

Supplementary Materials

A Bioresponsive Multimodal VR Interface for Personalized Heart Coherence Training through Spatially Mapped Biosignals

1 Animated Videos

This section provides the seven animations referenced throughout the paper. Table 1 maps each supplementary video (S1–S7) to the relevant figures and concepts in the main text.

ID (file)	Primary illustrated function / linkage in main paper
S1_brightness-saturation-change.mp4	Crossmodal mappings (brightness/saturation); Figure 2 (left panel)
S2_angularity-change.mp4	Crossmodal mappings (angularity→roundness); Figure 2 (middle panel)
S3_couplingvalues-change.mp4	Pattern symmetry via coupling kernel; Figure 2 (right panel)
S4_overall-coherence-change.mp4	System overview of coherence-driven aesthetics; Figure 1
S5_fast-breath-matching.mp4	Pacer-tracker: fast matching state; Figure 3
S6_slow-breath-match.mp4	Pacer-tracker: slow matching state; Figure 3
S7_slow-breath-mismatch.mp4	Pacer-tracker: slow but mismatched; Figure 3

Table 1: Mapping of supplementary animations to concepts and figures in the main paper.

1.1 Coherence Mapping (S1–S4)

S1: Brightness and Saturation Modulation with Coherence

File: S1_brightness-saturation-change.mp4

Description: Demonstrates the luminance and saturation channel used to convey cardiac coherence. As coherence rises, the particle field brightens and becomes more saturated; as coherence drops, the scene dims and desaturates. Complements Figure 2 (left panel).

S2: Angularity → Roundness Transition

File: S2_angularity-change.mp4

Description: Visualizes the shape transition from star-like/angular glyphs to smooth circular forms as coherence increases. Shows the affective mapping over time and how the shader drives perceived geometry. Complements Figure 2 (middle panel).

S3: Coupling-Driven Symmetry Patterns

File: S3_couplingvalues-change.mp4

Description: Sweeps oscillator coupling to reveal emergent symmetry on the geodesic particle layout. Higher coupling produces globally synchronized, symmetric motifs; lower coupling yields irregular configurations. Complements Figure 2 (right panel).

S4: Overall Coherence State Transitions (System Overview)

File: S4_overall-coherence-change.mp4

Description: A holistic pass through low→high coherence conditions, showing coordinated changes across brightness, saturation, angularity, and symmetry on the unified particle field. This is the moving counterpart to the system overview in Figure 1.

1.2 Breathing Pacer–Tracker (S5–S7)

The pacer–tracker visualization operates within the geodesic particle field by encoding breathing phase as concentric pentagon rings. The pacer defines the target rhythm as a propagating ring animation, while the tracker reflects the participant’s measured breathing. Synchrony causes the “gap” between pacer and tracker to collapse into a seamless animation; mismatch produces a visible dark band signaling phase error. Two pacing frequencies are illustrated: 6 s cycles (12 concentric rings) and 3 s cycles (6 rings), providing an intuitive spatial measure of inhalation and exhalation duration. This design delivers embodied guidance without numerical cues, enabling both fast and slow pacing, and highlighting how the system can adapt toward resonance-frequency optimization.

S5: Fast Breath, Matching

File: S5_fast-breath-matching.mp4

Description: Depicts a relatively fast guided tempo with the user’s tracked phase aligned to the pacer. The local pacer–tracker region shows minimal error coloration and fluid motion, while global coherence visuals remain unobstructed. Complements Figure 3.

S6: Slow Breath, Matching

File: S6_slow-breath-match.mp4

Description: Shows synchronized breathing at a slower tempo (near resonance). Error visualization is minimal, and the global scene moves into high-coherence aesthetics—guidance remains peripheral without numeric overlays. Complements Figure 3.

S7: Slow Breath, Mismatch

File: S7_slow-breath-mismatch.mp4

Description: Illustrates a slow pacer with user desynchronization. Phase mismatch appears as color-gradient and animation changes localized in the interstitial “gap” between pacer and tracker. Complements the mismatched states in Figure 3.

2 Technical Implementation: Real-time HRV Coherence Biofeedback System

We adapted scripts from three GitHub repositories—*Unity-Android-Bluetooth-Low-Energy* [1], *PolarBand2lsl* [2], and *dont-hold-your-breath* [3]—to build a real-time HRV biofeedback system that runs natively on the Meta Quest 3 headset. A Polar H10 chest strap streams interbeat intervals (IBIs) via Bluetooth Low Energy (BLE) directly into Unity, where data are parsed and processed on-device to compute a coherence metric that continuously drives visual feedback in VR. This on-headset design eliminates the need for external relays, minimizing latency and ensuring smooth integration with immersive rendering.

2.1 BLE Data Acquisition

Device communication relies on the Unity-Android-Bluetooth-Low-Energy plugin [1], which bridges Unity’s C# environment with Android’s native BLE APIs through a Java Native Interface. The plugin discovers the Polar H10, connects to the Heart Rate Service (UUID 0x180D), and subscribes to the Heart Rate Measurement characteristic (UUID 0x2A37). Notifications deliver IBI data directly to Unity, while the

plugin also manages Android 12+ Bluetooth permissions and handles asynchronous data transfer without compromising frame rendering.

2.2 Packet Decoding and Reference Implementation

To ensure reliable Polar integration, we drew on PolarBand2lsl [2], which, although originally designed for Lab Streaming Layer (LSL) output, documents the proprietary Polar protocol. Its implementation guided our C# routines for GATT service access, MAC address binding, robust device discovery, RR parsing, and error handling. By embedding these routines directly into Unity rather than relying on an external relay, we reduced dependencies and improved timing precision.

2.3 Signal Processing Pipeline

IBIs received from the Polar H10 are processed using methods adapted from the *dont-hold-your-breath* package [3] and closely aligned with the methodological pipeline reported by Blum et al. (2019) [4]. Data are buffered over a sliding 60–64 s window, resampled at 2 Hz (128 samples) via cubic-spline interpolation, detrended to remove the DC component, and tapered with a Hann window to minimize spectral leakage. A fast Fourier transform (FFT) is then applied to obtain the power spectral density. Artifacts are handled by rejecting IBIs that deviate more than 350 ms from the local moving average of the preceding ten beats, replacing them with the mean of that interval set, following the strategy described by Blum et al. Within the frequency domain, the dominant oscillatory component f^* is identified in the 0.06–0.26 Hz range with sub-bin refinement via spline interpolation. Coherence is then defined, consistent with HeartMath literature [5], as the ratio of narrowband power within ± 0.015 Hz of f^* to the total power across the 0.04–0.26 Hz band:

$$\text{Coherence} = \frac{\int_{f^* - 0.015}^{f^* + 0.015} P(f) df}{\int_{0.04}^{0.26} P(f) df}. \quad (1)$$

The resulting value ranges from 0 to 1 and is updated on every heartbeat, providing continuous feedback on cardiac rhythm synchrony. In addition to this continuous parameter, a dichotomous index is computed using a threshold of 0.5: values above this cutoff signify that at least half of spectral power is concentrated in a narrow band, corresponding to a sine-like HRV waveform and strong respiratory–cardiac coupling. While the computational steps replicate those of Blum et al. [4], our implementation diverges architecturally by executing BLE acquisition, packet decoding, artifact correction, and spectral analysis fully on the Meta Quest 3 headset rather than relying on a Windows-based relay. This preserves their validated coherence computation while reducing latency, minimizing external dependencies, and ensuring robustness for real-time immersive VR applications.

2.4 Adaptive Breathing Pacer Optimization

At the start of each session, the system establishes individual baselines before adaptive pacing begins. Participants first complete a 60-s spontaneous breathing phase to record their natural breathing rate and minimum coherence. This is followed by a 60-s guided slow-breathing phase, during which the system captures the longest sustainable breath duration and the maximum coherence achieved. Together, these phases define the participant’s personal coherence range, which is then used to normalize coherence values:

$$\text{Normalized Coherence} = \frac{C_{\text{current}} - C_{\text{baseline}}}{C_{\text{max}} - C_{\text{baseline}}}. \quad (2)$$

After baseline acquisition, the system evaluates coherence trends in real time. Coherence values computed on every heartbeat are logged in a circular buffer spanning 90 s. At the end of each buffer cycle, the average normalized coherence is calculated and mapped to adjustments in pacer rate. When average normalized coherence exceeds 0.7, the pacer is extended by 0.5–1.5 s to promote deeper practice. When coherence falls below 0.3, the pacer shortens by 0.3–0.8 s toward the baseline breathing rate, reducing strain. To avoid abrupt transitions, adjustments are limited to $\pm 15\%$ per cycle and smoothed with an exponential filter using an 8-s time constant.

Safety constraints ensure physiologically appropriate pacing. Breathing rates are restricted to $0.6\text{--}1.8\times$ the individual’s baseline, with absolute bounds of 3–12 breaths per minute. In addition, after three consecutive evaluation cycles with normalized coherence below 0.2, the pacer automatically reverts to the baseline rate, preventing sustained respiratory stress.

This procedure follows the adaptive guidance principles outlined in the ViBreathe protocol [6], but integrates them directly with real-time coherence metrics to provide individualized pacing that dynamically responds to the user’s physiological state.

References

- [1] Velorex. Unity-android-bluetooth-low-energy: A unity android plugin to support basic bluetooth low energy interactions, 2024. GitHub repository.
- [2] Mark M. Span. Polarband2lsl: Send polarband h10 data to an lsl, November 2023. GitHub repository.
- [3] Kieran Brennan. dont-hold-your-breath: Breathing analysis with polar h10 heart rate monitor, April 2023. GitHub repository.
- [4] Johannes Blum, Christoph Rockstroh, and Anja S Göritz. Heart rate variability biofeedback based on slow-paced breathing with immersive virtual reality nature scenery. *Frontiers in psychology*, 10:2172, 2019.
- [5] Rollin McCraty, Mike Atkinson, Dana Tomasino, and Raymond Trevor Bradley. The coherent heart heart-brain interactions, psychophysiological coherence, and the emergence of system-wide order. *Integral Review: A Transdisciplinary & Transcultural Journal for New Thought, Research, & Praxis*, 5(2), 2009.
- [6] Bin Yu, Pengcheng An, Sjoerd Hendriks, Ning Zhang, Loe Feijs, Min Li, and Jun Hu. Vibreathe: Heart rate variability enhanced respiration training for workaday stress management via an eyes-free tangible interface. *International Journal of Human-Computer Interaction*, 37(16):1551–1570, 2021.