “The MIDI Manual: A Practical Guide to MIDI in the Project Studio” by David Miles Huber.
- “MIDI Power! The Comprehensive Guide” by Robert Guerin.
- “Music Technology from Scratch” by Mortimer Rhind-Tutt.

This syllabus provides a blend of theoretical understanding and hands-on practice with MIDI technology, aiming to build proficiency in both technical skills and creative applications. Let me know if you'd like any adjustments!

2.4 Google Gemini

Syllabus: MIDI Fundamentals and Applications

Module Overview

This module aims to provide students with a comprehensive understanding of MIDI (Musical Instrument Digital Interface), its protocols, and applications in sound and music computing. Students will learn about the basics of MIDI, its data structure, and how to use MIDI devices and software to create, manipulate, and control musical sequences.

Learning Objectives

By the end of this module, students will be able to:

- Understand the fundamental concepts of MIDI and its history.
- Analyze MIDI messages and their structure.
- Use MIDI devices and software to create, edit, and control musical sequences.
- Integrate MIDI into music production and performance workflows.
- Explore advanced MIDI techniques and applications.

Course Outline

Week 1: Introduction to MIDI

- Introduction to MIDI: history, purpose, and basic concepts
- MIDI messages: types, structure, and functions
- MIDI devices: keyboards, controllers, sequencers, and sound modules

Week 2: MIDI Data Structures and Protocols

- MIDI data structures: events, channels, and messages
- MIDI protocols: UART, USB, and Ethernet
- MIDI system messages: exclusive, real-time, and system common

Week 3: MIDI Software and Hardware

- MIDI software: sequencers, editors, and virtual instruments
- MIDI hardware: interfaces, cables, and controllers
- MIDI system configuration and troubleshooting

Week 4: MIDI Applications and Techniques

- MIDI in music production and performance
- Advanced MIDI techniques: automation, synchronization, and control
- MIDI and other audio formats: integration and conversion

Teaching Methods

- Lectures and presentations
- Practical exercises and assignments
- Hands-on workshops and demonstrations
- Group discussions and Q&A sessions
- Guest lectures from industry professionals

Assessment

- Mid-term exam (theoretical knowledge)
- Final project (practical application of MIDI concepts)
- Class participation and assignments

Recommended Resources

Books:

- The MIDI Manual by David Benn
- MIDI: A Complete Guide by Paul Davies

Online resources:

- MIDI Association website
- Online tutorials and forums
- MIDI device and software manuals

Note: The specific content and emphasis of the module can be adjusted based on the students' background knowledge and the course's overall focus. Consider incorporating examples and case studies from various musical genres and applications to make the learning experience more engaging.

2.5 Claude 3.5 Sonnet

I'd be happy to help you create a teaching syllabus for a 24-hour module on MIDI as part of a Bachelor's Degree program in Sound and Music Computing. Let

me outline a comprehensive syllabus that covers the key aspects of MIDI.

MIDI Module Syllabus (24 Hours)

Course Description

This module provides a comprehensive introduction to MIDI (Musical Instrument Digital Interface) technology, its applications in music production, and its role in sound and music computing. Students will gain both theoretical knowledge and practical skills in working with MIDI.

Learning Objectives

By the end of this module, students will be able to:

1. Explain the fundamentals of MIDI technology and its historical context
2. Understand MIDI messages, channels, and protocols
3. Set up and configure MIDI systems
4. Use MIDI for music composition and production
5. Implement basic MIDI programming
6. Analyze and troubleshoot MIDI systems

Course Outline

Week 1 (8 hours)

Day 1 (4 hours)

1. Introduction to MIDI (2 hours)
 - Definition and historical context
 - MIDI 1.0 vs. MIDI 2.0
 - Role of MIDI in modern music production
2. MIDI Fundamentals (2 hours)
 - MIDI messages and events
 - Channels and polyphony
 - Note on/off, velocity, aftertouch

Day 2 (4 hours)

3. MIDI Hardware (2 hours)
 - MIDI controllers and interfaces
 - Synthesizers and sound modules
 - MIDI ports and connections
4. MIDI Software (2 hours)
 - Digital Audio Workstations (DAWs) and MIDI
 - Virtual instruments and plug-ins
 - MIDI editors and sequencers

Week 2 (8 hours)

Day 3 (4 hours)

5. Advanced MIDI Concepts (2 hours)
 - Control Change (CC) messages
 - Program changes
 - System Exclusive (SysEx) messages
6. MIDI Clock and Synchronization (2 hours)
 - MIDI Time Code (MTC)
 - MIDI Clock and tempo mapping
 - Synchronizing multiple devices

Day 4 (4 hours)

7. MIDI Programming Basics (2 hours)
 - Introduction to MIDI programming
 - MIDI libraries and APIs

- Simple MIDI input/output programming
- 8. MIDI for Composition and Arrangement (2 hours)
 - Using MIDI for musical notation
 - Quantization and humanization
 - MIDI orchestration techniques
- Week 3 (8 hours)
- Day 5 (4 hours)
- 9. MIDI and Audio Integration (2 hours)
 - Combining MIDI and audio in productions
 - MIDI-to-audio conversion
 - ReWire and inter-application MIDI routing
- 10. MIDI Troubleshooting and Best Practices (2 hours)
 - Common MIDI issues and their solutions
 - Optimizing MIDI performance
 - MIDI system management
- Day 6 (4 hours)
- 11. Future of MIDI and Emerging Technologies (2 hours)
 - MIDI 2.0 in-depth
 - OSC (Open Sound Control) and alternatives
 - MIDI in mobile and web applications
- 12. Final Project Workshop and Presentation (2 hours)
 - Students present their MIDI projects
 - Q&A and peer feedback session

Assessment

1. Practical assignments (40%): Weekly hands-on tasks demonstrating MIDI skills
2. Final project (40%): Create a musical piece or interactive system utilizing advanced MIDI techniques
3. Written exam (20%): Covering theoretical aspects of MIDI technology

Required Resources

- MIDI-capable keyboard or controller
- Digital Audio Workstation (e.g., Ableton Live, Logic Pro, or Reaper)
- MIDI programming environment (e.g., Max/MSP, Pure Data, or a programming language with MIDI library)

Recommended Reading

- The MIDI Manual: A Practical Guide to MIDI in the Project Studio by David Miles Huber
- MIDI Power!: The Comprehensive Guide by Robert Guérin
- Electronic and Experimental Music: Technology, Music, and Culture by Thom Holmes

I've created a comprehensive syllabus for a 24-hour MIDI module as part of a Bachelor's Degree program in Sound and Music Computing. This syllabus is designed to cover the essential aspects of MIDI technology, from its fundamentals to advanced concepts and practical applications.

The module is structured over three weeks, with two 4-hour sessions per week, totaling 24 hours of instruction. Here's a brief overview of the content:

1. Week 1 focuses on introducing MIDI, its fundamentals, and the hardware and software associated with it.
2. Week 2 delves into more advanced MIDI concepts, synchronization, basic programming, and composition techniques.
3. Week 3 covers the integration of MIDI with audio, troubleshooting, emerging technologies, and includes a final project presentation.

The syllabus also includes learning objectives, assessment methods, and recommended resources to support the students' learning.

Would you like me to elaborate on any specific part of the syllabus or make any adjustments to better fit your program's needs?

3. Analysis

3.1 Structure of the Answers

The three LLMs under examination provided responses sharing a very similar structure, that can be roughly split into three parts: 1. title and overview, 2. detailed course outline, and 3. final suggestions.

In the first part of the answers, all models proposed a title for the module and presented the general objectives of the course. *ChatGPT* was the only model that listed module prerequisites. It was also the only chatbot that did not expand the request into a more detailed description, whereas *Gemini* provided a "module overview" and, similarly, *Claude* proposed a "course description".

The second part of the answers, focusing on a more detailed course description, was articulated in weeks, days (only by *Claude*), and, finally, hours. This aspect is interesting since the prompt did not mention the concept of "week". Evidently, the various models have been trained on syllabi primarily organized into weeks, as it commonly happens in the U.S., U.K., Australia, New Zealand, and some European universities (specifically, those adhering to the Bologna Process). *ChatGPT* proposed 2 hours of lessons per week, which is coherent with the typical timetable of a semester-based system, such as the one in use at the Department of Computer Science, University of Milan. *Gemini* fitted the content into 4 weeks, suggesting that each week consists of 6 hours of lessons; this approach is consistent with the transition to a quarterly structure, which is curiously under discussion in this period at the Department of Computer Science, University of Milan. Finally, *Claude* indicated a structure made of 3 weeks, 2 days per week, 4 hours per day, which could be considered an intensive course.

In the last part of their answers, all LLMs agreed to propose forms of assessment that, once again, had not been explicitly requested and turned out to be aligned. Once again, it is plausible that LLM training took place

in educational systems quite similar to each other and, conversely, rather different from the Italian university system. Even if Italian university professors sometimes require the preparation of a final project and, more rarely, assign tasks to be completed during the course, this is not the rule. The emphasis placed on practical assignments and the final project seems overestimated for the Italian system, especially in the case of *Claude*, which allocated 80% of the overall grade to these activities. Finally, all LLMs recommended several software and hardware resources and provided a minimal bibliography.

3.2 Suggested Content

This section mainly aims to cross-compare the content of the answers, thus highlighting similarities and differences between LLMs. A critical analysis of the answers against the current version of the MIDI module, designed in a “traditional” way, will be reported in Section 3.3.

Concerning the module title, *ChatGPT* and *Gemini* made similar proposals: “MIDI: Foundations and Applications” and “MIDI Fundamentals and Applications”, respectively. Conversely, *Claude* did not provide a title. In both cases, an unexpected emphasis was placed on applications. Please note that practical aspects are difficult to reconcile with the idea of academic teaching in Italy. The consequences of this vision were also evident in the lesson plan, as better detailed below.

Concerning the course objectives, all LLMs agreed to start with the fundamental concepts and architecture of the MIDI protocol, then explore the technical details of MIDI messages, foster proficiency in using MIDI hardware and software (addressing sound synthesis, composition, and production), and gain experience with advanced MIDI techniques. No answer explicitly mentioned Standard MIDI Files, General MIDI and its evolutions, or the new MIDI 2.0 protocol extension, even if these subjects could be implicitly covered by other definitions. Only *Claude* cited basic MIDI programming, a subject that, interestingly, belonged to the syllabus of a previous edition of the course, taught when the module lasted 48 instead of 24 hours. It is worth underlining that, even if the students in the degree program have in-depth skills in general-purpose coding, it would be hard to introduce MIDI-oriented programming activities within a 24-hour module.

As a final remark, we noticed heterogeneity in the level of detail in the responses. The description provided by *Gemini* was much more concise and did not include specific references to the durations of the various teaching units. Needless to say, by adopting a technique different from zero-shot prompting, it would have been possible to elicit more detailed responses through additional interactions.

3.3 Comparison with Actual Module Content

After comparing the responses provided by the LLMs under examination, the next step involved comparing their suggestions with the current course structure, planned and structured in a traditional way based on years of experience and constant refinements. The results, graphically summarized in Figure 1, show some interesting similarities and several notable differences.

As a premise, it was not always possible to fully grab the topic of the various teaching units from the brief descriptions provided by the LLMs. For example, General MIDI, one of the key extensions made to the original protocol, was never mentioned; still, we cannot rule out the possibility that *ChatGPT* planned to cover it under the topics for Week 3 or *Gemini* under those for Week 4. The consequence in Figure 1 is that the column of General MIDI is present only in the actual edition of the course.

In a complementary way, the columns for “other” topics are present as LLM suggestions only, which is rather obvious if we consider that the current edition is the reference for defining what *other* content is. It is plausible that such a category, in addition to capturing concepts that do not fit within the current structure of the university course (e.g., MIDI troubleshooting), may also include topics whose placement is not entirely clear; if described with more specific terms, such content could be classified in the remaining categories.

The part of the course dedicated to the introduction and history of the protocol is quite similar in all proposals.

The fundamentals are much more prominent in *Gemini*’s syllabus, but this could be due to the poorly detailed response provided by this model.

One of the main differences between LLMs and the current course structure lies in the time dedicated to the description of specific MIDI messages, which is much greater in the current course edition.

Hardware and software aspects are not specifically addressed in the current syllabus, where these topics are distributed across several lessons. For example, the category of software tools known as MIDI monitors is mentioned during the discussion of MIDI messages, as monitors are useful to fully comprehend the bit-by-bit structure of messages. Similarly, sequencers and DAWs are introduced together with Standard MIDI Files. Conversely, the current syllabus does not cover MIDI hardware, except for some introductory concepts (e.g., how a MIDI cable is constructed) and trivial examples needed to describe basic scenarios (e.g., what are and how to connect controllers and synthesizers).

The topic of synchronization is interestingly covered in two hours in all proposals, both those from AI chatbots and the current edition.

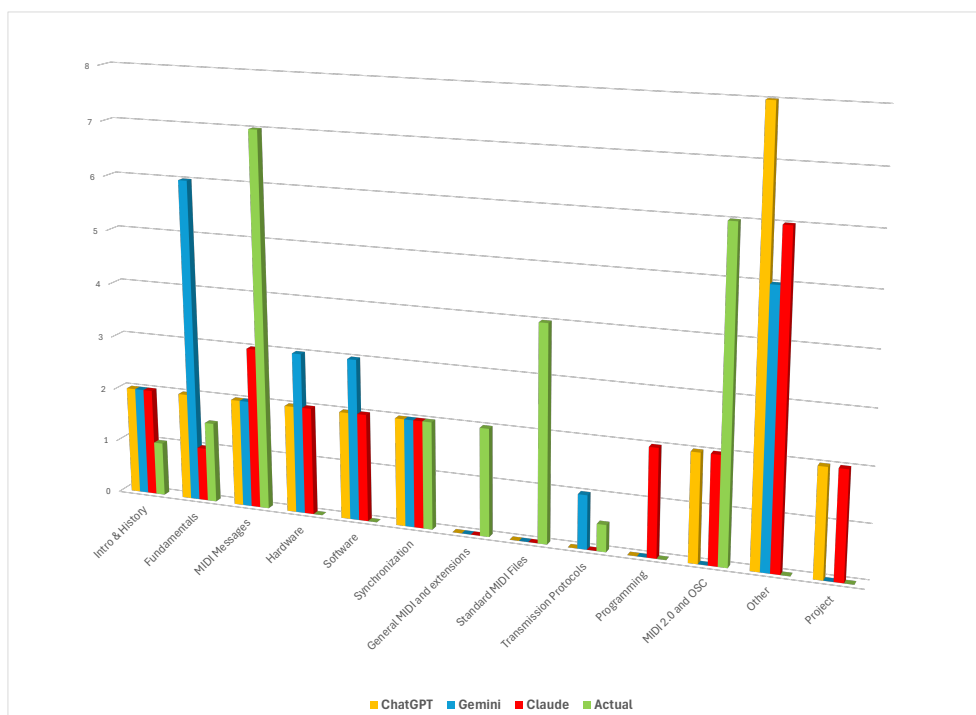


Figure 1 - Comparison of the number of hours dedicated to each lesson topic.

General MIDI, its extension, and Standard MIDI Files apparently characterize only the current syllabus, but these might be topics that LLMs do not mention explicitly, hiding them under other terms.

Transmission protocols, discussed only by *Gemini* among LLMs, are quickly covered also by the current syllabus.

As mentioned before, *Claude* is the only chatbot that proposes also a programming unit.

MIDI 2.0 and OSC are treated in a rather superficial way by LLMs, while teaching experience shows that they are rather complex topics and their understanding requires much more than 2 hours of lessons. In fact, in the current syllabus, MIDI 2.0 and OSC are covered in 6 hours, and students often complain that the subjects are very compressed.

Finally, LLMs propose practical activities and reserve some time to present the final project, have a Q&A session, and obtain peer feedback. Even if discussion is encouraged during lessons, the current course structure does not allocate a time slot for the mentioned activities, especially since the course does not include a lab or a final project.

Figure 2 provides an alternative graphical representation of the proposals made by chatbots with respect to the current course structure. Some blocks have comparable dimensions (e.g., the orange bars for introductory and historical aspects), some blocks appear only in chatbot proposals (e.g., the red bars referring to software aspects), some blocks are chatbot-specific (e.g., the light green bar for programming in

Claude), and, finally, some blocks highlight the different weights assigned to common topics (e.g., the olive bars for MIDI messages).

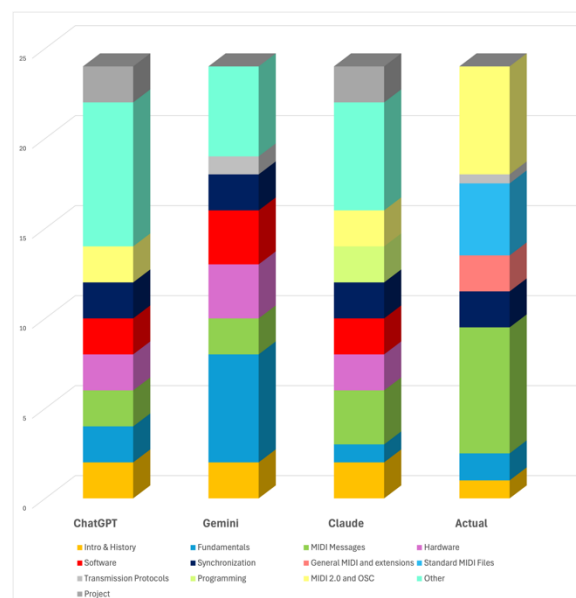


Figure 2 - The course structures, topic by topic, according to the various proposals.

4. Discussion and Conclusions

The final considerations start from the original question – “Is ChatGPT (and are other LLMs) better than me?”

– and decline such an inquiry into the three research questions listed at the end of Section 1. Most of the comments reported below are based on the experience gained by the author in more than a decade of teaching, so they have a strong foundation, but they are profoundly subjective, too.

[RQ1] “Can the suggestions made by LLMs help an instructor (or a student) without prior experience in planning an effective teaching/learning program on a well-focused subject?” The answer is yes. The LLMs, questioned with a rather generic prompt and in a zero-shot manner, were able to structure a rich and effective educational proposal, according to the knowledge accumulated by the author over years of teaching. Some aspects are barely sketched and some important keywords are missing, but, in general, the answers cover a wide range of topics. Under this perspective, the adoption of LLMs can work as a source of inspiration for those who want to approach a topic, even a very technical one, from scratch. Furthermore, LLMs can help a non-domain expert to structure an educational path in an organic way. Finally, it is worth underlining that this paper is based on very limited interactions with LLMs. When answers are considered unsatisfactory, incomplete, or too generic, new prompts can be launched, thus adopting strategies such as few-shot or chain-of-thought prompting.

[RQ2] “Are the suggestions made by LLMs valid from a didactic and pedagogical standpoint, in the eyes of an expert?” Here the answer becomes more complex. First, the expert must be open-minded towards the suggestions provided by LLMs. If the expert believes she is the best in her field, the experiment can be considered a failure from the start. From an agnostic point of view, some gaps found in the syllabus are serious. For example, the fact that no LLM mentions the topic of Standard MIDI Files is a critical issue: a domain expert could easily detect such a gap, but a non-expert would simply skip the topic, unaware of its existence and importance. Similarly, the weight given to some teaching subjects seems unlikely: covering MIDI programming (using what a paradigm and language?) or MIDI 2.0 extension in a few hours means keeping the presentation at an extremely superficial level, probably incomprehensible to the expected audience.

In general, the structure of the course proposed by LLMs shows a cultural bias, probably due to the training conducted on US and UK university systems. In fact, assignments are a regular part of the educational process in American and English universities. Differently from Italian academic standards, students are often given weekly or bi-weekly assignments in the form of essays, problem sets, projects, or reading tasks, and assignments contribute significantly to a student’s final grade. Many programs, particularly in STEM (science, technology, engineering, and math) fields, incorporate hands-on activities like lab work, projects,

internships, and practical applications. Especially in applied disciplines, project-based learning is emphasized, with students working in teams to develop solutions to real-world problems or creating innovative prototypes. Examples can be easily found in web-available syllabi of computer-related courses, such as those administered by Stanford University or MIT. [See <https://web.stanford.edu/class/cs107/syllabus> or <https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/pages/syllabus/> (visited on February 27, 2025)].

As a result of this “biased” training, some parts of the syllabi proposed by LLMs are more technical than theoretical, with plenty of practical topics (for example, MIDI routing and troubleshooting) and hands-on activities. However, it is worth noting that the introduction of lab and practical skills can be a good suggestion, also for the Italian system, to make lessons clearer and more engaging.

[RQ3] “Are the findings from this experiment generalizable to different educational contexts, e.g. other subjects or educational grades?” LLMs are certainly capable of providing answers on a wide range of topics. Of course, the prompts must be properly constructed to provide essential information about the students’ background and the target educational level.

An instructor is expected to be an expert in what she is teaching, not someone completely unfamiliar with the subject. This means that the function of LLMs is that of a co-pilot rather than a full replacement of the instructor. For example, a domain expert can easily detect and fill in the gaps in a course program. Similarly, she can recognize hallucinations, namely situations where the model generates information that is factually incorrect, misleading, or entirely fabricated, even though presented confidently (Perković G. et al., 2024).

Finally, the experience accumulated over years of teaching, the face-to-face interaction with students in a classroom environment, and the ability to build an empathetic relationship between teacher and learner go beyond the capabilities of an LLM.

In conclusion, “Is ChatGPT better than me?” Currently, it is not, and it is highly improbable that new versions of LLMs will be able to fully replace the role of an educator. However, AI-based chatbots can be helpful companions in revising consolidated syllabi, proposing new content, and pushing teachers to critically reflect on their choices. In this sense, analyzing the advantages and limitations of LLMs can play a fundamental role in being able to profitably use them.

References

Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram

- models of natural language. *Computational linguistics*, 18(4), 467-480.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3), 1-45.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Perković, G., Drobnjak, A., & Botički, I. (2024). Hallucinations in LLMs: Understanding and addressing challenges. 2024 47th MIPRO ICT and Electronics Convention (MIPRO), 2084-2088. IEEE.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2022). Finetuned language models are zero-shot learners. *International Conference on Learning Representations (ICLR 2022)*. <https://iclr.cc/virtual/2022/oral/6255>