# Coherence Sculpting Real-Time Neurofeedback Spec

**Overview and Goals**

**Coherence Sculpting** is an interactive neurofeedback feature that empowers users to shape large-scale brain connectivity in real time. It provides visual, audio, and optional haptic feedback to guide the user toward a **"useful edge" of criticality** – a brain state that is highly **ordered yet flexible**. At this edge, the brain exhibits strong functional connectivity without tipping into pathological hyper-synchrony or chaotic disintegration. The system does **not make clinical/medical claims**; instead, it offers a wellness and cognitive enhancement tool with proper safety guardrails. Product stakeholders can view this as a novel **engagement feature** that leverages advanced EEG/MEG signals and closed-loop interaction to potentially improve focus, creativity, or relaxation in users. The scope includes a full software pipeline (signal processing, metrics computation, feedback visualization), interface controls for users, and an evaluation plan to ensure it operates within safe, ethical bounds.

**Signals & System Context**

- **Neural Inputs:** Primarily high-density EEG (8–32+ channels, dry or wet electrodes) and optionally MEG if available. The system can also integrate **physiological channels** like heart rate variability (HRV) and respiration, to capture **cardio-cerebral coupling**. These additional signals enrich the feedback (e.g. mapping heart-brain coupling to visual elements) but are not required.
- **Frequency Bands:** The system monitors standard EEG bands δ, θ, α, β, γ, but each user's bands are **personalized** by anchoring to their Individual Alpha Frequency (IAF). Personalization accounts for individual differences in peak frequencies . For example, if a user's alpha peak is 9 Hz, the band definitions (theta, beta, sensorimotor rhythm, etc.) are shifted relative to 9 Hz rather than using generic 10 Hz. This ensures feedback targets are tailored to each brain's natural rhythms and developmental stage .
- **Technical Specs:** The EEG is sampled at ≥250 Hz to capture fast dynamics. A total end-to-end latency of ≤300 ms is targeted so that feedback feels immediate and relevant to the user's current brain state. To achieve this, data is processed in **sliding windows** (e.g. 1–4 second windows with 50–80% overlap) to balance temporal resolution against stable connectivity estimates. Hardware requirements include a capable EEG amplifier and a computing device (e.g. a tablet or PC) that can handle real-time signal filtering, feature extraction, and rendering of the 3D visualization at 30+ FPS. The software will be optimized for multi-threading (separating data acquisition, processing, and rendering threads) to meet the latency budget.

**Core Metrics and Neural Dynamics**

Real-time algorithms compute a suite of **connectivity and complexity metrics** from the EEG in sliding windows. These metrics quantify the brain's functional network state and guide the feedback. Key metrics include:

- **Phase Connectivity:** We measure phase synchrony between regions using metrics like **Phase-Locking Value (PLV)** and **weighted Phase Lag Index (wPLI)**. PLV provides a straightforward index of phase consistency between two signals, but it can be inflated by trivial zero-lag correlations (e.g. due to volume conduction). Therefore, wPLI or the **imaginary part of coherency** is used for robust feedback, since these metrics ignore or down-weight zero-phase differences . This ensures that the "coherence" we feedback reflects true network interactions rather than an artifact of one source bleeding into multiple sensors. Additionally, a **Phase Slope Index (PSI)** may be computed to estimate directionality of information flow (i.e. which region leads or lags in phase), giving insight into **causal connectivity** patterns. All phase metrics are anchored to the user's IAF when distinguishing bands (e.g. "alpha coherence" means coherence at the user's alpha frequency range).
- **Global Integration (Kuramoto Order):** The system computes the **Kuramoto Order Parameter R(t)** as a measure of global phase synchrony across all channels. R(t) ranges from 0 (completely desynchronized signals) to 1 (all channels phase-locked). This metric indicates **integration** level of the brain's oscillatory activity . We also derive **metastability**, defined as the standard deviation of R(t) over time . High metastability (i.e. large fluctuations in R) means the brain's synchrony is **constantly shifting**, reflecting a balance of integration and segregation . This state is theorized to underlie cognitive flexibility and adaptive processing (the brain neither locks into one rigid pattern nor falls into noise). In contrast, very low metastability means the brain is static (either persistently synchronized or persistently random), which could indicate boredom, rigid thought, or, at the extreme, pathological states. The target is

to gently guide users toward a **moderate-to-high metastability** regime associated with engaged, flexible mind states, without pushing into unstable extremes (we avoid driving the system into full **torus breakdown** chaos, as discussed below).

- **Network Graph Metrics:** Beyond pairwise phase metrics, the system can compute network-level measures on an EEG **connectivity graph** (sensors as nodes, coherence as edge weights). **Graph Laplacian energy** is one such measure – it captures the overall harmonics or "strain" in the network's connectivity pattern (related to the concept of **connectome harmonics** where certain connectivity patterns correspond to eigenmodes of the brain's structure). We also consider each channel's **participation coefficient** and the network **modularity Q**. Participation coefficient indicates how evenly a node's connections are spread across different network communities, while modularity Q measures the strength of community structure in the connectivity matrix. For example, a high Q might indicate the brain is segregated into distinct modules (which could correspond to focused processing in specific circuits), whereas a lower Q with high overall coherence might indicate more global integration. These graph metrics provide a richer picture of **dynamic functional connectivity** than any single pair of channels.
- **Cross-Frequency Coupling (CFC):** If enabled, the system monitors metrics like **phase-amplitude coupling (PAC)** to capture interactions between slow and fast oscillations. An example target could be **θ–γ coupling** in frontal areas, which has been linked to working memory and attention processes . PAC is computed by measuring how the phase of a low-frequency rhythm (e.g. theta) modulates the amplitude of a high-frequency rhythm (e.g. gamma). We might specifically track **n:m phase locking** events (where n cycles of a slow wave align consistently with m cycles of a faster oscillation). While CFC metrics are more complex and slower to compute, they can indicate integrative brain states (e.g. theta-driven gamma bursts during memory encoding ). These can serve as advanced targets for users interested in states like deep focus or meditation (many meditation practices naturally increase alpha–gamma coupling, for instance).
- **Complexity & Criticality:** To quantify how close the brain is to a critical point, we compute measures of signal complexity such as **Lempel–Ziv Complexity (LZC)** and **Permutation Entropy** from the EEG time series. These metrics gauge how unpredictable or novel the signal patterns are. Near criticality, brain signals often show 1/f-like spectra and high complexity (mix of order and randomness). We track the **1/f spectral slope** as well: a flatter 1/f slope (less negative) can indicate more high-frequency activity (potentially more arousal or chaos), whereas a steeper slope indicates dominance of slow waves (could mean drowsiness or too rigid order). By monitoring complexity measures alongside coherence, the system ensures the user is not driving the brain into overly simple, periodic dynamics (too ordered) or noise-like activity (too chaotic). The aim is to encourage a **sweet spot** where coherence is accompanied by rich, metastable dynamics – a hallmark of criticality.

Each of these metrics is computed in real-time on overlapping windows. Windows may be 2 seconds long with 1 second overlap (yielding an update every 1s), or adjustable based on user needs. Shorter windows (~1s) give more immediate feedback but less reliable coherence estimates, whereas longer windows (4s) smooth out noise but at the cost of latency. The default 2s/50% overlap is a compromise to stay within ~300 ms pipeline delay after each update.

**Interaction Model ("Sculpting" Controls)**

The user interacts with the system through intuitive controls, almost like "molding" their brain's connectivity in a virtual space. Rather than passively observing feedback, they actively **sculpt** the coherence patterns with guidance from the system:

- **Target Definition:** The system defines a **target connectivity profile** C* and related target metrics (R*, metastability M*, etc.) as a goal state. This target can be customized per session. For example, a user might set a target to **increase fronto-parietal beta coherence** (for focused attention), **maintain occipital alpha coherence** (for visual calm), and **boost theta–gamma PAC** (for memory or insight). The target profile is essentially a vector or matrix of desired connectivity values across certain channel pairs and frequency bands, along with desired global indices (e.g., "I want a moderately high R of 0.5 and metastability around 0.1"). We provide default target profiles corresponding to common cognitive states (e.g., a "Focus" preset might target higher frontal beta coherence and moderate global synchrony, while a "Relaxation" preset might target higher alpha coherence in posterior regions but lower overall R). Users can also paint their own targets (see Brush tool below).
- **User Controls:** Three main control modalities are provided to shape the coherence:
- **XY Pads:** The interface offers one or more two-dimensional sliders (XY pads) where each axis controls an aspect of connectivity. For instance, one pad might map **X = inter-areal coupling** (global connectivity level) and **Y = band emphasis** (shifting which frequency band is most pronounced in the coherence). By moving a point on this pad, the user simultaneously tweaks how globally synchronized vs. localized the activity is, and whether slow vs. fast rhythms dominate. Another pad could control **R vs. metastability**, effectively letting the user set a point on a continuum from stable synchronization (high R, low variance) to flexible fluctuations (moderate R, high variance). These pads let the user explore the **state space of brain dynamics** with intuitive gestures, with the system translating those gestures into target metric adjustments.

- **"Brush" Tool for Networks:** For finer control, the user can activate a "neural brush" mode. The EEG sensor layout (or a brain network diagram) is displayed, and the user can **paint** increases or decreases on connections or regions. For example, by brushing between two regions (say frontal and parietal), the user tells the system "I'd like more coherence here." The brush can also reduce connectivity: painting with a "cool" color might signify the user wants to **down-weight** or **de-emphasize** a connection (perhaps to quiet an overactive loop that might be linked with stress or rumination). Internally, this updates the target connectivity matrix C* by raising or lowering the weights for those specific edges the user brushed. It's analogous to an artist highlighting or shading connections in the brain's network according to how much synchrony they desire there.
- **Dials and Toggles:** Additional controls include dials for precise target values of **R** and **metastability**. For instance, a user can dial up the target metastability if they feel stuck in a mental rut and want more flexible variability, or dial it down if things feel too chaotic. A toggle for **"criticality mode"** could allow the system to automatically adjust targets to approach the empirically estimated edge-of-criticality for that user (based on complexity metrics), with safety margins. Another toggle might enable **"PAC boost"** where the system will selectively try to enhance cross-frequency coupling in a chosen band pair (say theta phase to gamma amplitude) once basic coherence targets are met. These controls give advanced users a way to personalize deeper, while novices can stick to presets and simpler XY pads.
- **Closed-Loop Adaptation Policy:** The core of Coherence Sculpting is a **closed-loop controller** that continuously compares the **measured connectivity** $\hat{C}(t)$ to the **target connectivity** $C^*$. The difference (error) drives adjustments in the feedback stimuli and, if applicable, in neuro-stimulation parameters:
- In its simplest form, the controller uses a **heuristic proportional policy**: for example, if a particular coherence (e.g. between channels Cz–Pz in beta band) is below target, the system might increase the salience of that connection in the feedback (making the user more aware of it) or suggest strategies (via on-screen prompts) like "try focusing on the feeling of that connection." Conversely, if a metric overshoots (too high), feedback could subtly discourage it (e.g. by reducing the reward or making the visualization less satisfying when overshooting). This proportional nudging is analogous to adjusting a thermostat – small errors lead to small adjustments to push the brain in the desired direction.
- For more advanced control, we have the option of a **Model Predictive Control (MPC)** or **Reinforcement Learning (RL)** based policy. In MPC mode, the system simulates or predicts how changes in stimuli (e.g. visual/auditory feedback parameters) will affect the coherence metrics over a short horizon, and chooses the optimal adjustments to minimize a cost function. The cost function can be defined as:

$$ J = \alpha \cdot | \hat{C}(t) - C^* |^2 + \beta \cdot (R(t) - R^*)^2 + \gamma \cdot (\text{Metastability}(t) - M^*)^2, $$

where the terms penalize deviations from target connectivity, target global synchrony, and target metastability respectively (with weights α, β, γ set by the user or defaults). The controller then outputs slight modifications to the feedback (or in an advanced scenario, to a concurrent brain stimulation amplitude if used) that will best reduce this cost. In RL mode, the system could treat the user's brain as an environment and try different feedback strategies, using reward signals (e.g. when error decreases) to learn an optimal policy over time. While full RL might be experimental, an **adaptive feedback difficulty** approach is implemented: if the user is consistently meeting targets easily, the system makes the task subtly harder (shifting targets closer to a more challenging state); if the user is struggling, it eases off. This kind of adaptive challenge tuning has been shown to facilitate learning in neurofeedback by keeping the user in a flow state of not too easy, not too hard .

- **Safety Guardrails:** The controller will **never push the brain outside safe ranges**. All target metrics have sane limits (e.g. do not drive global R above a conservative value like 0.6 on average, as excessively high synchronization could induce unwelcome effects or reflect hyper-focus to the point of discomfort). If unusual patterns are detected (e.g. signs of epileptiform activity or the user getting too close to a seizure-like state), the system will automatically pause feedback and alert the user. Additionally, if artifacts (blinks, muscle noise) spike, the controller pauses or ignores those windows to avoid reinforcing random muscle activity. These guardrails ensure the closed-loop system **stays within physiological norms** and avoids any "runaway" feedback scenarios.

**Feedback Mapping: Toroidal 3D Visualization & Multimodal Cues**

A signature element of Coherence Sculpting is its **toroidal ($T^2$/$T^3$) visualization** of the user's brain state. We map neural dynamics onto a donut-shaped 3D object (torus) that updates in real time, providing an intuitive yet rich representation of complex brain rhythms:

- **Toroidal Manifold Mapping:** Two angular dimensions of the torus are used to encode oscillatory phase information. For instance, the **major circle angle** (going around the donut's center) could represent the phase of a **slow wave** like theta or alpha. The **minor circle angle** (going around the tube of the donut) could represent the phase of a **faster oscillation** like beta or gamma. Thus, a single point on the torus surface can concurrently represent a specific phase relationship between a slow and fast rhythm – effectively visualizing cross-frequency coupling on the torus surface. When the user's brain exhibits stable coupling (e.g. a consistent theta phase at which gamma bursts occur), the visualization shows a coherent looping path on the torus (the point travels around in a regular orbit). This **closed orbit on the torus** indicates a stable phase-locking or periodic relationship. If the brain state becomes desynchronized or the phase relationship drifts, the point's path will **spread out** and fill the torus surface more randomly – an indication of losing that quasi-periodic structure (analogous to a "torus breakdown" route to chaos ). In summary, a **clean loop** on the torus = desirable stable coupling, whereas a breakdown (wandering all over the torus) signals either a transition to a new state or a loss of integration.

- **Multi-Torus Extension:** We can extend to **$T^3$ (three-torus)** by layering an additional oscillation or signal as a second torus linked together. Practically, this can be rendered as multiple tori interlinked or an evolving 3D shape. For example, a third angle could encode the phase of an even slower rhythm like delta or of the respiration cycle. This way, respiratory or infraslow phase becomes another dimension that can modulate the torus presentation (e.g. the whole torus could slowly twist as you breathe, aligning neural phase with breath phase). HRV or other measures could be mapped to color changes on the torus rim – for instance, a ring around the torus that pulses in sync with heartbeats, giving a subtle cue of heart-brain coupling.
- **Visual Encoding:** The torus is not just a phase plot; its **color, brightness, and texture** continuously change to reflect other metrics:
- **Color Hue** can represent the **Kuramoto order R** (global synchrony). For instance, cool colors (blue/green) when R is low (desynchronized), shifting to warm colors (orange/red) as R increases (more integration). This gives the user an immediate sense of how globally synchronized they are – the torus literally "heats up" with synchrony . Brightness or saturation can encode how close R is to the target value R* (hitting the sweet spot yields a vibrant, clear color, whereas being off-target makes the torus appear dim or grayish).
- **Tube Thickness** of the torus correlates with **band power** in a selected frequency (or overall EEG signal amplitude). If the user is increasing the power of a desired rhythm (say alpha amplitude during a relaxation exercise), the torus' cross-section fattens up. This gives tangible feedback: "bigger donut = more power in the desired band."
- **Surface Texture & Noise:** The torus surface may have a dynamic texture whose pattern randomness corresponds to **signal complexity or entropy**. When the brain is nicely at the critical edge (high complexity but not noise), the texture might display a pleasant, intricate pattern (like gentle fractal ripples). If the brain becomes too regular (low complexity), the texture might simplify into a smooth or repetitive pattern. If it becomes too noisy (very high entropy), the texture could appear as turbulent or erratic noise. Thus, users learn to recognize a balanced, critical pattern – for example, a subtle "woodgrain" or "marbling" texture might be the goal, whereas glassy smooth (too ordered) or static-like (too random) are states to move away from.
- **Audio Feedback:** In parallel with visuals, audio tones provide another feedback channel. The system generates a musical tone or ambience whose properties are tied to the brain metrics:
- The **base pitch** or carrier frequency of the audio can be linked to the user's **IAF** or another dominant frequency. This means the sound literally resonates at a frequency related to the user's own brain rhythm (e.g. a 10 Hz alpha might be mapped to a 440 Hz tone in the audible range, just as a reference). When the user successfully shifts their alpha rhythm, the sound might subtly shift in harmony, reinforcing the change.
- **Volume and Timbre:** As coherence error (deviation from target) decreases, the sound can become more harmonious, louder, or more melodious. Initially, when far from target, the audio might be sparse or discordant (e.g. broken chords or low drone). As the user "tunes" their brain closer to the target state, the music converges to a pleasing melody or rich harmonic tone, rewarding the brain for reaching the desired pattern. This use of **reward modulation** via audio is akin to how traditional neurofeedback beeps indicate success. Here we make it continuous and immersive – e.g. increasing phase alignment could be represented by a rising continuous tone whose clarity improves. Optionally, **binaural beats** or isochronic tones at theta frequency can be overlaid to **scaffold** the brain toward certain states (for non-medical facilitation of relaxation), but these are subtle and can be toggled by the user.
- **Rhythmic Pulses:** If the user's brain is approaching critical dynamics (high metastability), one might hear a slight increase in the **"flutter" or variability** in the sound (like a tremolo or rhythmic pulse that becomes more pronounced). This encodes metastability – e.g. a gentle rhythmic pulse in the music indicates the presence of rhythmic switching dynamics in the brain (which is good up to a point). If metastability overshoots safe ranges, the audio might start to sound **unsteady or broken**, cueing the user to relax focus and let things settle (preventing pushing into potentially erratic states).
- **Haptic Feedback (Optional):** For users with supported devices (wearables, VR suits, or simply a phone vibration), haptic patterns can reinforce changes. For example, slight vibrations on the left or right side could correspond to changes in left- vs. right-hemisphere coherence. Strong coherence increases might produce a gentle tapping sensation in sync with the brain's dominant frequency (like feeling your brain rhythm tactically). If coherence drops or the user loses a desired pattern, the haptic feedback could fade out or change pattern (indicating "loss of lock"). This modality is experimental but could add a grounding sensory channel – useful for users who close their eyes (they can still **feel** the feedback).

Together, these multimodal feedback channels create a closed-loop experience: the user's brain changes drive immediate changes in the torus visualization, soundscape, and possibly haptic pulses. The user learns, often implicitly, how to adjust their mental state to "sculpt" the feedback into a more ordered form (i.e. a stable torus orbit, warm colors, harmonious sounds). This is essentially a form of **brain-computer interface** training. Over repeated sessions, users may improve their ability to enter a desired brain state on demand, even without the feedback – the ultimate goal of neurofeedback training.

**Implementation Details & Pseudocode**

To realize this feature, a careful engineering plan is required, spanning signal processing, algorithm optimization, and UI integration. Below we outline the core computational pipeline and the controller logic in pseudocode, to guide developers:

**Real-Time Metric Pipeline**

The following pseudocode outlines how EEG data is transformed into the connectivity metrics in real time:

```
# Initialize parameters and data structures
sampling_rate = 250  # Hz (for example)
window_length = 2.0  # seconds for connectivity computation
step = 1.0           # slide window by 1s (50% overlap if window is 2s)
buffer = RingBuffer(size=window_length * sampling_rate)
band_filters = {band: DesignBandpassFilter(band) for band in [delta, theta, alpha, beta, gamma]}

# Data processing loop (runs continuously in real time)
for each new_eeg_sample in EEG_stream:
    buffer.append(new_eeg_sample)
    if buffer.is_full():
      # 1. Preprocess window (artifact removal could be inserted here)
      epoch = buffer.get_data()  # get last window_length seconds of data
      epoch = remove_artifacts(epoch)  # e.g., threshold EOG channels, ASR, etc.

      # 2. Compute analytic signals for each band and channel
      phase_matrix = {}  # phase_matrix[band][channel] will store instantaneous phase
      for each band in band_filters:
        band_epoch = band_filters[band].apply(epoch)      # band-pass filter each channel signal in this band
        analytic = HilbertTransform(band_epoch)           # complex analytic signal via Hilbert
        phase_matrix[band] = np.angle(analytic)           # instantaneous phase per channel in this band

      # 3. Compute connectivity metrics in this window
      connectivity = {}  # will hold matrices for PLV, wPLI, etc.
      # Example: compute PLV across all channel pairs for a specific band (say alpha)
      band = "alpha"
      num_channels = phase_matrix[band].shape[0]
      PLV_matrix = np.zeros((num_channels, num_channels))
      for i in range(num_channels):
        for j in range(i+1, num_channels):
          phase_diff = phase_matrix[band][i] - phase_matrix[band][j]  # vector of phase diffs over time
          PLV = abs(np.mean(np.exp(1j * phase_diff)))            # phase-locking value
          PLV_matrix[i,j] = PLV_matrix[j,i] = PLV
      connectivity["PLV_alpha"] = PLV_matrix
      # Compute wPLI or imaginary coherence similarly (using phase_matrix).
      connectivity["wPLI_alpha"] = compute_wPLI(phase_matrix["alpha"])
      # ... Repeat for other bands or specific region pairs as needed.

      # 4. Global measures
      # Kuramoto Order Parameter R(t): magnitude of average phase vector across oscillators
      all_phases = phase_matrix["alpha"]  # or could use a particular band or a weighted combination
      R_complex = np.mean(np.exp(1j * all_phases.flatten()))  # mean phase vector
```

```
R = abs(R_complex)
global_metrics = {}
global_metrics["R"] = R
# Update metastability (std of R over recent history)
metastability_series.append(R)
if metastability_series.length > some_history:
    metastability_series.pop_oldest()
global_metrics["metastability"] = np.std(metastability_series)

# 5. Cross-frequency coupling (example: theta phase to gamma amplitude coupling)
theta_phase = phase_matrix["theta"]
gamma_amp = np.abs(HilbertTransform(band_filters["gamma"].apply(epoch)))  # gamma amplitude envelope
global_metrics["theta-gamma_PAC"] = compute_PAC(theta_phase, gamma_amp)

# 6. Complexity measures
global_metrics["LZ_complexity"] = lempel_ziv_complexity(epoch)
global_metrics["perm_entropy"]   = permutation_entropy(epoch)
global_metrics["spectral_slope"] = estimate_1f_slope(epoch)

# The metrics (connectivity and global) are now available for feedback and controller.
 # Proceed to feedback update (handled in controller loop).

 buffer.slide(step * sampling_rate)  # advance the buffer by 'step' samples (overlap for next window)
```

In the above pseudocode, steps like remove_artifacts() would encapsulate artifact rejection (for example, using an algorithm like Artifact Subspace Reconstruction to remove transient noise or eyeblink components). The **Hilbert transform** is used for phase extraction after bandpass filtering , which is a common method for phase-based connectivity. We ensure these computations are vectorized and optimized (in practice, libraries like NumPy, SciPy or MNE can accelerate this, and one could also use the FFT-based convolution for filters to meet real-time speed). The connectivity calculations are done per frequency band of interest; depending on needs, we may not compute full matrices for all bands every update (we can focus on a subset of edges or summary metrics to reduce computation). Notably, by using measures like wPLI  and ignoring zero-lag phase differences, we account for volume conduction, meaning our feedback truly reflects functional coupling rather than artifacts of the reference or scalp conduction.

**Closed-Loop Controller Loop**

Once the metrics are computed, the controller uses them to update the feedback and guide the user. The controller runs at the same interval as metric updates (e.g. every 1 second, after each new window's metrics are available):

```
# Define target metrics (could be dynamically adjusted by user controls)
target_connectivity = load_target_profile(user_selection)  # e.g., target PLV for certain pairs/bands
R_target = user_defined_R_target   # e.g., 0.5 (moderate global synchrony)
M_target = user_defined_metastability_target  # e.g., 0.1 (desired std of R)

# Controller loop
while session_running:
   wait_for_new_metrics()  # synchronize with metric pipeline outputs
   current_connectivity = connectivity  # from the metric pipeline (e.g., PLV matrix)
   current_globals = global_metrics     # e.g., R, metastability, etc.

   # 1. Compute errors relative to targets
   conn_error = compute_matrix_error(current_connectivity, target_connectivity)
   R_error = current_globals["R"] - R_target
   M_error = current_globals["metastability"] - M_target

   # 2. Adapt feedback based on errors (Heuristic policy example)
   # For each relevant connection/band, adjust visual/audio cues proportionally
```

```
for each connection (i,j,band) in target_connectivity:
    error_ij = current_connectivity[band][i,j] - target_connectivity[band][i,j]
    # Example: adjust torus segment brightness or outline on that connection
    visualization.adjust_edge_highlight(i,j, band, -error_ij)
    # (If error is positive, overshooting target, so maybe cool down that edge;
    # if negative (below target), highlight it to draw user attention.)

# Adjust global visual properties
visualization.set_color_hue_based_on_R(current_globals["R"], R_target)
visualization.set_texture_noise_based_on_entropy(current_globals["perm_entropy"])
# ... similarly adjust audio: e.g., volume or pitch modulation from R_error/M_error.

# 3. Optional advanced control (MPC/RL)
if controller_mode == "MPC":
    # Define cost gradients for adjusting feedback parameters (simplified)
    cost = alpha*norm(conn_error)**2 + beta*(R_error**2) + gamma*(M_error**2)
    # Use an optimization method or pre-trained model to minimize cost.
    feedback_params = optimizer.solve_for_min_cost(cost, current_state)
    apply_feedback_parameters(feedback_params)  # e.g., change audio tone frequencies, visual gains, etc.

if controller_mode == "RL":
    # In reinforcement learning mode, we would update a policy based on reward
    reward = compute_reward(conn_error, R_error, M_error)
    policy.update(last_action, reward, current_state)  # imaginary policy update
    action = policy.sample_action(current_state)
    apply_feedback_parameters(action.feedback_params)

# 4. Safety checks
if artifact_detected() or current_globals["R"] > safe_R_max or unusual_EEG_detected():
    trigger_pause_and_alert()
    # (E.g., pause session if large EMG artifact or if R skyrockets unexpectedly indicating potential issue.)

# Loop continues for next metric update...
```

In this pseudocode, we show both a simple heuristic adjustment and placeholders for MPC/RL modes. In practice, the **heuristic mode** is likely sufficient and more interpretable: it essentially uses the error signals to make the feedback visualization/audio more or less intense, which indirectly nudges the user. For example, if a particular coherence is below target, the system might highlight that connection in the visualization (making it glow or throb), effectively telling the user's brain "pay attention here." Users often subconsciously adjust their mental focus to respond to such cues (e.g., trying different mental strategies until the feedback improves). On the other hand, the **MPC/RL modes** would be more "under the hood," potentially experimenting with different feedback mappings or difficulty settings to see what helps the user learn fastest. There is active research in using adaptive algorithms to improve neurofeedback efficiency , so we include these options to future-proof the system.

Throughout, the controller also monitors session timing (enforcing breaks if needed – e.g., a fatigue timer that suggests pausing after 20 minutes of intense training) and ensures **disclaimers and safety messages** remain visible (reinforcing that this is not a medical device and that users should not over-exert themselves). All feedback adjustments happen within the 300 ms window after new data arrives, to maintain a fluid real-time interaction.

**Validation, Evaluation and Deliverables**

To deliver this feature successfully, we outline the following components and an evaluation plan:

- **Functional Specification & UI Design:** We will produce a detailed spec (the document you're reading is part of it) and wireframes/mockups for the user interface. This will show the layout of the toroidal visualization, control panels (XY pad, brush tool, dials), and how feedback elements (colors, sounds) correspond to metrics. It will also detail the flow of a session (setup, calibration of IAF, live feedback, pausing, summary screen). We'll define performance requirements (the aforementioned latency budget, CPU usage targets) to ensure it runs on typical hardware.

- **Technical Pipeline & Pseudocode:** As shown above, pseudocode for the key algorithms (phase extraction, connectivity calculation, control loop) is provided. These serve as a blueprint for developers. A more formal **algorithmic description** (perhaps in a design document or as inline code comments) will accompany the implementation, explaining how each metric is derived. We will also define a **JSON schema** for saving/loading target profiles (so that power users or researchers can specify target connectivity matrices or load presets easily).
- **Literature Basis and Parameters:** We conducted a deep search of relevant neuroscience literature to ground our design choices. Coherence Sculpting builds on evidence that **EEG coherence can be trained** and linked to cognitive outcomes . For instance, Kober *et al.* (2020) showed that up-regulating EEG coherence in the low beta band led to improved memory performance, whereas down-regulating that coherence was difficult for users . This informs our decision to focus on **increasing beneficial connections** rather than forcing decreases (users may find it easier to raise coherence or create new coupling than to actively decorrelate specific regions). Another study in children with reading disorders found that 30 sessions of neurofeedback not only improved reading skills but also **normalized coherence patterns** (excessive slow-band coherence decreased, and helpful alpha coherence increased) . This suggests our feature could similarly guide users toward more optimal connectivity profiles over training. We selected metrics like wPLI due to their robustness in real time , and metastability as a proxy for cognitive flexibility . We will document these rationales and parameter choices (e.g., typical healthy ranges for wPLI, R, metastability) in an internal report so that stakeholders understand the scientific grounding. Typical target ranges will be set based on literature (for example, Kuramoto R usually hovers around 0.2–0.5 in eyes-open rest for EEG; we might use that as a baseline target for a relaxed state ). If we observe a user consistently exceeding or not reaching these ranges, it could indicate either exceptional ability or an issue with signal quality; the system can flag this.
- **User Testing and Evaluation Plan:** We plan to evaluate Coherence Sculpting in within-subject studies. A suggested evaluation is a **cross-over design** where users have sessions with and without the adaptive feedback to see if the feature accelerates learning or yields greater changes in coherence. Key metrics for success include:
- *Time in Target*: The percentage of session time the user's metrics stay within a defined threshold of the target state. We expect that with the interactive feedback, this time-in-target will increase over repeated sessions (learning effect).
- *Effect on Performance/State*: Although we do not make clinical claims, we can include some **behavioral or subjective measures**. For example, if a user is using a "focus" profile, we might administer a simple concentration task or self-report scale before and after training to see if they feel more focused. The literature suggests coherence training can transfer to cognitive improvements (e.g. memory improvement was noted after beta coherence training ). We will look for such improvements anecdotally and ensure any claims are conservative and grounded in evidence.
- *Retention*: We will check if users can maintain the trained state *without* feedback after a number of sessions. The ultimate test of neurofeedback is whether the brain can recall the state on its own. In some sessions, we might include "transfer blocks" (as some protocols do ) where the feedback is intermittently turned off. During those periods, we see if the user can sustain the desired coherence using the internalized skill. Successful retention would be a strong indicator of the feature's efficacy.
- *Adverse Events*: We will track any user-reported discomfort, such as headaches, dizziness, or unusual feelings. Neurofeedback is generally well tolerated, with very few side effects (perhaps mild fatigue or headache in some cases) . If any user experiences negative symptoms, we will adapt the protocol (e.g., lower the target intensities, shorten session length, or ensure more breaks). In our testing, we include a **fatigue questionnaire** at the end of each session and have an immediate abort option if the user feels unwell. No serious adverse effects are expected since the system is non-invasive and we are staying within normal brain activation ranges .
- **Safety and Ethics Checklist:** Prior to release, we will have the feature reviewed by an ethics or safety board. Key points include:
- Clear **user guidance and consent**, explaining that this is not a medical device and is for self-exploration/entertainment/education. The user should understand the intended use and limits.
- **Privacy:** EEG data is sensitive. All data processing can be local (on-device) to avoid cloud risks. If any cloud components exist (e.g., optional data logging for the user's tracking), data will be encrypted and opt-in.
- **No overclaiming:** Marketing and in-app text will avoid any suggestion that this can treat disorders. It will highlight potential benefits like "may help you relax or improve focus" with appropriate caution that individual results vary.
- **Session Limits:** Built-in defaults will limit sessions to ~20-30 minutes and at most 1-2 sessions per day, to prevent overuse. Neurofeedback overtraining is rare but possible; fatigue or frustration can set in if sessions are too long. The app will encourage regular breaks and show a countdown timer for how long the current session has been running.
- **Artifact Handling:** We will implement robust artifact rejection and **auto-pause** if signal quality is low (e.g., if many electrodes lose contact or the user moves too much). This prevents feeding back noisy data which could confuse or frustrate the user. The UI will gently prompt the user to check electrodes or relax muscles if artifacts are detected.
- **Transparency:** For advanced users, we'll provide an "open metrics" view where they can see the raw values (PLV, wPLI, etc.) in real-time if they wish. This caters to the prosumer or researcher community and allows verifying

that the system is doing what it claims. We might also open-source parts of the code (e.g., the signal processing pipeline) for community trust and contribution.

By covering the above points, we aim to create a feature that is **innovative yet responsible**. The deliverables will include this one-page spec document (serving both as a high-level overview for stakeholders and a semi-technical guide for the team), UI prototypes, and a working prototype of the algorithm ready for internal testing. With iterative feedback from test users, we will refine the experience to ensure Coherence Sculpting is engaging, effective, and safe.

**Conclusion**

Coherence Sculpting stands at the intersection of cutting-edge neuroscience and interactive design. It leverages proven metrics of brain network activity (like coherence and metastability) and wraps them in an intuitive feedback loop that users can manipulate. Early evidence that humans can learn to change their brain connectivity with feedback underpins the viability of this approach. Our design maximizes this potential by providing rich, **toroidal visual feedback** and comprehensive metrics that capture the elusive balance of order and chaos in the brain. If successful, this feature could open new frontiers in how people engage with their own neural activity – not just watching their brainwaves, but *sculpting* them towards desired patterns. It's a bold step beyond traditional neurofeedback, and with careful implementation and evaluation, it promises a unique value proposition for both passionate brain-hackers and curious newcomers alike. We proceed with both excitement and caution, grounded in research and guided by user well-being, to make Coherence Sculpting a reality.