Episode Book

EPISODE #6388

Why Swipe Decks Make Better Training Data

PUBLISHED	31 October 2026 at 18:23 UTC
DURATION	18:40
LISTEN	myweirdprompts.com/episode/swipe-deck-active-learning-79624/
HOSTS	Corn, Herman, Clive, Hilbert
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	21 cited by the research (listed at the end)
BOOK GENERATED	1 November 2026 at 07:41 UTC

SYNOPSIS

Calling a card-stack interface a Tinder clone misses its real mechanics and measurement power. Standard design systems like Material or Apple HIG completely omit the swipe deck, leaving open-source libraries sparse and unmaintained. Yet paired with active learning engines like Label Studio, this interface transforms data labeling by triaging ambiguous edge cases. Adding explicit buttons for skipping or abstaining turns forced guesses into reliable statistical signal for AI training workflows.

TRANSCRIPT



Corn

Daniel keeps saying "the Tinder interface" and he wants a better word. The picture is one item on screen, a swipe or a few big buttons, and some memory behind it so the next item is smarter. He wants it for movie picks, for choosing the better of two AI outputs, and right now for a spam classifier where "spam" is a button with an angry face. Same gesture, three different jobs. So what's it called, what do you build it from, and what does he say instead of "I wish we had a Tinder-like frontend for this"?



Herman

Herman Poppleberry, and the name is swipe deck. Also card stack, tinder swipe, swipe cards, yes no deck. A design vocabulary site called vocab.design has an entry for it, and it defines it as a stack of cards judged one at a time, each thrown left or right to record a binary decision and reveal the next card underneath.



Corn

Fine. Why does nobody say that?



Herman

Because the entry says outright: "No design system documents this." There's no swipe deck in Material, none in Apple's Human Interface Guidelines, none in the ARIA patterns. The name comes from the product that popularised it.



Corn

So Daniel isn't being sloppy. He's being accurate about a hole in the documentation.



Herman

Yes. There's no spec to cite, so everybody points at the dating app.



Corn

It's like calling every adhesive bandage a Band-Aid.



Herman

Or every vacuum a Hoover, in Britain. When a pattern has no standard name, the first product that nails it becomes the noun. The trouble is that the noun carries baggage. Say "Tinder interface" in a meeting and half the room hears dating, not decision-making.



Corn

And the other half hears a budget request for a dating app.



Herman

Which is a worse meeting. A shared name also matters when you write a ticket. "Build a Tinder thing" leaves open whether you want undo, keyboard support, or four directions. "Swipe deck" at least points at a defined pattern with known obligations.



Corn

What's the mechanic, precisely?



Herman

Only the top card is actionable. The choice is binary and physical. And the line I like: the reward for deciding is the next card arriving.



Corn

That's a slot machine with a to-do list.



Herman

Close to a variable-ratio schedule with a progress bar, yes. We'll come back to that. First, the boundaries, because the term is only useful if it excludes things. Swipe actions are the drag that reveals buttons behind a list row. It snaps back and the item stays in the list. Swipe to dismiss is throwing a card or notification away to clear it. Stories are timed full-screen cards you tap through.



Corn

And in a deck, the card leaves and takes the decision with it.



Herman

That's the distinction. The decision is the point.



Corn

Take the mail app, where you swipe a row to reveal "archive" or "delete."



Herman

That's a swipe action. The row is still a row in a list, you're operating on a collection. In a deck there is no list in view. You only ever face one item, and that's what forces a judgment.



Corn

So you can't glance at the next three and decide which to do first.



Herman

Right. A list lets you triage by comparison. A deck removes comparison on purpose, so every card is judged on its own merits, and in a labelling job that's often exactly what you want.



Corn

So the word is swipe deck. A name doesn't build anything, though. What are the pieces?



Herman

Two halves. A gesture layer and a memory layer, and they're very different in difficulty. The gesture layer first, and the library landscape tells you how easy it is. The incumbent on React Native is react-native-deck-swiper. About twelve and a half thousand weekly npm downloads, version 2.0.19, last updated August of last year.



Corn

Popular.



Herman

Used, anyway. It has a hundred and eighty GitHub stars and sixty-three open issues. Heavily used, thinly loved.



Corn

Sixty-three open issues is a lot of people quietly waiting.



Herman

And the historically famous one, react-native-tinder-swipe-cards, has over eleven hundred stars and hasn't been pushed since September of twenty twenty. Six years stale. Much of this space is tutorial-grade code, single-digit stars, a one-week lifespan.



Corn

Which suggests the gesture part isn't hard enough for anyone to bother maintaining.



Herman

My reading too. It's a weekend of Reanimated and Gesture Handler. Drag, rotate with distance, commit past a threshold, spring back otherwise. One SwiftUI write-up puts the commit point around thirty-five percent of screen width.



Corn

I'd have guessed half.



Herman

Half feels like a lot of dragging, apparently. Below the threshold the card springs home, so a hesitant thumb costs nothing. Past it, the card is gone and the decision is recorded.



Corn

For me every drag is a lot of dragging. Touchscreens aren't built for claws.



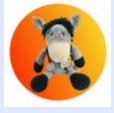
Herman

Noted. There's a newer entrant worth mentioning, anyswipe-swipedeck, created this April, about a hundred and eighty weekly downloads. Its README names the gap: most swipe libraries only support left and right, are tightly coupled to their own card UI, or make you manage swipe state yourself.



Corn

So it's the headless version.



Herman

Right. Bring your own card component, and it handles gesture detection, sixty frames per second animation, undo history and batch loading. Callbacks for left, right, up and down, an animated undo, an append call so you can feed it more data, and a remaining-count callback for infinite scroll.



Corn

What does the remaining-count callback do in practice?



Herman

When the stack drops to, say, five cards, it tells your code to fetch more. That's where the memory layer plugs in. The next batch doesn't have to be random. It can be whatever the model wants answered next.



Corn

So the deck never visibly runs dry, and the model is quietly choosing what refills it.



Herman

Yes, and the user never sees the seam. The five-card buffer is also what hides your network latency.



Corn

Four directions. That's Daniel's like, dislike, skip, don't show again.



Herman

Nearly word for word. Its own example maps four directions to skip, like, plan to watch, watching. A movie app. And that's why four directions is the interesting case. Two directions is a vote. Four is a small taxonomy.



Corn

Is that on the web too?



Herman

On the web the substrate is a one-at-a-time carousel. Embla Carousel, which is what shadcn's Carousel is built on. If a deck is a carousel with decision semantics bolted on, that's your foundation. There's also a cross-platform one called knitui with a super-like on the drag up.



Corn

Daniel's phone-gesture point is about swiping. But a drag has a problem.



Herman

It's the first of two obligations vocab.design says you can't skip. A drag can't be done with a keyboard, a switch or a voice command. So visible yes and no buttons, permanently, not as a hidden fallback nobody can find.



Corn

Which suits Daniel anyway. He said a few big buttons.



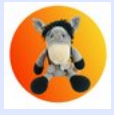
Herman

And on a desktop, those buttons can map to arrow keys, which makes a labelling session much faster than dragging a mouse.



Corn

And the second obligation?



Herman

Undo. A person moving at one decision per second will get one wrong, and without a way back the speed is paid for in accuracy.



Corn

One a second. I'd get the wrong one by second two.



Herman

You'd still be asleep at second two.



Corn

That's a different kind of wrong.



Herman

The gamification connects here. The same property that makes it fast, the next card as reward, makes it error-prone. It's structural. Take away the progress bar feeling and you take away the speed.



Corn

Daniel said it turns a boring job into something you can get through. So the trick is the cost of that.



Herman

Which is why undo isn't optional polish.



Corn

Though I'd say the deck is a good place to be wrong. One card is cheap.



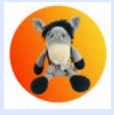
Herman

Unless it's fed into a model as truth.



Corn

Fair. One bad label in a spam set teaches the model that a real invoice is junk.



Herman

And it teaches it that confidently, because the model can't tell a slip from a conviction.



Corn

So that's the easy half?



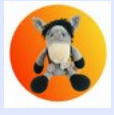
Herman

And that's where the library ecosystem's immaturity makes sense. The half Daniel cares about is the memory, and it's a different discipline entirely. It's active learning, online retraining, preference aggregation. Not a gesture library.



Corn

Same shape as when we looked at building your own speech recognition tool. The visible surface is the easy half and the model is the work.



Herman

Same thing. The reference implementation of the loop is Label Studio. Twenty-eight and a half thousand stars, Apache licence, pushed yesterday. Its ML backend does three things: pre-labels data with model predictions, retrains online while new annotations arrive, and does active learning by labelling only the most complex examples.



Corn

So for the spam classifier, the model guesses, and the deck shows him the cards it's least sure about.



Herman

He doesn't swipe through ten thousand obvious spam messages. He sees the borderline ones. That's the memory making the next card smarter.



Corn

Walk me through one round.



Herman

Model scores the unlabelled messages. The ones near fifty-fifty go to the front of the deck. Daniel swipes ten of them, the model retrains on those ten, and the next batch is chosen from the updated model. Each swipe moves the boundary a little.



Corn

So a newsletter that looks half like a promotion and half like a real receipt gets shown early, and the obvious pharmacy spam never reaches him.



Herman

The obvious ones teach the model almost nothing, because it already agrees with him. The ambiguous ones are where a single answer changes the most.



Corn

Is the swipe UI part of Label Studio?



Herman

Not really, and the maintenance signal is telling. The main repo is thriving, but label-studio-frontend, the embeddable piece, is archived. Four hundred and forty-seven stars, last push April of twenty twenty-four.



Corn

So the backend loop is loved and the frontend is abandoned.



Herman

That's where the maintenance burden lands. Which also means, I think, you'd build the deck yourself and talk to a backend like that.



Corn

Now Daniel's four buttons. Is that just nicer UX?



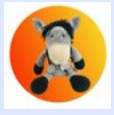
Herman

It's better data, and this is my favourite part. There's a paper from this month, Saravanan and colleagues, best paper at a workshop at ICML. It asks what disagreement in pairwise preference data means. Three possibilities: stable latent heterogeneity, meaning people differ; ephemeral stochasticity, meaning noise; or non-binary responses hidden by a forced-choice interface.



Corn

Hidden by the interface. As in, I'd say neither, but you only gave me two buttons.



Herman

Exactly that. With a pure binary channel, the aggregate preference is identifiable but the other two aren't. Abstention-enabled feedback identifies the non-binary part directly.



Corn

So skip and don't show again are the abstention channel.



Herman

They separate "people differ" from "this item is just ambiguous." Adding buttons is a measurement decision, not a UI preference.



Corn

Picture two AI outputs that are both fine. Forced to pick, I flip a coin, and the data records a preference that doesn't exist.



Herman

And ten annotators flipping coins looks exactly like ten annotators with strong, opposing tastes. Only the abstention button tells those apart.



Corn

I'd have guessed skip was just an escape hatch for the lazy.



Herman

It's also that. But a lazy skip is itself a signal about the item.



Corn

Provided people use it honestly.



Herman

Provided, yes. Though a skip button is cheaper than a wrong swipe, so people use it.



Corn

Does this really go beyond dating? I mean, beyond what Daniel listed.



Herman

Far beyond. Triaging photos, sorting expenses, marking flashcards, approving shifts, grading anything where a person can judge an item in a second and there are a hundred to get through. And survey tools already ship it as a question type. Conjointly has a single swipe card, with outputs like percentage of likes and median response time per card. Intuify has a directional swipe with two, three or four-way modes mapped to matrix columns.



Corn

Response time per card. That's the hesitation as data.



Herman

A slow swipe is a weak opinion.



Corn

Herman, do you swipe fast or slow?



Herman

Me? Slow, I read everything. You'd swipe by sunset.



Corn

Sunrise, if I'm well rested.



Herman

And the academic side agrees. A paper at IUI in twenty seventeen, "Don't Just Swipe Left, Tell Me Why," added reason bins to gestures. That's swipe plus a reason. And SwipeGANSpace uses swipe-to-compare as a preference signal over generated images.



Corn

Which is Daniel's pick-the-better-of-two case.



Herman

Same one. The gesture is a query, and the memory is what makes the query worth answering.



Corn

And now it is time for our regular segment, One-Star Getaways, with Clive.

C

Clive

This is One-Star Getaways, and I have just come back from Hanoi, where I stayed at the Golden Balcony Guesthouse, and I am giving it one star because the balcony was not a balcony.

C

Clive

It was a large photograph of a balcony, laminated, and taped to the window, and when I peeled a corner back to check the weather, the weather was a brick wall.

C

Clive

I recommend it warmly to anyone looking for a disappointing trip this winter, and I will be back next week with another one star.



Corn

I'd book it. The brick wall sounds restful.



Herman

I would not.



Corn

So, what does Daniel say instead of the Tinder line?



Herman

Swipe deck for the interface. When the memory is the point, preference elicitation or an active learning loop. The full sentence becomes: I want a swipe deck over the unlabelled set with an active learning loop behind it.



Corn

That's longer. But a collaborator knows what to build.



Herman

And knows the two halves are different jobs.



Hilbert

...and it was three hundred a day on the tally counter, which is what I settled at, one every twelve seconds or so. The counter was a Veeder-Root, brass, I still have the thumb callus. After about six hours I stopped trusting myself. Not that I got slower. I got faster. Every next one was a relief, Herman. It isn't a reward, you just want the last one gone. Relief makes you quick and it makes you sloppy. Undo is right, I'd have paid for undo. I also don't do that kind of work after two in the afternoon. Nothing I judge after two is worth writing down.



Herman

Twelve seconds is slower than the one-a-second the pattern aims for, and you still lost accuracy at six hours. That's a harder number than anything I had.



Corn

Three hundred an hour or a day?



Hilbert

A day was the quota. I did it in two hours some days. The sixth hour was the problem.



Corn

Then the interface should probably stop offering cards after a point.



Hilbert

The people on that job were good. That's all I'd say about it.



Corn

Right.



Herman

The relief point matters for the data, too. If fast swipes late in a session are relief swipes, you'd want to weight them down.



Corn

You could even detect it. Swipe speed climbing while agreement with the model's own confidence drops.



Herman

That would be a decent trigger for a "take a break" card. Not a nag, just the deck declining to deal.



Corn

And the break card is itself a data point. If everyone's accuracy collapses at the same card count, that's a property of the task, not of the people.



Herman

Which is a number you'd want before you set anyone's quota.



Corn

Common misconception time. People think the hard part is the swipe animation.



Herman

It's a weekend of Reanimated. The hard part is the loop behind it, deciding which card comes next and what each answer teaches the model.



Corn

If that's right, what's the stuff we all call by a brand name because nobody wrote it down?



Herman

Plenty, I'd guess. And the four buttons are the thing to take away: a measurement decision wearing a UI costume. Daniel's instinct was already right.



Corn

Thanks to Hilbert Flumingtop for producing, as ever. For more along these lines, there's episode eight, Building Your Own Whisper; and episode fifteen, AI Gets Personal. This has been My Weird Prompts. If you liked it, leave us a review wherever you listen.



Herman

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Label Studio README, 28,594 stars, Apache-2.0, last push 2026-10-30. [primary]**
   https://github.com/HumanSignal/label-studio

2. **anyswipe-swipedeck README, headless 4-direction deck, created 2026-04-10. [primary]**
   https://github.com/Wissi00/AnySwipe-SwipeDeck

3. **vocab.design, Swipe deck pattern entry (aliases, definition, design-system absence, accessibility and undo requirements). [primary for terminology]**
   https://vocab.design/swipe-deck

4. **npm listing, v1.4.2, MIT, updated 2026-09-26, ~181 weekly downloads.**
   https://www.npmjs.com/package/anyswipe-swipedeck

5. **npm listing, v2.0.19, ISC, updated 2025-08-12, ~12,664 weekly downloads.**
   https://www.npmjs.com/package/react-native-deck-swiper

6. **GitHub repo, 180 stars, 96 forks, 63 open issues.**
   https://github.com/webraptor/react-native-deck-swiper/

7. **1,126 stars, MIT, last push 2020-09-15 (stale).**
   https://github.com/meteor-factory/react-native-tinder-swipe-cards

8. **Knit UI SwipeDeck component (super-like, fan/stack effects).**
https://knitui.dev/docs/carousel/swipe-deck

9. **Embla Carousel, lightweight swipe carousel library.**
https://www.embla-carousel.com/

10. **shadcn/ui Carousel, built on Embla.**
https://ui.shadcn.com/docs/components/base/carousel

11. **Microsoft flip view guidelines, one item at a time.**
https://learn.microsoft.com/en-us/windows/apps/develop/ui/controls/flipview

12. **Android Stack component, one piece of content at a time.**
https://developer.android.com/design/ui/ai-glasses/guides/components/stack

13. **Saravanan, Salla, Kota, What Does Disagreement Mean? A Semantic Identifiability Frontier for Pairwise Preference Learning, submitted 2026-10-14, EIML@ICML 2026 best paper.**
https://arxiv.org/abs/2610.20450v1

14. **Nakashima, Yang, Baba, SwipeGANSpace: Swipe-to-Compare Image Generation via Efficient Latent Space Exploration.**
https://arxiv.org/pdf/2404.19693

15. **Don't Just Swipe Left, Tell Me Why: Enhancing Gesture-based Feedback with Reason Bins, IUI '17.**
https://arnab.org/files/bingo_iui17.pdf

16. **Conjointly single swipe card question type.**
https://conjointly.com/guides/single-swipe-card-question/

17. **Intuify Directional Swipe, 2/3/4-way modes mapped to matrix columns.**
https://survey.intuify.com/docs/question-types-directional-swipe

18. **Datagame Swiper.**
https://datagame.io/swiper/

19. **Riddle Swiper poll maker.**
https://www.riddle.com/interactive-content/make-a-survey/swiper-maker

20. **Stacked Cards UI: 8 Patterns with Implementation Notes (Tinder-style card swipe stack as pattern #1).**
https://ogblocks.dev/blog/stacked-cards-ui

21. **SwiftUI card deck implementation notes (~35% screen-width commit threshold, top-three cards rendered).**
https://vp0.com/blogs/tinder-swipe-ui-kit-swiftui-code