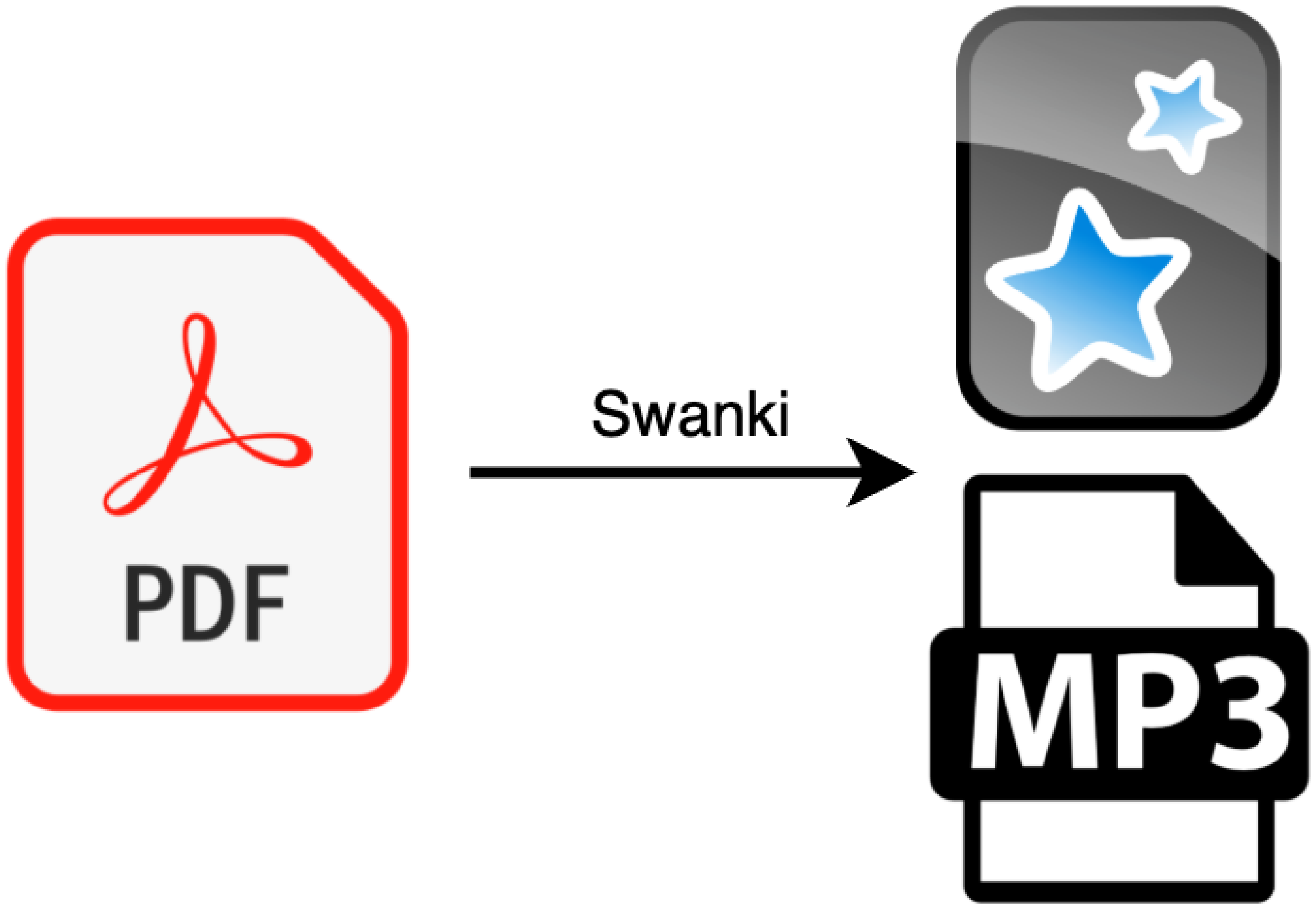
Michael Volk

ABSTRACT

Swanki (pronounced “swanky”) is an open-source Python application that leverages artificial intelligence to transform PDFs into Anki flashcards, addressing the underutilization of evidence-based learning techniques in STEM education. While spaced repetition flashcard applications have demonstrated effectiveness in medical education, showing Anki users achieving higher USMLE Step 1 scores, adoption remains limited in chemistry, engineering, and other STEM disciplines due to high content creation costs and misconceptions about flashcard utility. Swanki employs a modular pipeline architecture turning PDFs into high quality Anki flashcards and optional audio files. The system’s user controlled hierarchical configuration framework allows educators to customize AI prompts for specific learning objectives, generating three card types: standard question-answer, cloze deletion, and image-based cards that preserve LaTeX mathematical notation critical for STEM content. Four optional audio modes include complementary card narration, summaries, complete transcript readings, and lectures allowing for multimodal learning experiences. By automating the translation of educational materials into interactive study tools while maintaining educator oversight, Swanki directly supports the U.S. Executive Order on Advancing Artificial Intelligence Education for American Youth, enabling educators and students to both utilize and understand AI technologies. Future developments include support for chemical structure parsing via SMILE strings, programmatic visualization generation, and an Model Context Protocol server for natural language interaction helping democratize evidence-based educational content creation across chemistry subdisciplines and independent study more broadly.

GRAPHICAL ABSTRACT



KEYWORDS

#TODO find appropriate keywords on ACS

INTRODUCTION

Common learning and memory best practices including retrieval practice, spaced repetition, interleaved practice, and elaborative encoding are all supported by Anki, the spaced repetition flashcard application[1–5] . Anki is popular among medical students studying for United States Medical Licensing Exam (USMLE) Step exams where students can download pre-made flashcards developed by the community. Anki usage has been shown to be associated with higher USMLE Step 1 scores where more consistent users achieved higher scores[6] . Anki offers extensive digital content formats including text, images, audio, and mathematical notation with LaTEX. Despite effectiveness of Anki and other flashcard apps in learning and memory, they are underutilized in other STEM disciplines. This could be due to a variety of reasons like long chains of reasoning not being easily amenable to flashcards, but we suspect that that a large reason is simply due to the cost of card creation or a limited view of flashcards only supporting rote memorization[7] .

To fill this perceived gap we developed a python application named Swanki (pronounced "swanky"), as in fancy Anki, that takes PDFs as input and produces anki cards as output. We've additionally added optional audio features including audio summaries, audio reading, audio lectures, and complementary card audio that allow users to interact with material through their preferred medium.

There are recent anki plugins AnkiBrain for supporting the use of large language models (LLMs) for card generation from source material. There also a number of online services that now allow for similar types of content creation including NotebookLM, Limbiks, Anki-decks, Algor Education, Knowt, and many more. We differentiate primarily by making the entire content creation process configurable by the user so they can modify prompts to customize output for individual learning goals and by supporting the creation of image cards and other optional audio output options. During development we have found that it is difficult to know the depth of information that any one individual may desire. For example it can be difficult to create cards on implementation details of a multiline algorithm, but if the user specifies this by modifying configurable prompts this type of card creation becomes feasible.

METHODS

Architecture

Swanki is implemented as a modular python application built on a pipeline-driven architecture that converts PDFs into Anki flashcards through sequential processing stages (Figure 1A). The core pipeline orchestrates PDF splitting using PyPDF2, OCR conversion via Mathpix API, and AI-powered content generation using OpenAI's GPT models with Instructor for enforcing structured outputs. The system employs a sliding window approach over the PDF with configurable window sizes to maintain contextual coherence and for guaranteeing full coverage during card generation. Pydantic data models used by Instructor ensure content validation and type safety throughout the processing workflow.

Configuration

The software utilizes Hydra's hierarchical configuration system to provide comprehensive customization through automatically generated configuration files stored in `.swanki_config`. Users can modify processing parameters including window size, cards per page, image processing settings, and modify AI model prompts to their liking. The configuration system supports runtime overrides via command-line arguments and includes preset profiles (default, comprehensive, fast) for different use cases. Output files and Anki decks use citation keys so users can mix cards into large decks without losing context which supports elaborative encoding helping users relate their learnings across topics.

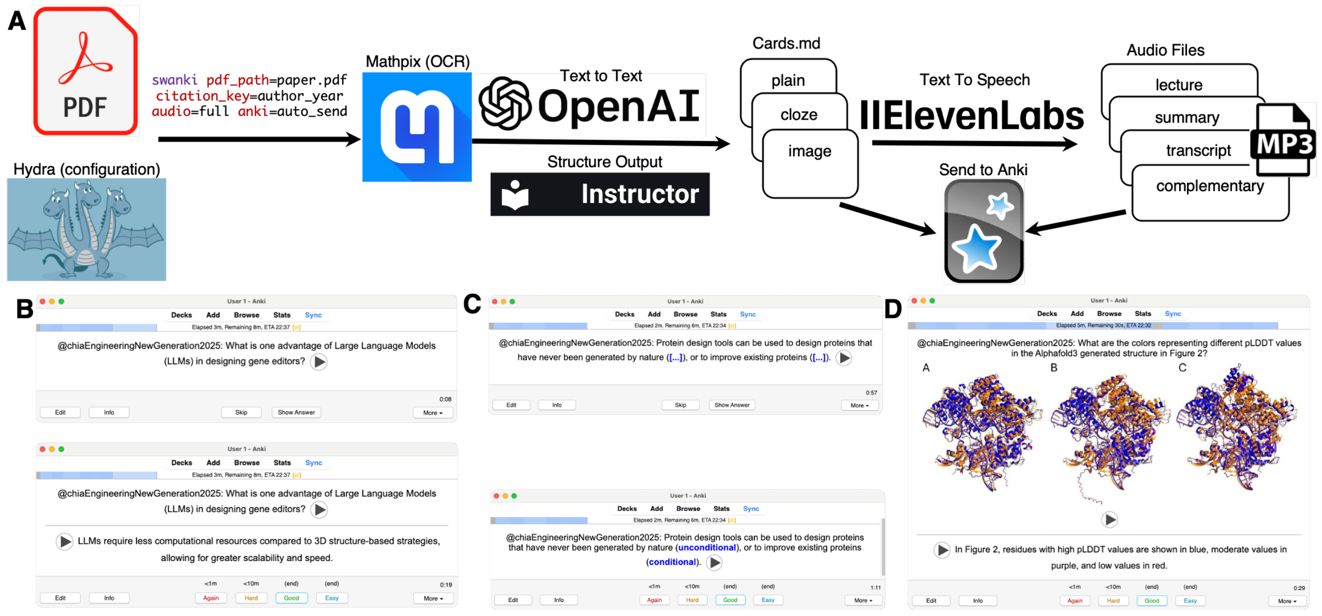
Cards

The system generates three primary card types: standard question-answer cards (Figure 1A), cloze deletion cards that mask information (Figure 1B), and image-based cards derived from figures and diagrams (Figure 1C). Cards are tagged using configurable hierarchical systems with citation keys automatically prepended to maintain source attribution. LaTeX mathematical notation is preserved for visual display and converted to natural language for audio compatibility.

Audio

Swanki's complementary audio system generates four distinct audio types: complementary audio providing front/back narration for each flashcard, summary audio for rapid content review, reading audio containing complete document narration, and lecture audio formatted as educational presentations. The system uses LLMs to convert academic text into transcripts for text-to-speech (TTS) models, automatically humanizing mathematical expressions and adding descriptions for images. Audio generation supports variable playback speeds, voice options, and other options configurable via the ElevenLabs API.

The audio system outputs enable different use cases. Lecture audio is good for first time listening and is similar to a solo podcast. Summaries can help for quick paper review, but are often difficult to digest with no previous familiarity with the PDF. Complete document narration can help serve as a pacer while reading and can help students from spending too much time reading instead of practicing via card review. Given a set amount of time, it is often a better use of time to reduce time spent reading and increase time being quizzed, that is reviewing cards. Lastly complementary card audio supports hands-free review sessions using gamepad controllers which can be easily setup with mobile devices with the Anki app. This can gamify the flashcard experience allowing users to participate in other activities like exercise as they listen to questions from the app and respond using small hand held controllers like the micro controller from 8BitDo.



```
swanki pdf_path=paper.pdf
citation_key=author_year
audio=full anki=auto_send
```

Elapsed 3m, Remaining 8m, ETA 22:37

@chiaEngineeringNewGeneration2025: What is one advantage of Large Language Models (LLMs) in designing gene editors?

LLMs require less computational resources compared to 3D structure-based strategies, allowing for greater scalability and speed.

Elapsed 2m, Remaining 6m, ETA 22:34

@chiaEngineeringNewGeneration2025: Protein design tools can be used to design proteins that have never been generated by nature ([...]), or to improve existing proteins ([...]).

@chiaEngineeringNewGeneration2025: Protein design tools can be used to design proteins that have never been generated by nature (unconditional), or to improve existing proteins (conditional).

Elapsed 5m, Remaining 30s, ETA 22:32

@chiaEngineeringNewGeneration2025: What are the colors representing different pLDDT values in the Alphafold3 generated structure in Figure 2?

In Figure 2, residues with high pLDDT values are shown in blue, moderate values in purple, and low values in red.

Figure 1. Swanki pipeline and example outputs from an open source ACS publication[8] . Play buttons on cards indicate that these cards were generated with complementary audio. Example cards are available on Github. (A) The Swanki pipeline starts with PDF input and the Hydra configuration from the user. A test default configuration is provided to users. Mathpix is then used for optical character recognition (OCR) converting the PDF to markdown and extracting images and equations. LLM's defaulting to using OpenAI API is then used with python library Instructor to create structured outputs improving the stability of the pipeline to create plain, cloze, and image cards. LLM and card outputs are then used to optionally construct lecture, summary, transcript, and complementary audio files. (B) Plain card example in the Anki App with the top image showing the front of the card and the bottom image showing the flipped card. (C) Cloze card example with the top image showing masked text and the bottom image showing flipped card revealing masked text. (D) Image card showing the flipped card.

DISCUSSION

Using LLMs poses a risk of hallucination but by providing source material in form of a PDF the LLMs task becomes easier by turning the task of educational material generation from scratch to a translation task, translating from PDF content to card format. Cards should always be reviewed prior to using them as serious educational content. One of the major goals of developing Swanki was to produce better cards, but this criteria is often ill-defined and depends on aesthetics and expertise of the educator. We choose to use a sliding window technique to get content over the entire document allowing the educator to prune cards according to their needs. This should help ease the burden of the educator as reviewing material is often faster than generating it from scratch.

On April 23, 2025 the executive order Advancing Artificial Intelligence Education for American Youth was signed in the United States[9] . The policy aims at increasing AI proficiency by integrating AI into education. Since Swanki can be used for any PDF we believe the software supports Sec. 6 "Improving Education Through Artificial Intelligence," as students and teachers can use the software to learn about AI by creating content on AI, but also by inspecting the open source software itself. We also believe that it supports Sec. 7 "Enhancing Training for Educations on Artificial Intelligence," by reducing the time intensive task of educational content creation. The app can also be used to help educators learn about modern AI techniques as these techniques are being used in other fields (Figure 1B, 1C, 1D).

While we push the limits of the types of content that Anki can natively represent, there are additional card types that we plan to support given enough community interest. To eliminate some of the manual cutting and stitching together of PDFs to remove superfluous information prior to sending the PDF to Swanki, implementing an model context protocol (MCP) server for use of Swanki could further eliminate some of the barriers to using the tool by allowing it to be called with only natural language. To further democratize the tool, our first step will be add support for open sourced models that can be run locally to reduce total cost of card generation. Additionally, we would like to find an open source drop in replacement for Mathpix, but this is difficult as bad optical character recognition at the start of the pipeline leads to garbled outputs. Additional cards could be supported such as image occlusion cards, where the content of interest in an image is occluded by a bounding box. This could be achieved using a semantic segmentation model to draw bounding boxes according to concepts elicited from the figure descriptions and the text. While the current pipeline can account for chemical structures via images, support of smile string rendering would be beneficial to parse long synthesis diagrams into component parts for more atomized card generation. Another feature would be to expand the scope of inputs to include videos. We find that PDFs cover most use cases since pptx, docx, etc. can often just be converted to PDF without any meaningful information loss. Video poses a larger challenge to synchronize transcripts with video frames to extract meaningful images and text for card generation. Cards generated from video could also allow for gifs which are supported by Anki. This would involve bounding frames of interest then simply converting video to gif and prompting for appropriately related questions. We show such a gif for linear algebra vector operations which can be found on github. Lastly, our most ambitious idea is to auto generate gifs programmatically via libraries such as manim that were developed for producing programmatic visualizations of mathematics. The idea is to go from text to program to gif output paired with appropriate questions. We envision that further feature development and open sourcing of swanki and similar educational tools will foster expertise of students in any domain.

ASSOCIATED CONTENT

The Supporting Information is available on the ACS Publications website at DOI: 10.1021/acs.jchemed.XXXXXXX. [ACS will fill this in.]

Example brief descriptions with file formats indicated are shown below; customize for your material.

We provide links here to the source:

- pypi-swanki
- github-Swanki
- docs-swanki
- #TODO Link to Open Source example cards on GitHub

AUTHOR INFORMATION

Corresponding Author *E-mail:

REFERENCES

1. Bradshaw, G. L. & Anderson, J. R. Elaborative encoding as an explanation of levels of processing. *Journal of Verbal Learning and Verbal Behavior* **21**, 165–174 (1982).
2. Polk, D. T. A. *The Learning Brain.*
3. Kornell, N. & Bjork, R. A. Optimising self-regulated study: The benefits—and costs—of dropping flashcards. *Memory* **16**, 125–136 (2008).
4. Santos-Ferreira, D. *et al.* Digital flashcards and medical physiology performance: A dose-dependent effect. *Advances in Physiology Education* **48**, 80–87 (2024).
5. PhD, B. O. *A Mind For Numbers: How to Excel at Math and Science (Even If You Flunked Algebra).* (Penguin, 2014).
6. Lu, M., Farhat, J. H. & Beck Dallaghan, G. L. Enhanced Learning and Retention of Medical Knowledge Using the Mobile Flash card Application Anki. *Medical Science Educator* **31**, 1975–1981 (2021).
7. Senzaki, S., Hackathorn, J., Appleby, D. C. & Gurung, R. A. R. Reinventing Flashcards to Increase Student Learning. *Psychology Learning & Teaching* **16**, 353–368 (2017).

8. Chia, B. S. *et al.* Engineering a New Generation of Gene Editors: Integrating Synthetic Biology and AI Innovations. *ACS Synthetic Biology* **14**, 636–647 (2025).

9. Orders, E. Advancing Artificial Intelligence Education for American Youth. *The White House* (2025).