Next-Generation Spaced Repetition: the White Paper

I. Introduction

The advent of computers has made spaced repetition available to the masses in the form of spaced-repetition software (or SRS), and provided a wealth of information on best practices for using it. The classic article/blogpost by Gwern includes a wide swath of citations to studies (both formal and informal) attesting to the usefulness of SRS, and will serve for our purposes as a literature review on the subject. SRS is used by tens of thousands of people for language-learning, mathematics, science and trivia, among other subjects. It is particularly heavily used by medical students, whose chosen profession demands the absorption of a punishingly large amount of content knowledge. The most commonly-used SRS platform is the open-source Anki, but alternatives such as Mnemosyne and SuperMemo are also reasonably popular. SRS is extremely effective when properly used; this Reddit user used Anki to teach his daughter to read by the age of 2 and his son to know his time tables up to 12×12 by the age of three. While this is something of an extreme example, it attests to the versatility and effectiveness of spaced repetition—a rare bright spot in a field (educational psychology) chock-full of half-baked results and failed replications.

However, despite its effectiveness, SRS remains little-used outside of a handful of niche contexts such as med school relative to the size of its potential market. I argue that this is due primarily to the deficiencies of the existing platforms, and that the time has come for an SRS system built for classroom usage rather than the disciplined autodidact. During a former stint as a Latin teacher, I attempted to include Anki in the classroom and encouraged students to use it, but soon ran up against a number of serious issues:

- The core of Anki's codebase dates to the early-to-mid-2000s, a time when software was generally downloaded and installed on the user's machine, and the user could be expected to configure and tweak its settings as needed. In the 2020s, most software is browser-side, and although Anki does allow the user to do reviews in the browser, the browser-side review platform is a mere ghost of the full program. **Most**

students today use Chromebooks onto which software cannot be installed locally and therefore cannot really use Anki; this is the biggest of a number of high hurdles.

- Anki gives serious users vast amounts of power to customize their cards and reviews, but the learning curve is steep, and the casual user is tempted to give up early on because the default settings are mediocre at best. This is perhaps most acute in the field of card and template design. It is possible to design custom cards with automatic formatting or other unusual behavior, but doing so requires the knowledge of at least HTML and in most cases also Javascript. There is a great deal of low-hanging fruit in the UI which could be picked with only a few lines of extra code.

  While Anki's card- and template-design abilities are powerful (considerably more so than those of competitors such as Mnemosyne), most of that power is inaccessible to casual users. Formatting options such as color must be selected manually and laboriously for every field, or implemented via custom scripts. Cloze deletions exist only for text. Language learners must switch between keyboard layouts every time they want to use diacritics, or copy-and-paste from a third-party keyboard. There is no native support for creating diagrams, which must be made in another application (such as Figma) and imported as images. The native math package, MathJax, allows virtually no additional formatting, which makes it difficult to use best-case practices when designing cards for mathematics, unless the user is willing to invest time and effort into writing unintuitive, tedious and kludgy LaTeX code. There is no built-in support for mass card manufacture via spreadsheet; users must create a CSV (comma-separated file), usually by converting and exporting from Excel.

- There is no support for complex relationships between cards other than sibling cards within the same template. Good cards mimic the structure of human memories, whose interrelations are much more complex than 'bury siblings until the following day'. Native support for this would make reviews both more effective, with much less friction during sessions.

- Aside from shared decks, Anki has no support for usage by groups of learners rather than individuals. The teacher cannot see whether or not students have been doing reviews, or how well they have done on them, except by checking their screens individually. Nor can a teacher tweak cards in students' decks after a shared deck has already been downloaded, make cards for students on an ongoing basis, or see which cards have been giving students trouble. As noted above, because Anki is downloadable rather than browser-based, it cannot be used by most students in contemporary classrooms anyways.

- Because Anki is open-source, non-profit software, there is very little incentive to change. Development is subsidized by iOS users, who must pay ($24.99 at the time of writing) to download the app version of Anki. It is unclear how many downloads the iOS app version of Anki has garnered, but it is currently the #1 paid app in the 'education' section. It has clearly done quite well for its main developers, but by the same token is stuck in a local maximum.

The above deficits also apply, sometimes to an even greater degree, to Anki's main competitors (Mnemosyne and SuperMemo), so I will pass over the alternatives and use Anki as my point of departure, drawing on a model of hierarchical chunking and review that I first sketched out in my blog series *Why Anki Works*. Anki is the best of a severely underoptimized batch, and the time has come for something new. This essay will dig into each of the above criticisms in turn and envision a competitor that does things differently.

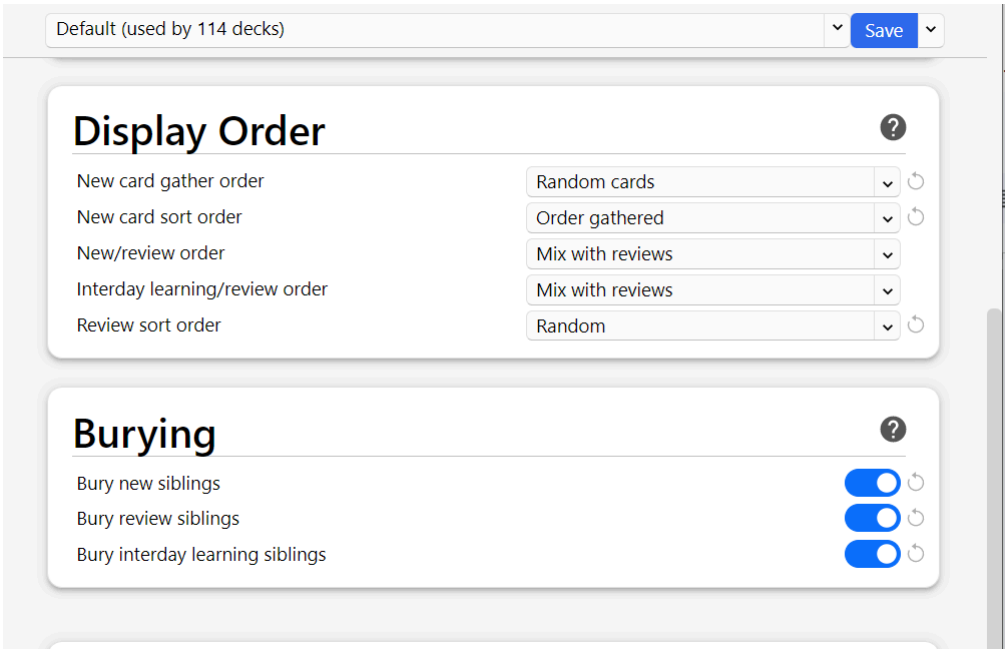
II. Platform

There is little to say here that has not been said. The software ecosystem of 2006, when Anki was released, was one of software distributed over the Internet or CD-ROM and installed locally on the user's computer. Nearly two decades on, most software is browser-side, and users often expect that software will be in-browser; students born since ~2010 or so are often bamboozled by the prospect of installing software locally. This alone would make a strong case for a native browser-side app.

More pressingly, however, most students in contemporary K-12 classrooms use Chromebooks, which cannot install software locally without carefully jailbreaking the device, which few students, parents or districts can or will want to do, not least because even a moderately experienced attempt is liable to brick the device. It is possible for Chromebook users to use AnkiDroid, the Android version of Anki, but—although AnkiDroid is superior to Anki's native web-based reviewer—it is still a far cry from the desktop version. The conclusion is therefore not merely strong but unassailable: **any SRS platform hoping to see wide classroom usage must be browser-based from day one**, and none of the available options are.

III. UI Design and the Learning Curve

Anki's user interface is kludgy. Virtually no power users use the default settings, which suggests that the default settings are in dire need of optimization[1]. Users wishing to improve their review experience are accosted with a bewildering array of dropdowns and buttons: *what is 'insertion order'? What is the difference between 'card gather order' and 'card sort order', and why does Anki make me care about this?* The tooltips explain these mechanics to some extent, but not why one might want to pick one option over another.

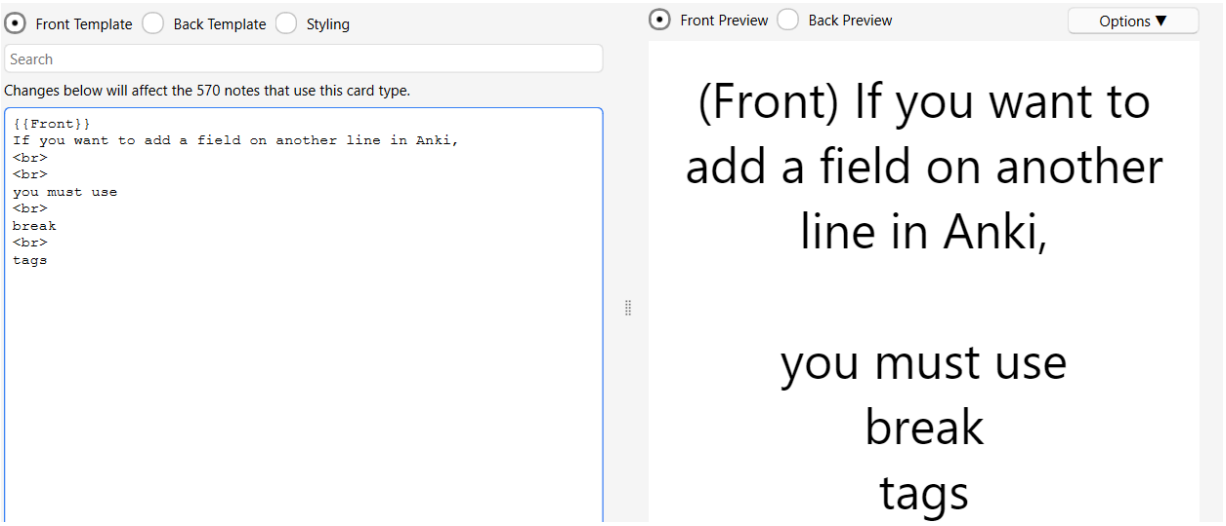


[1] However, the default settings have improved somewhat in the last few months since Anki switched to the FSRS algorithm. The FSRS algorithm is, happily, open-source, so we do not need to worry about exactly what the scheduling algorithm will look like—not, at least, until much later on.

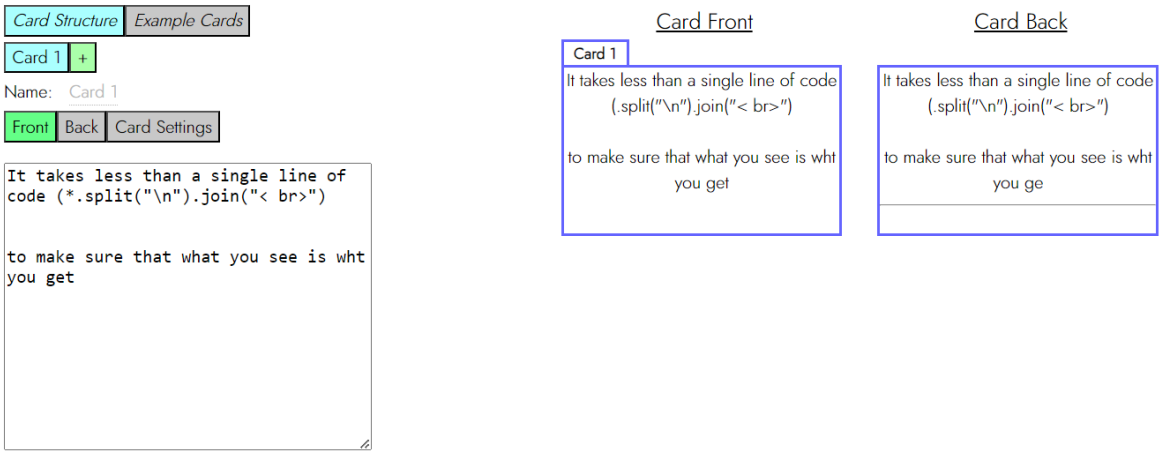
Most users therefore end up Googling for a set guide to improving their Anki settings and follow it blindly—in effect merely switching from one set of blind default settings to another.

A more serious problem is that of card and template design. There is wide agreement among serious SRS devotees that good card design is a make-or-break necessity for using spaced repetition effectively. Good cards fit comfortably within the contours of the user's cognition; bad cards create "cognitive friction" and make review sessions tedious and unpleasant. It is possible to make good cards with the default card templates, but for many purposes optimal cards require a customized template. Here Anki fails:

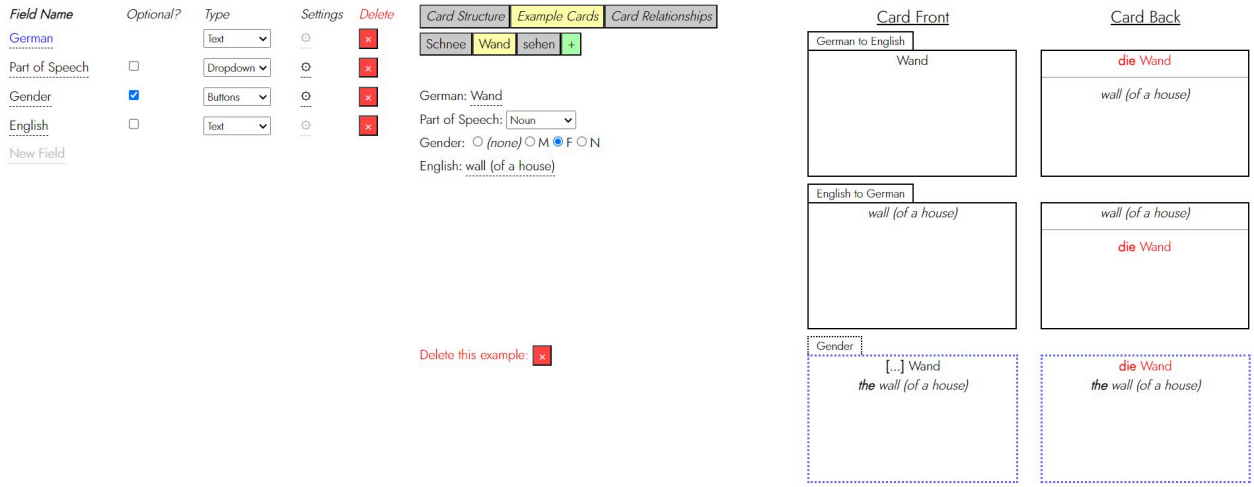
- The template-editor is raw HTML, not what-you-see-is-what-you-get (WYSIWYG):



But the basics of a WYSIWYG editor are not difficult:



- In a similar vein, consider advanced formatting options such as color. It is well known that the human mind is very good at remembering color and that color-coding makes it much easier to remember arbitrary information such as gender, tone or the identity of an atom on a molecular diagram. In Anki, color-coding must be done manually via a color-picker which creates a new window, which creates friction in card design. This becomes tedious when making more than a few cards in a session. I eventually got around this in my own vocabulary cards by adding a custom field to my vocabulary cards which takes a letter for gender (*m*, *f*, *n* or *e*[2]) and, via custom Javascript, applies corresponding coloring or highlighting to the text of the word. But it was not at all obvious that this is possible; it did not occur to me until after a few years of heavy Anki usage. We would prefer to do something like this:



Here, the 'gender' field has a 'value on card' field (*der, die, das*), and triggers color-coding of itself and the German lemma on the back side of all cards. Similarly, it should be possible to add basic formatting such as **boldness**, *italicization* or underlining to text via clicking a button, as in Microsoft Word, rather than using HTML tags[3].

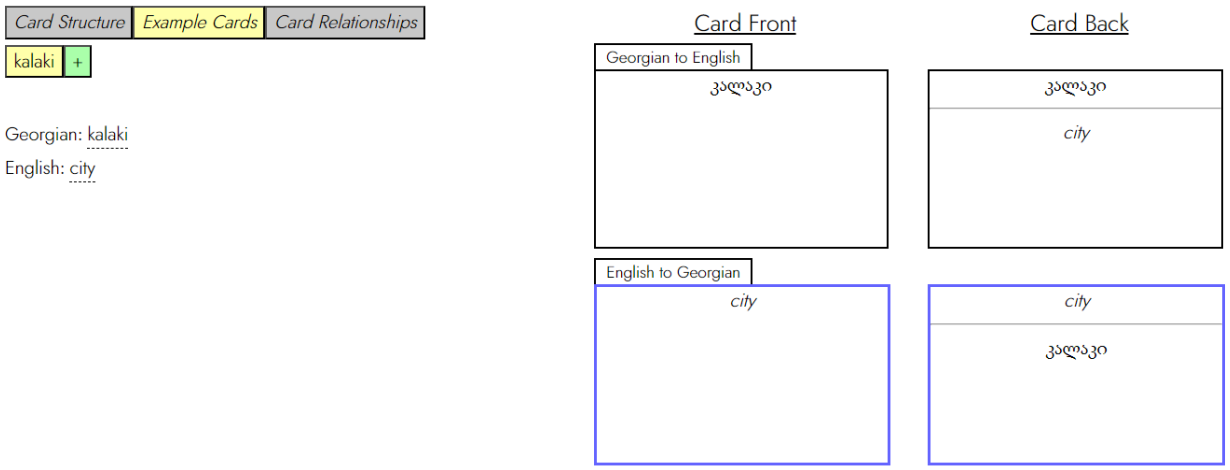
This is not to say that custom Javascript should not be possible. Power users of a replacement for Anki should, generally, be able to do anything power users of Anki

[2] For epicene nouns, e.g. Latin *sacerdōs* or Hebrew תְּהוֹם, which may be either masculine or feminine.
[3] This is possible in Anki's card-creator, but it cannot be done automatically in the card-template designer without using HTML.

currently can. But it is **much** more important that casual-to-medium-intensity users of the replacement be able to do *most* things that Anki power users can, including automatic custom formatting and good card design, without feeling overwhelmed.

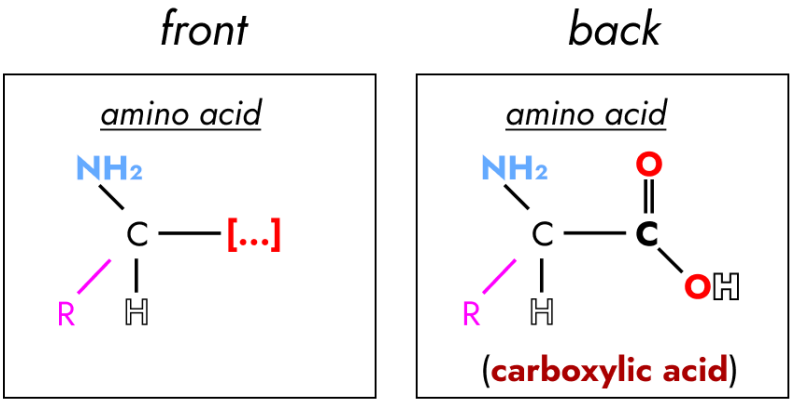
- Students of mathematics or physics—or any natural science where mathematical formulas are key—are also underserved by Anki. Anki's built-in MathJax library has no access to color-coding or other formatting; it allows cloze deletions, but these are black-and-white and thus difficult to see. More advanced math cards must be made with LaTeX, which adds yet another steep learning curve. A WYSIWYG math editor of the sort that WolframAlpha uses, with formatting options, is both feasible and desirable.

- Language-learners must switch keyboards, or copy and paste from a third-party program, to use diacritics or other alphabets; this, again, creates a great deal of friction in card manufacture. There is no reason that users should not be able to define or pick a custom text-replacement mapping that would apply to certain fields, as shown here for Georgian:



As with automatic coloring, this can be done in Anki with Javascript, but doing so is tedious, especially for alphabets other than Latin. It should not require Javascript to convert <chislo> to <число>, <mantha'nw> to <μανθάνω>, or even <k'Atab> to <כָּתַב> without the user having to give it more than half a second of thought[4]. Any and all "friction" in card creation and design should have a pedagogical purpose; **tediousness is the enemy**.

- Cloze deletions are powerful and efficient, but are built mainly for text; while Anki has Image Occlusion, it is new, and rather inelegant. There is no built-in diagram maker or any other way to handle diagrams or pictures other than as entirely static image files. Students of chemistry or biology might want to be able to do something like this:



Anki has the ability to do this ("image occlusion"), but it is weak and was only added very recently, almost two decades years after release; in its competitors (and in my day-to-day usage) two separate images must be designed in a third-party application such as Figma and copy-pasted into the card maker. While a viable replacement does not need the full powers of real image-editing software early on, a lightweight version for simple image-editing and diagram-making, built with common card formats in mind, is certainly feasible.

[4] Chinese characters or Japanese kanji will not, of course, be quite so simple; but text-to-character libraries are numerous.

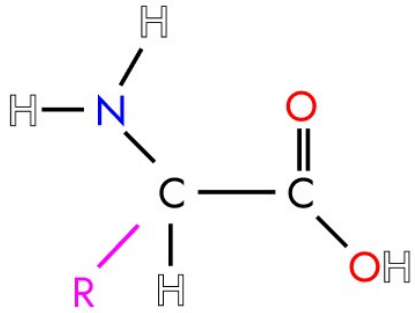
- Anki has no built-in way to manufacture cards *en masse*; users must use Excel and export a CSV file to do so. This is not a big ask for power users, but it represents yet another barrier to adoption by people who want things to Just Work—a population which includes most students and teachers.

Every single one of these deficiencies is solvable, and some (such as automatic text replacement or a WYSIWYG markdown editor) are very easy to solve. But Anki's developers have never thought to solve them, and Anki therefore remains a niche product.

This section would be incomplete without a discussion of LLMs and other artificial intelligence applications; many Anki users have used LLMs to great success in creating mnemonic images for things they need to learn. There is indubitably a place in a modern, classroom-based SRS platform for LLM usage; a carefully-tuned AI would further reduce friction in card construction and editing, make card construction much faster, and help pinpoint students' weaknesses in reviews (which, it bears repeating, students are quite bad at). Even then, however, there is a great deal of low-hanging fruit to be picked in SRS design that does not require the use of AI. Nothing I have discussed in this section requires an AI/LLM, and every other proposal in this white paper could have been built and deployed fifteen years ago.

IV. Card Relationships

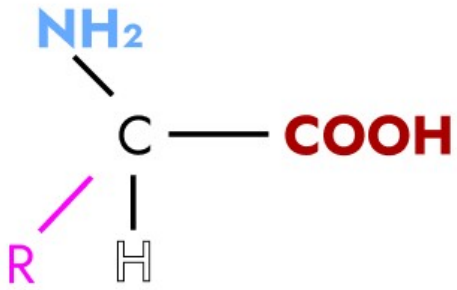
Consider, again, the structure of an amino acid, with atoms color-coded for convenience. How do we make a good card out of this?



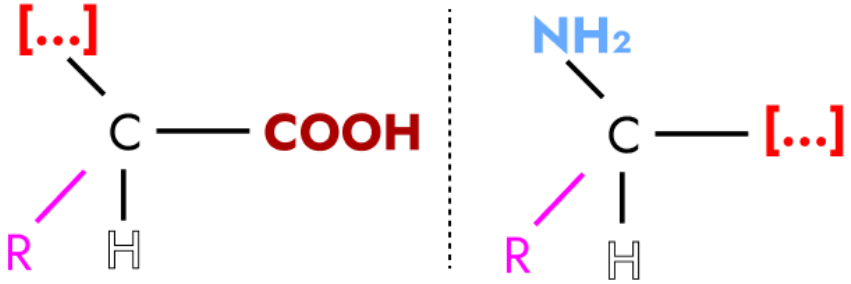
The general consensus among SRS enthusiasts is that 'draw an amino acid from scratch' is a bad prompt, because it's insufficiently atomic. It demands too much information of the user at once, and should be broken up into parts.

I disagree. A biochemistry student may indeed be asked to draw the structure of an amino acid from scratch on a quiz; that's a perfectly valid prompt and one that we might well wish to review. What *is* true is that if the student tries to learn the structure of an amino acid from scratch in one gulp, they are likely to fail.

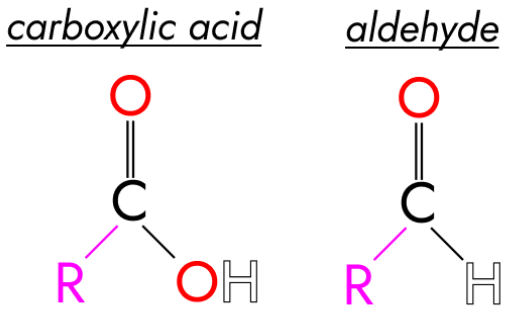
This is not because it is impossible to store the structure in one's head as a single chunk. Rather, packaging it into a single memory chunk is possible *only after its constituent sub-chunks have been learned*. An amino acid, like all organic molecules with more than a handful of atoms, is composed of multiple recognizable subcomponents. For an amino acid, the two recognizable functional groups are the amine to the "northwest" of the bonding carbon, and the carboxylic acid group to its "east".



Solidifying the structure of an amino acid into a single chunk thus has prerequisites: we must know the functional groups. We can imagine the following cloze deletions to test that knowledge:

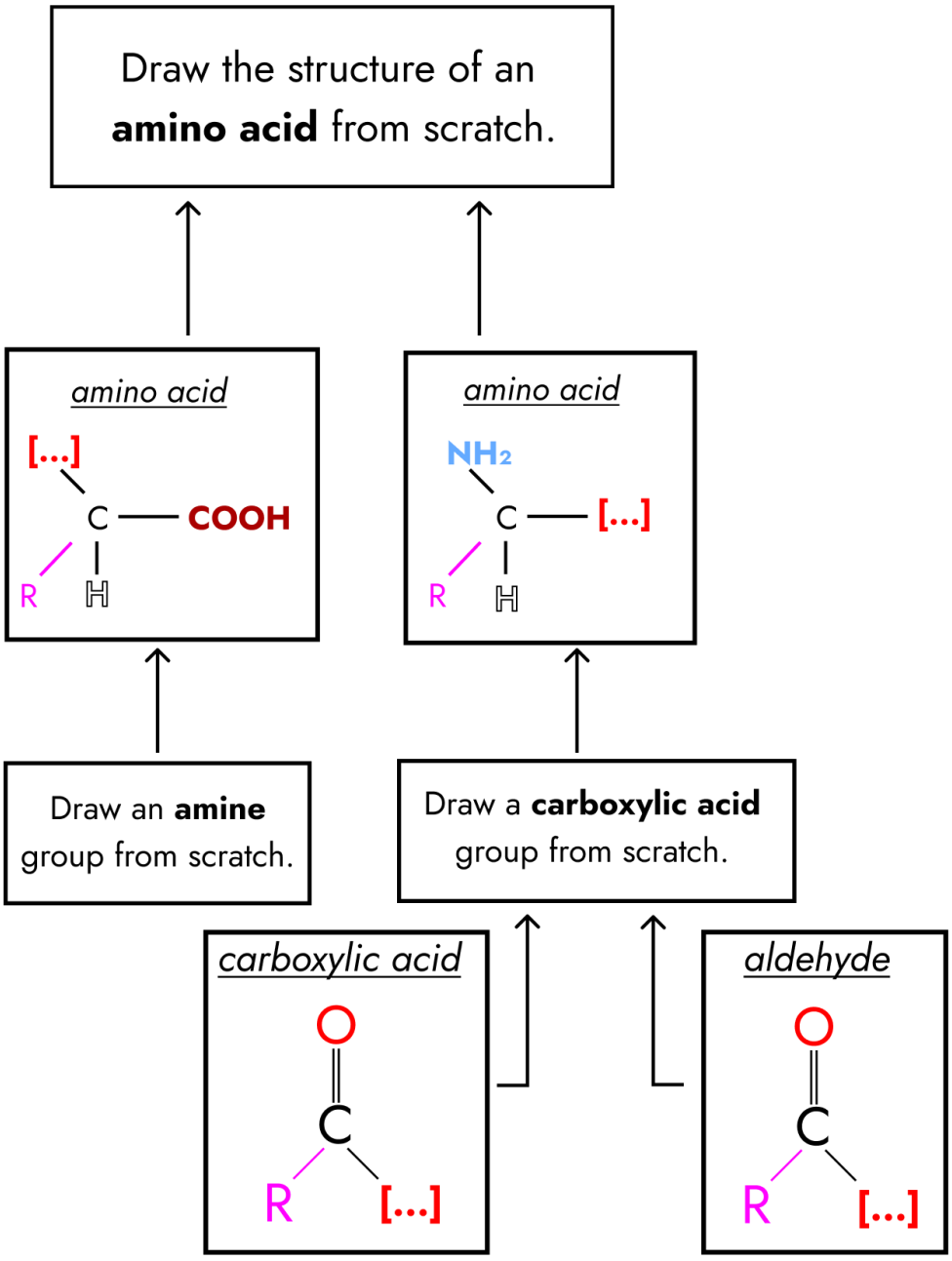


Even these might themselves have prerequisites. If a carboxylic acid group's hydroxide is swapped out for a hydrogen atom, the result is an aldehyde group.

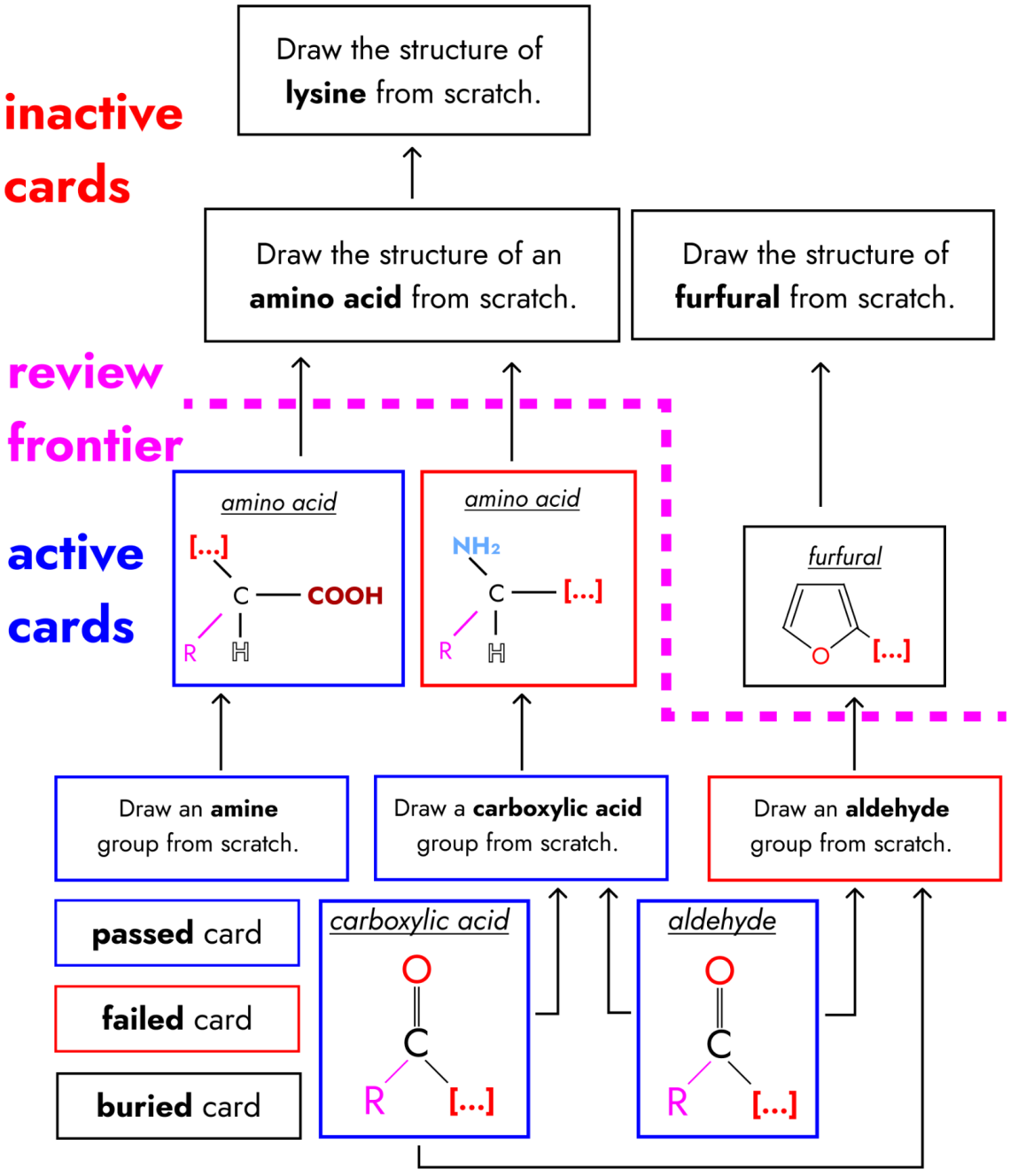


The structure of an amino acid can certainly be stored in a single mental chunk, but not until its component chunks have been 'solidified'—a process which itself relies on those components' subcomponents[5]. In the following diagram, an arrow *A* → *B* means 'card A is a prerequisite for card B':

[5] A more in-depth, albeit somewhat meandering, overview of this model of memory and its application to SRS software may be found in my earlier four-part blog series *Why Anki Works*.



We probably don't want to test a student on the full structure of an amino acid until we're fairly sure they have a decent handle on its prerequisites—doing so just creates mental friction. There will be a *frontier* of active cards based on whether or not all their prerequisites have been, if not mastered, at least marked as 'correct' on the last review.



Anki's card-relationship abilities begin and end with "sibling burial". If a template produces more than one card—e.g. a template for vocabulary cards which produces one English → Spanish card and one Spanish → English card—then those cards are considered

'siblings', and we can set Anki to 'bury' cards on days when one of their siblings is due for review. In effect, this tells Anki to show only one card from each note[6] per day.

But ideally, we'd like a replacement for Anki to 'follow the mind' and do a better job of imitating and reinforcing memories as they are actually structured: as a **network** of chunks and concepts rather than a collection of atomized facts. There is an add-on that allows this sort of hierarchical burial structure, but it only applies to cloze cards, and requires a certain amount of hacking Anki's review system into doing things it wasn't quite envisioned to do. There is no way to do this for other kinds of cards or for cards belonging to different notes, and even if you try to use the sibling-burial system to proxy it, you'd need to create a new template for every separate card-prerequisite hierarchy. The result is that the same chunk of information has to be added to, and reviewed in, cards from multiple templates, which means those cards interfere with each others' memory curves.

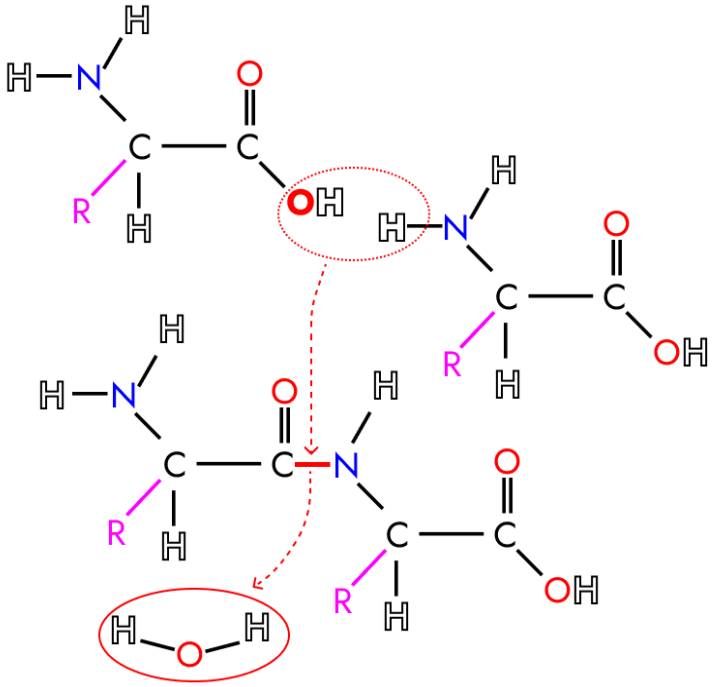
The skeptical reader might wonder whether or not this just adds needlessly to a student's workload: *Isn't it enough to just have a single card for the structure of an amino acid, and save yourself a lot of effort and time writing prerequisite cards?* As is often the case with new technology, it is all too easy to make intuitive but incorrect analogies with the physical world. There is a limit to the usefulness of physical flashcards, dictated by the size of one's collection and the increasing amount of time required to keep that collection organized as it grows. Even systems such as Leitner boxes, the first popular implementation of spaced repetition, are limited in practice to a couple thousand cards at most; beyond this number the user has to spend more time keeping track of cards than reviewing them. In 1972, there was no convenient method of tying a card on the structure of an amino acid to its prerequisites, and there were only so many prerequisite cards that made sense in practice before the deck collapsed under its own weight. An electronic card system, on the other hand, allows for a practically unlimited number of cards to be added and tracked; reports of Anki decks with a six-figure number of cards are not unheard of. The main limit is the user's patience for card manufacture (a process which, as I've established, is needlessly suboptimal in all established electronic study aids).

---

[6] A note is an instantiation of a template. In this example, *perro* → *dog* and *dog* → *perro* will be different cards, but belong to the same note (making them siblings).

This still leaves the question, however, of whether or not making large numbers of prerequisite cards is sensible on its own merits. My own experience and the consensus among power users of SRS software lead me to answer with a firm **yes**: new knowledge without prerequisites is virtually useless.

Suppose that, via brute-force repetition—spaced or otherwise—we have trained a high-school chemistry student to draw an amino acid from scratch, with no background knowledge other than the identities of its component atoms. Our hypothetical student has learned *something*, to be sure, but with little to no intermediate background knowledge, we cannot say that they 'know' the structure of an amino acid in ways that allow for further connection and application. Proteins, for example, are formed via peptide bonds between amino acids, in which the carboxylic acid of one amino acid and the amine of a second are joined to give a water molecule as a byproduct.



This knowledge is out of reach for the student who has only learned the structure as 'parrot-knowledge'. It requires the student to know, actively, that amino acids possess an amine group and a carboxylic acid group, what the structures of those groups are, and how they interact.

It is a common misconception that facts and content knowledge are separate from or secondary to critical thinking and analysis skills; on the contrary, they are prerequisites. The dreaded 'rote memorization' becomes a problem when a) it is pursued to the *exclusion* of analytical thinking, rather than as a necessary complement to it, and b) feeds students 'orphaned knowledge' like our hypothetical amino acid example, with no built-in connections on which analytical skills can operate. It is ultimately up to teachers to avoid the first failure mode, but tools such as SRS can greatly help with the second—**if** those tools are powerful and flexible while still being convenient and intuitive for normal users. Building such a tool constitutes the task before us.

V. Spaced Repetition for the Classroom

I used to teach middle-school Latin, and tried getting my students to use Anki. A few took to it like ducks to water and, as far as I know, are still using it to great success. Most didn't, for a number of reasons:

- As noted in section II, many of my students used Chromebooks, which meant Anki was inaccessible to them.

- Most schoolchildren make poor autodidacts. They are bad at recognizing what they're having trouble with and even worse at creating cards or study materials to hone in on that issue. For example, a student might have trouble remembering the difference between *dūcō* 'I lead' and *dīcō* 'I say'; they only differ in a single vowel, so the wires are easily crossed. A cloze deletion of the form ***d[...]cō*** is a good way to review this, but knowing that you need to do that requires a level of metacognition most students don't develop until at least high school. Teachers need to be able to do that for them…

- …but the only way for teachers to do so is to make shared decks for download. There is a consensus among Anki devotees that shared decks mostly work well for 'prepackaged' bodies of knowledge such as medical school courses; I've never been able to get anywhere with premade vocabulary decks (and it's probably not coincidental that a medical student's metacognition abilities are far better than the

average eighth-grader's). The only way to edit or tweak a shared deck after it's been downloaded is to ask students to download it again, which destroys their progress. Ideally, a teacher should be able to update, add or delete cards from students' decks as needed.

- Even if it were possible—in, say, a school where all the students are given a Macbook—to get everyone using Anki, students' progress must be checked manually, student by student. There is no easy way to see what cards are giving students trouble, or whether students have been doing reviews at all.

- Autodidacts can generally be trusted to be honest with their reviews. Students can't be; left to their own devices, they will just press 'easy' on any card they don't want to deal with (I would certainly have done this in high school if there were no checks). You could make all cards type-in-the-answer, but my experience has been that this creates friction and makes the experience unpleasant.

  A potential alternative is to check students on a *selection* of reviews using 'quiz mode'. At the end of a session, our SRS system can pull a handful of cards from that day's review and the most recent prior reviews that the student claims they passed, and ask them to type in the answer with a time limit[7]. Incorrect answers, or answers requiring the student to draw a diagram or speak into a microphone, can be sent to the teacher anonymously to be quickly graded (perhaps the student made a typo). Incorrect answers will be marked as a fail on the associated card; at the end of the quarter the teacher can pull up quiz-mode results and add them to the gradebook.

- Anki is not-for-profit, though as noted in the introduction, it seems to be something of a golden goose for its top developers. But a good SRS system designed for classroom usage is something at least some schools might pay for.

  I should emphasize that this does not mean that the design of such a system needs to revolve around the classroom to the exclusion of individual usage. As noted, there

[7] The time limit could, of course, be adjusted for students with extended-time accommodations.

are a number of simple possible improvements to Anki's design that would greatly enhance the individual experience as well. But, at this point in time, there is no SRS system on the market—open-source or otherwise—which is built for classroom usage by default[8]. Contemporary education is beset by the *bêtes noires* of learning loss and subpar content knowledge; compared to their counterparts twenty or even ten years ago, today's students are learning less at school, and forgetting more of what they do learn. Spaced repetition is not a panacea for these (no single platform or application can be), but it *does* constitute an effective tool that schools and teachers can and should be using—*if* that tool works out-of-the-box in a classroom context, is designed to minimize friction for teachers and students without sacrificing customizability, and is built for the browser.

VI. Prospects and Challenges

Edtech is a notoriously difficult sector to do well in; while public school districts have large budgets, much of which they waste by any halfway objective measure, their willingness to try something new is constrained by strict procurement rules. Early on, then, the target market is likely to be autodidacts, homeschooling pods, and private schools, in approximately that order.

However, if early hurdles in execution can be surmounted to reach that market, the fundamentals are excellent. American education is a gigantic market, with 54 million students in K-12 and about 18 million in higher education. Of those 54 million, about 9% are enrolled in private schools—but that proportion is set to grow considerably over the next

[8] I should note that in the past few years there have been tentative steps in this direction by a handful of established educational technology companies. Quizlet, for example, has recently introduced a spaced-repetition mode. But these are generally afterthoughts. Quizlet's structure is built for creating, and reviewing, a single deck of cards in one session for a specific unit—polyatomic ions, for example, or the vocabulary from Chapter 6 of the French textbook. This **does not** make Quizlet a true spaced-repetition system; the real power of Anki or Mnemosyne derives from the fact that a user may have four or even five figures' worth of cards in a single deck, but will only need to review a tiny fraction of that number on a given day.

decade or two as the voucher movement builds momentum in conservative and moderate states.

There are a number of highly successful study tools far less powerful than what I am outlining here. Quizlet, for example, claims that two out of three high school students in the United States use its platform. This seems exaggerated, but even if the true number is closer to half, this represents a market of over eight million students—and Quizlet is far less effective than it could be, as its card-architecture system is primitive and its spaced-repetition abilities are an afterthought. It is not enough to simply *build* a better mousetrap, of course, but when it comes to effective studying and content review, the whole is greater than the sum of its parts. In short, we are looking to combine the following:

1. A powerful SRS platform that can handle upwards of tens of thousands of cards (like Anki), based on the latest advances in spaced repetition. Anki has recently switched to a new algorithm called the Free Spaced Repetition Scheduler, or FSRS, which is represents a considerable improvement on Anki's old scheduler, and—best of all—is not proprietary.

2. A powerful, flexible template- and card-design system that allows multiple fields and custom formatting—like Anki's—but which is clear and intuitive enough that students and teachers who have never written a line of code can pick it up and use it with a minimum of friction. To my knowledge, **no competitors offer this.**

3. Web-based, classroom-centered review that allows teachers to design cards for students, see how students are doing, and actively check students' knowledge on a subset of cards to make sure they're doing reviews honestly. This exists on platforms such as Quizlet; Quizlet charges teachers extra to check on students' reviews, so the ability is something teachers and schools are willing to pay for.

4. Advanced card relationships that mimic the structure of knowledge in the brain, as a hierarchy of interconnected chunks rather than a pool of undifferentiated, isolated

factoids. I am unaware of **any** platform that offers this, but a simple dependency network between cards is not at all difficult to implement.

Electronic spaced-repetition systems have been around for over three decades at this point (since SuperMemo's 1991 debut)—so why is a new one necessary? I answer that, when it comes to study software, **the whole is greater than the sum of its parts**. Students and teachers are busy, and few are enthused by the prospect of writing Javascript on a card template. All SRS systems on the market today are either much weaker than they could be (such as Quizlet), or are hobbled by a great deal of friction in card manufacture and review that creates an unnecessary learning curve for new users and can make the process of studying frustrating even for experienced ones (such as Anki). There are, additionally, relatively simple improvements (such as the 'review frontier') that could be added to existing systems to great effect, but never have been. The competition is therefore less daunting than it seems, because we are not really trying to build another Quizlet or another Anki, but something that is considerably more useful and effective than either of them. We could describe a smartphone to someone from the year 2000 as 'a phone that can browse the Internet', but this would leave out what has made it so life-changing.

I have no doubt that a product that hits all of these requirements is one that could make hundreds of millions per year. Thirty to forty dollars per year per student is by no means extortionate for educational software, and schools will be willing to pay for a system that Just Works cleanly and easily. Given a rough approximation of 30 million students in grades 6-12 in the United States, an average price of $40 a year per student represents potential revenues of up to $1.2 billion a year. While it is unlikely that *every* school in the country would line up to buy a license, even a third of the age cohort would mean $400M in revenue a year. At an 80% profit margin and a price-to-earnings ratio of 25, this would be a $12.5B company.

Building a platform of this type will certainly be challenging. But it is feasible, and there has never been a better time to do so. There is little competition and a large, well-funded market facing serious challenges a modern SRS system can offer solutions to. It's time to build one.

Campbell Nilsen
www.twitter.com/nephew_jonathan
campbell.nilsen@gmail.com