Supplementary Information for On-Board Real-Time AI Compensation System for Nonlinear Drift in Electrochemical Seismometers

1 Supplementary Note 1: MET Mechanism and Small-Signal Model

The small-signal frequency response of the MET can be obtained by cascading the mechanical vibration pickup subsystem and the electrochemical conversion subsystem. The mechanical subsystem converts external vibration into volumetric flow rate of the electrolyte in the flow channel, and the electrochemical subsystem further converts the flow-induced ion concentration perturbation into electrode current. Therefore, the overall transfer function can be written as Eq. (S1).

$$H(s) = H_{\text{mech}}(s) \cdot H_{\text{ec}}(s) \tag{S1}$$

where $H_{\text{mech}}(s)$ denotes the transfer function of the mechanical vibration pickup, and $H_{\text{ec}}(s)$ represents the transfer function of the fluid mechanics and electrochemical conversion.

Under small-signal conditions, the mechanical vibration pickup can be approximated as a linear second-order system composed of rubber membrane elasticity, fluid inertial mass, and hydrodynamic damping. Let the volumetric displacement of the fluid in the flow channel be V and the external vibration acceleration be a, the dynamic equation can be written as Eq. (S2).

$$\frac{d^2V}{dt^2} + \frac{R_{\text{h}} S_{\text{ch}}}{\rho L}\frac{dV}{dt} + \frac{k_{\text{r}}}{\rho L}V = -S_{\text{ch}}a \tag{S2}$$

where S_{ch} is the channel cross-sectional area, k_{r} is the membrane elasticity coefficient, ρ is the electrolyte density, L is the channel length, and R_{h} is the hydrodynamic resistance. Define the volumetric flow rate as $Q(t) = dV(t)/dt$, and the external vibration velocity satisfies $a(t) = dv(t)/dt$. Under zero initial conditions, applying the Laplace transform to Eq. (S2) yields the transfer function of the mechanical subsystem from vibration velocity to volumetric flow rate:

$$H_{\text{mech}}(s) = \frac{Q(s)}{v(s)} = \frac{-S_{\text{ch}}s^2}{s^2 + \frac{R_{\text{h}}S_{\text{ch}}}{\rho L}s + \frac{k_{\text{r}}}{\rho L}} \tag{S3}$$

Eq. (S3) shows that in the mechanical subsystem, $\frac{k_r}{\rho L}$ determines the small-signal natural frequency, and $\frac{R_h S_{ch}}{\rho L}$ governs the damping term. Thus, the rubber membrane stiffness, hydrodynamic resistance, and equivalent fluid mass jointly control the low-frequency transition characteristics.

After the mechanical flow enters the vicinity of the electrodes, it alters the ion concentration field and the concentration gradient at the electrode surface. The small-signal electrochemical conversion process can be jointly described by the incompressible fluid equation, the Nernst–Planck transport equation, and the electrode current expression:

$$\begin{cases} \nabla \cdot \mathbf{u} = 0, \\ \rho \dfrac{d\mathbf{u}}{dt} = -\nabla P + \mu_{\text{elec}} \nabla^2 \mathbf{u} + \rho \mathbf{a}, \\ \dfrac{\partial C_k}{\partial t} = z_k m_k N_f \nabla \cdot (C_k \nabla \varphi) + D_k \nabla^2 C_k - \mathbf{u} \cdot \nabla C_k, \\ I = -D_{I_3} q_e \oint \left(\nabla C_{I_3^-} \cdot \mathbf{n} \right) dS_e. \end{cases} \tag{S4}$$

where $\mathbf{u}$ is the electrolyte velocity vector, P is the pressure, μ_{elec} is the dynamic viscosity, $\mathbf{a}$ is the external acceleration, C_k is the concentration of the k-th ion species, z_k is the charge number, m_k is the ion mobility coefficient, φ is the electrolyte potential, D_k is the diffusion coefficient, N_f is the Faraday constant, q_e is the elementary charge, $\mathbf{n}$ is the outward normal vector of the electrode surface, and S_e is the electrode surface area. Under small-signal linearization and ideal electrode assumptions, the coupled process reduces to a first-order lag from volumetric flow rate to current output:

$$H_{ec}(s) = \frac{I(s)}{Q(s)} = \frac{2 q_e n_c S_e}{1 + \frac{s}{\omega_d}} \tag{S5}$$

where n_c is the charge carrier concentration, $\omega_d = D_{I_3}/h^2$ is the characteristic angular frequency of ion diffusion, and h is the electrode spacing. The overall small-signal transfer function is thus obtained:

$$H(s) = \frac{-S_{ch} s^2}{s^2 + \frac{R_h S_{ch}}{\rho L} s + \frac{k_r}{\rho L}} \cdot \frac{2 q_e n_c S_e}{1 + \frac{s}{\omega_d}} \tag{S6}$$

Eq. (S6) indicates that the ideal small-signal response of an MET sensor is jointly shaped by a mechanical second-order high-pass stage and an electrochemical first-order low-pass stage: the mechanical parameters govern the low-frequency end and the resonance region, while ion diffusion near the electrodes controls the high-frequency end, and together they determine the passband sensitivity.

$$H(s) = S \frac{4\pi \zeta \eta s}{s^2 + 4\pi \zeta \eta s + 4\pi^2 \eta^2} \tag{S7}$$

Here, η is the equivalent natural frequency, S is the equivalent sensitivity near the peak, and ζ is the equivalent damping ratio. Under the ideal small-signal approximation, $\eta = \frac{1}{2\pi}\sqrt{\frac{k_r}{\rho L}}$, S is jointly determined by the mechanical mid-band gain and the electrochemical low-frequency gain, and $\zeta = \frac{R_h S_{ch}}{2\sqrt{\rho L k_r}}$. Therefore, experimentally observed drifts in η and S directly reflect the variation of mechanical-electrochemical coupling parameters with input magnitude.

As the input magnitude increases, the small-signal linearization condition underlying Eq. (S6) gradually deviates from the actual operating state. Mechanical geometric nonlinearity, fluid inertial nonlinearity, multiplicative ion-transport coupling, and the exponential nonlinearity of electrode reaction kinetics jointly alter the local frequency response.

First, the rubber membrane exhibits geometric nonlinearity under large displacements. Because the membrane edge is constrained by the transducer housing, an increase in the center displacement z introduces additional tensile stress, and the restoring force can be approximately expressed in Duffing form:

$$F_{\mathrm{m}} = k_1 z + k_3 z^3 + \dots \tag{S8}$$

where k_1 is the linear stiffness coefficient and k_3 is the cubic nonlinear stiffness coefficient. The cubic term increases the equivalent stiffness at large amplitudes, shifting the equivalent η in Eq. (S7) toward higher frequencies.

Second, high flow rates amplify the fluid inertial term. The complete fluid dynamic process can be written as:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla P + \mu_{\mathrm{elec}} \nabla^2 \mathbf{u} + \rho \mathbf{a} \tag{S9}$$

Here, $\rho(\mathbf{u} \cdot \nabla)\mathbf{u}$ is the convective inertial term. As the characteristic flow velocity increases, the linear proportionality between pressure drop and flow rate breaks down, and the equivalent hydrodynamic resistance R_{h} becomes amplitude-dependent, thereby affecting S and ζ in Eq. (S7).

Third, the convection-diffusion process in ion transport induces a frequency-dependent multiplicative coupling. The convection term in the Nernst–Planck equation can be written as Eq. (S10):

$$\frac{\partial C_k}{\partial t} = z_k m_k N_{\mathrm{f}} \nabla \cdot (C_k \nabla \varphi) + D_k \nabla^2 C_k - \mathbf{u} \cdot \nabla C_k \tag{S10}$$

Here, $-\mathbf{u} \cdot \nabla C_k$ multiplies the flow velocity and concentration gradient, making the perturbation of the concentration field near the electrode depend on both input amplitude and frequency. When the excitation frequency approaches $\omega_{\mathrm{d}} = D_{\mathrm{I_3}^-}/h^2$, the relative contributions of diffusion and convection change, causing the frequency response drift to exhibit amplitude-frequency coupling.

Finally, the charge transfer process at the electrode surface satisfies the Butler–Volmer exponential relationship:

$$J = J_0 \left[\exp\left(\frac{\alpha_{\mathrm{ct}} n_{\mathrm{e}} N_{\mathrm{f}} \eta}{RT} \right) - \exp\left(-\frac{(1-\alpha_{\mathrm{ct}}) n_{\mathrm{e}} N_{\mathrm{f}} \eta}{RT} \right) \right] \tag{S11}$$

Here, J is the current density, J_0 the exchange current density, α_{ct} the charge transfer coefficient, n_{e} the number of electrons transferred in the electrode reaction, η the overpotential, R the ideal gas constant, and T the absolute temperature. Large-signal excitation alters the surface concentration and overpotential, driving the electrode current beyond the small-signal linear regime and further shifting the passband sensitivity.

The four amplitude-frequency coupling mechanisms described above respectively alter the membrane equivalent stiffness, fluid damping, ion concentration distribution layer, and electrode reaction rate, ultimately manifesting as local shifts in η, S, and ζ with input magnitude in Eq. (S7).

2 Supplementary Note 2: Parallel Wiener Equivalent Modeling

Although the full-range frequency response of the MET exhibits significant magnitude dependence, its measured frequency response at a fixed single magnitude can still be approximated as a local linear second-order system. Based on this local linear characteristic, this paper introduces a three-branch parallel Wiener equivalent model that combines multiple linear dynamic kernels with static nonlinear mappings to model the variation of the system frequency response with input amplitude based on measured data. The input–output relationship of this model is written as

$$y(t) = \sum_{i=1}^{3} f_i\left((h_i * x)(t)\right), \tag{S12}$$

where h_i denotes the i-th local linear dynamic branch and $f_i(\cdot)$ represents the corresponding static nonlinear mapping. Each dynamic branch is approximated by a second-order bandpass element:

$$W_{\mathrm{w},i}(s) = \frac{A_i s}{s^2 + C_i s + B_i}. \tag{S13}$$

The low-, medium-, and high-amplitude branches cover the small-signal response, the peak-frequency migration transition zone, and the large-signal passband gain uplift region, respectively. The superimposed model captures the rightward shift of the natural frequency and the sensitivity increase at 100 Hz as the input amplitude rises.

Fig. S1(a) illustrates the basic structure of the three-branch parallel Wiener equivalent model: each branch consists of a local linear dynamic kernel and a static nonlinear mapping, and the superposition of multiple branches describes the variation of the frequency response with amplitude. Fig. S1(b) compares the parallel Wiener equivalent model with the measured amplitude-sweep results. The model achieves a center-frequency MAE of 1.87 Hz and a sensitivity MAE of 4.53at 100 Hz across 25 magnitude points. This result demonstrates that an architecture combining local linear dynamics with static nonlinear mapping effectively characterizes the amplitude-frequency coupling drift of the MET, and confirms the physical plausibility of feature extraction based on the Wiener model.

3 Supplementary Note 3: Dataset Protocol Details

3.1 Dataset

The dataset obtained from the above experiments contains 300 frequency–magnitude operating conditions, with 8000 steady-state sampling points saved per condition. It uses paired time-domain sequences as basic samples, preserving both amplitude-frequency feature points and the pointwise waveform, phase, and local dynamic information required for end-to-end compensator training. The dataset can be represented as an $F \times M$ matrix $\mathcal{D}$, as shown in Eq. (S14), where $i = 1, \ldots, F$ denotes the frequency index f_i, $j = 1, \ldots, M$ denotes the magnitude index m_j, and $n = 0, \ldots, N-1$ denotes the sampling point index in the output signal sequence.

$$\mathcal{D} = \left\{ \left(\tilde{\mathbf{Y}}_{i,j}, \overset{\circ}{\mathbf{Y}}_{i,j} \right) \,\middle|\, i = 1, \ldots, F, j = 1, \ldots, M \right\} \tag{S14}$$

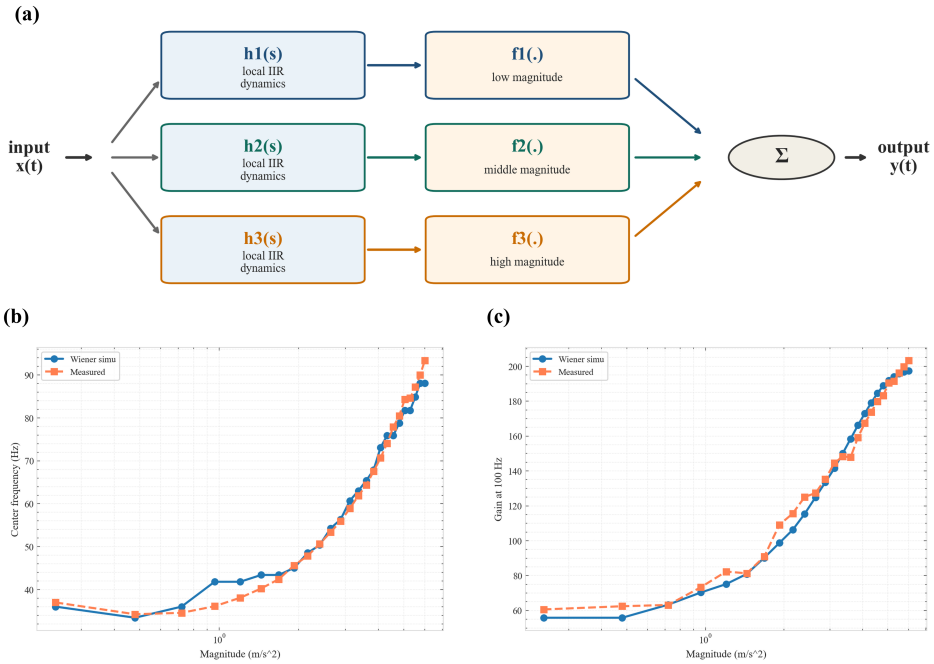

Fig. S1 Parallel Wiener equivalent modeling of the amplitude-dependent MET response. (a) Principle schematic of the three-branch parallel Wiener equivalent model. Each branch combines a local linear dynamic kernel $h_i(s)$ with a static nonlinear mapping $f_i(\cdot)$, and the branch outputs are summed to reproduce the amplitude-dependent response. (b) Center-frequency trajectory reproduced across 25 magnitude points from 0.24 to 6.0 $\mathrm{m/s^2}$. (c) 100 Hz gain trajectory reproduced over the same magnitude range.

$$\begin{cases} \tilde{\mathbf{Y}}_{i,j} = \left[\, \tilde{y}_{i,j}[0],\ \tilde{y}_{i,j}[1],\ \ldots,\ \tilde{y}_{i,j}[N-1] \right] \\ \overset{\circ}{\mathbf{Y}}_{i,j} = \left[\, \overset{\circ}{y}_{i,j}[0],\ \overset{\circ}{y}_{i,j}[1],\ \ldots,\ \overset{\circ}{y}_{i,j}[N-1] \right] \end{cases} \tag{S15}$$

In dataset $\mathcal{D}$, each element $\left(\tilde{\mathbf{Y}}_{i,j}, \overset{\circ}{\mathbf{Y}}_{i,j}\right)$ represents a paired signal sequence: $\tilde{\mathbf{Y}}_{i,j}$ is the measured raw MET output sequence, and $\overset{\circ}{\mathbf{Y}}_{i,j}$ is the output sequence of an ideal linear system under the same frequency and magnitude conditions. Both sequences have length N. The target sequence $\overset{\circ}{\mathbf{Y}}_{i,j}$ is generated by an ideal second-order linear reference system identified from the low-amplitude input regime. The sampling rate is 2000 Hz, each steady-state record contains 8000 points, and the total number of frequency–magnitude operating conditions is 300. This definition assigns a clear objective to the compensation task: the model output should approximate the ideal linear system in the time domain while preserving the target sensitivity and stable natural frequency in the frequency domain. The original tests cover sinusoidal steady-state excitation with per-frequency magnitude sweeps, capturing amplitude variations from weak vibrations to stronger near-field vibrations encountered in subsurface exploration. The data preprocessing pipeline includes steady-state segment extraction,

Table S1 Experimental setup.

Experimental setup	
Vibration Table Distortion Rate	$\leq 1\%$
Data Acquisition Resolution	16-bit
Environment Temperature	25°C
Test Frequency Range	10$\sim$128 Hz
Magnitude Range	0.24$\sim$6.0 $\mathrm{m/s^2}$
Sampling Frequency	2000 Hz
Number of Frequency Sequences F	12
Number of Amplitude Sequences M	25
Sampling Points per Sequence N	8000

zero-mean correction, frequency and magnitude condition encoding, per-condition temporal training/validation splitting, input/target sequence normalization, and metric recomputation based on the same split. Main-text Fig. 2(b) illustrates the complete data construction chain, from vibration table frequency–magnitude sweeps and measured-response-to-ideal-reference pairing to steady-state clipping, temporal splitting, and normalization.

Table S2 Dataset definition and preprocessing protocol.

Item	**Value**
Target curve source	ideal second-order response fitted from the low-magnitude calibration sweep
Operating points	300
Samples per record	8000
Train/validation split	per-condition 50%/50% temporal training/validation split
Temporal halves	300 / 300
Split unit	temporal halves within each frequency–magnitude operating condition
Frequency range	10–128 Hz
Magnitude range	0.24–6.0 $\mathrm{m/s^2}$
Scenario mapping	frequency–magnitude matrix with paired measured and ideal waveforms
Preprocessing steps	steady-state clipping, per-condition temporal splitting, and input/target normalization to [-1, 1]

Tab. S2 summarizes the dataset size, target sequence construction, and preprocessing pipeline shared by all experiments. All records are normalized to the range $-1 \sim 1$. For each frequency–magnitude condition, 8000 steady-state sampling points are first extracted and then divided into training and validation temporal halves according to per-condition 50%/50% temporal training/validation split. This split is used to evaluate compensation performance within the current experimental matrix.

4 Supplementary Note 4: KAN and Wiener-Layer Implementation Details

The proposed on-board real-time AI compensation system is illustrated in main-text Fig. 3(a). Although the parallel Wiener model effectively characterizes the amplitude-frequency coupling of the system, directly deriving its full-scale, wideband inverse model using conventional analytical approaches presents severe mathematical difficulties. Assume the uncompensated output of the MET system is $y(t)$, as expressed in Eq. (S12), which is composed of multiple branches of local dynamics h_i and static nonlinearities f_i. To design a feedforward compensator that restores the ideal linear response $\overset{\circ}{y}(t) = (\overset{\circ}{h} * x)(t)$, a conventional analytical strategy is to construct a series structure that combines linear filtering with static nonlinear inversion. Suppose the linear part of the compensator adopts the impulse response corresponding to the relative transfer function $\overset{\rho}{h}_j = \overset{\circ}{h} * h_j^{-1}$. When the uncompensated signal $y(t)$ passes through this linear part, the intermediate output $z_j(t)$ of the j-th branch can be expanded as:

$$z_j(t) = (\overset{\rho}{h}_j * y)(t) = \overset{\rho}{h}_j * f_j\big(h_j * x\big) + \sum_{i \neq j} \overset{\rho}{h}_j * f_i\big(h_i * x\big) \tag{S16}$$

In Eq. (S16), conventional analytical inversion faces two severe mathematical challenges. The first is the **non-commutative residual**. Because the linear dynamic operator $\overset{\rho}{h}_j$ and the static nonlinear operator f_j do not commute, i.e., $\overset{\rho}{h}_j * f_j(\cdot) \neq f_j(\overset{\rho}{h}_j * \cdot)$, the main branch response cannot be simplified to the ideal $f_j(\overset{\circ}{h} * x)$. Expanding f_j to the third-order Taylor term $f_j(v) \approx k_1 v + k_3 v^3$, the main branch output contains $k_3\big[\overset{\rho}{h}_j * (h_j * x)^3\big]$. This deviates significantly from the desired purely static distribution $k_3(\overset{\circ}{h} * x)^3$, producing a frequency-dependent memory nonlinear error. The second is the **cross-branch crosstalk residual** on the right side of Eq. (S16), representing complex coupling among different local dynamics. If the compensator's subsequent stage relies solely on a single-dimensional static analytical inverse mapping $f_j^{-1}(\cdot)$, it faces the non-distributability of the nonlinear inverse operator over multi-path mixed signals, i.e., $f_j^{-1}(A + B) \neq f_j^{-1}(A) + f_j^{-1}(B)$. This causes severe nonlinear intermodulation between the main branch signal and the cross-branch crosstalk. Therefore, neither a Hammerstein compensator (nonlinearity first, then linearity) nor a Wiener compensator (linearity first, then nonlinearity) can fully cancel the dynamic memory residuals arising from the above operator properties (non-commutativity or non-distributability) under multi-channel parallel summation. It is precisely this mathematical blind spot that motivates Wiener-KAN to fully neuralize the compensation structure: a multi-kernel Wiener linear layer extracts frequency response features $\{z_j(t)\}$ at each operating point to retain local physical priors, while the independent one-dimensional static inversion is discarded in favor of a multi-layer, multi-input KAN network. The multi-layer KAN receives multi-dimensional features and relies on the high-dimensional nonlinear expressiveness of deep splines to fit and eliminate the complex dynamic residuals and cross-branch crosstalk through end-to-end learning, thereby achieving high-quality global mapping reconstruction over a wide bandwidth and full dynamic range. Compared to traditional force feedback approaches, this pure feedforward structure requires no

additional mechanical feedback unit, thus avoiding the closed-loop oscillation problem caused by nonlinearity in force feedback systems and supporting operation over a wider magnitude range and bandwidth. In implementation, the compensator's front-end uses linear recurrent weights initialized with relative frequency responses. During training, multi-frequency, multi-magnitude vibration signals are fed simultaneously to the MET and the ideal linear system. The loss function is computed by comparing the measured discrepancy, and backpropagation updates the KAN network and other trainable back-end parameters.

4.1 KAN-Based Nonlinear Mapping with Sign and Symmetry Constraints

To construct a compensator that integrates structured physical constraints with nonlinear modeling capability, selecting an appropriate neural network architecture to characterize the sensor input/output relationship is critical. Although global polynomials can describe certain static nonlinearities, they struggle to stably capture nonlinear migration across different magnitude intervals over an extremely wide input dynamic range. To address this, this paper introduces trainable b-spline activation functions via KAN, leveraging their piecewise local approximation property to directly learn the sensor calibration curve. Traditional MLPs rely solely on linear weights and fixed activation functions (e.g., ReLU, tanh) for implicit approximation, which not only increases training difficulty and overfitting risk but also makes it hard to directly incorporate sensor-specific mathematical constraints. Unlike MLPs that use fixed-shape activation functions for indirect fitting, the sign- and symmetry-constrained KAN nonlinear mapping adopted here optimizes the activation function shape through learnable b-spline coefficients, thereby enhancing the single-neuron modeling capacity for sensor nonlinearity. On this basis, we further introduce constraints such as odd symmetry and positivity, as formulated in Eq. (S17).

$$\phi(x) = \text{sign}(x) \sum_{k=1}^{K} c_k B_k(|x|), \quad B_i(x) \geq 0 \, \forall x \in \mathbb{R}, \quad c_k \geq 0 \tag{S17}$$

where $B_k(x)$ denotes the b-spline basis function and c_k represents the trainable weights. After training, these weights adaptively approximate the sensor calibration curve under the above constraints:

1. **Odd symmetry** $\phi(-x) = -\phi(x)$: implemented by computing the activation value on the absolute input and then restoring the sign;
2. **Positivity** $\phi(x) > 0$ $(x > 0)$: ensured by non-negative basis functions $B_k(\cdot) \geq 0$ and constrained weights $c_k \geq 0$ to maintain consistent sign direction between input and output.

For MET sensors, the electrochemical input-output relationship typically exhibits specific symmetry and positivity characteristics. Directly incorporating corresponding mathematical constraints into the model helps maintain physically plausible predictions in data-sparse regions. Conventional MLPs can only indirectly impose such soft constraints through penalty terms in the loss function, and the constraint strength is difficult to balance during training. In contrast, KAN can achieve structural constraints by directly constraining the b-spline coefficient space.

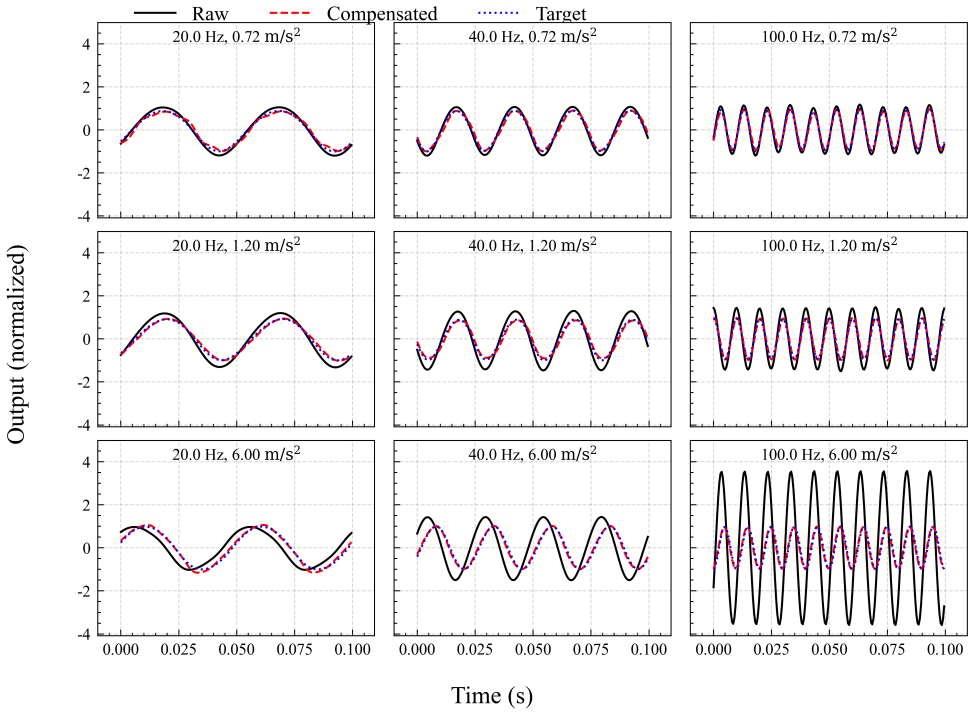

Fig. S2 Representative time-domain compensation outputs under selected frequency-magnitude conditions. Rows correspond to magnitudes 0.72, 1.20, and 6.00 m/s^2; columns correspond to 20.0, 40.0, and 100.0 Hz. Black solid, red dashed, and blue dotted curves denote raw, compensated, and target outputs, respectively.

From the perspective of structural modeling of sensor characteristics, when describing complex nonlinear relationships, MLPs often require increasing network depth or neuron count. In contrast, the b-spline activation function used in KAN possesses a piecewise polynomial structure, which bears mathematical similarity to the piecewise response characteristics exhibited by MET sensors across different magnitude intervals. From a function approximation standpoint, piecewise polynomials may offer better parameter efficiency when handling functions with interval-dependent variations. Fig. S2 shows representative time-domain compensation outputs under selected frequency and magnitude conditions. The compensated output is closer to the target linear output in both phase and amplitude, indicating that the reduction of frequency-response drift is also reflected in time-domain waveform consistency. To address MET nonlinear compensation problems, this paper adopts a multi-layer KAN structure as expressed in Eq. (S18). Given an input vector $\mathbf{x}$, its output is written as $\mathrm{KAN}_{\mathrm{con}}(\mathbf{x})$.

$$\mathrm{KAN}_{\mathrm{con}}(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \cdots \circ \mathbf{\Phi}_0)\,\mathbf{x} \tag{S18}$$

Here, $\mathbf{\Phi}_l$ denotes the function matrix of the l-th layer, whose element $\phi_{l,j,i}$ represents the constrained activation function connecting the i-th node in layer l to the j-th node in layer $l+1$, as defined in Eq. (S17), and L is the total number of KAN layers. However, KAN alone cannot

directly capture frequency-domain information. Therefore, this paper cascades a Wiener linear layer before the KAN to explicitly provide the network with frequency response information.

4.2 Magnitude-Conditioned Local Transfer Functions for Wiener Layer Initialization

This paper implements the linear dynamic part of the Wiener structure as a Wiener linear layer initialized using magnitude-conditioned local transfer functions. Under local operating conditions, the MET can be approximated as a linear system. Therefore, for each fixed magnitude, this paper identifies its corresponding local linear frequency response from measurement data. Furthermore, based on the relative transfer function method for linear systems, the corresponding local relative transfer function of the compensator at the same operating point can be calculated. By using these magnitude-varying local transfer functions as the initialization basis for the Wiener linear layer, this paper introduces signal processing domain knowledge into the model to enhance its frequency-dependent compensation capability. Specifically, this Wiener linear layer converts these local linear response parameters into the initial weights of a linear RNN and organizes multiple operating-point linear kernels into multi-output responses of the same layer. The calculation process for mapping these local linear responses to a discrete difference equation is given below. First, Eq. (S7) is used to fit the MET frequency response at a specific magnitude, yielding the measured transfer function $\tilde{H}$. Compared to an ideal system, $\tilde{H}$ exhibits nonlinear drift characterized by the natural frequency $\tilde{\eta}$, sensitivity $\tilde{S}$ at $\tilde{\eta}$, and damping ratio $\tilde{\zeta}$. To construct the target response, this paper predefines the transfer function of an ideal linear system $\overset{\circ}{H}$, whose parameters include the natural frequency $\overset{\circ}{\eta}$, sensitivity $\overset{\circ}{S}$ at $\overset{\circ}{\eta}$, and damping ratio $\overset{\circ}{\zeta}$. According to linear system theory, frequency response compensation at a specific magnitude can be achieved by cascading a relative transfer function $\overset{\rho}{H}$ before $\tilde{H}$, i.e., $\overset{\circ}{H} = \tilde{H} \cdot \overset{\rho}{H}$. The expression for $\overset{\rho}{H}$ is shown in Eq. (S19).

$$\begin{aligned} y[n] =& b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \\ & - a_1 y[n-1] - a_2 y[n-2] \end{aligned} \tag{S19}$$

where $y[n]$ denotes the output at time step n, $x[n]$ denotes the input at time step n, b_0, b_1, b_2 are the filter zero coefficients, and a_1, a_2 are the pole coefficients. For the selected h local response slices, this paper directly constructs mutually independent discrete linear kernels from the extracted difference equation parameters. By applying these local linear filtering kernels in parallel to the same input signal $x[n]$, a single-input multiple-output (SIMO) multi-kernel Wiener linear feature extraction layer WienerLayer$(x[n])$ is formed. Finally, cascading the Wiener linear layer with the constrained KAN yields the Wiener-KAN model for frequency response nonlinear compensation of the MET system, as expressed in Eq. (S20).

$$\text{Wiener-KAN}(x[n]) = \text{KAN}_{\text{con}} \circ \text{WienerLayer}(x[n]) \tag{S20}$$

5 Supplementary Note 5: AFMAE Derivation

5.1 AFMAE Loss for Amplitude-Frequency Response

Traditional mean absolute error (MAE) and mean squared error (MSE) are widely used in regression tasks, but they primarily measure pointwise error accumulation and tend to be overly sensitive to high-frequency random noise. For nonlinear system modeling, researchers are more concerned with preserving the overall frequency response characteristics of the signal. Therefore, this paper designs the amplitude-frequency mean absolute error (AFMAE) to quantify the difference between the compensated output and the ideal output in the frequency domain. To derive the expression for AFMAE, this paper first examines the energy-amplitude relationship of a sinusoidal signal. For a sine wave $A\sin(\omega t)$ with period $T = \frac{2\pi}{\omega}$, its energy E can be expressed as Eq. (S21).

$$E = \int_0^T A^2 \sin^2(\omega t)\, dt = \frac{1}{2} A^2\, T \tag{S21}$$

In the discrete case, given a sampling frequency f_s and N sample points, the period is $T = \frac{N}{f_s}$. The signal energy can then be approximated by Eq. (S22).

$$E \approx \frac{1}{f_s} \sum_{n=0}^{N-1} \big(y[n]\big)^2 \tag{S22}$$

Thus, the output signal amplitude can be written as $A \approx \sqrt{\frac{2E}{T}}$. For a sinusoidal input with frequency f_i and magnitude m_j, its frequency response is defined as Eq. (S23).

$$\begin{aligned} \left|\hat{H}(f_i, m_j)\right| &= \frac{\hat{A}(f_i, m_j)}{m_j} = \sqrt{\frac{2}{T}} \frac{\sqrt{\hat{E}(f_i, m_j)}}{m_j} \\ &= \sqrt{\frac{2}{T f_s}} \frac{\sqrt{\sum_{n=0}^{N-1} \hat{y}_{i,j}[n]^2}}{m_j} \end{aligned} \tag{S23}$$

Here, $\hat{H}$ denotes the network-predicted frequency response, $\hat{A}$ and $\hat{E}$ are the amplitude and discrete energy estimate of the network output, and $\hat{y}_{i,j}[n]$ is the sample predicted by the network for input magnitude m_j at frequency f_i. To quantify the discrepancy between the predicted and true frequency responses, this paper computes the logarithmic difference between the true frequency response $H(f_i, m_j)$ and the network-predicted frequency response

$\hat{H}(f_i, m_j)$, yielding Eq. (S24).

$$\begin{aligned} &\log |H(f_i, m_j)| - \log \left|\hat{H}(f_i, m_j)\right| \\ =&\log \sqrt{\frac{\sum_{n=0}^{N-1} y_{i,j}[n]^2}{\sum_{n=0}^{N-1} \hat{y}_{i,j}[n]^2}} \\ =&\frac{1}{2}\left(\log \sum_{n=0}^{N-1} y_{i,j}[n]^2 - \log \sum_{n=0}^{N-1} \hat{y}_{i,j}[n]^2\right) \end{aligned} \tag{S24}$$

Here, H denotes the true frequency response. The logarithmic form provides a more symmetric measure of the output discrepancy on the logarithmic amplitude axis (dB scale), thus maintaining a relatively balanced penalty when the predicted value is either higher or lower than the true value, while also avoiding explicit computation of m_j. The constant factor from the amplitude-energy relation is absorbed into the loss weight β, so $\mathcal{L}_{\text{AFMAE}}$ uses the equivalent scaled form shown in Eq. (S25).

$$\mathcal{L}_{\text{AFMAE}} = \frac{1}{FM}\sum_{i=1}^{F}\sum_{j=1}^{M}\left|\log \sum_{n=0}^{N-1} y_{i,j}[n]^2 - \log \sum_{n=0}^{N-1} \hat{y}_{i,j}[n]^2\right| \tag{S25}$$

Here, F is the number of frequency samples, and M is the number of magnitude samples. This method characterizes the amplitude-frequency response efficiently with a single energy statistic. In practice, a weighted combination of MAE and AFMAE yields superior results. This paper introduces a weight hyperparameter β into the total loss function, as shown in Eq. (S26).

$$\mathcal{L}_{\text{total}} = (1-\beta)\,\mathcal{L}_{\text{MAE}} + \beta\,\mathcal{L}_{\text{AFMAE}} \tag{S26}$$

6 Supplementary Note 6: LUT Inference and Embedded Validation Details

6.1 Precomputed LUT Inference for Real-Time Embedded Deployment

In typical seismic monitoring applications, continuous high-frequency sampling demands that the single-step inference latency of the compensation algorithm must be strictly less than the sampling period. This strict real-time constraint forces us to abandon the dense matrix multiply-accumulate operations of neural networks. The online inference cost of KAN mainly arises from evaluating the learnable activation function on each edge. For the activation function $\phi_{l,j,i}(x)$ connecting input i to output j in layer l, the exact path requires computing the b-spline basis functions and the weighted sum of coefficients at each sampling point. To meet hard real-time requirements, this paper offline-samples this activation function into a precomputed lookup table after training. This converts continuous spline evaluation directly into extremely low-latency memory address mapping, table reading, and optional first-order linear interpolation during online inference.

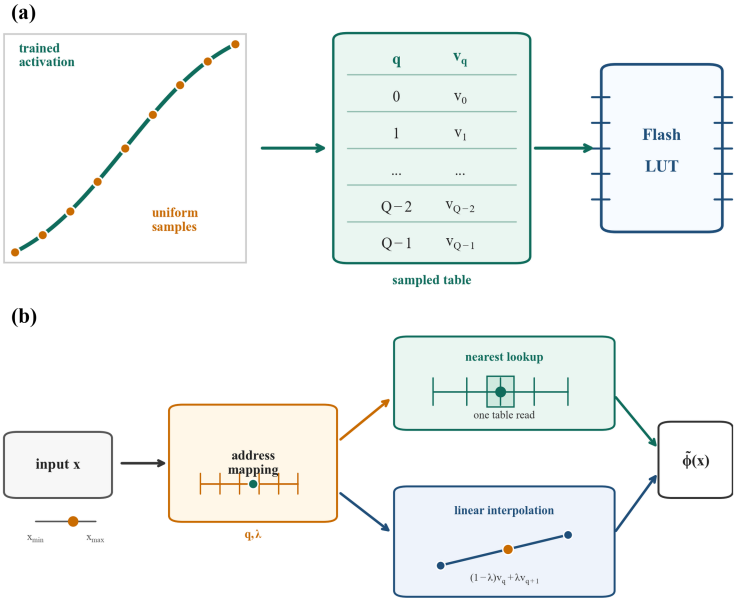

Fig. S3 LUT inference principles for embedded Wiener-KAN deployment. (a) Offline table construction samples the trained KAN activation into uniform entries and stores the resulting table in Flash. (b) Runtime MCU lookup maps the input to a table index and uses either nearest lookup or linear interpolation, illustrated by the selected table cell and the interpolation segment.

Let the LUT range be $[x_{\min}, x_{\max}]$, with Q sampling points and a step size $\Delta_{\mathrm{LUT}} = (x_{\max} - x_{\min})/(Q-1)$. The q-th entry is

$$v_q = \phi\left(x_{\min} + q\Delta_{\mathrm{LUT}}\right), \quad q = 0, 1, \ldots, Q-1. \tag{S27}$$

Nearest-neighbor LUT query maps input x to an integer table address

$$q(x) = \mathrm{clip}\left(\mathrm{round}\left(\frac{x - x_{\min}}{\Delta_{\mathrm{LUT}}}\right), 0, Q-1\right), \quad \tilde{\phi}(x) = v_{q(x)}. \tag{S28}$$

First-order linear interpolation computes the fractional grid coordinate $r = (x - x_{\min})/\Delta_{\mathrm{LUT}}$, sets $q = \lfloor r \rfloor$ and $\lambda = r - q$, then obtains

$$\tilde{\phi}(x) = (1-\lambda)v_q + \lambda v_{q+1}. \tag{S29}$$

Therefore, the nearest-neighbor path obtains the activation output via table address mapping and lookup; the linear interpolation path adds limited addition and multiplication between adjacent entries to improve continuity. Fig. S3 illustrates these two query paths. This deployment path differs fundamentally from the computational structure of an MLP. In an MLP layer $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$, weights participate in online computation for every sample point through matrix multiplication-addition, with the input and output dimensions jointly determining the number

of multiplications and additions. Edge weights in KAN are expressed by trainable functions $\phi_{l,j,i}(\cdot)$; after training, the function shapes are precomputed into table values v_q, and the online path mainly revolves around table address mapping, lookup, and optional interpolation. Consequently, the B-spline grid size and order primarily affect offline function expressiveness and table value generation, while the real-time feasibility of the derived implementation is directly evaluated by the target board throughput. This property gives KAN a deployment advantage on sensor edge devices lacking GPU/TPU matrix acceleration units. In the embedded implementation, the LUT tables, together with normalization constants, linear recurrent kernels, and output scaling parameters, are written into the C-side read-only data region. During the online phase, the Wiener linear layer and KAN query path are executed point-by-point within a fixed sampling window; therefore, the primary evaluation metric uses points/s obtained from the target board serial port benchmark loop. This procedure directly covers compiler optimization, memory access, and function call overhead, making it suitable for evaluating the real-time execution capability of the final exported kernel.

6.2 On-board inference test flow

On-board inference testing verifies the complete pipeline of Wiener-KAN from the trained model to the embedded C implementation. After training, the weights, normalization constants, and LUT tables are exported as C code. The same C implementation is first run in QEMU and compared with the TensorFlow reference output, then flashed onto the STM32F405 target board. Throughput is collected via a serial benchmark loop, and Flash and RAM usage are read from the Keil build results.

Fig. S4(a)–(b) illustrate the embedded verification path. The validation window is first normalized and encapsulated as a C array. The exported C implementation is then executed on QEMU and the target board, with its predicted output aligned to the TensorFlow reference to compute QEMU-MAE and KEIL-MAE. The hardware path obtains throughput records via a repeated UART benchmark loop, with Flash and RAM usage reported from Keil build results. Fig. S4(c) presents the RAM–throughput–KEIL-MAE relationship for the lateral export model and three Wiener-KAN implementation variants, where both LUT nearest and LUT interp use 769 quantization points. Fig. S4(d) further fixes the same training weights and shows the QEMU-MAE–Flash relationship over 65–769 quantization points, demonstrating that the interpolation path maintains lower QEMU-MAE across the entire sweep.

Table S3 On-board implementation variants of the same Wiener-KAN weights. KEIL speed is measured in processed points per second, so higher values indica te higher throughput.

Implementation	**KEIL speed** (points/s)	**KEIL-MAE**
LUT nearest	6986.7	7.114e-03
LUT + linear interp	3844.8	1.902e-04
No LUT exact	45.0	6.517e-07

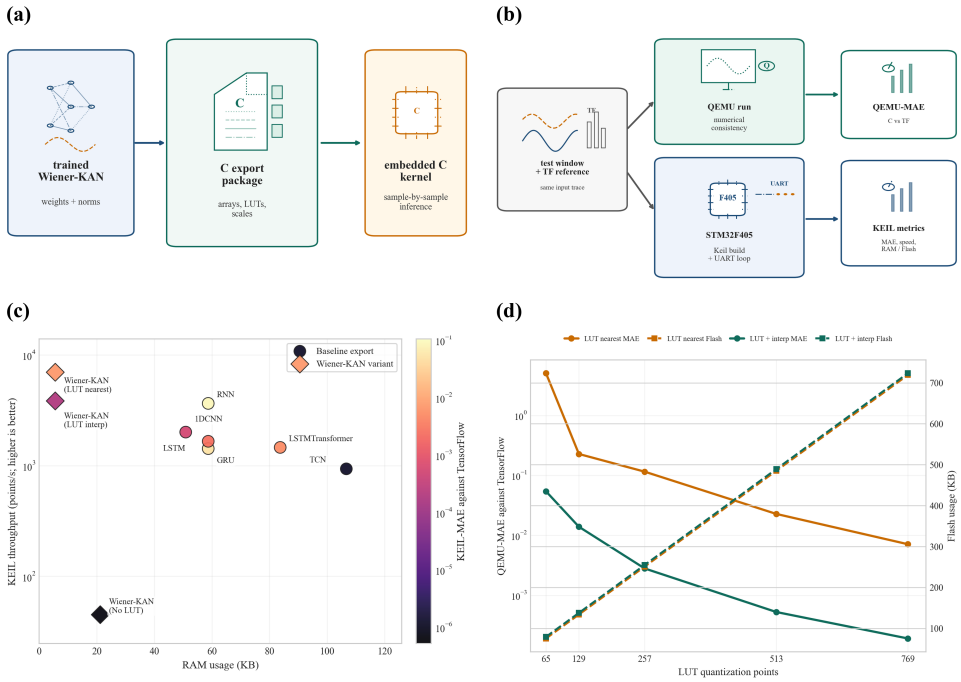

Fig. S4 Embedded inference validation workflow and performance. (a) The trained Wiener-KAN project is exported once as a C implementation with weights, normalization constants and LUT tables. (b) The same C implementation is validated in QEMU for numerical consistency and on STM32F405 for numerical, throughput and memory metrics. (c) Embedded performance of exported baseline models and Wiener-KAN implementation variants; marker position reports RAM usage and measured KEIL throughput, while marker color encodes KEIL-MAE. The LUT nearest and LUT interp variants in this panel both use 769-point tables. (d) LUT quantization-point sweep of the same Wiener-KAN weights; the x axis gives the LUT point count, the left y axis reports QEMU-MAE, and the right y axis reports Flash usage for nearest lookup and linear interpolation.

On-board benchmarking on the STM32F405 shows that the primary Wiener-KAN implementation achieves 6986.7 points/s with 5.6 KB RAM. Tab. S3 reveals a speed–error trade-off under the same training weights across LUT nearest-neighbor, LUT linear interpolation, and exact spline paths: nearest-neighbor lookup yields the highest throughput, linear interpolation significantly reduces KEIL-MAE, and the exact spline path provides the lowest C-level approximation error. Fig. S4(c) presents the RAM–throughput–KEIL-MAE relationship for the lateral export model and three Wiener-KAN implementation variants. Fig. S4(d) shows the deployment accuracy–storage trade-off under the same training weights over 65–769 LUT quantization points. As the number of quantization points increases, Flash grows nearly linearly while QEMU-MAE decreases monotonically; linear interpolation maintains a lower error curve than nearest-neighbor lookup across the full sweep.

7 Supplementary Note 7: Extended Ablations and Hyperparameter Sensitivity

7.1 Structural ablation of the Wiener and KAN blocks

The Struct group in main-text Tab. 2 isolates two core structural modules of Wiener-KAN. CNN-KAN replaces the Wiener linear block with a convolutional front-end while retaining the same KAN nonlinear mapping, comparing explicit local frequency response slices against a data-driven temporal front-end. Wiener-MLP retains the same Wiener front-end but replaces the physics-constrained KAN mapping with an MLP mapping, comparing spline-type single-neuron nonlinearity against conventional fully-connected mapping. The Struct results indicate that the two substitutions correspond to the linear dynamic skeleton and the nonlinear compensation body of Wiener-KAN, respectively. CNN-KAN approaches the baseline configuration in frequency drift and sensitivity drift, yet exhibits a marked increase in linearity error, suggesting that the Wiener front-end initialized with local frequency response is more conducive to preserving the stable shape of the in-band compensation curve. Wiener-MLP lags behind Wiener-KAN on all three physical metrics, indicating that after sharing the Wiener front-end, the physics-constrained KAN mapping still undertakes the primary expressive capability for amplitude-dependent nonlinear compensation.

7.2 Physical sign and symmetry constraints

The constraint ablation study examines odd symmetry and positivity separately. Odd symmetry corresponds to the MET sensor's direction-consistent response to positive and negative vibration inputs, while positivity corresponds to monotonic direction preservation within the positive input interval. All constraint variants share the same Wiener front-end, spline order, grid number, and training data, so the constraint group in main-text Tab. 2 directly reflects the impact of structural constraints themselves on frequency drift, sensitivity drift, and linearity error. The *Odd / No positive basis* variant in main-text Tab. 2 retains odd symmetry but removes the direction-preserving constraint on the positive input interval, making its optimization process more prone to deviate from a stable compensation mapping. This variant exhibits markedly increased errors in frequency drift, sensitivity drift, and linearity error simultaneously, indicating that the positivity constraint provides Wiener-KAN with a stable functional shape prior. The Baseline retains both odd symmetry and positivity, making the nonlinear compensation mapping more consistent with the device response characteristics, thus maintaining more stable convergence across all three physical metrics.

7.3 Wiener front-end initialization and trainability

The Wiener front group in main-text Tab. 2 evaluates whether the Wiener front-end benefits from local frequency response initialization and whether the injected RNN-style IIR layer is suitable for training. The frozen system prior setting yields the best frequency drift, sensitivity drift, and linearity error. The randomly initialized and frozen front-end still allows the back-end model to train, but all three physical metrics increase. The trainable IIR front-end causes a marked increase in frequency drift, indicating that the local frequency response prior is most effective as a stable linear dynamic skeleton. Accordingly, the reported Wiener-KAN uses the frozen system-prior front-end.

7.4 Hyperparameter sensitivity

The hyperparameters of Wiener-KAN include the number of relative frequency response slices h in the Wiener linear layer, the number of layers l and units per layer u in the constrained KAN, and the order and grid number of the b-spline activation function. This paper conducts a single-factor sensitivity analysis around the canonical $h8u6l6g8s2$ configuration, with training epochs, learning rate, loss function, data split, and evaluation pipeline held constant. The baseline configuration achieves Freq Drift $= 2.34$ Hz, Sens Drift $= 9.09$ V $\cdot$ s/m, and Linearity $= 0.511\%$.

Supplementary Fig. S5(a)–(e) presents single-factor sensitivity results around the selected configuration; the unnumbered legend panel in the middle of the second row provides the colors, markers, and baseline absolute values used for normalization. Increasing the KAN depth to $l = 7$ further improves all three physical metrics; this paper adopts the $h = 8, u = 6, l = 6$ model as the representative configuration and regards $l = 7$ as a higher-accuracy candidate. The spline grid g and spline order s mainly affect the offline expressiveness of the spline function, while the actual runtime performance during LUT deployment is jointly determined by the exported implementation and on-board throughput.

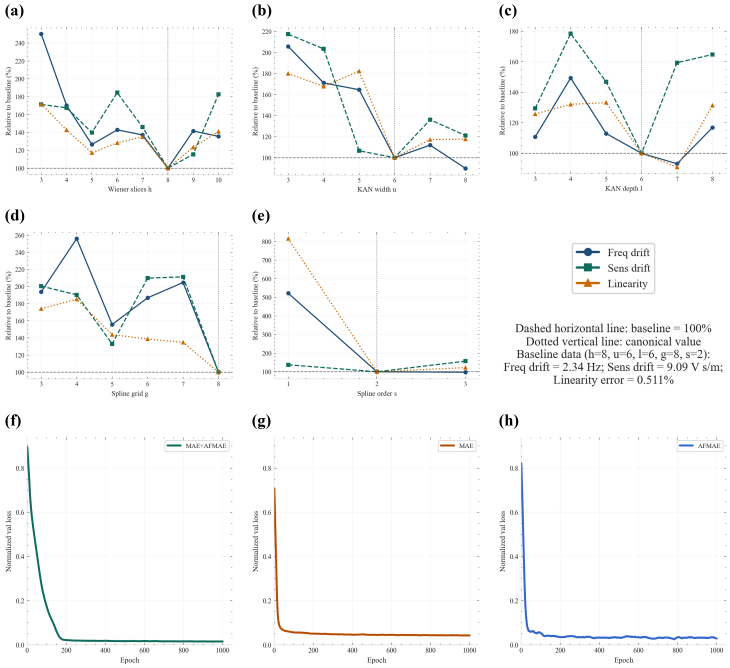

Fig. S5 One-factor hyperparameter sensitivity and loss-objective convergence around the canonical Wiener-KAN configuration. (a) Wiener slices h. (b) KAN width u. (c) KAN depth l. (d) Spline grid g. (e) Spline order s. The unnumbered center panel in the second row summarizes the legend and the absolute baseline values used for normalization. (f) Joint MAE+AFMAE objective. (g) MAE-only objective. (h) AFMAE-only objective. Each sensitivity panel reports the relative changes in frequency drift, sensitivity drift and linearity error with respect to the baseline; the loss panels report normalized validation-loss trajectories.

7.5 Loss-function ablation

The training objective is jointly defined by a time-domain error term and an amplitude-frequency error term. MAE directly constrains pointwise waveform error, while AFMAE constrains amplitude statistics under the same frequency and magnitude conditions. Their combination simultaneously enforces waveform consistency and frequency response consistency. To isolate the contribution of each loss function, this paper compares MAE, AFMAE, and MAE+AFMAE under identical data splits, model structures, initialization schemes, and training epochs. The Loss group in main-text Tab. 2 and Supplementary Fig. S5(f)–(h) jointly evaluate convergence epochs and final compensation quality under a 1k-epoch fixed learning rate setting. MAE+AFMAE yields the most balanced results across frequency drift, sensitivity drift, and linearity error, indicating that pointwise time-domain error and amplitude-frequency response error must jointly participate in the training objective. With MAE alone, the pointwise error objective imposes weak constraints on amplitude-related frequency drift. With AFMAE alone, the amplitude-frequency statistical constraint improves frequency drift, but sensitivity drift and linearity error remain higher than those of the joint objective. The joint loss retains both constraints, keeping the convergence curve and final physical metrics consistent.