Brief Description of the Drawings

FIG. 1 is a block diagram of a system for bias-blind behavioral pattern recognition, showing multi-modal data input, a baseline establishment engine, a cultural-context module, a calibration processor, a reporting module, and a quality assurance module.

FIG. 2 is a flowchart illustrating a method of baseline calibration, including collecting behavioral data, initializing with ephemeral initialization vectors, computing intra-individual statistics, applying calibration updates, detecting convergence, and generating reports.

FIG. 3 is a diagram showing the operation of ephemeral initialization vectors (EIVs) over time, including initialization, transition, and individual-only phases with weighting decay.

FIG. 4 is a schematic representation of an adjustment vector framework, showing the application of amplification, dampening, threshold, and relationship vectors to behavioral input data.

FIG. 5 is a graph illustrating convergence of variance over time, showing an exponential decay toward a stable value.

FIG. 6 is a block diagram of an adversarial detection module, including cross-modal coherence checks, temporal stability analysis, physiological plausibility testing, statistical anomaly detection, and flagging of adversarial conditions.

Detailed Description of the Drawings

FIG. 1 illustrates an exemplary system 100 for bias-blind behavioral pattern recognition. The system 100 includes a multi-modal data input module 110 configured to receive behavioral signals such as speech, visual engagement, response timing, gestures, keystroke dynamics, and handwriting. The system further includes a baseline establishment engine 120 configured to compute intra-individual statistics including mean and variance, while not defining or computing any between-group variance. A cultural-context module 130 generates a context vector C based on one or more communication style dimensions such as directness, authority, expressiveness, and feedback. A calibration processor 140 applies adjustment vectors and drift detection to align incoming data with the individual's baseline. A reporting module 150 generates configurable outputs including session reports and alerts. A quality assurance module 160 performs integrity checks, including sensor validation, status monitoring, and adversarial detection.

FIG. 2 illustrates a method 200 of baseline calibration. At step 210, behavioral data are collected from one or more modalities. At step 220, the method initializes with one or more ephemeral initialization vectors (EIVs). At step 230, intra-individual mean and variance values are computed. At step 240, calibration updates are applied using amplification, dampening, threshold, and relationship parameters. At step 250, convergence is evaluated based on variance stability thresholds. If convergence is not achieved, the method loops back to step 240. If convergence is achieved, the method proceeds to step 260, in which EIV weighting is reduced to zero. At step 270, the system generates session reports or receipts.

FIG. 3 illustrates the operation of EIVs 300 over time. During an initialization phase 310, occurring within approximately 0–5 minutes, EIVs provide broad tolerance ranges. During a transition phase 320, occurring within approximately 5–15 minutes, the individual baseline gradually overrides the EIV weighting. During an individual-only phase 330, beginning after approximately 15 minutes, the weighting of the EIVs decays to zero, leaving the calibration entirely dependent on the individual's baseline.

FIG. 4 illustrates an adjustment framework 400. An input vector X is provided to an amplification matrix A and a dampening vector D, resulting in scaled input values. A threshold vector T is applied to shift the scaled input. A relationship matrix R is applied to capture correlations across input features. The resulting adjusted vector X′ is output as $X' = A\cdot(D\circ X) + T + R\cdot X$.

FIG. 5 illustrates a convergence curve 500. The vertical axis represents variance σ^2, and the horizontal axis represents time. The curve 510 shows exponential decay of variance according to $\sigma^2(t) = \sigma^2_0\cdot e^{(-t/\tau)} + \sigma^2\_stable$. The curve approaches a stable variance σ^2_stable after a convergence interval, such as approximately 20 minutes.

FIG. 6 illustrates an adversarial detection module 600. The module 600 includes a cross-modal coherence checker 610 configured to compare variance across modalities, a temporal stability analyzer 620 configured to monitor variance changes over time, a physiological plausibility tester 630 configured to assess signal validity against physiological ranges, and a statistical anomaly detector 640 implementing sequential probability ratio tests or Mahalanobis distance checks. A flagging unit 650 receives outputs of the analyzers and produces adversarial condition indicators, which may be logged or used to terminate a session.

Claim–Figure Support Matrix (BB-BPRF)

Claim / Element	Supporting Figure(s)
Independent Method Claim 1 – Collect behavioral data; compute μ, σ²_individual only; no σ²_between; generate calibration coefficients; apply real-time adjustments	**Fig. 1** (System Overview: Baseline Establishment + Calibration Processor), **Fig. 2** (Baseline Calibration Flow)
Claim 2 – Variance-reduction coefficients (σ²_pre –	**Fig. 2** (Flow: baseline → recalibration)

Claim / Element	Supporting Figure(s)
σ^2_post)/σ^2_post	
Claim 3 – Between-group variance σ^2_between undefined	**Fig. 1** (Baseline Establishment Engine)
Claim 4 – Optimization with Lipschitz continuity; monotone convergence	**Fig. 5** (Convergence Curve)
Claim 5 – Alerts configurable: routine / urgent / predictive	**Fig. 1** (Reporting & Alerts Module)
Claim 6 – Compare current session vs prior baselines	**Fig. 2** (Flow loop: recalibration + drift detection)
Claim 7 – Compile-time & runtime enforcement against cross-user ops	**Fig. 1** (QA / Integrity Module)
Claim 8 – Demographic blindness enforced structurally	**Fig. 1** (System Overview: exclusion of σ^2_between, proxy guards)
Claim 9 – Generate EIVs as calibration placeholders	**Fig. 3** (EIV Weight Decay), **Fig. 2** (Initialize with EIV placeholders)
Claim 10–12 – EIV decay weight → 0, confidence growth, 90-second convergence	**Fig. 3** (Decay curve), **Fig. 5** (90-sec convergence confidence)
Claim 13 – Baseline calibration with EIVs evidencing crypto init (no demographics)	**Fig. 3** (EIV transition)
Claim 14 – Configurable reports excluding identifiers, referencing unlinkable IDs	**Fig. 1** (Reporting & Alerts Module)
Claim 15–16 – Collect system status / QA checks	**Fig. 1** (QA / Integrity Module), **Fig. 6** (Adversarial Detection)
Claim 17–18 – Context-conditioned statistics & Bayesian updating	**Fig. 2** (Baseline Flow + update loop)
Claim 19 – Drift score: deviation beyond threshold triggers recalibration	**Fig. 2** (Loopback recalibration step)
Claim 20 – Convergence via successive adjustment vectors	**Fig. 5** (Convergence Curve: variance decay)
Claim 21 – Sub-millisecond low-complexity inference	**Fig. 4** (Adjustment Vector Framework: A/D/T/R)
Claim 22 – Applies identically across modalities	**Fig. 1** (Multi-Modal Data Inputs)
Claim 23 – Cultural-context module generates vector C = [c1,c2,c3,c4]	**Fig. 1** (Cultural-Context Module), **Fig. 4** (Adjustment applied with C inputs)
Claim 24 – Human feedback integration (bounded ≤0.2, reject proxies)	**Fig. 1** (QA/Integrity Module, Feedback Path)
Claim 25 – Multi-modal fusion weights from within-individual variances	**Fig. 1** (System Overview, fusion at Calibration Processor)
Claim 26 – Text evaluation: prompt-specific baselines, no cross-prompt normalization	**Fig. 1** (Multi-Modal Data Input; Baseline Engine)
Claim 27–29 – Information utilization advantage 6–12× vs demographic approaches	**Fig. 5** (Information Content Advantage curve)
Claim 30–32 – Convergence acceleration; network-driven scaling effects	**Fig. 5** (Convergence scaling with sessions)
Claim 33–34 – Adversarial detection: cross-modal	**Fig. 6** (Adversarial Detection Module)

Claim / Element	Supporting Figure(s)
coherence, weighting history 70/30	
Claim 35 – Validity checks: exclude invalid params >30%	**Fig. 6** (Out-of-bounds detection, termination path)
System Claim 37 – Multi-modal input, baseline engine, cultural-context, calibration processor	**Fig. 1** (System Overview, all modules)
System dependent claims (QA module, reporting module, EIV generator, proxy-detector, secure enclave, cryptographic receipt generator, etc.)	**Fig. 1** (QA & Integrity, Reporting), **Fig. 3** (EIV decay), **Fig. 6** (Adversarial / proxy checks)
Medium Claim 62 – Instructions for collecting data, establishing intra-individual baselines, applying coefficients	**Fig. 2** (Flowchart), **Fig. 4** (Adjustment Equation implementation)
Medium dependent claims (system status checks, QA rejection, cross-user prohibition, configurable reports)	**Fig. 1** (QA, Reporting, Enforcement), **Fig. 6** (Adversarial checks)

Figure 6 – Adversarial Detection Module

```
Inputs ---> Cross-Modal Coherence
        |-> Temporal Stability Checks
        |-> Physiological Plausibility
        |-> Statistical Tests (SPRT, Mahalanobis)
                |
                v
        +--------------------+
        | Adversarial Flags  |
        |  - Weight history  |
        |  - Session reject  |
        |  - Alerts          |
        +--------------------+
```

Figure 5 – Convergence Curve

```
Variance σ²(t)
|
|\
| \              σ²(t) = σ²₀·e^(−t/τ) + σ²_stable
|  \
|   \       ~95% convergence by 20-25 minutes
|    \
|     \
|      ------------------> Time
```

Figure 4 – Adjustment Vector Framework

```
Input Vector X ---> [Amplification A] ---\
                  [Dampening D] -----------*--> + [Threshold T] --> + [Relationship R·X] --> X_adjusted
(Equation reference:  X_adjusted = A·(D∘X) + T + R·X )
```

Figure 3 – Ephemeral Initialization Vectors (EIV) Operation

```
EIV Weight vs Time
|
|\
| \       Phase 1: Initialization (0-5 min)
|  \      Broad tolerance ranges
|   \
|    \    Phase 2: Transition (5-15 min)
|     \   Individual overrides EIV
|      \
|       \  Phase 3: Individual Only (15+ min)
|        \ Weight_EIV = 0
|
+-------------------------------- Time
```

Figure 2 – Baseline Calibration Flowchart

```
Start
  |
  v
Collect behavioral data
  |
  v
Initialize with EIV placeholders
  |
  v
Compute intra-individual μ, σ²_individual
  |
  v
Apply calibration updates (A, D, T, R)
  |
  v
Convergence reached?
  |--- No --> Loop back to Apply updates
  |--- Yes
  v
Zero out EIV weighting (Weight_EIV → 0)
  |
  v
Generate reports / receipts
  |
  v
End
```

Figure 1 – System Overview

```
+-------------------------+
| Multi-Modal Data Input  |
|  - Speech               |
|  - Visual               |
|  - Response Timing      |
|  - Gestures/Keystrokes  |
+-----------+-------------+
            |
            v
+-------------------------+
| Baseline Establishment  |
| Engine                  |
|  - Compute μ, σ²_indiv  |
|  - No σ²_between        |
+-----------+-------------+
            |
            v
+-------------------------+
| Cultural-Context Module |
|   C = [c1,c2,c3,c4]     |
| (Directness, Authority, |
|  Expressiveness, Feedback)|
+-----------+-------------+
            |
            v
+-------------------------+
| Real-Time Calibration   |
| Processor               |
|  - Apply A/D/T/R        |
|  - Drift Detection      |
+-----------+-------------+
            |
            v
+-------------------------+
| Reporting & Alerts      |
|  - Configurable reports |
|  - Session metadata     |
|  - Urgent/routine notif |
+-------------------------+
            |
            v
+-------------------------+
| QA / Integrity Module   |
|  - Sensor checks        |
|  - Status validation    |
|  - Adversarial detection|
+-------------------------+
```