Advanced Leak Correlation With AI Integration (v1)

You already have “classic” two-sensor correlation with decent pre-filtering and stacking, so the next gains come from:

1. making the physics explicit (c, dispersion, pipe graph),
2. using more informative correlation variants,
3. putting a probabilistic wrapper on top (uncertainty, priors),
4. integrating your leak classifier to **gate** and **weight** the correlation.

---

1. Make It Physics-Aware Instead of Pure “τ-Peak” Correlation

1.1. Joint {Position, Velocity} Estimation

Let pipe length be **L**, leak position be **x** from sensor A, and the effective propagation velocity be **c**.

Classic formula:

$$\tau(x, c) = \frac{L - 2x}{c}$$

Instead of assuming **c** and solving for **x**, search jointly over x, c and maximize a correlation score.

1.2. Dispersion-Aware Multi-Band Model

Split into K frequency bands, compute a delay τ_k per band, and fit a frequency-dependent velocity model:

$$c_k = c_0 + \alpha f_k$$

Solve for leak position **x** and dispersion parameters simultaneously.

---

2. Use More Informative Correlation Variants

2.1. GCC-PHAT, Roth, SCOT

Use several generalized cross-correlation kernels and fuse:

- PHAT
- Roth
- SCOT
- Classical

2.2. Multi-Resolution / Wavelet Correlation

Perform correlation at multiple resolutions using decimated signals or wavelet-based coherence extraction.

---

3. Coherence-Driven Leak Band Selection

Compute magnitude-squared coherence:

$$C_{xy}(f) = \frac{|E[X(f)Y^*(f)]|^2}{E[|X(f)|^2]E[|Y(f)|^2]}$$

Use coherence to auto-detect leak-relevant frequency bands and weight correlations accordingly.

---

4. Bayesian Leak Position Estimation

Define:

- prior $p(x)$
- likelihood $p(D|x) \propto \exp(\beta s(x))$

Compute posterior:

$$p(x|D) \propto p(D|x)p(x)$$

Return:

- MAP estimate
- credible intervals
- entropy-based quality index

---

5. AI Integration Into the Correlation Pipeline

There are **four levels** of AI integration:

1. **Window gating and weighting** using CNN leak probabilities.
2. **Learned cross-channel feature extraction** (learned GCC).
3. **Correlation peak referee** (MLP to validate peaks and correct bias).
4. **Sequence model over corrgrams** (CNN/Transformer over R_t(τ) images).

5.1. Level 1 — Window Gating + Weighting

Run the leak CNN per window to produce:

$$p_{leak}(w)$$

Use it to weight correlation windows:

$$R(\tau) = \frac{1}{\sum w} \sum_w p_{leak}(w) R_w(\tau)$$

Combine with SNR/coherence:

$$w_{final}(w) = p_{leak}(w)^\gamma \cdot SNR(w)^\delta$$

5.2. Level 2 — Learned Cross-Spectral Mask (Learned GCC)

Construct input tensor:

- |X|
- |Y|
- phase(X)
- phase(Y)

Train a CNN/UNet to predict a time-frequency mask $M_w(f,k)$ to weight cross-spectral products. Inverse FFT yields a learned correlation.

5.3. Level 3 — Correlation Peak Referee

Extract features from stacked $R(\tau)$:

- normalized curve
- top-K peak stats
- peak width, PSR
- band-wise delay consistency

Feed into an MLP that outputs:

- corrected leak position $\hat{x}$
- uncertainty σ_x
- false-positive probability

5.4. Level 4 — Corrgram-Based Sequence Model

Build corrgram:

$$C[t,i] = R_t(\tau_i)$$

Train a CNN/Transformer over (time × lag) to predict leak position and confidence.

---

6. Robust Stacking and Statistics

Upgrade stacking:

- weighted trimmed mean
- median stacking
- Huber M-estimator

Use peak stability metrics:

- peak-to-sidelobe ratio
- curvature at peak
- multi-band peak alignment

---

7. Multi-Sensor Pipe-Graph Correlation

Generalize to N sensors:

$$\tau_{ij}(x,c) = \frac{|x - x_j| - |x - x_i|}{c}$$

Solve for x via least-squares across all sensor pairs or on a branching pipe graph.

---

8. Simulation-Based Bias Correction

Simulate synthetic leak events, run full pipeline, learn a correction model:

$$x_{true} = f(x_{est}, features)$$

Apply f() in production to remove systematic bias.

---

9. Recommended Architecture Integration

1. Preprocess: filtering, resampling, adaptive band selection.
2. Window-level CNN → p_leak(w).
3. Compute per-window GCC-PHAT correlations.
4. Weighted robust stacking using p_leak × SNR.
5. Feed R(τ) into correlation referee AI.
6. Optionally construct corrgram and feed into a 2D CNN.

This yields a modern, hybrid physics + ML correlation engine.