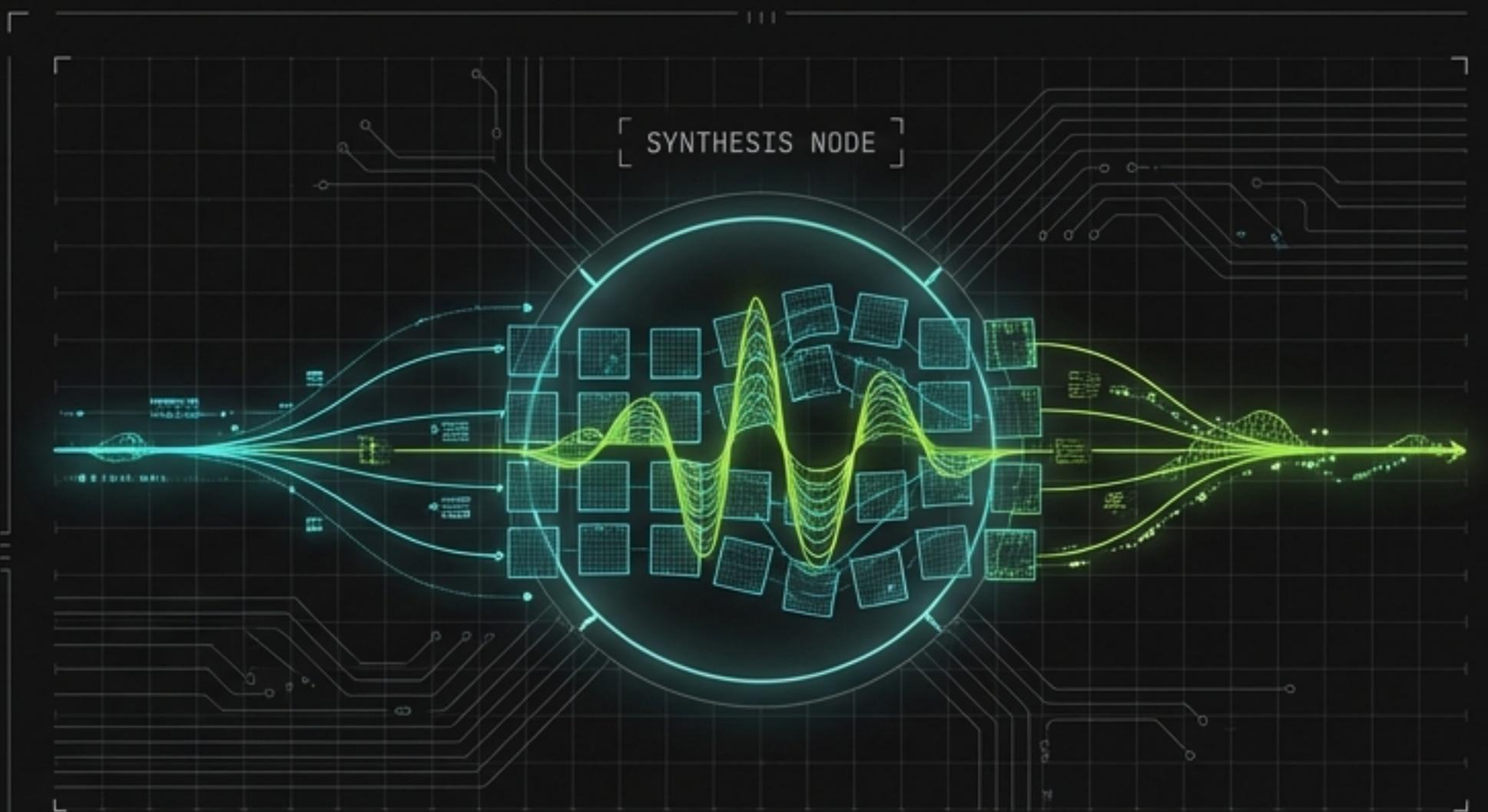
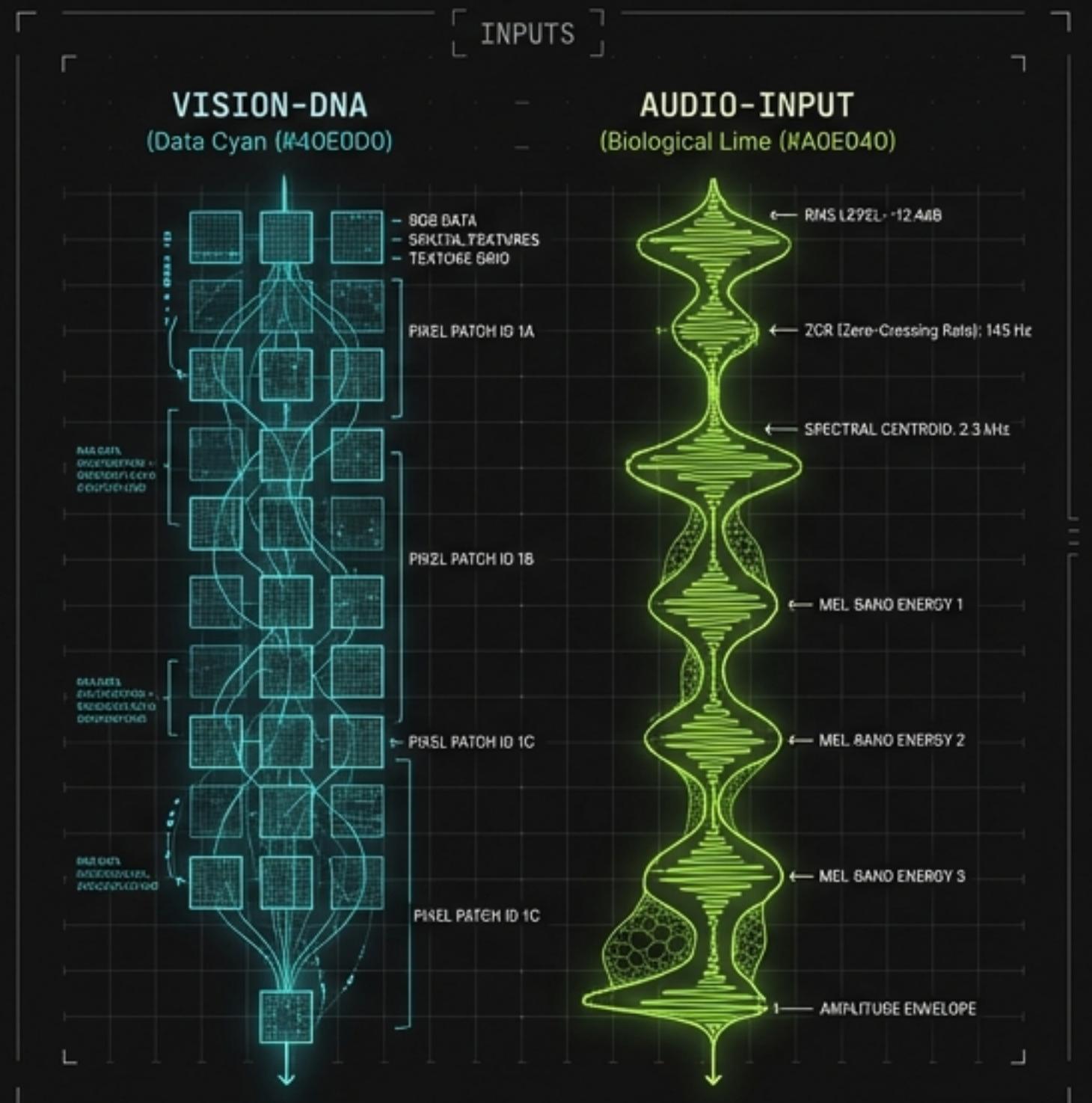


Mycelia DNA Learner

Vision-DNA trifft Klangästhetik: Eine neue Ära der digitalen Synästhesie.

- Wir präsentieren einen lebenden digitalen Organismus. Ein System, das Bilder nicht nur speichert, sondern ihre DNA lernt – und sie durch den Einfluss von Schallwellen neu träumt. Kein Filter. Keine bloße Überlagerung. Eine algorithmische Verschmelzung. •

Wenn ein Bild hören kann

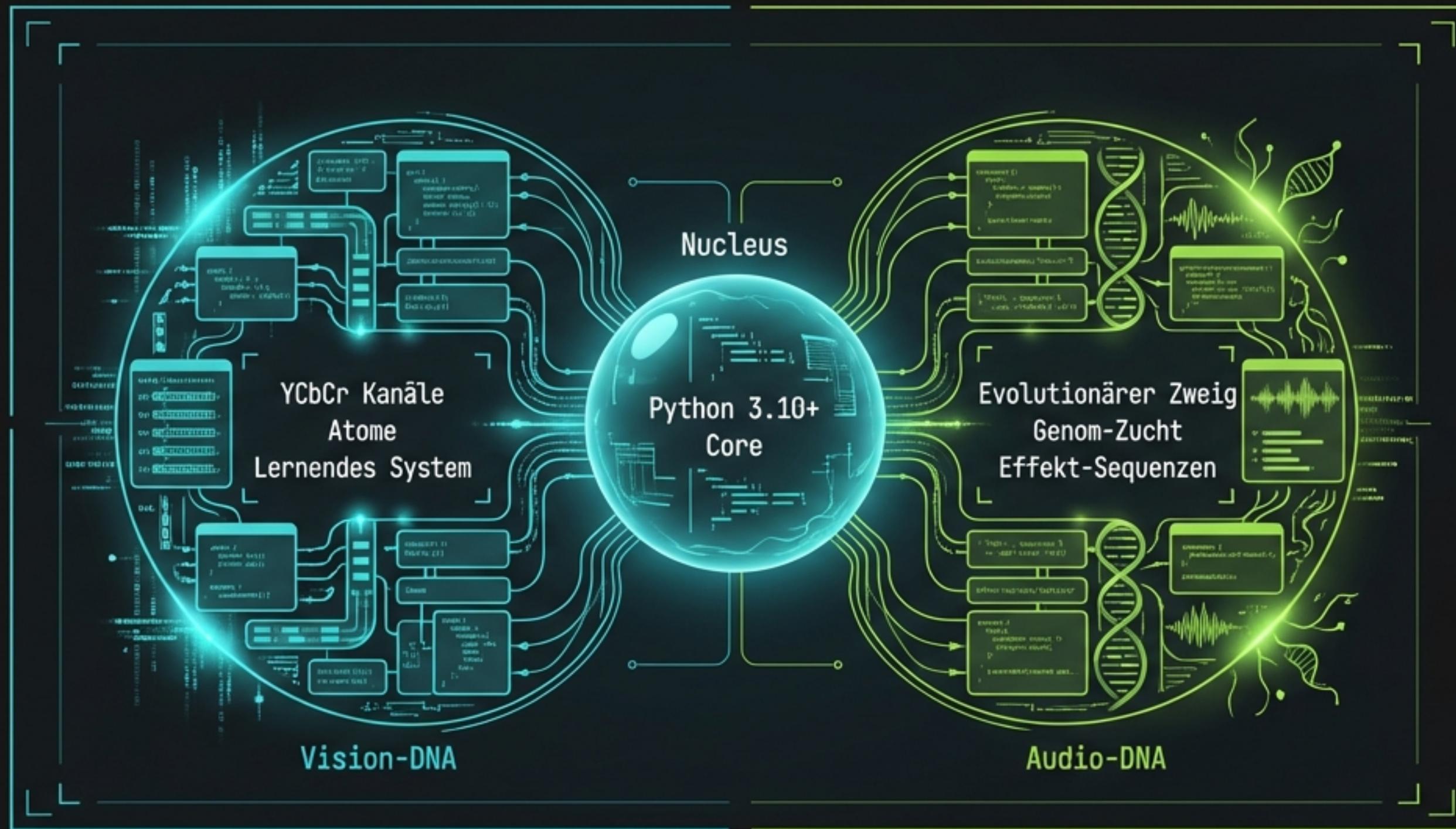


Das Prinzip: Audio-Modulierte Bildrekonstruktion

Das System mischt Audiodaten nicht als Pixel in das Bild. Es nutzt die Audioeigenschaften, um den Prozess der Bildrekonstruktion fundamental zu steuern.

“Das Resultat ist eine visuelle Neuinterpretation, geformt durch den Charakter des Klangs.”

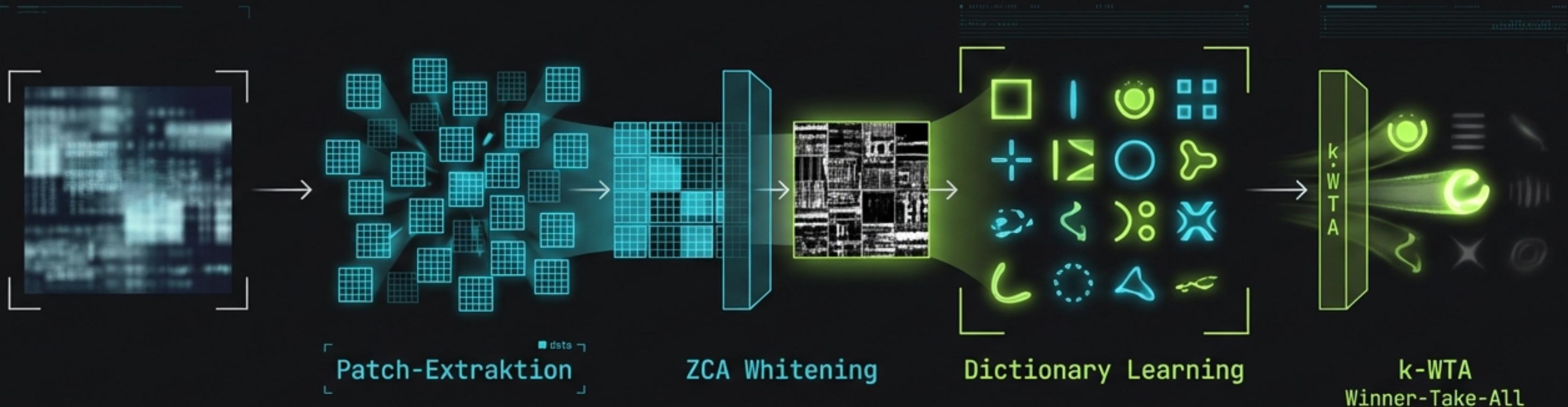
Die Biologische Architektur



- **Vision-DNA:** Ein mehrkanaliges, lernendes System (YCbCr), das visuelle "Atome" speichert.
- **Audio-DNA:** Ein evolutionärer Zweig, der Effekt-Sequenzen („Genome“) züchtet und optimiert.
- **Bio-Inspiration:** Der Code imitiert natürliche Prozesse – von der sparsamen neuronalen Aktivierung bis zur genetischen Mutation von Parametern.

Implementiert in Python 3.10+. Inspiriert von der Natur, gebaut für die Maschine.

Vision-DNA: Wie der Algorithmus sieht



Patch-Extraktion

Zerlegung des Bildes in fundamentale Bausteine.

Dictionary Learning

Das System lernt ein 'Wörterbuch' visueller Strukturen.

ZCA Whitening

Normalisierung der Daten, ähnlich der Vorverarbeitung im menschlichen Auge.

k-WTA (Winner-Take-All)

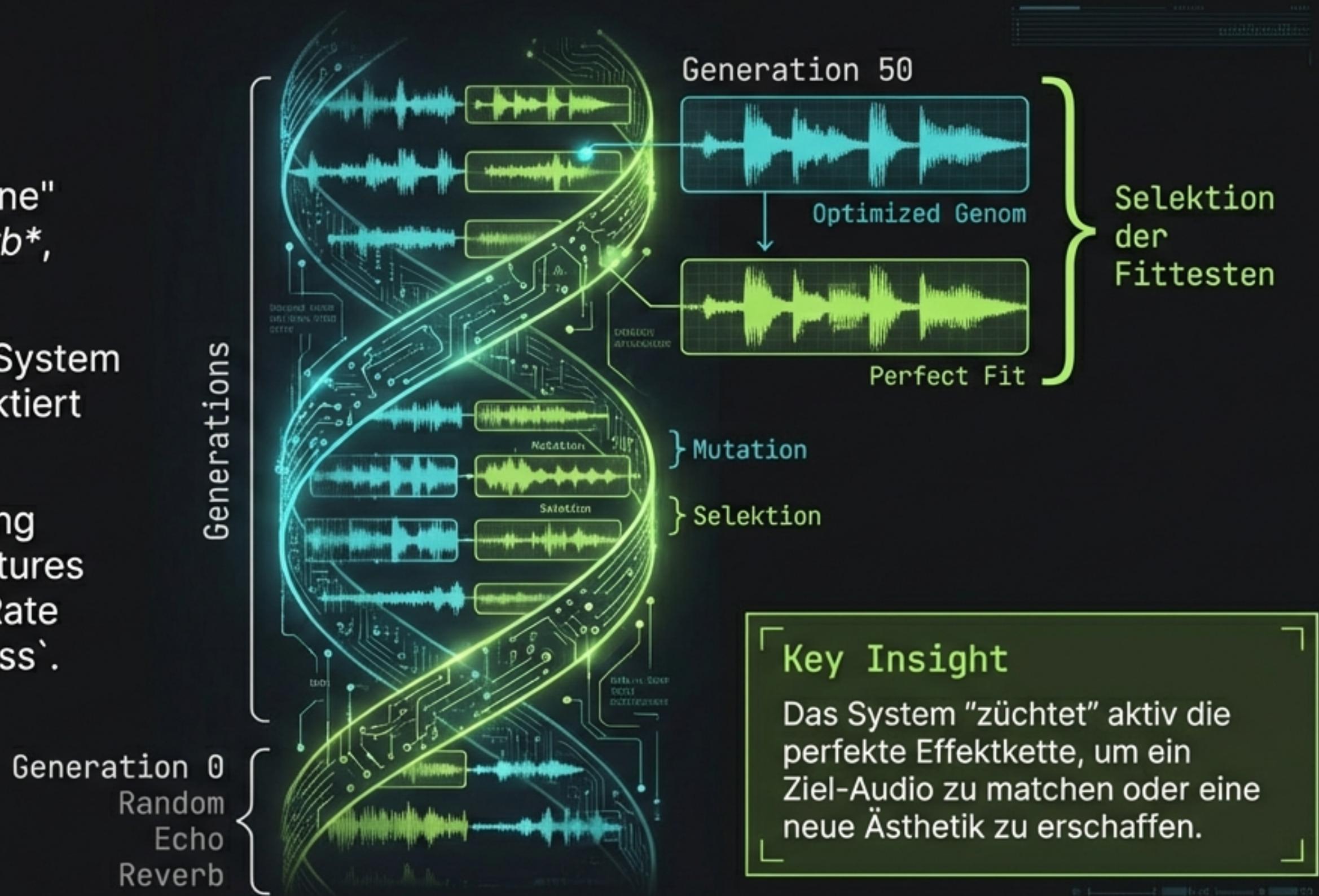
Neuronale Sparsamkeit. Nur die stärksten Signale (die 'Gewinner') dürfen feuern.

Basis: `SparseChannelModel` mit Dictionary-Atomen und bio-inspirierter Modulation.

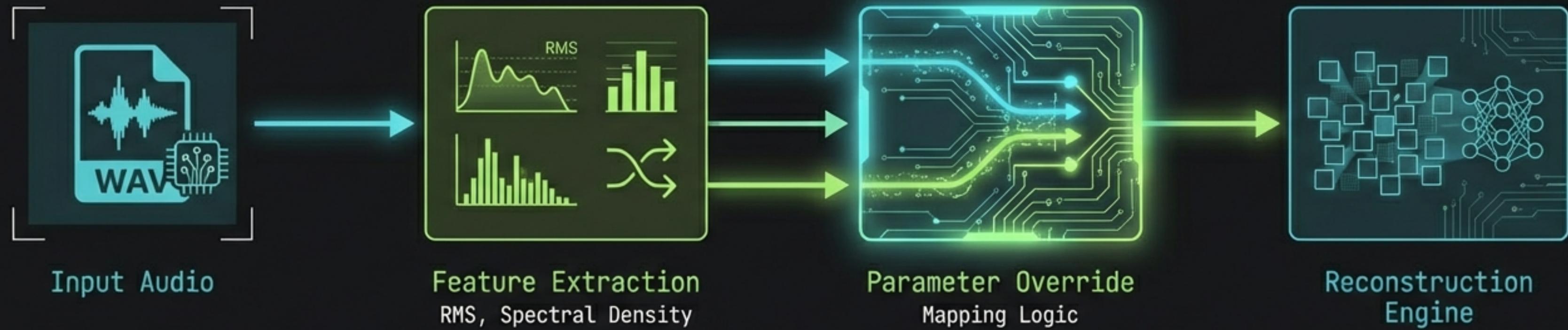
Audio-DNA: Evolutionäre Klangformung

Der Prozess

- **Generation 0:** Zufällige "Gene" (Effekte wie **Echo**, **Reverb**, **Distort**).
- **Mutation & Selektion:** Das System variiert Parameter und selektiert die fittesten Genome.
- **Fitness-Funktion:** Bewertung basierend auf robusten Features wie 'RMS', 'Zero Crossing Rate (ZCR)' und 'Spectral Flatness'.



Die Schnittstelle: Wenn Klang auf Bild trifft



Die Mechanik:

- **Analyse:** Extraktion akustischer Merkmale.
- **Mapping:** Übersetzung von Audio-Werten in Vision-Parameter.
- **Override:** Der Audio-Input überschreibt die Standard-Einstellungen der Bild-Engine.

Example Mapping:

Lautes Audio (High RMS) -> Erhöhte `detail_strength`
Rauschhaftes Audio (High Flatness) -> Angepasstes `chroma_smooth`

Input-Analyse: Der Herzschlag der Daten

```
[audio_control] source=data\audio files=1 sr=16000 sec=45.00 strength=0.65
[audio_control] features rms=0.1354 zcr=0.1000 flatness=0.5827
[audio_control] bands(L/M/H)=0.142/0.449/0.409
```

RMS (Energie des Signals)

Flatness (Komplexität und Rauschen)

Das System lauscht. Es quantifiziert die 'Seele' des Audios.

Die Transmutation: Mapping & Override

Detected Audio Features

mapped noise = **0.458**

mapped tonal = **0.400**

mapped energy = **0.376**

Parameter Override Execution

Parameter	Original Value	New Value	
Zeitliche Dichte (tp_len)	1	→ 2	Information aus Nachbar-Frames
Pooling Modus (tp_mode)	max	→ avg	Weichere Durchschnittswerte
Detailgrad (detail_strength)	0.180	→ 0.181	Subtile Schärfung
Farbglättung (chroma_smooth)	1.000	→ 1.239	Massive Erhöhung der Farbweichheit

Das Ergebnis: Vision im Klangbad



Input: Original Vision-DNA



Output: Audio-Modulierte Rekonstruktion

Beachten Sie die veränderte Ästhetik. Die Kantenführung und das Shading wurden durch die Audio-Parameter (`'chroma_smooth'`, `'tp_mode'`) neu interpretiert.

Ein Bild, dessen Atmosphäre durch Klang diktiert wurde.

Die Syntax des Lebens

```
@dataclass
class VisionDNAModel:
    y: SparseChannelModel
    cb: SparseChannelModel
    cr: SparseChannelModel

    def derive_audio_vision_controls(
        feat: np.ndarray,
        tp_len: int,
        tp_mode: str,
        cfg: ReconstructionPostConfig,
        strength: float,
    ) -> tuple[int, str, ReconstructionPostConfig, dict[str, float]]:
        # ... logic implementation ...
```

Typ-Sicherheit

Modularität

Kontrolle

Sauberer, typ-sicherer Python Code (Python 3.10+). Nutzung von `Enum`-Logik für strikte Prozesskontrolle. Modularer Aufbau ermöglicht unendliche Erweiterbarkeit der DNA.

Die Macht der Konsole

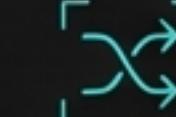
`train_vision`

Trainiert das visuelle Gedächtnis
(Dictionary Learning).



`reconstruct`

Erzeugt das Bild neu – optional
mit `--audio_control`.



`learn_audio`

Startet die evolutionäre Suche
nach dem perfekten Klang.



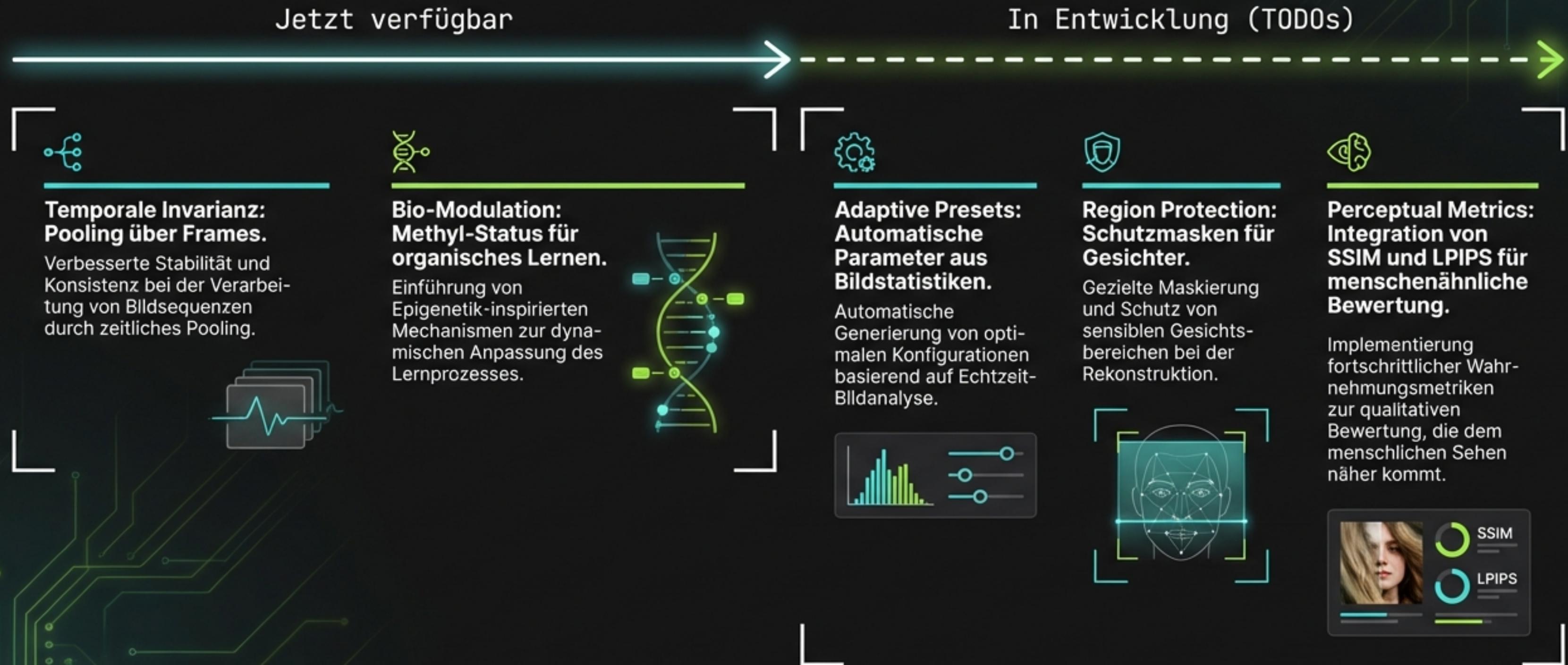
`reconstruct_seq`

Wendet die Logik auf ganze
Bildsequenzen an (Video-Potenzial).

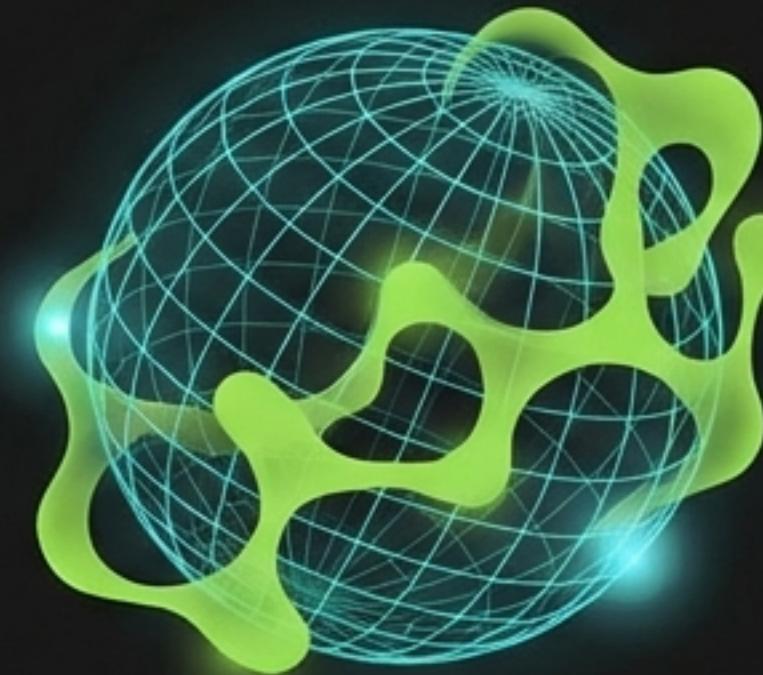


Volle Kontrolle über alle Hyperparameter via CLI.

Roadmap & Features: Die nächste Stufe



Eine neue Ära der Medienverarbeitung



Mycelia DNA Learner ist mehr als Software. Es ist der Beweis, Beweis, dass binärer Code organische Züge annehmen kann.

Erleben Sie die Symbiose aus visueller Präzision und akustischer Emotion.

Starten Sie die Evolution. Experimentieren Sie mit der DNA.