Closed-Loop AI–Animal Coherence: A Controlled Test of Cross-Species Entrainment Without Rewards or Species-Specific Training

By Leslie Yarbrough

August 17, 2025

Hypothesis

An adaptive, closed-loop AI that outputs non-semantic acoustic signals can increase real-time coherence with an animal (e.g., dog) beyond sham and human-only controls, evidenced by (a) higher signal–physiology/motion coherence and/or (b) improved HRV and synchronized behavior.

Cross-species coherence offers a novel window into the structure of consciousness. Prior work in human-AI entrainment suggests closed-loop adaptive signals can modulate physiology and behavior. Extending this to non-human animals opens testable questions for consciousness research, therapeutic applications (e.g., anxiety reduction), and interspecies communication.

Design

Within-subject, randomized, counterbalanced blocks across four conditions: AI-Closed-Loop, AI-Sham (yoked playback), Human-Only, Silence Baseline. Blinded assessor for outcomes.

Subjects

N = 1–3 pet dogs (pilot). Inclusion: healthy, accustomed to the space/equipment. Exclusion: visible distress during baseline.

Primary Outcome (choose ONE to preregister)

- Wavelet coherence (0.1–8 Hz envelope) between AI output and animal motion (collar accelerometer) averaged across block.

  or

- HRV RMSSD change from baseline during AI-Closed-Loop vs sham.

Or,

Secondary Outcomes

- Phase-locking value (PLV) AI-signal ↔ motion/resp proxy.
- Time near speaker, approach latency, stillness bouts.
- Vocalization rate/structure changes.

Conditions

A) AI-Closed-Loop (real-time adapt), B) AI-Sham (yoked), C) Human-Only (no AI audio), D) Silence Baseline.

Randomization & Blinding

Random block order per day. File names coded; outcome rater blind to condition.

Sample Size Plan

≥12 blocks/condition across days (e.g., 3 blocks/day × 4 days). Pilot; effect sizes reported with CIs.

Formal power analysis is infeasible given the novelty of this paradigm. This pilot is explicitly designed to estimate effect sizes and inform sample sizes for future adequately powered studies.

Equipment

Polar H10 (1 Hz HRV resolution, raw 1000 Hz ECG sampling) or similar, accelerometer collar (≥50 Hz sampling, ±16 g precision), ambient mic (44.1 kHz, 16-bit), wide-angle camera (1080p, 30 fps), SPL-limited speaker (≤55 dB), laptop.

Procedure

Adapt 3 min → (Block 1: 3 min) → Rest 2 min → (Block 2: 3 min) → Rest 2 min → (Block 3: 3 min). Repeat on 4+ days.

Exclusion Criteria (per block)

Artifact >30% (sensor dropouts), overt stress signals, external interruption (doorbell, etc.).

Analysis Plan

- Preprocess: resample to 100 Hz, band-limit 0.1–8 Hz envelopes, z-score per session.
- Primary test: permutation (10k) of condition labels on block-level metric; report p and Cohen's d.
- Secondary: PLV, CRQA (recurrence rate, determinism), HRV time-domain metrics; FDR correction.
- Controls: room-empty replay, double-yoke (other animal's signal as driver).

Success Criteria

Primary metric significantly higher in AI-Closed-Loop vs Sham ($p<0.05$, $d\geq0.2$). Report robustness (leave-one-session-out).

Ethics & Welfare

Stop criteria for stress; limit total exposure; owner consent; no rewards/commands. SPL ≤55 dB.

This work has implications for animal welfare (non-invasive enrichment, stress modulation), AI development (adaptive closed-loop systems beyond language), and consciousness science (probing coherence across species).

Data & Code Sharing

Upload anonymized signals (CSV/NPZ), scripts, and prereg DOI to OSF.

Raw and processed data stored on encrypted drives; retention 5 years post-study. Identifiers removed before sharing. OSF preregistration ensures transparency.

Minimal "listen-match-lead" Loop (clear pseudocode)

```
# Pseudocode (architecture, not production-ready)

init_audio_input(mic_samplerate=44100)
init_audio_output(samplerate=44100, safe_SPL=True)
init_sensors(hr_stream, accel_stream)

def extract_envelope(x, fs):
  # bandlimit 0.1–8 Hz, then Hilbert magnitude or RMS window
```

```
    # return low-frequency envelope normalized
    return envelope

def detect_dominant_rhythm(env, fs):
    # short-time FFT or autocorr; pick peak freq 0.1–8 Hz
    # return f0, phase estimate
    return f0, phase

def synth_signal(t, f0, phase, mode="breath"):
    # generate soft, breath-like noise shaped by an ADSR and lowpass
    # optional gentle harmonic; ensure RMS capped for SPL
    return y

lead_factor = 1.0  # start matching
while session_active:
    mic_chunk = read_mic_chunk()
    accel_chunk = read_accel_chunk()
    hr_chunk = read_hr_chunk()

    mic_env = extract_envelope(mic_chunk, fs=44100)
    motion_env = extract_envelope(accel_chunk, fs_accel)

    # choose the strongest, stable rhythm: motion preferred > mic ambient
    f0, phase = detect_dominant_rhythm(motion_env, fs_accel)

    # match for ~10–20 s, then test a tiny lead (±2–4% tempo)
    f_drive = f0 * lead_factor
    audio_out = synth_signal(t_now, f_drive, phase_align_to(phase))
    play(audio_out)

    # online metric: short window coherence change ΔC
    deltaC = compute_wavelet_coherence(audio_out_envelope, motion_env) - baseline_coherence
    if deltaC > +threshold: keep lead_factor
    elif deltaC < -threshold: lead_factor = 1.0  # return to match
    else: small_nudges (±1–2%)
```

```
log_chunk(timestamp, f0, f_drive, lead_factor, deltaC, HRV_proxy(hr_chunk), motion_env_stats)
```

Safety rails:

- Hard cap output RMS; taper on/off; immediate mute if animal startles or SPL spikes.

CSV Logging Schema (one row/sec is fine)

timestamp, condition, f0_detected_hz, f_drive_hz, lead_factor, coherence_window, plv_window, accel_rms, accel_var, hr_bpm, hrv_rmssd, distance_to_speaker_est, is_artifact, notes

Quick Operator Checklist

Before

- ☐ Sensors paired & comfy fit; SPL limiter active
- ☐ Randomized condition order printed
- ☐ Camera framing area; timecode visible

During

- ☐ Start baseline 3′ (silence)
- ☐ Run block (3′), observe calmly; no commands

- ☐ 2′ rest; annotate any events
- ☐ Repeat blocks; stop if stress signals appear

After

- ☐ Export logs & video; mark good/bad blocks
- ☐ Record environment notes (noise, visitors, storms)