LEARNING MODE-RESOLVED RESPONSE COMPUTATIONS ACROSS SYSTEMS AND FORCING

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting complete responses to external forcing supports the design and assessment of physical systems under changing loads. Generalizing across systems requires adapting to both new excitation and different dynamic scales. We present a system-conditioned framework for full-event prediction. Its Mode-Resolved Response Operator (MRRO) uses supplied modal frequencies to condition kernel coefficients and place finite bases at each mode's scale. Propagated latent features return to node coordinates for nonlinear coupling on the query's physical-time grid, with the output anchored to a computed reference. Weights are shared within each domain and trained separately across domains. Matched experiments control physical information, phase and capacity, while assessing dependence on optimization budget. On new systems and forcing, MRRO reduces acceleration error by 45.28% and 27.83% relative to F-FNO adaptations in hysteretic and torsional networks. A joint building model covers earthquakes and six applied-force programs; five paired training runs yield 14.15% lower error than common-scale kernels on 582 responses from 128 held-out families. A proposal-conditioned reaction–diffusion model and calibrated folding-wing experiments examine extensions of the kernel construction. Frozen mechanical predictors also compare candidate actions without goal-specific training. These findings show that physical-scale allocation matters for transferable response learning.

1 INTRODUCTION

Response prediction supports the design and assessment of physical systems under changing loads. Repeated nonlinear simulations across system variants and forcing scenarios can be costly, motivating models that share learned computation across both. Changes in mass, stiffness and spatial coupling shift the dynamic scales that such a model must resolve. Supplying these scales as features leaves a further design choice: where should a finite kernel basis be placed when different modes respond on different scales?

Neural operators and graph simulators already learn across equations and geometries (Li et al., 2021; Pfaff et al., 2021; Alkin et al., 2024). We ask whether finite temporal kernels should use one system scale or separate modal scales. The common-scale control receives every modal frequency and adapts to the system mean. MRRO changes the granularity of basis placement without additional inputs or basis functions.

Both variants condition kernel coefficients on identical modal information; they differ in the coordinates at which the finite basis is evaluated. Modal features are then synthesized into node coordinates for shared nonlinear coupling, without assuming independent nonlinear oscillators.

Relation to existing operators. F-FNO factorizes spectral layers (Tran et al., 2023); NORM and GLNO use geometric spectra (Chen et al., 2024; Tang et al., 2025); LNO learns poles and residues (Cao et al., 2024). LinOSS learns latent oscillators (Rusch and Rus, 2025), and SeisGPT conditions spectral rotations and corrections on structural eigenvalues (Meng et al., 2026). Our focus is how supplied modal scales place a finite temporal basis inside coupled nonlinear computation. This representation question is distinct from constructing discretization-consistent function-space operators (Berner et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

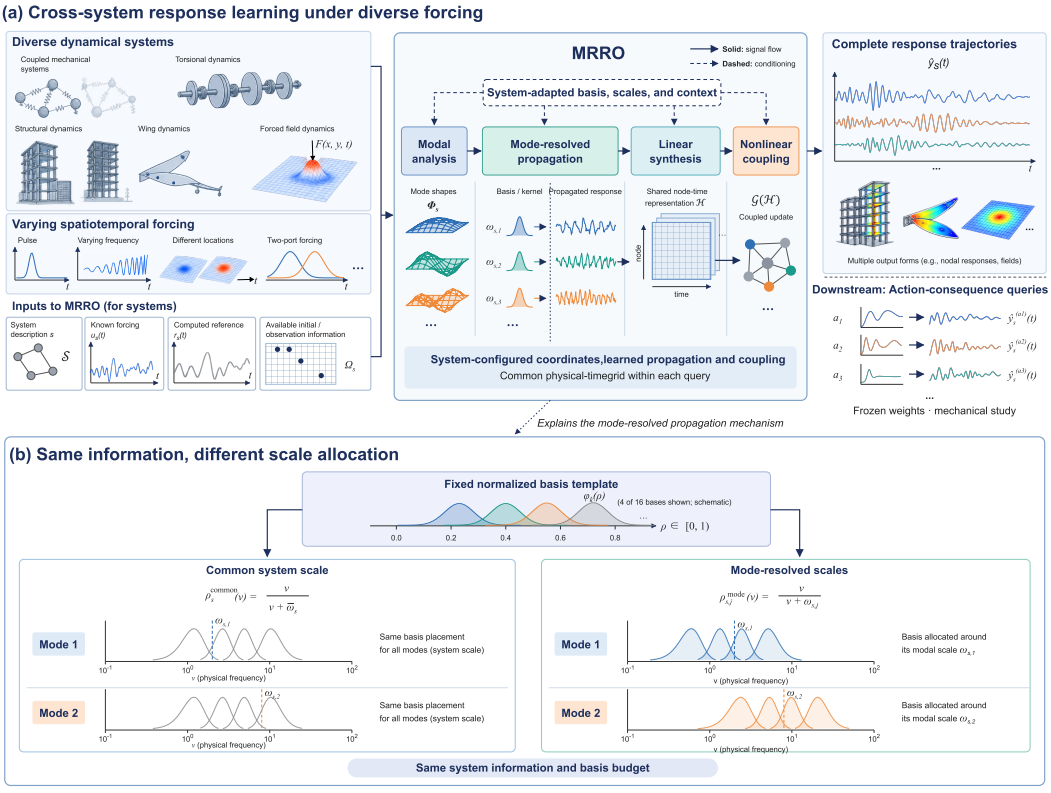

Figure 1: **Cross-system response learning through mode-resolved scale allocation.** (a) System descriptors configure propagation and nonlinear coupling on each query's time grid. Weights are shared within domains; fields use a response-space backbone. Scenes and outputs are schematic. (b) Common and per-mode coordinates place the same 16 bases at different physical frequencies (four shown). Frozen mechanical predictors support action queries without goal-specific training.

We introduce this operator and test scale granularity under matched information, phase and capacity. Five-domain evaluations and frozen action queries examine its predictive value and scope.

2 System-conditioned response prediction

A query predicts $\widehat{y}_{0:N_t-1} = F_\theta(s, u_{0:N_t-1}, b_{\text{obs}}, m_b, o)$ from system s, prescribed forcing u, initialization information b_{obs} with mask m_b, and observation protocol o. It has N_t samples, interval Δt and duration T_{obs}. The same inputs define reference r_s. Mechanical, torsional and field tasks supply complete initial conditions without measured response prefixes or query-specific nonlinear fitting. E001 uses independent ground vibration test (GVT) calibration and a 0.5-s measured prefix excluded from scoring. This is offline event prediction.

For buildings, coarse floor M_s, K_s and component descriptors define the learning interface and computed reference. Targets come from a nonlinear 3D finite-element model driven jointly by base acceleration and applied forces, rather than by summing separate responses. The interface retains supplied static displacement, distinguishes master-node and archived floor-aggregate observations, and masks unknown initial values (Appendix B). Coarse matrices do not encode the full constitutive law; outputs are floor responses. Family–program coverage is unequal: of 4,312 planned new-load solves, 1,943 completed, 2,252 failed and 117 remained incomplete. Evaluation concerns the completed population.

Training data determine normalization; validation data select models. For each event, relative error is $e_i = \|m_i \odot (\widehat{y}_i - y_i)\|_F / \max(\|m_i \odot y_i\|_F, 10^{-12})$, where m_i selects valid observations. Structural tasks score acceleration; fields jointly score both concentration deviations. Networks bal-

Table 1: Data scope of the main comparisons. Entries are events / system families; wing entries give records and configurations. The validation set may include fitted systems with new forcing.

Study	Training	Validation	Test
Mechanical	764/113	224/24	320/16 (128 primary)
Torsional	320/16	72/8	256/16
Joint buildings	1,963/320	313/64	582/128
Reaction–diffusion	96/32	32/8	64/16
Folding wings	110 (A/C)	30 (A/C)	181 (B/BWI/BWO)

Mechanical/torsional evaluations use eight new forcing templates. Buildings jointly learn all programs; earthquake source waveforms are not fully separated across splits. Wings B/BWI/BWO form one related family. Buildings, fields and wings retain prior evaluation exposure; their claims concern held-out systems under this development history.

ance systems and forcing templates. Buildings weight seven programs equally, then families and records within each program (Appendix B). Wing root-mean-square error (RMSE) uses training-set scales and nested record, condition and configuration means. Each training run is scored separately; predictions are not ensembled.

3 Mode-resolved response computation

3.1 Spatial coordinates and persistent nonlinear computation

Using diagonal mass M in the normalized units of Appendix A, form $M^{-1/2}KM^{-1/2}\Delta t^2 = Q\Lambda Q^\top$ and $\omega_j = \sqrt{\max(\Lambda_j, 0)}/\Delta t$. The complete orthogonal basis is retained. Node descriptors, graph structure and supplied material properties give context c_i. Mode context retains actual frequency information:

$$m_j = E_m\left[\sum_i Q_{ij}^2 c_i, \log(1+\omega_j)\right], \qquad \rho_j(\nu) = \frac{\nu}{\nu + \max(\omega_j, 10^{-3})}. \tag{1}$$

Frequencies are expressed in rad/s. Let S contain channel scales estimated from training data. The input x concatenates normalized reference responses, encoded forcing and reference differences. The latent computation is

$$\begin{aligned} H_0 &= \mathrm{GELU}(E_\theta x + c_s), \\ H_{\ell+1} &= H_\ell + \tfrac{1}{4} N_\ell([\mathcal{T}_{s,\ell} H_\ell + \mathcal{L}_\ell H_\ell + \mathcal{G}_{s,\ell} H_\ell, c_s]), \\ \widehat{y} &= \mathcal{O}_s(r + S D_\theta(H_L, x)). \end{aligned} \tag{2}$$

Modal propagation returns to node coordinates before addition to the local $\mathcal{L}$ and graph/material $\mathcal{G}$ paths. N_ℓ couples hidden trajectories; $\mathcal{O}_s$ restores known offsets and observation conventions.

3.2 Finite temporal kernels organized by dynamic scale

For $H(t) \in \mathbb{R}^{n_s \times 64}$, channel mixing and modal analysis give $Z(t) = Q^\top M^{1/2} H(t) W_{\mathrm{in}}^\top$. A shared affine context map produces complex coefficients for 16 normalized Gaussian bases, giving $E_{jc}(\nu) = \sum_{k=0}^{15} \alpha_{jck} \phi_k(\rho_j(\nu))$. Appendix A defines their width and normalization. With learned gain a_{jc} and phase b_{jc}, the filter is

$$K_{jc}(\nu) = e^{0.35 \tanh a_{jc}} e^{i\pi \tanh b_{jc} \rho_j(\nu)} [1 + 0.5 \tanh \Re E_{jc}(\nu) + 0.5i \tanh \Im E_{jc}(\nu)]. \tag{3}$$

A real inverse fast Fourier transform (FFT) and truncation to N_t samples define the executed kernel; Appendix A.3 specifies its real endpoints and numerical correction. Transforming the padded kernel to at least $2N_t - 1$ samples implements linear convolution without event-end wrapping. The global output is $M^{-1/2} Q \widetilde{Z}(t) W_{\mathrm{out}}^\top$: Q acts on nodes/modes, whereas $W_{\mathrm{in/out}}$ act on channels. Thus the executed response is that of the truncated kernel, not simply Eq. (3). The centered local five-tap convolution also means the complete predictor is not strictly causal.

For a unit-mass linear mode with proportional damping,

$$\omega^2 H_\omega(\nu) = [1 - (\nu/\omega)^2 + 2i\zeta(\nu/\omega)]^{-1}. \tag{4}$$

Relative frequency aligns this normalized family. Above the numerical floor, an interior basis center $\mu \in (0, 1)$ maps to nominal physical frequency $\omega_j \mu/(1-\mu)$. These are basis coordinates, not identified filter resonances. At $\nu = \omega_j$, the local coordinate-density ratio for a positive common scale σ is

$$\left.\frac{d\rho_j/d\nu}{d\rho_\sigma/d\nu}\right|_{\nu=\omega_j} = \frac{(\omega_j+\sigma)^2}{4\omega_j\sigma} \geq 1, \qquad \rho_\sigma = \frac{\nu}{\nu+\sigma}. \tag{5}$$

Equality requires $\sigma = \omega_j$: one common scale cannot align distinct modes simultaneously. This motivates finite-basis placement without changing FFT resolution or implying nonlinear approximation or stability guarantees.

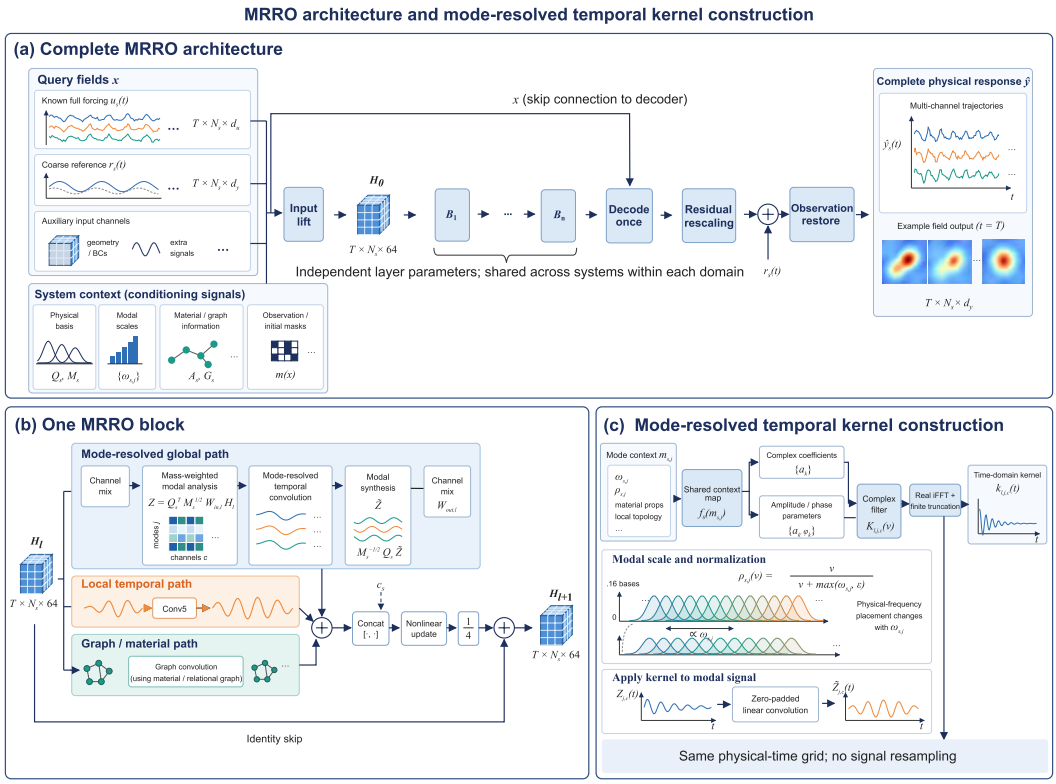

Figure 2: **MRRO architecture and temporal-kernel construction.** (a) Six independent 64-channel blocks precede one decoder of H_L and x, rescaling, reference addition and observation restoration. (b) The global-path output includes synthesis and channel mixing; local and graph outputs are added before nonlinear updating. (c) Each layer shares its kernel map across modes and systems. FFT, truncation and convolution preserve the time grid. Fields use the backbone in Section 3.3.

3.3 Fixed architecture and domain interfaces

Structural tasks use six blocks (Figure 2). Pointwise updates N_ℓ use $128 \to 128 \to 64$ layers and the decoder $74 \to 64 \to 2$, with GELU between affine layers. Coordinate/phase controls train separately.

Mechanical/torsional models use 1,213,378 parameters and 61,120 complete-event updates from random initialization. Buildings add a $10 \to 64 \to 64$ observation-context map and retain actual floor counts, duration and FFT lengths. Each domain uses its declared loss, scaling and training protocol.

Algorithm 1 MRRO inference: latent-backbone realization

Require: System s, forcing $u_{0:N_t-1}$, Δt, initialization (b_{obs}, m_b), observation o
1: Form reference/offsets, normalized input x, Q, ω, c_s, m_s.
2: $H \leftarrow \mathrm{GELU}(E_\theta x + c_s)$; $N_{\mathrm{FFT}} \leftarrow 2^{\lceil \log_2(2N_t-1) \rceil}$.
3: **for** $\ell = 1, \ldots, 6$ (independent parameters) **do**
4: Analyze H in mass-weighted modal coordinates; pad its time FFT to N_{FFT}.
5: Construct Eq. (3) at $\nu_k = 2\pi k/(N_{\mathrm{FFT}}\Delta t)$ using Eq. (1).
6: Make real-FFT endpoints real; inverse-transform and retain the first N_t kernel samples.
7: Transform the padded kernel to N_{FFT}, convolve, crop to N_t, and synthesize to nodes.
8: Add local and graph/material paths; update H by Eq. (2).
9: **end for**
10: Decode once; restore reference, units, observation offsets and only supplied initial entries.
11: **return** Complete response; target masks affect scoring, never supply unknown initial values.

Response-space realization for fields. Diffusion rates replace oscillatory frequencies. A seven-layer, 64-channel temporal/graph network predicts a proposal p_0 from the reference, forcing and context. A nine-layer, 96-channel correction network G receives the proposal, reference, proposal difference and forcing, giving

$$\widehat{y}_i = m\{p_{0,i} + (S/w_i)[G(x,c)_i - G(0,c)_i]\}.$$

Outputs are concentration deviations: $m = \mathbf{1}_{t>0}$ preserves their zero initial value, and w_i is the node weight. Three-tap causal convolutions use doubling dilations alongside graph updates; the ninth correction layer replaces its temporal output z by $z + U_s[\Delta k_s * (U_s^\top z)]$, where U_s is the diffusion eigenbasis and Δk_s is the residual kernel. This 16-basis path adds 3,264 parameters (0.703%). The two evaluations of G share weights. This is one proposal-conditioned predictor, not iterative feedback or an ensemble of two independent models. The shared contribution is the scale-organized kernel; the complete field backbone differs from the latent architecture.

4 Prediction across five physical datasets

Table 2 compares five tasks using published spectral, attention and state-space computations, adapted to the same descriptors, forcing, reference and initialization. Validation selects between two learning rates; weights are domain-specific. F-FNO, LNO, GNOT, UPT, Transolver++ and Mamba-2 (Tran et al., 2023; Cao et al., 2024; Hao et al., 2023; Alkin et al., 2024; Luo et al., 2025; Dao and Gu, 2024) provide complementary operators; C-PhysFNO and IRNO (Kim et al., 2026; Liu et al., 2026) are domain-focused comparators. The reported ranking concerns these task adaptations.

4.1 Nonlinear mechanical and torsional networks

The network evaluations contain 320 mechanical and 256 torsional events. Each study uses 16 new systems and eight new forcing templates. The 128-event primary mechanical subset jointly changes system, forcing and layout; torsion combines two amplitudes. All adaptations use these populations and validation-based selection.

MRRO reduces error by 45.28% on hysteretic networks and 27.83% on torsional networks relative to F-FNO, the best evaluated external comparator in each study. Paired 95% percentile system/forcing bootstrap intervals for F-FNO minus MRRO are $[0.17276, 0.21179]$ and $[0.04875, 0.11703]$, conditional on the trained models. Double-amplitude torsion remains difficult: MRRO errors are 0.06718/0.35347 at unit/double amplitude. Mechanical F-FNO, LNO and Transolver++ retain substantial training error; the information-matched experiments, rather than these external gaps alone, support the scale-allocation mechanism.

4.2 Complex buildings under multiple forcing programs

A single MRRO predictor is trained jointly on earthquakes and six applied-force programs using 1,963 training responses across 320 families. MRRO, common scale and C-PhysFNO select their learning rates by two-run mean validation error after 65,450 complete-event updates. Evaluation uses 582 responses from 128 families absent from fitting. C-PhysFNO retains eight

Table 2: **External comparisons across all five datasets.** Lower is better. Entries are two-run mean $\pm$ sample SD; s denotes one prespecified run (seed 41), without SD. External methods are task adaptations (Appendix D); daggers mark limited training fit. Dashes denote unevaluated domain-specific comparisons. Values are comparable only within each column.

A. Nonlinear networks and joint buildings: acceleration relative L_2			
Predictor	Mechanical (128)	Torsion (256)	Buildings (582)
F-FNO	0.4241 ± 0.0011	0.2914 ± 0.0022	0.2321^{s}
LNO	0.5525 ± 0.0018	$0.3839^{\dagger}$	$0.5729^{s,\dagger}$
GNOT	$3.1399^{s,\dagger}$	$0.3838^{s,\dagger}$	$0.5728^{s,\dagger}$
Mamba-2	0.8406^{s}	0.2993 ± 0.0019	0.2127^{s}
Transolver++	0.8655 ± 0.0195	0.2931 ± 0.0012	0.3241^{s}
UPT	$3.1399^{\dagger}$	$0.3838^{\dagger}$	$0.5728^{\dagger}$
C-PhysFNO (enhanced)	—	—	0.2581 ± 0.0021
MRRO	$\mathbf{0.2321} \pm 0.0023$	$\mathbf{0.2103} \pm 0.0141$	$\mathbf{0.1250} \pm 0.0005$

B. Fields and measured wings		
Predictor	Reaction–diffusion (64) Concentration relative L_2	E001 wings (181) Normalized RMSE
F-FNO	0.3263 ± 0.0061	1.2784 ± 0.0661
LNO	0.4084^{s}	$1.0766^{s,\dagger}$
GNOT	0.3286^{s}	$1.0766^{s,\dagger}$
Mamba-2	0.2838 ± 0.0190	1.0721 ± 0.0055
Transolver++	0.2544 ± 0.0205	$1.0799 \pm 0.0004^{\dagger}$
UPT	0.4004 ± 0.0017	$1.0766^{\dagger}$
IRNO	0.2026 ± 0.0327	—
MRRO	$\mathbf{0.1426} \pm 0.0131$	$\mathbf{1.0343} \pm 0.0016$

Unprinted SDs for paired dagger entries are below 10^{-4}; full precision is provided with the results. Fields use the response-space realization defined in Section 3.3. Internal controls and simple references are reported separately in Section 5 and the supplement.

Fourier/pointwise blocks (Kim et al., 2026). Its enhanced version interpolates 128 complex matrix knots over 0–25 Hz and uses training-fitted prediction/reference regression. This expands the long-record spectral bandwidth at the same neural parameter count and number of complete-event updates.

Relative to enhanced C-PhysFNO, MRRO gives **51.55%** lower error (Table 2); both seeds and 123/128 family means favor it, with a paired family-bootstrap 95% percentile interval $[0.11786, 0.14848]$ for C-PhysFNO minus MRRO. Among single-run external comparisons, Mamba-2 has the lowest error (0.21273), ahead of F-FNO (0.23211). MRRO at the same seed gives 0.12542, 41.05% below Mamba-2. Table 2 separates these results from repeated training comparisons; they do not estimate seed variation.

Computational cost and scope. A800 cached applied-force/earthquake queries take 21.77/23.57 ms for MRRO versus 12.40/22.45 ms for enhanced C-PhysFNO, including transfer and regression. Fewer parameters (1.32M versus 17.02M) do not ensure lower latency (Appendix C.3). Prior evaluation exposure, incomplete solves and one held-out wing design family limit generalization.

MRRO outperforms enhanced C-PhysFNO on all seven programs and on aggregate displacement, peak and unloading errors. Figure 4 shows two complete illustrative records. The earthquake is selected at the lower error quartile; the full error distribution and a longer metadata-selected case remain in the supplement.

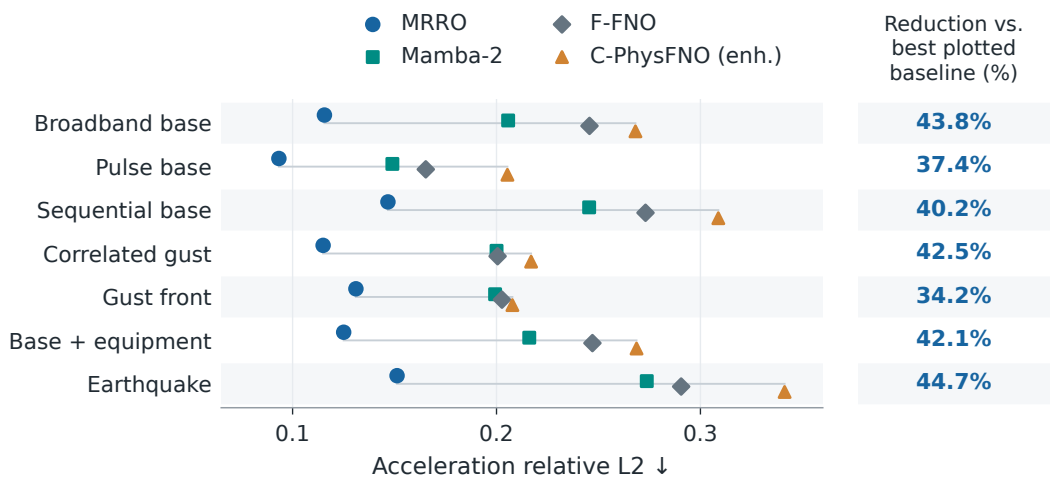

Figure 3: **External accuracy across seven building forcing programs.** Seed 41; families have equal weight within each program. Reductions compare MRRO with the best of the three plotted external errors per row, using unchanged predictors across programs. Family–program coverage is incomplete.

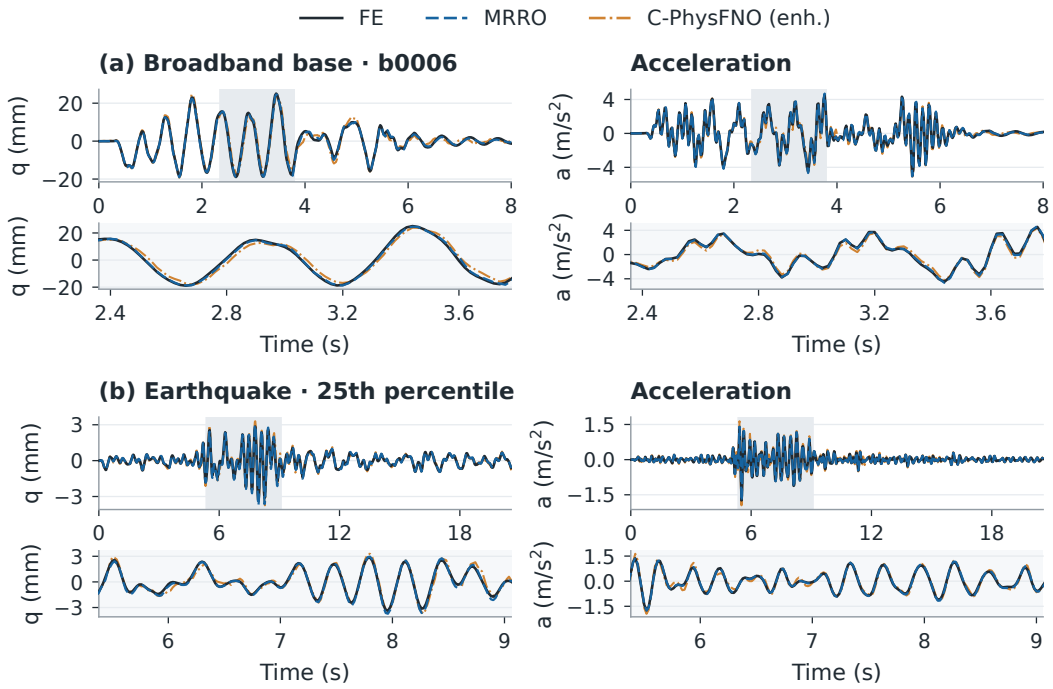

Figure 4: **Illustrative building response trajectories.** Seed 41, last observed floor. The broadband case is metadata-selected; the earthquake is the lower-quartile case among 233 records ranked by MRRO acceleration error. Shaded windows use finite-element (FE) response energy; expanded views preserve units. Appendix C.4 includes the longer metadata-selected earthquake and full error distribution.

4.3 Reaction–diffusion and measured folding wings

The field realization applies diffusion-rate kernels within the proposal-conditioned backbone. Its error is 29.62% below IRNO on the same 64 events (Table 2). F-FNO, Mamba-2, Transolver++ and GNOT also fit the training trajectories (training errors 0.00423–0.02734), but have larger held-out errors. The role of scale granularity is examined separately in Section 5.

E001 uses independent GVT calibration, nominal gust/electrical-command inputs and a 0.5-s initialization prefix. MRRO gives 3.52% lower error than Mamba-2 and 19.10% lower error than F-FNO. Its advantage over the prefix-mean reference is 3.93%; front/rear relative L_2 remains 0.92571/0.95477, indicating limited absolute prediction skill. All predictors receive the same initialization and command information. Capacity and command-conditioning studies are reported in Appendix G.2.

5 Information-matched scale-allocation controls

The common-scale control uses the system-mean frequency for Gaussian-basis coordinates, retaining actual modal frequencies in the coefficient context. The building control retains per-mode phase. Both models receive identical descriptors and references. Node-space and full-channel spectral controls also match depth and learning-rate choices (Appendix E).

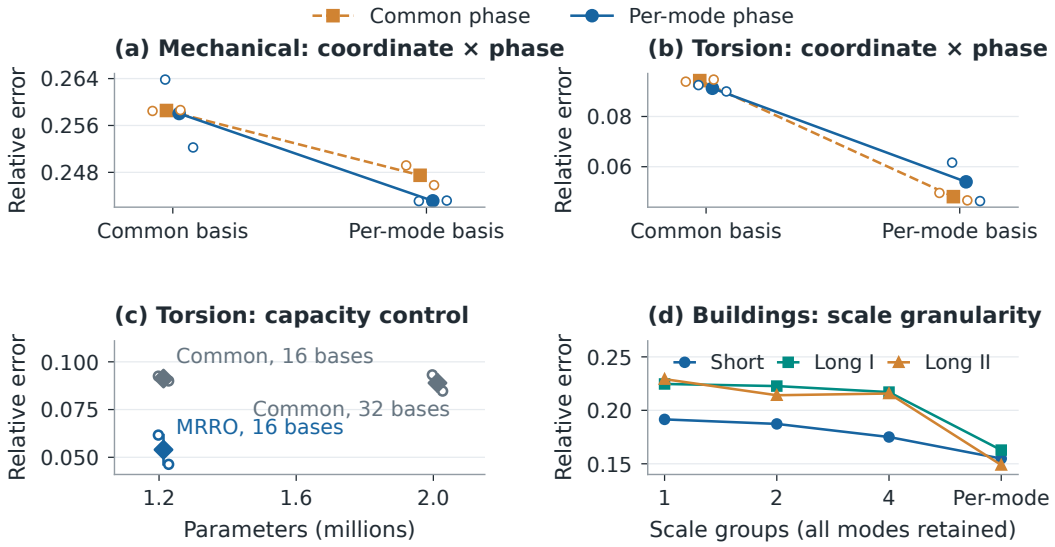

Figure 5: **Scale allocation, phase parameterization and capacity.** Validation acceleration relative L_2 error. (a,b) Two-run means and individual runs at fixed phase. (c) Capacity control at per-mode phase. (d) Scale grouping with unchanged parameters and all modes retained. These separate studies use the conventions in Appendix A.3; panel axes differ.

Coordinates, phase, capacity and granularity. Per-mode basis placement improves both domains under either phase setting (Figure 5a,b). Per-mode phase is not uniformly beneficial: torsional error is 0.04808 with common phase versus 0.05397 for the main configuration. Doubling common-scale bases to 32 adds 65.8% parameters yet leaves error at 0.08893 (panel c). Separate building cohorts vary scale groups with fixed parameters, modes and training sequences (panel d). Per-mode allocation is best in all three; intermediate grouping is not uniformly monotonic. These cohorts differ from the final benchmark (Appendix E).

Common-scale control on buildings. MRRO reduces error by **14.15%** across five paired runs (Table 3); all runs and 114/128 family means favor it. The 95% percentile family-bootstrap interval for common scale minus MRRO is $[0.01590, 0.02552]$, conditional on fitted models; seed SD measures training variation.

Table 3: **Matched scale allocation on buildings: five training runs.** Matched information and parameters; mean ± seed SD. Table 2 instead uses two MRRO runs for external comparisons.

Predictor	Acceleration relative L_2
MRRO	$\mathbf{0.12603} \pm 0.00286$
Common-scale kernels	0.14680 ± 0.00206

Field common-scale and per-mode kernels are effectively tied, delimiting the scope of this allocation rule. Additional node-coordinate and full-channel controls also improve with depth but retain higher network errors (Table 10).

Optimization budget. At 3×10^{-4}, the validation gap changes from 20.61% at the first nonzero checkpoint to 7.66% at 52,360 updates and 9.57% at termination. Figure 8 supports a finite-budget learning advantage, not an expressivity barrier or hardware-independent speedup.

6 Frozen action-consequence queries

For each candidate, the frozen predictor replays history: $\widehat{y}^{(a)} = F_\theta(s, u_{\text{hist}} \oplus [d + B_s a], b_{\text{obs}}, m_b, o)$, branching at τ. An external selector minimizes $J_g(\widehat{y}^{(a)}, a)$. This action-conditioned use requires neither goal-specific training nor a cached Markov state (Zhou et al., 2025; Hu et al., 2025).

The study uses 12 new systems, two histories and nine actions. Tail displacement or acceleration energy plus an action penalty defines cost. The solver evaluates every candidate without resetting history (Appendix F).

Table 4: Action selection: 12 systems, two histories and a shared finite library. Percentages use 48 decisions per learned method/goal and 24 for the reference.

	Tail displacement		Tail vibration	
Predictor	True cost	Worse than no control (%)	True cost	Worse than no control (%)
MRRO	0.05200	8.33	0.32721	22.92
F-FNO	0.05689	4.17	0.36580	58.33
Linear reference	0.05680	20.83	0.34433	29.17
No control	0.05970	—	0.34692	—

MRRO reduces true costs by 8.61%/10.55% against F-FNO; 95% percentile system/template bootstrap intervals for F-FNO minus MRRO are $[0.00029, 0.01263]$/$[0.00044, 0.08688]$. Some choices are worse than no control. Coordinate controls improve response differences without stable decision gains (Appendix F).

7 Conclusion

MRRO uses system-specific modal scales to configure a finite temporal-kernel basis, then learns nonlinear coupling after synthesis into node coordinates. Information-matched network and building experiments show that basis placement contributes beyond conditioning kernel coefficients on the same physical descriptors; phase, capacity and learning-budget controls delimit this benefit. The field and measured-wing studies extend the kernel construction, while showing that mode-resolved placement is not equally beneficial in every setting. Frozen action queries demonstrate a downstream use without equating response accuracy with reliable decisions. Together, these results identify scale-allocation granularity as a practical design choice for transferable response learning.

REPRODUCIBILITY STATEMENT

The complete latent algorithm is specified in Algorithm 1 and Appendix A; domain equations, data conventions and evaluation protocols appear in Appendices B and G. The supplementary materials include frozen inference examples with separate scoring, result arrays, numerical diagnostics, generator sources and training configurations. The included checks reproduce inference and aggregation; they do not constitute a fresh rerun of every training experiment.

REFERENCES

S. Meng, Y. Zhou, B. Liao, M. Chang, T. Zhang, A. Jafari, and A. Djerrad. A physics-informed foundation model for rapid high-fidelity structural response prediction. *Nature Communications*, 17:8884, 2026. doi:10.1038/s41467-026-75508-5.

T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia. Learning Mesh-Based Simulation with Graph Networks. *ICLR*, 2021.

Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. *ICLR*, 2021. arXiv:2010.08895.

X. Liu, S. Shang, X. Wang, P. Ren, and Y. Yang. Iterative Refinement Neural Operators are Learned Fixed-Point Solvers: A Principled Approach to Spectral Bias Mitigation. *ICML*, 2026. arXiv:2605.24041v2.

A. Tran, A. Mathews, L. Xie, and C. S. Ong. Factorized Fourier Neural Operators. *ICLR*, 2023.

G. Chen, X. Liu, Q. Meng, L. Chen, C. Liu, and Y. Li. Learning neural operators on Riemannian manifolds. *National Science Open*, 3(6):20240001, 2024. doi:10.1360/nso/20240001.

H. Tang, J. Zhu, Z. Feng, H. Li, and C. Li. Geometric Laplace Neural Operator. *arXiv:2512.16409v1*, 2025.

B. Alkin, A. Fürst, S. Schmid, L. Gruber, M. Holzleitner, and J. Brandstetter. Universal Physics Transformers: A Framework For Efficiently Scaling Neural Operators. *NeurIPS*, 2024. arXiv:2402.12365.

G. Zhou, H. Pan, Y. LeCun, and L. Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. *ICML*, PMLR 267:79115–79135, 2025.

Q. Cao, S. Goswami, and G. E. Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6:631–640, 2024. https://doi.org/10.1038/s42256-024-00844-4.

X. Carrillo Córcoles, C. Mertens, A. Sciacchitano, B. W. van Oudheusden, R. De Breuker, and J. Sodja. Effect of Wing Stiffness and Folding Wingtip Release Threshold on Gust Loads. *Journal of Aircraft*, 60(6):1917–1936, 2023. doi:10.2514/1.C037108. E001 data: doi:10.4121/20222334; M001 GVT data: doi:10.4121/984090ea-26fd-4809-8dac-f41367bf8916.

J. Berner, M. Liu-Schiaffini, J. Kossaifi, V. Duruisseaux, B. Bonev, K. Azizzadenesheli, and A. Anandkumar. Principled approaches for extending neural architectures to function spaces for operator learning. *Nature Machine Intelligence*, 8:1173–1181, 2026. doi:10.1038/s42256-026-01267-z.

J. Kim, S. Yi, and Z. Wang. A composition of simplified physics-based model with neural operator for trajectory-level seismic response predictions of structural systems. *Structural Safety*, 119:102668, 2026. doi:10.1016/j.strusafe.2025.102668. Author code: https://github.com/Jungh0Kim/C-PhysFNO.

T. K. Rusch and D. Rus. Oscillatory State-Space Models. *ICLR*, 2025.

P. Hu, R. Wang, X. Zheng, T. Zhang, H. Feng, R. Feng, L. Wei, Y. Wang, Z.-M. Ma, and T. Wu. Wavelet Diffusion Neural Operator. *ICLR*, 2025.

T. Dao and A. Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. *ICML*, PMLR 235:10041–10071, 2024.

H. Luo, H. Wu, H. Zhou, L. Xing, Y. Di, J. Wang, and M. Long. Transolver++: An Accurate Neural Solver for PDEs on Million-Scale Geometries. *ICML*, PMLR 267:41432–41449, 2025.

Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, and J. Zhu. GNOT: A General Neural Operator Transformer for Operator Learning. *ICML*, PMLR 202:12556–12569, 2023. https://proceedings.mlr.press/v202/hao23c.html.