DOT Music Understanding Upgrade Roadmap

From technical metering to musical perception, comparison, experimentation, and learned taste

DOT already has a useful metering layer. The next objective is to add musical timescales, source-aware analysis, perceptual representations, structure, preference learning, and a bounded processing loop.

1. Go beyond one-second measurements

Keep DOT's existing measurements, but do not make one-second windows its only representation. Music operates over multiple timescales.

- Milliseconds: transients, attacks and drum hits
- Beats: groove and syncopation
- Bars: riffs and chord patterns
- Phrases: melody and arrangement
- Sections: verse, chorus, drop, bridge and similar structural regions
- Whole song: overall production and style identity

```
TRACK
|- waveform measurements
|- transient/event timeline
|- beat/bar timeline
|- harmonic timeline
|- stem timeline
|- section/structure timeline
`- perceptual embeddings
```

2. Give DOT instrument-aware ears

Add source separation so DOT can investigate whether a measured change is associated primarily with bass, drums, vocals, or other material rather than treating the stereo mix as a single source.

```
Instead of: 60-120 Hz is +3.1 dB
Aim for: The difference is predominantly in the bass stem rather than the drum stem.
```

Open-source source-separation ecosystems include Demucs-family systems and BS-RoFormer variants. Under the strict licensing policy, the exact software license and model-weight license must be audited separately before installation.

3. Add perceptual understanding

A perceptual representation complements physical DSP measurements. LAION CLAP is an example of a system designed to extract audio and text representations for downstream tasks. This kind of layer can help represent relationships associated with descriptors such as dark, aggressive, dreamy, dense, electronic, acoustic, percussive, atmospheric, distorted, or cinematic.

```
DSP layer = What physically happened?
Perceptual layer = What does the audio resemble?
DOT reasoning = What does that combination mean?
```

Checkpoint and training-data rights should still be verified independently of a repository's code license.

4. Teach DOT rhythm

- tempo and beat locations
- onset and transient locations
- transient density
- beat-strength curve
- inter-onset timing

- rhythmic similarity across sections and tracks

The goal is not merely BPM. DOT should build a rhythmic timeline that can identify when rhythmic activity, density, or emphasis changes.

5. Teach DOT harmony

- estimated key
- chroma and pitch-class distribution
- chord sequence estimates
- harmonic-change rate
- harmonic density
- consonance/dissonance proxies where useful

This lets DOT distinguish tonal-balance similarity from harmonic similarity. Two tracks may have comparable spectra while behaving very differently harmonically.

6. Teach song structure

DOT should construct section boundaries and, where confidence is sufficient, semantic labels such as intro, verse, chorus, bridge and outro. When labels are uncertain, the structural segmentation remains useful.

```
SECTION A 0:00-0:18
SECTION B 0:18-0:47
SECTION C 0:47-1:12
SECTION B' 1:12-1:41
```

This permits section-to-section comparison instead of averaging an entire song into one set of statistics.

7. Turn the LDM feel into a learned concept

An off-the-shelf analyzer will not inherently know the user-specific meaning of LDM. DOT should learn it from labeled examples and preferences supplied by Pedram.

```
LDM PROFILE
Atmosphere 0-10
Darkness 0-10
Energy 0-10
Aggression 0-10
Mechanical feel 0-10
Melodic emphasis 0-10
Density 0-10
Bass character 0-10
Space/depth 0-10
Rawness 0-10
```

These dimensions are examples. The final rubric should contain only characteristics that meaningfully correspond to Pedram's definition of LDM.

8. Prefer pairwise comparisons for subjective learning

Instead of relying only on absolute 1-100 scores, ask comparative questions such as: Which track is more LDM? Which master has more punch? Which sounds more finished? Repeated pairwise choices can form a preference ranking.

```
Parasites > My Friendly Hand
Parasites > Track C
Track D > Parasites
```

```
My Friendly Hand > Track E
```

The same approach can be used for darkness, punch, vocal character, heaviness, spaciousness, emotional intensity and perceived production quality.

9. Make comparisons evidence-based

DOT should generate a layered target-versus-reference report and keep objective measurements separate from model estimates and subjective preference predictions.

Layer	Examples
Technical	loudness, dynamics, low end, high end, stereo
Rhythm	tempo, density, transient strength
Harmony	key estimate, harmonic density, chord behavior
Sources	drums, bass, vocals, other/instruments
Perceptual	darkness, aggression, atmosphere, overall similarity
Personal	LDM profile and learned preference predictions

Estimated chords, source assignments, semantic attributes and preference predictions should carry confidence or uncertainty rather than being presented as ground truth.

10. Give DOT a bounded audio-processing action layer

- EQ and dynamic EQ
- compression, limiting and expansion
- transient shaping
- saturation and distortion
- stereo width and mid/side processing
- reverb, delay and convolution
- filters
- pitch and time stretching
- layering
- gain automation

Reference matching should be hypothesis-driven rather than blindly reversing every numerical difference. DOT should propose processing, render candidates, reanalyze them, and select or refine based on measured outcomes and user preference.

11. Candidate experimentation loop

```
BASELINE
|
analyze target + reference
```

```
|
form hypotheses
|
+---------+---------+
| | |
A B C
| | |
+---------+---------+
|
reanalyze every candidate
|
reject artifacts / regressions
|
compare technical + perceptual deltas
|
Pedram rating / preference
|
store successful experience
|
next iteration
```

This turns DOT from a passive analyzer into an experimental audio engineering agent while keeping changes measurable and reversible.

12. Target architecture

```
AUDIO
|
TECHNICAL ANALYSIS
|-- spectrum
|-- dynamics
`-- stereo
|
MUSICAL ANALYSIS
|-- rhythm
|-- harmony
`-- structure
|
STEM ANALYSIS
|-- vocals
|-- bass
|-- drums
`-- other
|
PERCEPTUAL REPRESENTATION
|
DOT REASONING
|
TARGET vs REFERENCE
|
PROCESSING HYPOTHESES
|
A / B / C CANDIDATES
|
REANALYZE + CHECK
|
PEDRAM'S RATING
|
EXPERIENCE MEMORY
`----> next track
```

Bottom line

DOT currently has meters. The upgrade path is to give it meters + perceptual representations + musical concepts + memory + learned taste + controlled experimentation. Pedram's ratings are especially important because they can teach DOT what the LDM character means in this specific creative system.

Licensing constraint

For the strict CC0/public-domain requirement, treat every AI component as at least two separately verified artifacts: the software/code and the model weights or data. Do not assume that an open-source repository automatically makes its pretrained model weights or training data acceptable for DOT.

Research references used in the preceding response

- LAION-AI/CLAP repository: audio/text representation project; repository shows CC0-1.0.
- Open-source source-separation ecosystem references: source-separation tutorial and a 2026 public toolkit listing Demucs-family and BS-RoFormer resources.
- The roadmap itself is an engineering proposal derived from DOT's stated limitations; specific implementation choices should be license-audited before deployment.