



Conducting Between Roller Coasters

JUNE 2026

5 min read • 1,228 words

I almost didn't write this down, because the sentence sounds like a brag and it isn't one. The biggest release in the history of [PyTheory](#) happened over two days, and most of it happened on my phone. On vacation. With two thumbs.

The interesting part isn't the phone. It's what the phone stopped being in the way of.

[Last week I wrote about taking PyTheory's gate down](#), the `pip install` wall that kept everyone who isn't a programmer standing outside the library. That essay was about access for other people. This one is about something that happened to me while I was supposedly resting.

What shipped

A lot. More than belongs in a paragraph, so here is the paragraph anyway.

The one I'm proudest of is thirty-six Hindustani ragas, in real shruti tuning. Not just the ten parent thaats the library already knew, but the living ragas: Yaman, Bhairav, Malkauns, Darbari, each with its ascending and descending line, its catch-phrase, its time of day, its rasa. And they play back in just intonation off the twenty-two shruti ratios, so a Ga sits where it is actually meant to sit, about fourteen cents under the piano, where it is supposed to ring with that aching

sweetness. Daksh, a friend who actually plays this music, asked for it, and getting that one detail right mattered to me more than the rest of the release combined.

Then the theory itself grew up. Negative harmony, the Ernst Levy mirror that turns a chord into its shadow. Reharmonization, the move a jazz pianist makes on a tired progression. Cadence detection, secondary dominants, non-chord-tone analysis, neo-Riemannian transformations, twelve-tone rows with the full matrix.

The quieter additions, for completeness: scalable fretboard diagrams you can drop into a video, a metronome with a tempo trainer that ramps the BPM while you practice, the circle of fifths, and the diatonic chords sorted by what they actually do instead of just listed in order. Boring, useful, done.

Most of this I taught myself. I half-attended one theory class in high school and learned nothing from it; my real training is as a percussionist, classical, which means rhythm lives in my hands and the harmony I picked up later, on my own and in pieces. It is a strange feeling to hand a machine the gaps in your own education and get them back filled in, tested, and playable.

And none of it stayed put. The same features went everywhere PyTheory lives, and the places they landed got real upgrades of their own in the process. [The browser playground](#) grew up alongside the library, so all of this runs with nothing to install. [The Ableton Live extension](#) did too, so you can reharmonize a clip or print a raga without leaving your session. And underneath all of it the rendering engine got the biggest overhaul of the bunch: a stereo-linked master bus with a soft-knee limiter, reverb that is finally stereo, a real convolution hall, tails that ring out instead of clipping, and rendering about twice as fast.

Here is the whole two days, release by release, because I can't pretend I'm not showing off:

- **v0.53.0**

- Twenty living Hindustani ragas, in just-intonation shruti tuning.
- SVG fretboard diagrams — chord boxes, the five pentatonic and seven diatonic scale shapes, arpeggio maps.
- A metronome with chord-stab clicks and a tempo trainer that ramps the BPM.

- The circle of fifths at the key level.
- Diatonic chords grouped by harmonic function.
- Negative harmony.

- **v0.53.1**

- Sixteen more ragas, thirty-six in all.
- A hall reverb on playback, bleeding each note's tail into the next.

- **v0.54.0**

- Reharmonization, for a single chord or a whole progression.
- Cadence detection.
- Secondary-dominant detection.
- Chord-scale theory, with avoid-notes.
- Non-chord-tone analysis.
- A part-writing checker — parallel fifths, octaves, voice crossing.
- Neo-Riemannian P/L/R transformations, with Tonnetz paths.
- Twelve-tone rows with the full 12×12 matrix.
- A pitch-class-set toolkit.
- Thirty-four built-in progressions, behind a fixed case-sensitive Roman-numeral parser.
- A `pytheory analyze` command.
- `--json` and `--play` across the CLI.
- Headless `Score.render()` and `to_wav()`.
- Reverb tails that ring out instead of clipping.
- A real convolution "hall" reverb.
- Genuinely stereo reverb.
- A stereo-linked master bus with a soft-knee limiter.
- Rendering about twice as fast.

- **v0.54.1**

- The corrected Roman-numeral progressions, finally matching their docs on PyPI.

\$ uv add pytheory , if you want to see for yourself.

The part I can't get over

I have shipped software for twenty years. I have never once been able to do work like this from a bench with a phone and a roller coaster screaming past fifty feet away.

Here is how it actually went. I would be waiting in line for a coaster, or waiting for my family to come back from one I had sat out, and I would open [Claude Code](#) on my phone and ask it the same plain question I kept coming back to. How could this be better? It would think, and propose things, a feature to add, something to cut, a rough edge worth hardening, and we would sort through them right there in the queue and decide what to make. By the time the ride let out it was often done. Committed, tested, documented, with a note explaining what it had decided while my wife and stepkids were upside down over Virginia. The one thing it couldn't do from the cloud was push the release to PyPI, so that last button waited for me.

It did not feel like coding. It felt like conducting.

I want to be honest about the seams, because the seams are the whole story. The machine made mistakes. It reached for a synth that was wrong for a solo line, set a detune so wide the chord smeared, forgot that marching music is always at one hundred twenty. The judgment stayed mine. The ear stayed mine. What changed is that the distance between an idea and a working, released feature collapsed to almost nothing. I could stand in that small distance from anywhere, including a place I had traveled to specifically in order to stand nowhere near my work.

There is something tender in that, and I don't fully know what to do with it yet. For most of my life, making things meant sitting alone in a room with my head down, paying a real and sometimes ruinous cost for the privilege. This was the opposite. This was loose and unserious and somehow social, even though it was

only me and a phone and the first coasters being run empty against the morning. I would think of a sound, and a few minutes later I could hear the sound. The grief that usually lives in the gap between those two moments was simply gone.

That is the whole thesis of for humans, the phrase I have been chasing since [a library about HTTP](#). The tool should meet you where you are. I did not expect to learn that "where you are" could be a roller-coaster line, mildly sunburned, thinking about a flat third in a raga, and that the work would meet me there anyway, and seem glad to.

The voice in all of it is still mine. The stamina was borrowed. That feels like a fair trade for a vacation that, for once, I actually came home from rested.

Go make something. It's a good time to be a person who wants to hear an idea out loud.