SPECTRAL ATTENTION OPERATOR FOR LEARNING BOUNDARY-DRIVEN HEMODYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Patient-specific simulation of boundary-driven hemodynamics requires solving the governing PDEs for every inlet waveform and physical parameter setting. We introduce the Spectral Attention Operator (SAO), which learns the map from an inlet waveform to the response at any queried location and parameter setting. SAO couples temporal Fourier modes according to the inlet and adapts its temporal functions to location and physical parameters through Feature-wise Linear Modulation. It predicts complete trajectories in one forward pass without temporal rollout. Across three viscous Burgers and two hemodynamics benchmarks, SAO attains the best overall accuracy in comparisons with five capacity-matched operator families, with relative L^2 errors near or below 1%. On the multi-frequency and hemodynamic flow-rate tasks, SAO reduces error by 46–71% against the strongest external baseline. Ablations isolate the contributions of cross-mode attention and query conditioning, and physics-based diagnostics show accurate shock localization and hemodynamic fields consistent with the governing relations. For hemodynamics, SAO is hundreds to thousands of times faster than numerical simulation on CPU and four to five orders of magnitude faster on GPU.

1 INTRODUCTION

Simulating physical systems governed by partial differential equations (PDEs) often requires repeatedly solving the same equations for different boundary signals and parameter settings. For example, in arterial hemodynamics, each heartbeat generates an inlet waveform that propagates through the vascular network as coupled pressure, flow, and cross-sectional area waves. The shapes of these waves depend on vessel geometry, wall stiffness, and downstream resistance (Kissas et al., 2020; Schiavazzi et al., 2017). They therefore carry information about physiological properties that are difficult or impossible to measure directly.

Tasks such as parameter inference, sensitivity analysis, and uncertainty quantification (Melis et al., 2017; Schiavazzi et al., 2017) often require repeated model evaluations across inlet waveforms and physiological parameters. One-dimensional hemodynamics offers an efficient forward model by representing the vascular network as connected one-dimensional deformable tubes and resolving pressure, flow, and area along each vessel over time. Although far cheaper than three-dimensional simulation (Pfaller et al., 2022; Benemerito et al., 2024), a single network simulation can still take minutes or even hours, depending on the complexity of the network and its physical parameters.

Neural operators reduce repeated-simulation cost by learning solution maps over families of input functions (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). Once trained, the same operator can be evaluated for new inputs without solving the PDE again. DeepONet represents an operator through learned input-dependent coefficients and coordinate-dependent basis functions, while the Fourier neural operator (FNO) learns global convolutions in the Fourier domain (Lu et al., 2021; Li et al., 2021). Improved DeepONets and MIONet extend branch–trunk factorizations and multiple-input maps (Wang et al., 2022; Jin et al., 2022); wavelet and convolutional operators alter the basis or preserve multiscale local structure (Tripura & Chakraborty, 2023; Raonić et al., 2023); geometry-aware FNOs generalize spectral representations beyond regular grids (Li et al., 2023b). Galerkin transformers, OFormer, GNOT, and Transolver instead use attention to couple input samples, latent tokens, or geometric states (Cao, 2021; Li et al., 2023a; Hao et al., 2023; Wu et al., 2024).

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.

However, most neural-operator work on time-dependent PDEs is organized around initial-value problems: the model receives an initial state, sometimes with forcing, coefficients, or physical parameters, and predicts the ensuing field under fixed boundary conditions. This work considers a boundary driven initial boundary value problem in which an entire time-varying boundary waveform drives the evolution and the model returns a complete response waveform at any requested location.

A smaller line of work makes boundary data variable, but targets different settings. For example, BV-NORM learns families of PDE solutions as boundary values and domain geometry vary (Deng et al., 2025). Physics-informed models reconstruct cardiovascular fields from sparse measurements, and graph-based reduced-order models learn cardiovascular surrogates across vascular geometries (Kissas et al., 2020; Arzani et al., 2021; Pegolotti et al., 2024). These models incorporate boundary data, geometry, or physiological parameters, but organize the learned map around different inputs, outputs, or computational axes (Table 1.1).

Table 1.1: **How each operator supports boundary-driven trajectory learning.** ✓: native; △: supported through our interface adaptation; ×: not supported without an additional mechanism.

Formulation property	**DeepONet**	**FNO**	**OFormer**	**BV-NORM**	**SAO** (ours)
Boundary waveform as functional input	△	△	△	✓	✓
Query-conditioned response history	✓	△	✓	×	✓
Direct full-trajectory prediction	✓	△	✓	×	✓
Input-adaptive frequency coupling	×	×	×	×	✓
Query-adaptive temporal representation	×	△	✓	×	✓

In this setting, a complete inlet waveform determines a query-conditioned response waveform along the same temporal coordinate. We denote this map as

$$\mathcal{G} : \big(g(\cdot), \mu, x\big) \longmapsto y(x, \cdot),$$

where $g(\cdot)$ is the inlet waveform, x is the queried location, μ contains physical parameters, and $y(x, \cdot)$ is the requested response trajectory. The input and output vary along the same axis, time; space and physical parameters specify which output signal to predict. How a model represents and combines these inputs imposes an inductive bias on the learned map (Battaglia et al., 2018). Nonlinear propagation steepens and reflects waves, redistributing signal energy across temporal frequencies according to the inlet's amplitude and spectral content (LeVeque, 2002; Blackstock, 1966; Anliker et al., 1971). A standard Fourier layer, however, uses the same learned frequency-mixing weights for every input. Moreover, the output waveform varies with location and material properties. If the temporal functions are fixed across queries, the model can capture those changes only by reweighting them. These considerations motivate inlet-adaptive frequency interactions and query-adaptive temporal functions.

Thus, we introduce the **Spectral Attention Operator (SAO)** to learn this boundary-to-response map along time. Its Dynamic Net converts the inlet into latent temporal functions using spectral blocks with self-attention across retained Fourier modes. Because each inlet determines its attention weights, spectral coupling changes with the driving waveform. A Static Net maps the query $c = (x, \mu)$ to synthesis coefficients and Feature-wise Linear Modulation (FiLM) parameters (Perez et al., 2018). FiLM modifies the temporal features throughout the Dynamic Net before they are assembled into the response. Given the complete inlet waveform, SAO returns the complete response trajectory in one forward pass. This avoids the per-step inference cost and error accumulation of autoregressive neural PDE solvers (Brandstetter et al., 2022; Lippe et al., 2023).

Our contributions are:

- We formulate fixed-topology boundary-driven simulation as learning an operator from an inlet waveform to response trajectories conditioned on location and physical parameters. The operator acts along the **temporal coordinate** shared by its input and output.
- We propose SAO, combining inlet-dependent attention across temporal Fourier modes with query-dependent FiLM conditioning. Its temporal representation adapts to both the driving waveform and the requested response.

- We evaluate SAO against five capacity-matched operator families on five benchmarks, with dense-grid testing after training on 10 probes per simulation. We conduct extensive ablations to isolate the roles of its main components and use physics-based measures to assess shock resolution and hemodynamic consistency.

2 PRELIMINARIES

We consider 1D hyperbolic systems of conservation laws of the general form

$$\partial_t U(x,t) + \partial_x F(U(x,t)) = S(U(x,t)), \qquad x \in \Omega \subset \mathbb{R},\ t \in (0,T], \tag{1}$$

where $U \in \mathbb{R}^m$ is the vector of conserved states, $F : \mathbb{R}^m \to \mathbb{R}^m$ is the flux, and S collects source terms. Such systems, including gas dynamics, shallow-water flow, and hemodynamics, exhibit finite-speed wave propagation, nonlinear steepening, and the formation of steep gradients (Appendix B).

Problem formulation. The dynamics in (1) are driven by a time-dependent inlet boundary condition $g(t) \in \mathbb{R}^{d_{\text{in}}}$, $t \in [0,T]$, and optionally by system parameters $\mu \in \mathbb{R}^{d_\mu}$ (e.g., vessel-wall stiffness). Let $\mathcal{X}$ denote the space of admissible inlet waveforms (functions $g : [0,T] \to \mathbb{R}^{d_{\text{in}}}$), $\mathcal{M} \subseteq \mathbb{R}^{d_\mu}$ the parameter set, and $\mathcal{Y}$ the space of spatio-temporal responses $y : \Omega \times [0,T] \to \mathbb{R}^{d_{\text{out}}}$. Our goal is to learn a parametric operator

$$\mathcal{G} : \mathcal{X} \times \mathcal{M} \to \mathcal{Y}, \qquad y = \mathcal{G}(g,\mu), \tag{2}$$

mapping inlet waveforms and parameters to the spatio-temporal response $y(x,t) \in \mathbb{R}^{d_{\text{out}}}$ throughout the domain. The operator $\mathcal{G}$ acts on the continuous inlet $g(\cdot) \in \mathcal{X}$; the network receives its samples on a uniform temporal grid, a vector $\mathbf{g} \in \mathbb{R}^{d_{\text{in}} \times N_t}$, and we write g for the function and $\mathbf{g}$ for the sampled vector. In practice, we learn the pointwise evaluation: given a query location $x \in \Omega$, the neural operator returns the temporal response at that location,

$$\widehat{y}(x,\cdot) \;=\; \widehat{\mathcal{G}}_\theta(g(\cdot),\, \mu,\, x) \;\approx\; y(x,\cdot). \tag{3}$$

For each benchmark the learned map is conditional on the fixed problem specification used to generate the data (initial state, non-inlet boundary conditions, geometry or topology, constitutive constants, observation interval), which are not varied inputs. Because the complete sampled inlet is available, the task is offline full-trajectory emulation rather than causal prediction at intermediate times.

3 METHODS

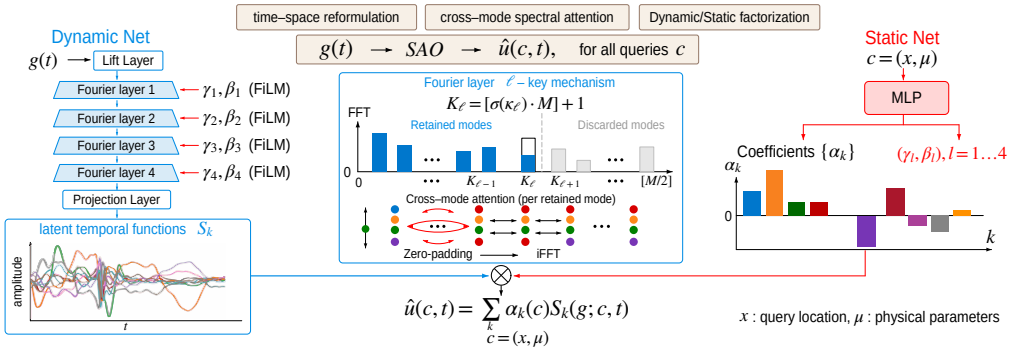

Figure 3.1: **Overview of SAO.** *Left:* the Dynamic Net turns the inlet waveform into latent temporal functions. *Centre:* one spectral block, with its retained modes and cross-mode attention. *Right:* the Static Net turns the query into coefficients and FiLM parameters. *Bottom:* their combination gives the prediction (Sections 3.2–3.4).

We propose **SAO**, a neural operator with an inlet-driven Dynamic Net and a query-driven Static Net (Figure 3.1). Given an inlet time series $g(\cdot)$ and a conditioning input c, the Dynamic Net turns g into temporal functions and the Static Net turns c into synthesis coefficients and FiLM parameters. The FiLM parameters condition the temporal functions, and the coefficients combine them into $\hat{y}(c, \cdot)$.

3.1 Design for boundary-driven hyperbolic dynamics

SAO implements the pointwise operator in Eq. (3): it takes the complete inlet waveform $g(\cdot)$ and returns the response trajectory $y(x, \cdot)$ at a query specified by c. The prediction requires **no** temporal rollout, and the shared temporal coordinate of the inlet and response allows the Dynamic Net to apply its spectral blocks along time. Many response trajectories admit compact temporal representations (Benner et al., 2015; Greif & Urban, 2019). SAO learns a set of temporal functions from the inlet and combines them with query-dependent coefficients. Within each spectral block, the mode gate selects the retained bandwidth and cross-mode attention adapts interactions among the retained frequencies to the current hidden representation.

3.2 Query conditioning (Static Net)

Because location and physical parameters determine the requested response, the Static Net maps the conditioning input $c \in \mathbb{R}^{d_c}$ to coefficients that weight the temporal functions:

$$\alpha_\theta(c) \in \mathbb{R}^{V \times K}, \tag{4}$$

where $\alpha_{\theta,v,k}(c)$ weights temporal function k for variable v (V=3 for blood flow). The Static Net is an MLP with GELU activations.

Spatial query encoding. For all benchmarks, we replace the raw scalar coordinate x with a $(2K_x{+}1)$-dimensional sinusoidal encoding,

$$\mathbf{p}(x) = \big[x,\ \sin(\pi x),\ \cos(\pi x),\ \ldots,\ \sin(K_x \pi x),\ \cos(K_x \pi x)\big], \quad K_x = 8, \tag{5}$$

giving a 17-dimensional positional encoding. The complete conditioning vector is $c = \mathbf{p}(x)$ for Burgers, $c = [\mathbf{p}(x), E, R]$ for H1, and $c = [\mathbf{p}(x), \mathbf{e}_j, E_1, E_2, E_3, R_1, R_2, R_3]$ for H2, where $\mathbf{e}_j \in \mathbb{R}^3$ identifies the queried vessel. Thus d_c is 17 for Burgers, 19 for H1, and 26 for H2. This encoding provides features at several spatial frequencies for learning the dependence on query location.

FiLM conditioning. FiLM lets the query alter the temporal features before they are combined into the response. For each of the L spectral blocks, the Static Net produces scale and shift vectors $(\gamma_\ell, \beta_\ell) \in \mathbb{R}^C$. After block ℓ, the hidden representation is modulated as

$$h_\ell \ \leftarrow \ (1 + \gamma_\ell) \odot h_\ell + \beta_\ell, \tag{6}$$

where $\odot$ denotes elementwise multiplication broadcast over time. The FiLM (Perez et al., 2018) parameters are zero-initialized so that the modulation initially acts as the identity (Appendix E.6).

3.3 Dynamic Net (conditioned waveform-to-function map)

Let $g \in \mathbb{R}^{N_t}$ denote the discretized inlet waveform on a fixed time grid. Conditioned by the FiLM parameters from the Static Net, the Dynamic Net produces

$$S_\theta(g; c) \in \mathbb{R}^{K \times N_t}, \tag{7}$$

where the k-th row $S_{\theta,k}(g; c) \in \mathbb{R}^{N_t}$ is a learned temporal function that depends on both the inlet and the query. We select the latent synthesis width K separately for each benchmark; the values and reduced-basis analogy are in Appendix E.2.

The waveform is lifted from one channel to width C by a linear layer, then passed through L stacked *Spectral Attention Blocks*. Each block performs four operations:

1. computes an FFT along the temporal dimension, then truncates to $K_{\mathrm{eff},\ell}$ retained Fourier modes, where $K_{\mathrm{eff},\ell} = \lfloor \sigma(\kappa_\ell) \cdot M \rfloor + 1$ is controlled by a per-layer learnable scalar κ_ℓ;

2. applies multi-head self-attention across the retained Fourier modes, representing each mode by the real and imaginary parts of its channel vector, and adds the attention output residually in the frequency domain;
3. applies complex-valued spectral mixing weights to the retained modes, with the fractional part of $K_{\mathrm{soft},\ell}{=}\sigma(\kappa_\ell)\cdot M$ used as a differentiable weight on the last mode;
4. zero-pads back to M modes and applies an inverse FFT.

For the retained-mode token matrix $Z_\ell \in \mathbb{R}^{K_{\mathrm{eff},\ell}\times 2C}$ in the second step, the attention operation is

$$\mathrm{Attn}(Z_\ell) = \mathrm{softmax}\left(\frac{(Z_\ell W_Q)(Z_\ell W_K)^\top}{\sqrt{d}}\right) Z_\ell W_V, \tag{8}$$

where each token contains the real and imaginary channel values of one retained Fourier mode and d is the dimension of one attention head. Standard FNO uses complex spectral weights that are fixed after training, whereas the attention in each SAO block computes activation-dependent couplings among the retained Fourier modes. A parallel 1×1 convolution in each block adds pointwise mixing, GELU activations separate the layers, and two final linear projections map the hidden representation to K temporal functions. Algorithm 1 in Appendix E.3 gives the full pseudocode for the Spectral Attention Block. Appendix E.4 analyzes its computational complexity relative to standard FNO.

3.4 Temporal synthesis

The predicted response at query c is obtained by contracting the static coefficients with the query-conditioned temporal functions:

$$\hat{y}_v(c,\cdot) \;=\; \sum_{k=1}^{K} \alpha_{\theta,v,k}(c)\, S_{\theta,k}(g;c) \;\in\; \mathbb{R}^{N_t}, \qquad v=1,\dots,V. \tag{9}$$

For hemodynamics, one SAO model predicts P, Q, and A jointly: the three outputs share the temporal functions $S_\theta(g;c)$ and use separate coefficient vectors $\alpha_{\theta,v}(c)$. This sharing reflects the coupling of the three fields in the governing equations while preserving a separate synthesis for each output. Appendix H.6 compares this shared construction with field-specific temporal functions. We refer to S_θ as the "dynamic" functions and α_θ as the "static" coefficients. FiLM gives $S_{\theta,k}(g;c)$ joint dependence on the inlet and query. Consequently, K is the latent synthesis width and does not bound the rank of the assembled space–time field. Appendix E.1 relates this construction to reduced-basis methods and DeepONet, while Appendix I isolates the strictly separable variants for which a rank bound does apply.

4 Experiments

4.1 Experimental setup

Benchmarks. We evaluate five benchmarks from two systems (full formulations in Appendix B). Viscous Burgers isolates nonlinear advection, wave steepening, and cross-frequency transfer, directly probing the input-dependent spectral interactions central to SAO. B1–B3 are viscous Burgers cases, a scalar conservation law ($m{=}1$), generated with `Clawpack` (Mandli et al., 2016); the inlet is one, ten or twenty sinusoids with random amplitudes, frequencies and phases. The summed expected amplitude is matched across the three, so wave count and effective driving amplitude vary together (root-mean-square inlet amplitude 2.94, 0.93 and 0.66). The 1D blood-flow equations are established network-scale models of arterial pulse propagation and form the target application, where conventional repeated simulation remains costly (Pfaller et al., 2022; Benemerito et al., 2024). H1 and H2 are 1D blood-flow cases ($m{=}2$, state $[A,Q]^\top$) generated with `openBF` (Benemerito et al., 2024): H1 isolates coupled propagation in a single artery, while H2 adds wave splitting and junction interactions in a Y-bifurcation. Both are driven by physiological inlet waveforms, with wall stiffness E and radius scale R sampled per vessel, and the three fields P, Q and A predicted together. The cases hold 500, 2000, 2000, 1000 and 1000 simulations, split 70/20/10 at the simulation level. Parameter ranges, grids and solver settings are in Appendix D.

Baselines and ablations. To test SAO against distinct operator designs, we compare it with spectral, branch–trunk, attention-based, and fixed-basis models: FNO-T (Li et al., 2021), DeepONet* (Lu et al., 2021), POD-DeepONet (Lu et al., 2022), OFormer (Li et al., 2023a), and POD+MLP. FNO-T applies the FNO along time with the query broadcast as input channels. DeepONet* uses the inlet to generate the basis and the query to supply the coefficients. POD-DeepONet and POD+MLP predict coefficients over a fixed proper orthogonal decomposition (POD) basis, the latter as a regression control. Each model receives (g, x, μ), predicts $y(x, t)$, and is matched to SAO within $\sim 0.4\%$ in parameter count. The models use identical data splits, optimizer, schedule, and evaluation points (Appendices H.1–H.3). Four ablations test query adaptation, input-dependent mode coupling, bandwidth selection, and temporal feature learning by removing FiLM (`SAO-noFiLM`), cross-mode attention (`SAO-noAttn`), the learnable mode gate (`SAO-fixedModes`), or the spectral Dynamic Net (`SAO-MLP`), respectively. `SAO-MLP` has fewer parameters than SAO by construction; implementation and capacity details are in Appendices E.3–E.6 and H.2. Appendix A collects the definitions of every variant named in this paper.

Implementation details. Every model is trained on 10 random probe locations per simulation (per vessel on H2). Sparse per-simulation probes require the model to recover waveform–location pairs absent from that simulation's training targets. Evaluation uses all 100 non-inlet locations for Burgers, 101 locations for H1, and 303 locations across the three H2 vessels. Inputs are z-score normalized. For hemodynamics, the training objective averages the relative L^2 error over the three fields and adds the gate penalty:

$$\mathcal{L}_{\text{hemo}} = \frac{1}{V} \sum_{v=1}^{V} \frac{\|\hat{y}_v - y_v\|_{\mathcal{B}}}{\|y_v\|_{\mathcal{B}}} + \lambda_{\text{gate}} \sum_{\ell=1}^{L} \sigma(\kappa_\ell)^2, \qquad V = 3, \tag{10}$$

where the norm spans the minibatch and all time steps after the predictions have been returned to physical units. Burgers uses mean-squared error in place of the first term (Appendix E.5). All models use AdamW (Loshchilov & Hutter, 2019) with warmup and cosine annealing, gradient clipping, and early stopping. SAO uses C=64, L=4 blocks, M=101 modes, and four attention heads. The per-case hyperparameters are in Appendix F.4 and the full training protocol in Appendix F. The test metric is $\text{RL2} = \|\hat{u} - u\|_F / \|u\|_F$, computed per field for hemodynamics and averaged over five seeds. Checkpoints are selected by validation RL2; Appendix D.2 specifies the probe protocol and dense evaluation grid.

4.2 Main results

Tables 4.1 and 4.2 report five-seed test errors for SAO, its ablations, and the five external baselines. Figure 4.1 shows error maps for reference simulations from B2 and the H2 parent vessel. The remaining cases and the H2 daughter vessels are in Appendix G.3. SAO reaches 0.90–1.03% RL2 on Burgers and 0.041–0.161% across the H1 fields. Its error is 46–71% below the best external baseline on B2, B3, and both flow-rate columns. SAO is not the lowest in two comparisons: OFormer matches it on B1, and DeepONet* has a slightly lower H2 area error within seed variation (Appendix G.4). Per-seed values and parameter counts are in Appendices G.5 and H.2.

Table 4.1: Test RL2 (%) on Burgers, 5-seed means. Best in bold, second best underlined; the last rows give SAO's relative gain over the best ablation and the best baseline.

Variant	B1	B2	B3
SAO (ours)	<u>0.93</u>	**1.03**	**0.90**
`SAO-noFiLM`	22.56	11.50	2.42
`SAO-noAttn`	1.33	4.00	3.99
`SAO-fixedModes`	3.62	<u>2.06</u>	<u>1.75</u>
`SAO-MLP`	6.94	8.59	8.10
OFormer	**0.92**	3.39	3.12
DeepONet*	6.55	14.98	11.87
POD-DeepONet	15.58	24.87	19.30
FNO-T	4.22	5.19	5.14
POD+MLP	7.05	13.57	14.14
vs. ablations (%)	+30	+50	+49
vs. baselines (%)	-1	+70	+71

Learning rate sensitivity. Learning rate searches on representative model–case pairs preserve the ordering on B2 and H2. On B1, the selected rates give 0.88% error for OFormer and 1.00% for SAO (Appendix H.4).

Table 4.2: Test RL2 (%) on hemodynamics, 5-seed means of one three-field model per row (P, Q, A and their mean). Best in bold, second best underlined; the last rows give SAO's relative gain over the best ablation and the best baseline (paired-seed tests in §G.4).

	H1 (single artery)				H2 (Y-bifurcation)			
Variant	**P**	**Q**	**A**	**Mean**	**P**	**Q**	**A**	**Mean**
SAO (ours)	<u>0.088</u>	**0.161**	**0.041**	**0.097**	<u>0.139</u>	**0.129**	<u>0.052</u>	**0.107**
`SAO-noFiLM`	0.350	1.010	0.264	0.541	0.228	0.457	0.419	0.368
`SAO-noAttn`	0.149	0.394	0.072	0.205	0.231	0.563	0.102	0.299
`SAO-fixedModes`	**0.070**	<u>0.190</u>	<u>0.042</u>	<u>0.101</u>	**0.112**	<u>0.154</u>	0.067	<u>0.111</u>
`SAO-MLP`	0.178	0.411	0.084	0.224	0.283	0.978	0.237	0.499
OFormer	0.240	1.259	0.246	0.581	0.237	1.138	0.210	0.528
DeepONet*	0.151	0.341	0.070	0.187	0.171	0.241	**0.051**	0.154
POD-DeepONet	0.118	0.365	0.226	0.236	0.168	0.296	0.199	0.221
FNO-T	0.191	0.698	0.154	0.348	0.320	0.853	0.248	0.473
POD+MLP	0.176	0.470	0.291	0.312	0.230	0.686	0.488	0.468
vs. ablations (%)	-26	+15	+2	+4	-24	+16	+22	+4
vs. baselines (%)	+25	+53	+41	+48	+17	+46	-2	+31

Table 4.3: Physics-based error measures for the three-field models (%), 5-seed mean ± sd. Best in bold, second best underlined; the true-Q row is the ground-truth junction residual. Baseline rows are computed from their saved test predictions with the same formulas (Appendix H.7).

	H1 (K=10)		H2 (K=20)		
Model	**Derived U**	**Tube-law cons.**	**Derived U**	**Tube-law cons.**	**Junction mass**
SAO (ours)	**0.178**±0.007	**0.027**±0.003	**0.165**±0.012	<u>0.173±0.088</u>	0.192±0.013
`SAO-noFiLM`	1.072±0.037	0.532±0.020	0.744±0.015	2.057±0.073	0.529±0.069
`SAO-noAttn`	0.421±0.010	0.084±0.006	0.646±0.015	0.433±0.019	0.921±0.052
`SAO-fixedModes`	<u>0.208±0.022</u>	0.057±0.045	<u>0.200±0.012</u>	0.308±0.090	0.215±0.033
`SAO-MLP`	0.446±0.016	0.071±0.006	1.077±0.027	1.128±0.149	0.678±0.072
OFormer	1.355±0.208	0.479±0.078	1.239±0.143	0.833±0.236	0.277±0.081
DeepONet*	0.370±0.005	<u>0.051±0.005</u>	0.269±0.009	**0.113**±0.006	**0.146**±0.008
POD-DeepONet	0.458±0.007	0.486±0.002	0.371±0.008	0.779±0.010	<u>0.156±0.003</u>
FNO-T	0.757±0.010	0.333±0.010	1.029±0.035	1.179±0.047	1.567±0.095
POD+MLP	0.595±0.003	0.636±0.012	0.889±0.029	2.110±0.126	0.757±0.046
True-Q reference	—	—	—	—	0.206

Table 4.4: Physics-based error measures for Burgers (%). SAO values are 5-seed mean ± sd; the best-other columns report the lowest mean among the ablations and baselines.

Diagnostic	**Case**	**SAO**	**Best other**	**Model**
Steep-window RL2	B1	2.33±0.08	2.52	OFormer
	B2	1.34±0.04	2.76	`SAO-fixedModes`
	B3	0.93±0.01	1.82	`SAO-fixedModes`
Shock location (% domain)	B1	0.016±0.009	0.92	`SAO-noAttn`
	B2	0.000	0.011	`SAO-fixedModes`
	B3	0.000	0.005	`SAO-fixedModes`

Physical fidelity. Tables 4.3 and 4.4 evaluate properties absent from the training loss. On Burgers, SAO maintains low error in the steepest-gradient cells and locates the B2 and B3 shocks on the reference grid point. On hemodynamics, it has the lowest derived-velocity error on both cases and the lowest tube-law inconsistency on H1. The H2 junction residuals of DeepONet*, POD-DeepONet, and SAO all lie below the reference simulation's 0.206% residual, so this diagnostic

does not reliably rank their conservation fidelity. SAO retains lower three-field and derived-velocity errors than those baselines on H2. Appendix H.7 gives the metric definitions and paired tests.

Reference-discretization error. We refine the reference grids to distinguish surrogate error from discretization error in the numerical targets. On 20 H2 test simulations re-solved at twice the resolution, the reference's own grid error is $0.024/0.118/0.006\%$ on $P/Q/A$ against SAO's $0.136/0.138/0.046\%$. The flow-rate error is therefore comparable to the measured grid discrepancy. For 20 Burgers test simulations, doubling the reference grid changes the solutions by 0.50%, 1.12%, and 1.81% on B1–B3; quadrupling it changes them by 0.64%, 1.55%, and 2.57%. SAO's corresponding errors are 1.01%, 1.08%, and 0.83%. Thus the measured reference-grid discrepancy reaches or exceeds the model error on B2 and B3 (Appendix H.8).

Parameter extrapolation. Holding out the tails of each physical-parameter range tests whether the learned map extends beyond the parameter values seen during training. Errors increase by 1.08–$1.55\times$ on three of four case–parameter pairs when the models are trained on the middle 80% of one parameter and tested on its tails. On the H2 radius axis, one of three seeds reaches 1.03% error on the small-radius tail, compared with 0.19% for the other two seeds (Appendix H.10).

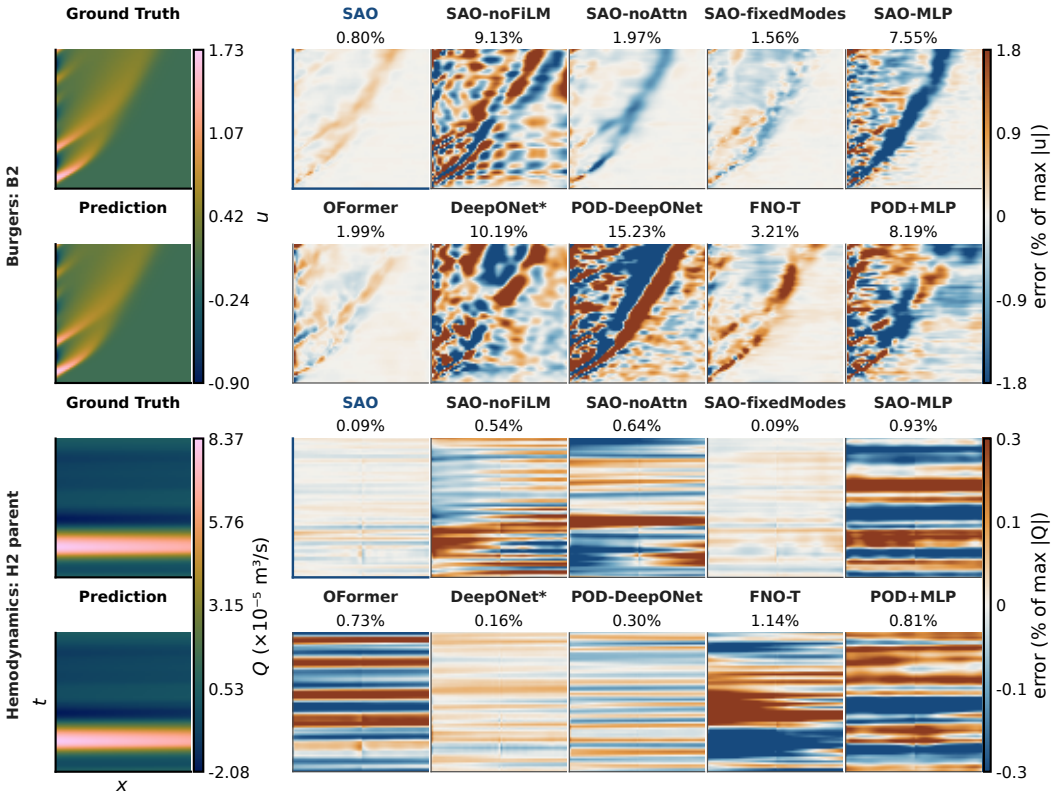

Figure 4.1: **Model comparison on the reference simulations of B2 and the H2 parent vessel.** Each block: ground truth, SAO's prediction, and every model's signed error with its RL2 (top row ablations, bottom row baselines), on that block's colour scale. B1, B3, H1, the H2 daughters and the scale rule are in Appendix G.3.

4.3 Model Analysis

Temporal reformulation. The spatial FNO control measures the contribution of organizing the operator along time. A spatial FNO given the complete (x, t) field under the same probe supervision and capacity beats FNO-T by $2.1\times$ on B2 and $3.2\times$ on H2. It remains $2.4\times$ and $1.4\times$ less accu-

rate than SAO, so the temporal reformulation alone does not explain SAO's advantage (Appendix Table H.2).

Query-dependent conditioning. Removing FiLM (`SAO-noFiLM`) inflates RL2 by $24\times$ on B1, $11\times$ on B2, $2.7\times$ on B3, $6.3\times$ on H1, and $3.5\times$ on H2 (Q field; Tables 4.1 and 4.2). The Burgers margin is largest on B1, whose travelling steep front produces strong spatial variation. In hemodynamics, the margin shows the value of conditioning on location and physical parameters. The FiLM modulation is evaluated at every non-inlet grid location of a held-out simulation, including the $\sim$90 of 100 locations that were not probes for it. Because the coordinates recur across simulations, this test measures interpolation to unseen *(waveform, location)* pairs on the training grid (Appendix D.2).

Spatial variation in hemodynamics. The hemodynamic fields vary less across space than time (Figure 4.1; $\mathrm{Var}_x[Q]/\mathrm{Var}_t[Q] = 0.031$ on H1 and $\approx$0.012 per H2 vessel), making them a stringent test of query dependence. Yet no x-independent predictor can go below 14.7% RL2 on H1 flow rate or 9.9% per H2 vessel, and a three-field control without positional encoding reaches 14.9% and 10.2% (Appendix H.9). SAO's corresponding errors are 0.161% and 0.129%. On the smoother pressure and area fields, the bounds are 2.3% and 1.0% on H1 and 0.22% and 0.05% on H2. SAO's margin accordingly shrinks to $1.6\times$ on H2 pressure and parity on H2 area, indicating that flow rate carries most of the spatial structure in this benchmark.

Cross-mode attention. SAO outperforms `SAO-noAttn` by $1.4\times$, $3.9\times$, $4.4\times$ on B1–B3 and $2.4\times$, $4.4\times$ on H1–H2 (Q field). The ablation shows the contribution of input-dependent cross-mode attention under the reported protocol.

Spectral bandwidth selection. SAO outperforms `SAO-fixedModes` (all modes kept) by $3.9\times$, $2.0\times$, $1.9\times$ on B1–B3 and $1.2\times$ on H1 and H2, a hemodynamic margin present only on Q (Appendix G.4). On B1–B3, a fixed per-layer cutoff at the learned bandwidth performs as well as or slightly better than the gate (Appendix G.1). The gate automates the selection of a compact bandwidth per layer, retaining 4–6% of the spectrum on Burgers and 18–23% on hemodynamics.

Computational efficiency. On one H100 at matched capacity, SAO's per-query inference (0.013–0.019 ms) is 5.6–$8.2\times$ faster than OFormer and $1.8\times$ to two orders of magnitude faster than DeepONet*, whose trunk rereads the inlet at every output time. The POD surrogates remain faster. Table 4.5 reports SAO's CPU and GPU speedups over the numerical solvers. Including data generation and training, SAO becomes cheaper after approximately 1,000 additional H1 or H2 simulations and after 1,540–5,540 additional Burgers simulations (Appendix G.2).

Table 4.5: Solver wall time per simulation against SAO's inference on one core of the same node or on one GPU; speedup is their ratio (Appendix G.2).

Case	Solver	SAO CPU core	SAO GPU	Speedup CPU	Speedup GPU
B1	2.35±0.11 s	91 ms	1.84 ms	26×	∼1,300×
B2	2.28±0.12 s	94 ms	1.32 ms	24×	∼1,700×
B3	2.33±0.13 s	91 ms	1.44 ms	26×	∼1,600×
H1	0.75±0.05 min	115 ms	1.91 ms	∼390×	∼24,000×
H2	12.4±1.0 min	320 ms	3.77 ms	∼2,300×	∼200,000×

5 Conclusion

We introduced the Spectral Attention Operator (SAO) for boundary-driven wave responses, with 1D hemodynamics as the target application. SAO combines inlet-driven temporal functions with query-dependent coefficients and FiLM, adapting the representation to the inlet and requested response. It attains 0.90–1.03% error on Burgers and the lowest flow-rate error on both hemodynamics benchmarks. Ablations show that FiLM and cross-mode attention improve every benchmark, while a fixed-cutoff control shows that the gate automates bandwidth selection. SAO is $390\times$–$2{,}300\times$ faster than the hemodynamics solver on one CPU core and recovers its data-generation and training cost after approximately $1{,}000$ simulations. The evidence covers offline full-trajectory emulation on fixed 1D domains with the complete inlet available. New inlet families, off-grid queries, resolution transfer, and topology changes are untested.

Reproducibility Statement

Reference solutions for the Burgers benchmarks are generated by Strang operator splitting of advection and diffusion, with the advective sub-step solved by a Clawpack/PyClaw finite-volume monotonic upstream-centred (MUSCL) Godunov scheme on a fine N_x=1024 grid (downsampled to 101 points) and the diffusive sub-step solved implicitly. The hemodynamics benchmarks use the openBF 1D blood-flow solver with an explicit MUSCL shock-capturing discretization (Appendix D.1). The appendices document the data-generation settings, architecture, training hyperparameters, simulation-level split, five training seeds, baseline adaptation, and evaluation protocol. The external baselines inherit SAO's optimizer and schedule rather than being tuned per model. Exact reproduction also requires the implementation, generated split indices, waveform-augmentation configuration, and complete solver configuration; these materials are not part of the present manuscript packet. Code, checkpoints, and data-generation scripts are planned for public release.

Ethics Statement

All data used in this work are synthetic, generated by numerical simulation of viscous Burgers dynamics and 1D physiological blood-flow models; no human or animal subjects, personal data, or clinical records are involved. The hemodynamics benchmarks use physiologically inspired but non-patient-specific inlet waveforms and vessel parameters, so no privacy or consent concerns arise. Any future clinical application of SAO-style surrogates to patient-specific hemodynamic prediction would require independent validation against clinical data and appropriate regulatory and ethical review, which is outside the scope of this work.

AI Use Statement

Large-language-model-based coding assistants were used to assist with experiment code and manuscript editing under the authors' direction. The authors designed and verified all methods, experiments, and claims and reviewed all generated content.

References

David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008. doi: 10.2514/1.35374.

Max Anliker, Robert L. Rockwell, and Eric Ogden. Nonlinear analysis of flow pulses and shock waves in arteries. Part I: Derivation and properties of mathematical model. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 22(2):217–246, 1971.

Amirhossein Arzani, Jian-Xun Wang, and Roshan M D'Souza. Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Physics of Fluids*, 33(7):071905, 2021.

Ivo Babuška and Manil Suri. The p- and h-p versions of the finite element method, an overview. *Computer Methods in Applied Mechanics and Engineering*, 80(1–3):5–26, 1990.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. doi: 10.48550/arXiv.1806.01261.

Ivan Benemerito, Alessandro Melis, Antoine Wehenkel, and Alberto Marzo. openBF: an open-source finite volume 1D blood flow solver. *Physiological Measurement*, 45(12):125002, 2024.

Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

David T. Blackstock. Connection between the Fay and Fubini solutions for plane sound waves of finite amplitude. *The Journal of the Acoustical Society of America*, 39(6):1019–1026, 1966.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24924–24940, 2021.

Zhiliang Deng, Qinglu Meng, Yingguang Li, Xu Liu, Gengxiang Chen, Lu Chen, Changqing Liu, and Xiaozhong Hao. BV-NORM: A neural operator learning framework for parametric boundary value problems on complex geometric domains in engineering. *Engineering Applications of Artificial Intelligence*, 144:110109, 2025. doi: 10.1016/j.engappai.2025.110109.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. doi: 10.1137/S0895479895290954.

Constantin Greif and Karsten Urban. Decay of the Kolmogorov N-width for wave problems. *Applied Mathematics Letters*, 96:216–222, 2019. doi: 10.1016/j.aml.2019.05.013.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, volume 202, pp. 12556–12569. PMLR, 2023.

Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.

Pengzhan Jin, Shuai Meng, and Lu Lu. MIONet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022.

Georgios Kissas, Yibo Yang, Eileen Hwuang, Walter R Witschey, John A Detre, and Paris Perdikaris. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 358:112623, 2020.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Samuel Lanthaler, Roberto Molinaro, Patrik Hadorn, and Siddhartha Mishra. Nonlinear reconstruction for operator learning of PDEs with discontinuities. In *International Conference on Learning Representations*, 2023.

Randall J LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, 2002.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023b.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67398–67433, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022.

Kyle T Mandli, Aron J Ahmadia, Marsha Berger, Donna Calhoun, David L George, Yiannis Hadjimichael, David I Ketcheson, Grady I Lemoine, and Randall J LeVeque. Clawpack: building an open source ecosystem for solving hyperbolic PDEs. *PeerJ Computer Science*, 2:e68, 2016.

Alessandro Melis, Richard H Clayton, and Alberto Marzo. Bayesian sensitivity analysis of a 1D vascular model with Gaussian process emulators. *International Journal for Numerical Methods in Biomedical Engineering*, 33(12):e2882, 2017.

Luca Pegolotti, Martin R Pfaller, Natalia L Rubio, Ke Ding, Rita Brugarolas Brufau, Eric Darve, and Alison L Marsden. Learning reduced-order models for cardiovascular simulations with graph neural networks. *Computers in Biology and Medicine*, 168:107676, 2024.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018.

Martin R Pfaller, Jonathan Pham, Aekaansh Verma, Luca Pegolotti, Nathan M Wilson, David W Parker, Weiguang Yang, and Alison L Marsden. Automated generation of 0D and 1D reduced-order models of patient-specific blood flow. *International Journal for Numerical Methods in Biomedical Engineering*, 38(10):e3639, 2022.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200, 2023.

Julius Reiss, Philipp Schulze, Jörn Sesterhenn, and Volker Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344, 2018. doi: 10.1137/17M1140571.

Daniele E Schiavazzi, Alessia Baretta, Giancarlo Pennati, Tain-Yen Hsia, and Alison L Marsden. Patient-specific parameter estimation in single-ventricle lumped circulation models under uncertainty. *International Journal for Numerical Methods in Biomedical Engineering*, 33(3):e02799, 2017.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Improved architectures and training algorithms for deep operator networks. *Journal of Scientific Computing*, 92(2):35, 2022.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *International Conference on Machine Learning*, volume 235, pp. 53681–53705. PMLR, 2024.

Appendix Contents

Supplementary Material for SAO

A Model Variants – – –

A MODEL VARIANTS

Table A.1 defines every model variant named in this paper and points to the section that describes it. Names of the form `SAO-`x denote SAO with one component changed; the four ablations of §4.1 each remove one component, and the remaining SAO variants are controls used in the appendices.

Table A.1: Model variants and the section that defines each one.

Name	Definition	Defined in
SAO	Full model: Dynamic Net (four Fourier layers with cross-mode attention and a learnable mode gate) and Static Net (coefficients and FiLM parameters from the query).	§3
`SAO-noFiLM`	FiLM removed; the Static Net supplies only the coefficients, so the temporal functions depend on the inlet alone (strict factorization).	§4.1, E.6
`SAO-noAttn`	Cross-mode attention removed; the spectral mixing weights are fixed after training.	§4.1, E.3
`SAO-fixedModes`	Learnable mode gate removed; all M Fourier modes are retained.	§4.1, E.5
`SAO-MLP`	The spectral Dynamic Net replaced by an MLP with an order of magnitude fewer parameters.	§4.1, H.2
`SAO-slimMix`	Spectral mixing tensor allocated for 48 modes instead of 101.	G.2
`SAO-noMix`	Spectral mixing removed; the gate, cross-mode attention and pointwise convolution are kept.	G.2
`SAO-splitBasis`	$K/2$ private temporal functions per field added to the shared basis.	H.6
`SAO-muFiLM`	FiLM parameters computed from the physical parameters μ only, never from the query location; strictly factorized in x.	I.2
`SAO-shift`	Strict factorization plus a per-basis time shift $\tau_k(x)$.	I.3
`SAO-shiftInlet`	As `SAO-shift`, with the shifts conditioned on the inlet as well, $\tau_k(x, g)$.	I.3
`SAO-filterInlet`	As `SAO-shift`, with a waveform-conditioned complex filter per basis (one seed).	I.3
FNO-T	Fourier neural operator applied along the time axis, with the query broadcast as extra input channels.	H.1
Spatial FNO-BVP	Two-dimensional FNO over (x, t) with the inlet broadcast along x; a control.	H.5
DeepONet*	DeepONet with SAO's input assignment: the trunk net builds the temporal basis from the inlet and time, the branch net supplies the coefficients from the query.	H.1
DeepONet* (shared basis)	DeepONet* constrained to one basis shared by the three fields.	H.6
POD-DeepONet	DeepONet whose trunk is replaced by a fixed POD basis.	H.1
OFormer	Transformer operator: inlet tokens in the encoder, query-conditioned decoder.	H.1
POD+MLP	Fixed POD basis with an MLP mapping the inlet and query to its coefficients.	H.1

B DETAILED PDE FORMULATIONS

This appendix provides the full mathematical formulations of the hyperbolic conservation laws used as benchmarks in this work.

B.1 HYPERBOLIC CONSERVATION LAWS: BACKGROUND

In one spatial dimension a scalar conservation law reads

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = 0, \qquad x \in \Omega \subset \mathbb{R},\ t \in (0,T], \tag{11}$$

and a system of m conservation laws

$$\partial_t U(x,t) + \partial_x F(U(x,t)) = 0, \qquad U \in \mathbb{R}^m,\ F : \mathbb{R}^m \to \mathbb{R}^m, \tag{12}$$

with flux Jacobian $A(U) = \partial F/\partial U$. The system is hyperbolic when $A(U)$ has real eigenvalues $\lambda_1(U), \ldots, \lambda_m(U)$, the characteristic wave speeds, so information propagates at finite speed. A

nonlinear flux steepens smooth data into shocks, across which the Rankine–Hugoniot condition $s\,(u_R - u_L) = f(u_R) - f(u_L)$ fixes the shock speed s and an entropy condition selects the physical weak solution (LeVeque, 2002). The two problems studied here, the viscous Burgers equation and the 1D blood-flow system, are specified in the following subsections together with their boundary conditions and notation.

B.2 1D BURGERS EQUATION

We use the 1D viscous Burgers equation

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = \nu\, \partial_{xx} u(x,t), \qquad x \in [0,L],\ t \in (0,T], \tag{13}$$

where $u(x,t) \in \mathbb{R}$ is the scalar state and $\nu = 0.01$ is the kinematic viscosity. The flux is

$$f(u) = \frac{1}{2}u^2, \tag{14}$$

so (13) is equivalently written in quasilinear form as

$$\partial_t u + u\, \partial_x u = \nu\, \partial_{xx} u. \tag{15}$$

The characteristic speed is $a(u) = f'(u) = u$, so regions with larger u travel faster and steepen the waveform, while viscosity regularizes the resulting gradients. For multi-wave inlets, this nonlinear steepening couples frequency components, allowing us to test whether cross-mode attention improves operator learning.

B.3 1D BLOOD-FLOW (HEMODYNAMICS) SYSTEM

To assess the proposed approach on a physically grounded hyperbolic *system*, we consider a standard 1D model of blood flow in a compliant vessel. Let $A(x,t) > 0$ denote the cross-sectional area and $Q(x,t)$ the volumetric flow rate. Under the assumptions of long-wave (slender) flow, axisymmetry, and a velocity profile closure, the governing equations can be written in conservation-law form as

$$\partial_t U(x,t) + \partial_x F(U(x,t)) = S(U(x,t))\,, \qquad U = \begin{bmatrix} A \\ Q \end{bmatrix}, \tag{16}$$

where the flux is

$$F(U) = \begin{bmatrix} Q \\ \dfrac{Q^2}{A} + \mathcal{P}(A) \end{bmatrix}, \tag{17}$$

and $S(U)$ collects source terms (e.g., viscous/frictional losses and possibly gravity). Here $\mathcal{P}(A)$ is the *elastic* (pressure) contribution induced by vessel wall compliance, defined such that

$$\mathcal{P}'(A) = \frac{A}{\rho}\,\frac{dp}{dA}, \tag{18}$$

with ρ the blood density and $p = p(A)$ a chosen tube law relating pressure to area (e.g., $p(A) = p_{\text{ext}} + \beta(\sqrt{A} - \sqrt{A_0})$ for reference area A_0 and stiffness parameter β). With (17), (16) expands to

$$\partial_t A + \partial_x Q = 0, \tag{19}$$

$$\partial_t Q + \partial_x \left(\frac{Q^2}{A} + \mathcal{P}(A) \right) = S_2(A,Q), \tag{20}$$

where S_2 denotes the momentum source term (often modeled as a friction term proportional to Q/A).

Neglecting source terms, the system is hyperbolic: the flux Jacobian $A(U) = \partial F/\partial U$ has two real eigenvalues corresponding to forward- and backward-traveling waves. In particular, defining the mean velocity $v = Q/A$ and the wave speed

$$c(A) = \sqrt{\frac{A}{\rho}\,\frac{dp}{dA}}, \tag{21}$$

the characteristic speeds are $v \pm c(A)$, corresponding to forward- and backward-travelling pulse waves in compliant arteries. Such waves are central to cardiovascular assessment because their pressure and flow profiles carry information about arterial stiffness and geometry (Kissas et al., 2020). We use a standard single-vessel setting to isolate pulse-wave propagation and a bifurcating network to include wave splitting and junction interactions.

C OPERATOR LEARNING TARGET: FULL FORMULATION

With $\Omega = [0, L]$ and $t \in [0, T]$, the inlet time series $g(t) \in \mathbb{R}^{d_{\text{in}}}$ ($u(0,t)$ for Burgers, $Q(0,t)$ for blood flow), the parameter vector $\mu \in \mathbb{R}^{d_\mu}$ (wall stiffness and radius scale on H1 and H2, empty on B1–B3) and the response $y(x,t) \in \mathbb{R}^{d_{\text{out}}}$ ($y = u$ for Burgers; $y = [P, Q, A]^\top$, in that order, for blood flow), the learning target is the queried operator

$$\widehat{y}(x, \cdot) \;=\; \widehat{\mathcal{G}}_\theta(g(\cdot),\, \mu,\, x) \;\approx\; y(x, \cdot), \qquad x \in \Omega, \tag{22}$$

trained on pairs $\{(g^{(n)}, \mu^{(n)}),\, y^{(n)}\}_{n=1}^{N}$ from the reference solvers by minimising the relative L^2 discrepancy between predicted and reference time series at the probed locations (§4.1), and queried at test time at every non-inlet grid location of held-out simulations. The map is conditional on the fixed problem specification used to generate the data (initial state, non-inlet boundary conditions, geometry or topology, constitutive constants, observation interval), and because the complete inlet is available the task is offline trajectory emulation rather than causal prediction.

D DATA GENERATION

For each benchmark, we sample inlet waveforms and, where applicable, physical parameters, then compute the corresponding solution on a fixed space–time grid. These numerical solutions provide the training targets and test references.

D.1 REFERENCE SOLVERS

Burgers. For the 1D viscous Burgers equation ($\nu = 0.01$), we generate reference solutions with second-order Strang splitting: a half-step of advection, a full step of diffusion, and a second half-step of advection. We solve the inviscid sub-step with `Clawpack`/`PyClaw` using its wave-propagation formulation, Burgers Riemann solver, van Leer Total Variation Diminishing limiter, and CFL-controlled time steps. The diffusive sub-step is solved implicitly through a tridiagonal linear system. We compute solutions on $N_x = 1024$ cells and downsample them to 101 spatial points for storage (Mandli et al., 2016).

Blood flow. For the 1D blood-flow system, we use `openBF`, which evolves cross-sectional area and flow rate with an explicit finite-volume MUSCL scheme. The current implementation is second-order accurate in space and time and uses piecewise-linear reconstruction, Godunov fluxes, slope limiting, and CFL-controlled time steps. Prescribed inlet waveforms and lumped-parameter outlet models, such as Windkessel models, provide the boundary conditions. The solver reports pressure, area, and flow rate; pressure is obtained from the tube law in Appendix B.3 (Benemerito et al., 2024).

D.2 DATASET CONSTRUCTION

The Burgers datasets progress from a single sinusoidal inlet to superpositions of 10 and 20 sinusoids. The single-wave case yields smooth transport, whereas the multi-wave inputs introduce broader spectra and nonlinear interactions among frequency components. The hemodynamics datasets progress from a single compliant vessel to a Y-bifurcation, adding wave splitting and junction interactions. The following sections specify the sampling distributions, numerical grids, and sample counts.

Common data format. For each case, we generate samples of the form

$$\left(g^{(n)}(\cdot),\, \mu^{(n)}\right) \;\mapsto\; \left\{y^{(n)}(x_j, t_k)\right\}_{j=1,k=1}^{N_x, N_t}, \qquad n = 1, \ldots, N, \tag{23}$$

where $g(t)$ is the inlet time series, μ denotes system parameters (e.g., Young's modulus), y is the target field (u for Burgers; the three-component field $[P, Q, A]^\top$ for blood flow), while $\{t_k\}$ and $\{x_j\}$ are temporal and spatial grids, respectively.

Random probe protocol. For each Burgers simulation, 10 spatial probe locations are assigned at dataset construction time by sampling uniformly at random (without replacement) from the 100 non-inlet grid points $\{x_j\}_{j=1}^{100}$ (99 interior points plus the outlet), using a deterministic seed derived from the simulation index using a fixed global seed. This procedure is applied identically to all simulations before the train/val/test split, so all three subsets carry per-simulation random probe sets rather than a shared fixed grid. Training and validation use these pre-assigned 10-probe samples directly; the validation RL2 monitored for early stopping is therefore a 10-probe metric. At test time, the model is queried at all 100 non-inlet spatial locations (99 interior points plus the outlet); the reported RL2 is this dense evaluation and is the primary accuracy measure for all Burgers results. Because probe locations vary across simulations, the Static Net sees broad spatial coverage during training, preventing memorization of a fixed probe coordinate. H1 likewise uses 10 probes per simulation, while H2 uses 10 probes independently on each vessel; their spatial grids are specified below.

Case B1: Burgers, simple sine-wave inlet. For the scalar Burgers equation with flux $f(u) = \frac{1}{2}u^2$, we construct a baseline dataset using a randomized single-frequency sinusoidal inlet. Reference solutions are obtained via Strang operator splitting: the hyperbolic step uses `PyClaw`/Clawpack with a Burgers Riemann solver and the van Leer TVD limiter (`ClawSolver1D`), and the parabolic step applies implicit tridiagonal diffusion with $\nu = 0.01$. The computational domain is $\Omega = [0, 1]$ with solver grid $N_x = 1024$, downsampled to 101 uniformly spaced spatial output points. The initial condition is $u(x, 0) = 0$, so dynamics are driven entirely by the inlet. For each of the $N = 500$ simulations, the inlet waveform is

$$u(0, t) = A \sin(2\pi f t), \qquad A \sim \mathcal{U}(2.0,\, 6.0), \quad f \sim \mathcal{U}(1,\, 3)\ \text{Hz}, \tag{24}$$

with the right boundary using extrapolation (outflow) conditions. The amplitude range is set large enough that waves fully traverse the domain within the simulation horizon $T = 1.0$. The solver outputs $N_t = 201$ uniformly spaced time steps over $[0, T]$. Each simulation is assigned 10 randomly chosen interior probe locations (see Appendix D.2), yielding $N = 500 \times 10 = 5{,}000$ samples.

Case B2: Burgers, 10-wave composite inlet. To study operator learning at intermediate spectral complexity, we construct Case B2 with $N_w = 10$ sinusoidal inlet components:

$$u(0, t) = \sum_{k=1}^{10} A_k \sin\big(2\pi f_k t + \phi_k\big), \tag{25}$$

where $A_k \sim \mathcal{U}(0.2, 0.6)$, $f_k \sim \mathcal{U}(1, 3)$ Hz, and $\phi_k \sim \mathcal{U}(0, 2\pi)$. The per-wave amplitude is set so that $N_w \cdot \mathbb{E}[A_k] = 10 \times 0.4 = 4.0$ matches the expected total amplitude of B1. This equalizes the summed expected per-wave amplitude only; it does not fix the driving signal RMS, which is 0.931 for B2 versus 2.944 for B1 (§4.1). The time horizon is $T = 2.0$ to allow multi-frequency interactions to develop. All numerical settings (Strang operator splitting, `PyClaw` van Leer advection, implicit tridiagonal diffusion, $N_x = 1024$ downsampled to 101, $N_t = 201$, $\nu = 0.01$) are identical to Case B1. We generate $N = 2000$ simulations with 10 randomly assigned probe locations each (Appendix D.2), yielding 20,000 samples.

Case B3: Burgers, 20-wave composite inlet. Case B3 extends to $N_w = 20$ sinusoidal components:

$$u(0, t) = \sum_{k=1}^{20} A_k \sin\big(2\pi f_k t + \phi_k\big), \tag{26}$$

where $A_k \sim \mathcal{U}(0.1, 0.3)$, $f_k \sim \mathcal{U}(1, 3)$ Hz, and $\phi_k \sim \mathcal{U}(0, 2\pi)$. The per-wave amplitude is set so that $N_w \cdot \mathbb{E}[A_k] = 20 \times 0.2 = 4.0$ matches B1 and B2 in summed expected amplitude, though the driving signal RMS is 0.658 for B3, lower than B2's 0.931 and B1's 2.944 (§4.1). Each individual wave has amplitude $\mathbb{E}[A_k] = 0.2$, half that of B2; since Burgers nonlinearity scales as $u\,\partial_x u \propto A^2$, B3 is approximately four times less nonlinear per wave than B2, though aggregated over all sinusoids the expected total squared amplitude ($\sum_k \mathbb{E}[A_k^2]$) ratio between B2 and B3 is only $2\times$, since B3 has twice as many waves. All numerical settings are identical to B2. We generate $N = 2000$ simulations with 10 randomly assigned probe locations each (Appendix D.2), yielding 20,000 samples.

Inlet waveform generation (FFT-based augmentation). For each blood-flow network, we begin with the measured inlet waveform $y(t)$ provided by Benemerito et al. (2024). We compute its real FFT $Y = \mathcal{F}(y)$ from the samples $\{(t_i, y_i)\}_{i=1}^{N_t}$ and perturb it in two ways: a phase ramp shifts the waveform in time, and band-limited phase and gain perturbations modify its low- and mid-frequency components while preserving smoothness. The inverse FFT gives an augmented waveform $\tilde{y}(t)$. We generate $N_{\text{in}} = 1000$ inlet waveforms for each network and use them directly as boundary conditions.

Case H1: Blood flow, single vessel (aorta-like). We first consider a single compliant vessel as a baseline hemodynamics benchmark. Reference solutions are generated with `openBF`, which solves the 1D blood-flow balance laws for cross-sectional area and flow rate (Appendix B.3) using an explicit finite-volume MUSCL shock-capturing discretization with slope limiting and CFL-controlled time stepping.

For each simulation, the inlet boundary condition is prescribed by one realization from the FFT-augmented waveform ensemble described in Appendix D.2. To account for variability in arterial compliance and vessel geometry, we treat two physical parameters as inputs: the vessel-wall Young's modulus E and a lumen-radius scale factor R, sampled independently for each run as

$$E \sim \mathcal{U}(200,\, 800)\ \text{kPa}, \qquad R \sim \mathcal{U}(0.8,\, 1.2), \tag{27}$$

drawing $N = 1000$ realizations. The radius scale R multiplies the nominal lumen radius r_0 of the vessel, modulating the reference cross-sectional area $A_0 = \pi(Rr_0)^2$ and characteristic impedance. For each tuple $(g^{(n)}(\cdot), E^{(n)}, R^{(n)})$, we run one forward simulation with `openBF` and record the resulting spatio-temporal pressure, flow rate, and cross-sectional area $[P, Q, A](x, t)$ over a fixed space–time grid. Each simulation is assigned 10 randomly chosen interior probe locations (Appendix D.2), yielding $1000 \times 10 = 10{,}000$ samples total.

Case H2: Blood flow, Y-bifurcation. Case H2 extends H1 to a Y-shaped bifurcation consisting of one parent artery and two daughter vessels (3 segments total). The parent inlet is driven by one realization from the FFT-augmented waveform ensemble (Appendix D.2). Young's modulus and lumen-radius scale are sampled independently per vessel:

$$E_i \sim \mathcal{U}(200,\, 800)\ \text{kPa}, \qquad R_i \sim \mathcal{U}(0.8,\, 1.2), \qquad i = 1, 2, 3, \tag{28}$$

drawing $N = 1000$ realizations. For each simulation, we run `openBF` on the full 3-vessel network and record $[P, Q, A](x, t)$ for each vessel. Each vessel is assigned 10 randomly chosen probe locations independently, yielding $1000 \times 3 \times 10 = 30{,}000$ samples total. The Static Net receives a one-hot vessel identifier, a 17-dimensional sinusoidal positional encoding, and the E and R vectors for all vessels as context.

D.3 Dataset statistics

All datasets use a presplit 70/20/10 protocol: simulations are divided into train, validation, and test subsets at the simulation level before any samples are constructed, preventing information leakage across splits. For Burgers cases (B1/B2/B3), probe locations are assigned randomly per simulation before the split (Appendix D.2); validation RL2 used for early stopping is a 10-probe metric, while the reported RL2 is evaluated at all 100 non-inlet spatial locations (99 interior points plus the outlet) at test time only. Table D.1 summarizes the resulting sample counts.

Table D.1: Dataset statistics. Samples = simulations $\times$ probe locations. Split is 70/20/10 at the simulation level.

Case	N_{sim}	N_{samples}	Train / Val / Test sims	Train / Val / Test samples
B1	500	5,000	350 / 100 / 50	3,500 / 1,000 / 500
B2	2,000	20,000	1,400 / 400 / 200	14,000 / 4,000 / 2,000
B3	2,000	20,000	1,400 / 400 / 200	14,000 / 4,000 / 2,000
H1	1,000	10,000	700 / 200 / 100	7,000 / 2,000 / 1,000
H2	1,000	30,000	700 / 200 / 100	21,000 / 6,000 / 3,000

E ARCHITECTURE DETAILS

E.1 CONNECTION TO REDUCED-BASIS METHODS AND DEEPONET

The synthesis in Eq. (9) is structurally analogous to classical reduced-order modeling techniques such as Proper Orthogonal Decomposition (POD) and reduced-basis (RB) methods (Benner et al., 2015). In those frameworks, a high-fidelity solution is approximated as $u \approx \sum_{k=1}^{K} \alpha_k \, \phi_k$, where $\{\phi_k\}$ is a fixed orthonormal basis extracted offline from solution snapshots via SVD, and the coefficients $\{\alpha_k\}$ are recovered online via Galerkin projection onto the governing equations. SAO replaces both steps with learned neural maps: the Dynamic Net produces a set of K basis functions $\{S_{\theta,k}(g)\}$ that adapt to the inlet waveform g, while the Static Net predicts the corresponding coefficients $\{\alpha_{\theta,k}(c)\}$ from the conditioning input c, requiring no access to the governing equations at inference time.

Classical POD/RB uses a single global basis. SAO uses an *input-dependent* basis whose Dynamic Net produces a distinct set of basis functions for each inlet waveform without enlarging K. Before query-dependent FiLM is applied, the synthesis formula (9) resembles the K-term DeepONet output formula (Lu et al., 2021). The Dynamic Net plays the role of the *branch* net because it consumes the input function g, and the Static Net plays the role of the *trunk* net because it consumes the query coordinate x and parameters μ. SAO's waveform path emits an entire temporal dictionary rather than scalar coefficients. FiLM makes that dictionary query dependent, so the correspondence is structural rather than a rank guarantee for the assembled field.

E.2 SELECTION OF THE LATENT BASIS SIZE

The choice of K, the number of latent temporal functions, parallels basis selection in finite-element and reduced-basis methods. There, the solution is approximated in a K-dimensional trial space, $u_h = \sum_{k=1}^{K} \alpha_k \phi_k$, and K depends on solution regularity (h-, p-, or hp-refinement (Hughes, 2000; Babuška & Suri, 1990)) or the singular-value decay of a snapshot matrix (Benner et al., 2015). SAO has the same synthesis form, with $\phi_k \leftrightarrow S_{\theta,k}(g)$ and $\alpha_k \leftrightarrow \alpha_{\theta,k}(c)$, but learns an inlet-dependent trial space.

The selected values increase with inlet complexity: B1 uses K=6, whereas B3, with 20 input waves, uses K=20. The hemodynamics cases show that temporal smoothness alone does not determine K, because one basis represents three fields and, for H2, three vessels. We therefore select K for each case using validation performance (Table F.1).

E.3 CROSS-MODE SPECTRAL ATTENTION

Let $X_k \in \mathbb{C}^C$ denote the vector of Fourier coefficients at retained mode index k for a given sample, and let $\psi(\cdot)$ map a complex channel vector to a real feature vector by concatenating real and imaginary parts. If K_{eff} modes are retained, each becomes a token $z_k = \psi(X_k) \in \mathbb{R}^{2C}$, and the tokens are stacked into

$$Z = [\, z_0;\, z_1;\, \ldots;\, z_{K_{\text{eff}}-1} \,] \in \mathbb{R}^{K_{\text{eff}} \times 2C}, \tag{29}$$

and we compute multi-head self-attention *across the modes*

$$\text{Attn}(Z) \;=\; \text{softmax}\!\left(\frac{QK^\top}{\sqrt{d}}\right) V, \qquad Q = ZW_Q,\;\; K = ZW_K,\;\; V = ZW_V, \tag{30}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{2C \times d}$ are learned (multi-head) projections, d is the per-head dimension, and $QK^\top \in \mathbb{R}^{K_{\text{eff}} \times K_{\text{eff}}}$. The softmax is taken over the *mode* dimension: every retained frequency attends to every other, so the attention forms input-adaptive couplings *across frequencies*, while channel information enters through the token features and the projections. The attention output is projected back to complex form and added residually to the standard spectral mixing output. Algorithm 1 gives the complete pseudocode.

Algorithm 1 Spectral Attention Block (ℓ-th layer)

Require: Hidden state $h \in \mathbb{R}^{B\times C\times N_t}$; gate parameter κ_ℓ; spectral weights $W_\ell \in \mathbb{C}^{C\times C\times M}$; attention projections W_Q, W_K, W_V; FiLM parameters $\gamma_\ell, \beta_\ell \in \mathbb{R}^C$

Ensure: Updated state $h' \in \mathbb{R}^{B\times C\times N_t}$

// Gate: compute soft and integer mode counts

$K_{\text{soft}} \leftarrow \sigma(\kappa_\ell)\cdot M$

$K_{\text{eff}} \leftarrow \lfloor K_{\text{soft}} \rfloor + 1; \quad \text{frac} \leftarrow K_{\text{soft}} - \lfloor K_{\text{soft}} \rfloor$

// FFT and truncation

$H \leftarrow \text{rfft}(h)$ ▷ $\in \mathbb{C}^{B\times C\times M}$

$\widetilde{H} \leftarrow H[\,\cdot\,,\,\cdot\,, :K_{\text{eff}}]$ ▷ retain K_{eff} modes

// Attention across retained modes (inner residual)

$Z \leftarrow \psi(\widetilde{H}) \in \mathbb{R}^{B\times K_{\text{eff}}\times 2C}$ ▷ each retained mode is a token; features = real/imag of all channels

$A \leftarrow \text{MultiheadAttn}(Z;\, W_Q, W_K, W_V)$ ▷ softmax over modes ($K_{\text{eff}}\times K_{\text{eff}}$)

$\widetilde{H} \mathrel{+}= \phi(A)$ ▷ inner residual, per mode

// Spectral mixing and fractional gate

$H_{\text{mix}}[\,\cdot\,, c, k] \leftarrow \sum_{c'} W_\ell[c, c', k]\cdot \widetilde{H}[\,\cdot\,, c', k]$ ▷ for $k = 0, \ldots, K_{\text{eff}} - 1$

$H_{\text{mix}}[\,\cdot\,,\,\cdot\,, K_{\text{eff}} - 1] \times= \text{frac}$ ▷ differentiable weight on last mode

// Zero-pad, iFFT, skip connection, activation

$H_{\text{out}} \leftarrow \text{ZeroPad}(H_{\text{mix}},\, M)$ ▷ pad back to M modes

$h_{\text{spec}} \leftarrow \text{irfft}(H_{\text{out}})$

$h_{\text{skip}} \leftarrow \text{Conv1d}(h)$ ▷ parallel 1×1 pointwise mixing

$h_{\text{out}} \leftarrow \text{GELU}(h_{\text{spec}} + h_{\text{skip}})$

// FiLM conditioning (residual scale-shift)

$h' \leftarrow (1 + \gamma_\ell)\odot h_{\text{out}} + \beta_\ell$ ▷ broadcast γ_ℓ, β_ℓ over N_t

E.4 Computational complexity

Let N_t be the sequence length, C the channel width, M the number of retained Fourier modes ($M \le N_t/2 + 1$), and d the per-head attention dimension. Table E.1 compares per-layer cost between an FNO block and an SAO block.

Table E.1: Per-block computational cost for FNO and SAO. B = batch size. This local comparison does not include the number of spatial queries.

Operation	**FNO**	**SAO**
FFT / iFFT	$\mathcal{O}(CN_t \log N_t)$	same
Spectral mixing	$\mathcal{O}(C^2 M)$	same
Mode attention	—	$\mathcal{O}(C^2 K_{\text{eff}} + K_{\text{eff}}^2 C)$
Pointwise conv	$\mathcal{O}(C^2 N_t)$	same
Total	$\mathcal{O}(C^2 N_t)$	$\mathcal{O}(C^2(N_t + K_{\text{eff}}) + K_{\text{eff}}^2 C)$

The attention overhead is $\mathcal{O}(C^2 K_{\text{eff}} + K_{\text{eff}}^2 C)$, where $K_{\text{eff}} = \lfloor \sigma(\kappa_\ell)\cdot M \rfloor + 1$ is the learned mode count. For our default settings (C=64, N_t=201) the learned K_{eff} ranges from ≈2 to ≈40 across layers and benchmarks (Appendix G.1), over which the attention cost stays below the pointwise convolution (at most ≈0.35× it, at the largest observed K_{eff}), so the pointwise convolution remains the dominant term. Compared to spatial attention in prior operators (Cao, 2021; Wu et al., 2024) which scales as $\mathcal{O}(N_t^2 C)$ or $\mathcal{O}(N_t C^2)$, SAO's frequency-domain cross-mode attention is cheaper at the tested sequence length when $K_{\text{eff}} \ll N_t$. Because FiLM makes the Dynamic Net query dependent, complete-field cost also scales with the number of queries; the per-block expression above is not a claim of query-side amortization.

E.5 LEARNABLE MODE GATE

Each spectral block ℓ contains a scalar gating parameter κ_ℓ (initialized to 0) passed through a sigmoid, yielding a soft mode count $K_{\text{soft}}{=}\sigma(\kappa_\ell)\cdot M \in (0, M)$. The integer truncation $K_{\text{eff}}{=}\lfloor K_{\text{soft}}\rfloor + 1$ determines how many Fourier modes are retained in the attention and mixing operations. The fractional part $\text{frac}{=}K_{\text{soft}} - \lfloor K_{\text{soft}}\rfloor$ is used as a differentiable multiplier on the last retained mode, so that the accuracy loss propagates gradients directly through $\text{frac} \to K_{\text{soft}} \to \kappa_\ell$, unlike a hard floor that blocks this path. To encourage parsimonious spectral representations, we add a gate regularization term to the training loss:

$$\mathcal{L}_{\text{gate}} \;=\; \lambda_{\text{gate}} \sum_{\ell=1}^{L} \sigma(\kappa_\ell)^2, \tag{31}$$

where $\sigma(\cdot)$ is the sigmoid function and $L = 4$ in our Dynamic Net. For the single-output Burgers cases the full training objective is

$$\mathcal{L} \;=\; \frac{1}{|\mathcal{B}|} \sum_{(g,x,y)\in\mathcal{B}} \big\|\hat{y}(x,\cdot) - \tilde{y}\big\|^2 \;+\; \mathcal{L}_{\text{gate}}, \tag{32}$$

where $\tilde{y}$ is the normalized ground-truth output. The three-field hemodynamics models instead use the mean of the per-variable relative L^2 errors,

$$\mathcal{L} \;=\; \frac{1}{V} \sum_{v=1}^{V} \frac{\big\|\hat{y}_v - y_v\big\|_{\mathcal{B}}}{\big\|y_v\big\|_{\mathcal{B}}} \;+\; \mathcal{L}_{\text{gate}}, \qquad V = 3, \tag{33}$$

where $\|\cdot\|_{\mathcal{B}}$ is the Euclidean norm taken over the whole minibatch and all timesteps, and both $\hat{y}_v$ and y_v are in raw physical units: the prediction is un-normalized (differentiably) before the error is formed, so this is not an MSE on normalized values. Each variable thus contributes a scale-free term, and P, Q, and A, whose physical magnitudes differ by orders of magnitude, weigh comparably. In both objectives $\lambda_{\text{gate}} = 0.005$ for Burgers and $\lambda_{\text{gate}} = 0.001$ for hemodynamics. Because the cutoff changes the attention token set at integer thresholds, we do not rely on global differentiability of the selection rule as a contribution.

E.6 FILM CONDITIONING ALGORITHM

The Static Net jointly produces basis coefficients and FiLM modulation parameters from the conditioning input c. A single two-layer MLP maps $c \in \mathbb{R}^{d_c}$ to an output of dimension $V \cdot K + L \cdot 2C$, which is split into:

- $\alpha \in \mathbb{R}^{V\times K}$: coefficients for the latent temporal synthesis, one length-K vector per output variable ($V{=}1$ for Burgers, $V{=}3$ for hemodynamics). Widening this head is the only architectural change between the single- and multi-output models; the Dynamic Net and the FiLM path are identical. Sharing one basis across variables suits the hemodynamics case, where the same wave dynamics couple P, Q, and A and so give them common temporal structure.
- $(\gamma_\ell, \beta_\ell) \in \mathbb{R}^C$ for $\ell = 1, \ldots, L$: scale and shift vectors for each spectral layer.

The last linear layer's FiLM columns are zero-initialized, so $\gamma_\ell = \beta_\ell = \mathbf{0}$ at the start of training, making the modulation an identity map initially. Algorithm 2 gives the full forward pass.

F TRAINING DETAILS

F.1 NORMALIZATION

For the Burgers cases, all time series are normalized using per-feature z-score statistics computed from the training split. Specifically, the mean $\bar{\mu}$ and standard deviation $\bar{\sigma}$ are estimated by concatenating the inlet waveforms and output time series from training samples along the sample axis, yielding per-timestep statistics over the joint distribution of inputs and outputs. Both inlet g and

Algorithm 2 SAO forward pass with FiLM conditioning

Require: inlet waveform $g \in \mathbb{R}^{N_t}$, conditioning input $c \in \mathbb{R}^{d_c}$
Ensure: prediction $\hat{y} \in \mathbb{R}^{V \times N_t}$

$[\alpha;\ \mathrm{film}] \leftarrow \mathrm{MLP}(c)$ ▷ $\alpha \in \mathbb{R}^{V \times K}$; $\mathrm{film} \in \mathbb{R}^{L \times 2 \times C}$
$\gamma_\ell \leftarrow \mathrm{film}[\ell, 0, :], \quad \beta_\ell \leftarrow \mathrm{film}[\ell, 1, :]$ ▷ for $\ell = 1, \ldots, L$
$h \leftarrow \mathrm{Lift}(g)$ ▷ $(B, 1, N_t) \to (B, C, N_t)$
for $\ell = 1, \ldots, L$ **do**
    $h \leftarrow \mathrm{SpectralBlock}_\ell(h;\ \kappa_\ell, \gamma_\ell, \beta_\ell)$ ▷ gate + FFT + attn + mix + iFFT + FiLM (Alg. 1)
end for
$S \leftarrow \mathrm{Project}(h) \in \mathbb{R}^{K \times N_t}$ ▷ two linear layers
$\hat{y}_v \leftarrow \sum_{k=1}^{K} \alpha_{v,k}\, S_k$ **for** $v = 1, \ldots, V$ ▷ latent temporal synthesis, shared functions S

output y are normalized as $\tilde{z} = (z - \bar{\mu})/\bar{\sigma}$. The spatial query coordinate x is scaled to $[0, 1]$. The statistics $(\bar{\mu}, \bar{\sigma})$ are stored alongside the model checkpoint and applied at inference time to recover physical-unit predictions.

For the three-field hemodynamics models the inlet is z-scored per timestep with its own training statistics, and the output statistics are computed *per variable*: variable v has a per-timestep mean curve $\bar{\mu}_v \in \mathbb{R}^{N_t}$, which is subtracted, and a single scalar scale $\bar{\sigma}_v$ (the standard deviation of that variable over all training samples and timesteps), by which the result is divided. The stored tensors are therefore $\bar{\mu} \in \mathbb{R}^{V \times N_t}$ and $\bar{\sigma} \in \mathbb{R}^{V \times 1}$. On H2 both are computed per (variable, vessel) pair. As with Burgers they are stored with the checkpoint and applied at inference time; during training the prediction is un-normalized with them before the loss (33) is formed.

F.2 Optimizer and learning rate schedule

We use AdamW (Loshchilov & Hutter, 2019) with base learning rate 3×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay 10^{-4}. The learning rate schedule consists of a linear warm-up phase over the first 500 steps (ramping from $0.1\times$ to $1\times$ base rate) followed by cosine annealing to a minimum of 10^{-6}. Gradients are clipped to a maximum ℓ_2 norm of 2.0 at each step.

F.3 Data splits and early stopping

All results (SAO, the ablations and the external baselines) average the same five seeds $\{42, 123, 7, 0, 1\}$. Data are split $70/20/10$ into training, validation and test at the simulation level with a fixed seed (per-case counts in Appendix D.3); batches hold 64 samples; the epoch caps are 2000 on B1–B3, 3000 on H1 and 4500 on H2 (Table F.1). Early stopping uses a patience of 500 epochs, a 200-epoch minimum and a 10^{-3} relative-improvement threshold on the validation relative L^2, and the checkpoint with the lowest validation loss is kept. On B1–B3 the mean best epoch is 1736–1904, so the 2000-epoch cap rather than the patience criterion ended every run; on H1 and H2 (Appendix G.2) the patience criterion fired on two of the five H1 seeds and four of the five H2 seeds, and the remaining seeds ran to their cap, so for those the reported accuracy is a conservative estimate.

F.4 Hyperparameter summary

Table F.1 summarizes the key architectural and training hyperparameters for all cases.

Table F.1: SAO hyperparameters per experimental case. The H1 and H2 columns describe the three-field (P, Q, A) models; the strictly factorized models of Appendix I.2 use the same configuration.

Hyperparameter	**Symbol**	**B1**	**B2**	**B3**	**H1**	**H2**
Basis functions	K	6	10	20	10	20
Output variables	V	1	1	1	3	3
Channel width	C	64	64	64	64	64
Spectral blocks	L	4	4	4	4	4
Max Fourier modes	M	101	101	101	101	101
Attention heads	H	4	4	4	4	4
Per-head dim	d	32	32	32	32	32
Attn dropout		0.3	0.3	0.3	0.3	0.3
Batch size				64		
Max epochs		2000	2000	2000	3000	4500
Early stop patience		500	500	500	500	500
Learning rate				3×10^{-4}		
Weight decay				10^{-4}		
Warmup steps				500		
Gradient clip				2.0		
λ_{gate}		0.005	0.005	0.005	0.001	0.001

G Additional experiments

G.1 Learned spectral cutoffs

Table G.1 reports the learned $K_{\text{soft}} = \sigma(\kappa_\ell) \times M$ values for each spectral layer ($M = 101$ modes) across all benchmarks, averaged over five seeds.

Table G.1: Learned K_{soft} per spectral layer ($M = 101$), 5-seed means; $\lambda_{\text{gate}} = 0.005$ on Burgers and 0.001 on H1 and H2. The H1 and H2 rows are the three-field checkpoints (K=10 and K=20).

Case	L_1	L_2	L_3	L_4	**Avg**
B1 (single-wave)	4.2	5.2	5.9	7.3	5.7
B2 (10-wave)	1.8	4.2	6.0	10.0	5.5
B3 (20-wave)	1.8	2.8	5.0	8.0	4.4
H1 (single artery)	12.6	18.4	22.7	40.1	23.4
H2 (Y-bifurcation)	15.7	16.2	18.2	22.1	18.0

A coarse-to-fine hierarchy emerges uniformly: L_1 always retains the fewest modes, acting as an aggressive low-pass filter, while L_4 retains the most. For Burgers, the gate retains only ∼4–6% of the full spectrum on average (4–10 modes). The three-field hemodynamics models keep a wider band at every layer, which is what one would expect of a single shared basis that has to serve three physical fields: H1 rises from $L_1 \approx 13$ to $L_4 \approx 40$ modes (avg $\approx 23\%$ of the spectrum) and H2 from $L_1 \approx 16$ to $L_4 \approx 22$ (avg $\approx 18\%$). The wider H1 band is consistent with the multi-harmonic structure of physiological aortic waveforms. The gate discards >80% of Fourier coefficients on every Burgers case and on H2, and ≈77% on H1.

Fixed-cutoff control. To separate the low-pass effect from any benefit of *learning* the cutoff, we retrained SAO on B1–B3 (five seeds) with the gate replaced by a static per-layer cutoff fixed at the seed-averaged integer mode count retained by the learned gate (Table G.1), rounded to the nearest integer. The fixed-cutoff variant reaches RL2 $0.85/0.93/0.85\%$ on B1–B3, against the learnable gate's $0.93/1.03/0.90\%$ (Table 4.1); both are far below `SAO-fixedModes`'s $3.62/2.06/1.75\%$, which retains every mode. A well-chosen cutoff therefore matches or slightly beats the gate, and

the gate's contribution is the automatic selection of that cutoff and the resulting interpretable mode hierarchy rather than accuracy.

G.2 Computational efficiency

Hardware. Burgers data generation runs on one CPU core (AMD Threadripper PRO 5975WX) and Burgers training on one NVIDIA RTX 6000 Ada GPU; the openBF hemodynamics simulations run serially on CPU cores of an HPC cluster, and the hemodynamics models train and infer on one NVIDIA H100 NVL GPU. The solver wall times of Tables 4.5 and G.3 and SAO's single-thread CPU inference are measured on one core of the hemodynamics data-generation node (Intel Xeon Gold 6548Y+): the Burgers solver over 20 test simulations per case at the data-generation grid (N_x=1024), openBF over the 1000 simulations of each data set. All GPU inference times in Tables 4.5 and G.3 are from one NVIDIA H100 NVL of the same node: B1–B3 in one timing pass covering SAO and every baseline, with a warm-up pass excluded; H1–H2 from the test runs.

Resource cost. Table G.2 lists, per benchmark, the number of training samples, the allocated and active parameter counts, the early-stopping epoch and the wall-clock training time of SAO over the five seeds. On B1–B3 the 2000-epoch cap ended every run; on H1 and H2 the patience criterion fired on two and four of the five seeds respectively, and the wide H2 spread reflects one seed whose best checkpoint came at epoch 1304 and one at 4256. Wall times are per run on one GPU (a workstation for B1–B3, a shared cluster for H1 and H2) and are reported for cost context, not as a controlled benchmark.

Table G.2: SAO resource cost per benchmark, mean ± sd over five seeds. Training samples are simulations × 10 probes × the 70% split; active parameters exclude the spectral weights above each layer's gate cutoff (Appendix G.1). Epoch caps: 2000 (B1–B3), 3000 (H1), 4500 (H2) for every model.

Case	Training samples	Params (alloc.)	Active params	Best epoch	Train wall (min)
B1 (single-wave)	3,500	3.67M	0.56M	1904±28	40.8±0.2
B2 (10-wave)	14,000	3.67M	0.57M	1736±157	134.7±0.8
B3 (20-wave)	14,000	3.67M	0.53M	1879±88	135.1±0.6
H1 (single artery)	7,000	3.67M	1.15M	2461±205	69.3±4.8
H2 (Y-bifurcation)	21,000	3.68M	0.98M	2755±1097	221.9±68.0

Parameter efficiency. The spectral mixing tensor accounts for 90% of SAO's parameters (827K per layer, compared with 66K for attention and 4K for the pointwise convolution), although each forward pass uses only the weights below the learned cutoff K (Appendix G.1). Allocating the tensor for 48 modes (`SAO-slimMix`) covers the largest learned cutoff of 41, reduces the parameter count by 47%, and leaves accuracy unchanged within seed variation on H1 and H2 (three-field means: 0.106% versus 0.097%, paired p=0.13; 0.104% versus 0.107%, p=0.62). The learned cutoff in each layer remains within one mode of the reference model's cutoff. Removing spectral mixing entirely (`SAO-noMix`) retains the gate, cross-mode attention, and pointwise convolution but increases error by $7.1\times$ on H1 and $1.7\times$ on H2. Thus, the allocated mixing tensor can be reduced substantially, but frequency-dependent channel mixing remains important for accuracy. We match the external baselines to SAO's 3.67M allocated parameters rather than its 0.53–1.15M active parameters (Table G.2), giving the baselines the larger parameter budget.

End-to-end timing. Table G.3 gives the reference-solver time per simulation, SAO's per-query and per-simulation inference cost, the resulting speedup, and, on H1 and H2, the per-query cost of the four closest alternative operators (defined in Appendix H.1) measured on the same GPU. Per-simulation inference is the measured per-query cost times the number of dense query rows in one simulation (100 on B1–B3, 101 on H1 and 303 on H2, each query returning all output fields at once). The solver-to-surrogate ratios in Table G.3 are CPU-to-GPU comparisons; Table 4.5 adds the same-core ratios and Table G.4 the amortized cost. On H1 and H2, SAO is 5.7–$8.2\times$ cheaper per query than OFormer and two orders of magnitude cheaper than DeepONet*, whose trunk re-reads the whole inlet at every output time, but it is not the cheapest model: POD-DeepONet and

POD+MLP, which evaluate a frozen basis (0.017 and 0.008 ms on H1 and H2), and the `SAO-MLP` ablation, with an order of magnitude fewer parameters (0.010 and 0.006 ms), are faster. On B1–B3, timed in the same pass, SAO is 5.6–7.6× cheaper per query than OFormer and 1.8–2.3× cheaper than DeepONet*, FNO-T is within 1.2× of SAO, and the two POD surrogates are again faster.

Table G.3: End-to-end timing per benchmark. Solver time is the reference solver's wall time per simulation on one Xeon Gold 6548Y+ core (Burgers: 20 test simulations per case at the data-generation grid N_x=1024; openBF: mean ± sd over 1000 simulations). SAO per-query cost is the 5-seed mean on one H100 (B1–B3 in one timing pass with a warm-up pass excluded, H1–H2 from the test runs), per-simulation cost multiplies it by the dense query rows, and speedup is solver over SAO; the H1 per-query spread comes from one seed. Alternative-model costs are timed the same way on the same GPU; *DeepONet is the inlet-to-basis assignment, whose trunk re-reads the inlet at every output time.

		SAO (ms)			Alternatives (ms/query)			
Case	**Solver time**	**query**	**sim.**	**Speedup**	**FNO-T**	**DeepONet***	**POD-DeepONet**	**OFormer**
B1	2.35±0.11 s	0.018	1.84	∼1,300×	0.021	0.033	0.012	0.102
B2	2.28±0.12 s	0.013	1.32	∼1,700×	0.015	0.030	0.006	0.101
B3	2.33±0.13 s	0.014	1.44	∼1,600×	0.016	0.031	0.007	0.102
H1	0.75±0.05 min	0.019	1.91	∼24,000×	0.036	1.730	0.017	0.108
H2	12.4±1.0 min	0.012	3.77	∼200,000×	0.023	1.920	0.008	0.102

Same-device cost and amortization. The solver-to-GPU ratios above compare devices. Timed on one core of the solver node (Xeon Gold 6548Y+, single PyTorch thread), SAO's per-simulation inference takes 91, 94, 91, 115 and 320 ms on B1–B3, H1 and H2, so on one CPU core it is 24–26× faster than the Burgers solver and 390× and 2,300× faster than openBF. On an older Xeon E5-2660 v4 the same inference takes 193–656 ms with one thread and 18–64 ms with 28 threads. Table G.4 then charges the SAO route in full, data generation on CPU plus training on one GPU plus inference, against solving every query. The routes break even at $N^* = (T_{\mathrm{data}} + T_{\mathrm{train}})/(t_{\mathrm{solver}} - t_{\mathrm{SAO}})$ additional simulations: about 1,000 on H1 and H2, where the training data are over 90% of the SAO cost, and 1,540 to 5,540 on B1–B3, where a simulation costs 2.3 s and training is the larger share. Because training time is GPU time, the table also gives N^* with one GPU-hour charged as eight CPU-hours. At 10^5 queries the SAO route saves 94–99% of the cost on every benchmark; at 10^3 it saves nothing anywhere. A cheaper solver is not an alternative at SAO's accuracy: on H2, halving the grid to 99 nodes cuts the solver time to 5.5 min per simulation but raises its flow-rate error against the 399-node solution to 0.346%, above SAO's 0.182% (Appendix H.8).

Table G.4: Amortized cost of the SAO route: T_{data} solver time for the training set (CPU-hours), T_{train} mean SAO training time (GPU-hours), t_{solver} and t_{SAO} the per-simulation costs of Table 4.5 (SAO on the GPU). N^* is the number of additional simulations beyond which the SAO route is cheaper, charging one GPU-hour as one or as eight CPU-hours; the last columns give the cost saved at 10^4 and 10^5 queries.

Case	T_{data}	T_{train}	t_{solver}	t_{SAO}	N^* (1:1)	N^* (1:8)	saved, 10^4	saved, 10^5
B1	0.33	0.68	2.35 s	1.84 ms	1,540	8,820	85%	98%
B2	1.27	2.24	2.28 s	1.32 ms	5,540	30,300	45%	94%
B3	1.29	2.25	2.33 s	1.44 ms	5,480	29,900	45%	94%
H1	12.42	1.16	44.7 s	1.91 ms	1,090	1,740	89%	99%
H2	206.11	3.70	742 s	3.77 ms	1,020	1,140	90%	99%

G.3 Model comparison on the reference simulations

Figures G.1, G.2 and G.3 complete Figure 4.1 with H1, the H2 daughter vessels, and B1 and B3, respectively, in the same layout. The reference simulation of each case is the test simulation with median SAO error (flow-rate error on the hemodynamics cases) on the stored test metrics of seed s_{42}. Each figure shows the ground truth and SAO's prediction on that simulation, then the signed

error of every model, prediction minus truth as a percentage of the peak field value, with the model's RL2 on that simulation above each panel. In every block x runs left to right and time upward. Blue and orange denote negative and positive error, respectively, and near-zero error is neutral. The colour scale of each figure spans three times the 99.5th percentile of the smallest model error, so the larger errors saturate. The annotated values are for the plotted simulation only; five-seed test-set means are in Tables 4.1 and 4.2. Ablations occupy the top row of each block and external baselines the bottom row. On H2 the errors are normalized by the peak flow rate of the whole network, so the three vessels share one scale.

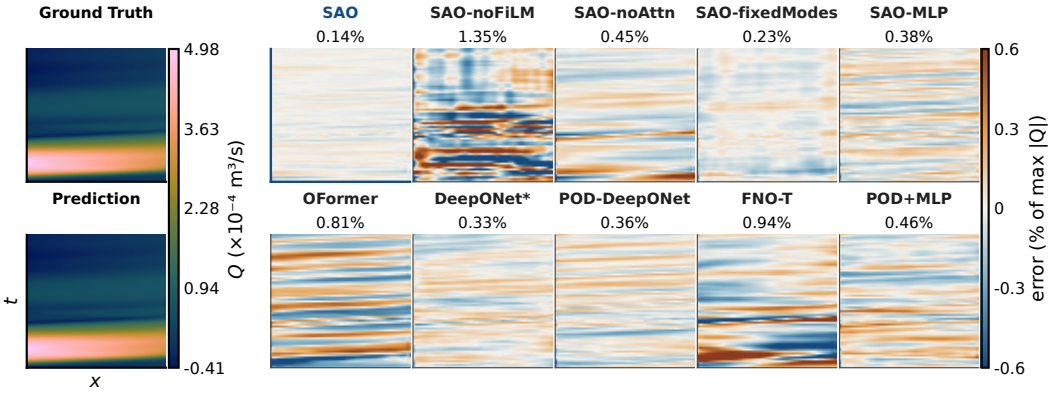

Figure G.1: **H1 reference simulation** (test simulation 22, seed s_{42}): flow rate Q, with the same model ordering and error convention as Figure 4.1.

G.4 Paired-seed comparisons for hemodynamics

Table 4.2 reports the three-field means, Table G.5 the seed variation, and Table 4.3 the physics-based error measures. Paired t-tests over the same five seeds qualify the numerical ordering. The four ablations each remove one component of SAO (FiLM, cross-mode attention, the learnable mode gate, or the spectral Dynamic Net); Appendix A defines every variant. Against `SAO-fixedModes`, the only robust accuracy difference is on the Q field, on both cases (p=0.018 on H1, p=0.006 on H2). On the three-field mean the two are indistinguishable (p=0.60 on H1, p=0.59 on H2, with mixed per-seed signs), and although `SAO-fixedModes` attains the lower mean pressure error on both cases, that difference is within seed variation as well (p=0.083, p=0.201). Against DeepONet*, SAO wins every field on H1 (Q $p < 0.0001$, mean p=0.0001, P p=0.0047, A p=0.0034) and wins the Q field ($p < 0.0001$) and the three-field mean (p=0.0013) on H2, where the P (p=0.13) and A (p=0.91) differences are within seed variation. The gaps to `SAO-noAttn`, `SAO-noFiLM`, `SAO-MLP` and to the other four external baselines run from 2.3× to 8.8× on Q and need no such hedging. No single ablation ordering holds across the two cases: `SAO-noAttn` is the milder of the two on H1 (0.394 against `SAO-noFiLM`'s 1.010 on Q) and the harsher on H2 (0.563 against 0.457).

G.5 Seed-to-seed variation

Table G.5 reports the seed-to-seed spread behind every row of Tables 4.1 and 4.2: for each cell the sample standard deviation and the range of the five per-seed values, seeds $\{42, 123, 7, 0, 1\}$, at the rounding of the main table. Every row (SAO, the four ablations and the five external baselines) is evaluated over the same five seeds; no row is averaged over a smaller set. On H1 and H2 the columns are the Q field and the three-field mean of the three-field models. Reported $\pm$ is the sample standard deviation (Bessel-corrected, $\mathrm{ddof} = 1$) across the same 5 seeds; this estimator is used uniformly wherever this paper reports a spread.

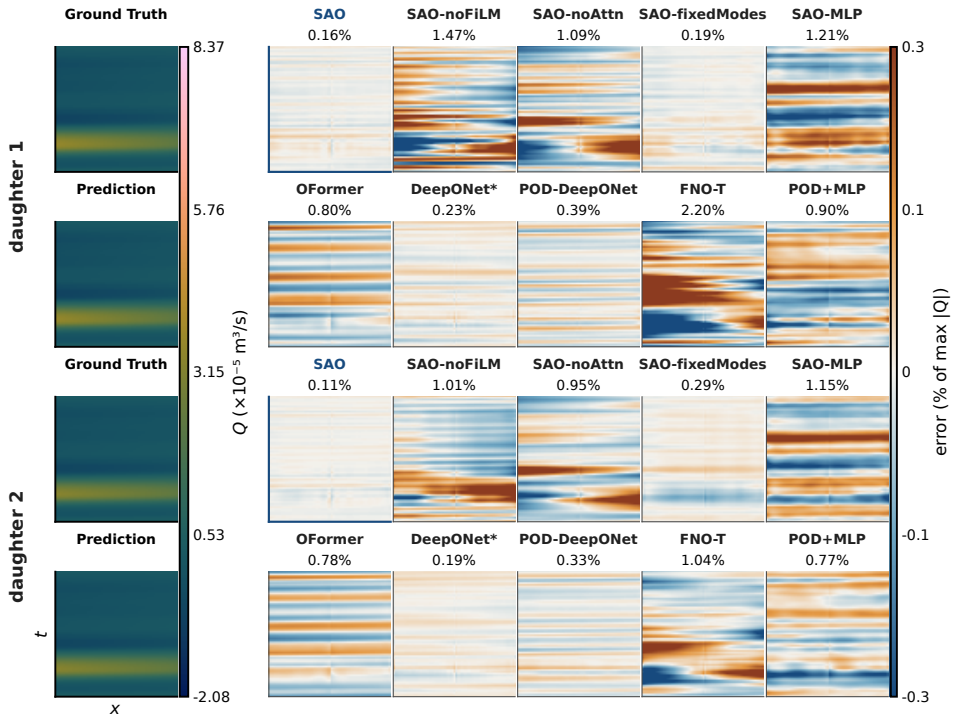

Figure G.2: **H2 reference simulation** (test simulation 97, seed s_{42}): flow rate on the two daughter vessels, using the whole-network field and error scales also used for the H2-parent block in Figure 4.1.

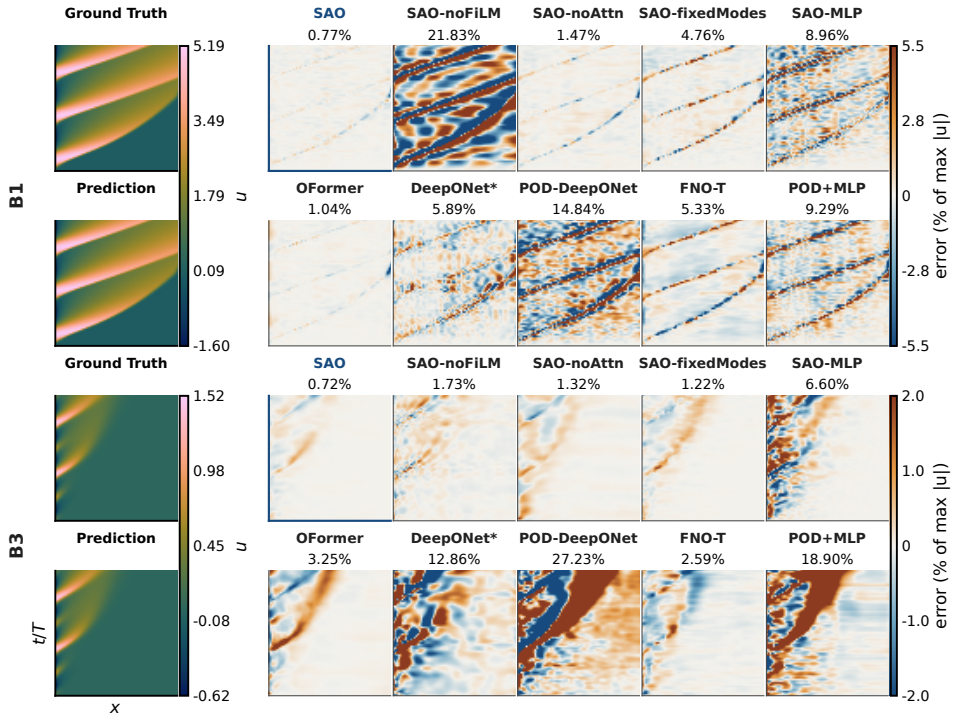

Figure G.3: **Additional Burgers reference simulations** (seed s_{42}): B1 (test simulation 28) and B3 (70), field u over normalized time t/T, each with its own scales.

Table G.5: Seed-to-seed spread of the dense test RL2 (%) of Tables 4.1 and 4.2: sample sd over the five seeds, with the per-seed range beneath. *as in Table 4.2.

Variant	B1	B2	B3
SAO (ours)	0.03 (0.90–0.98)	0.04 (0.98–1.07)	0.02 (0.87–0.92)
SAO−noFiLM	0.09 (22.44–22.66)	0.10 (11.34–11.60)	0.24 (2.18–2.77)
SAO−noAttn	0.09 (1.20–1.43)	0.34 (3.63–4.42)	0.23 (3.74–4.31)
SAO−fixedModes	0.20 (3.47–3.93)	0.03 (2.03–2.09)	0.06 (1.69–1.86)
SAO−MLP	0.48 (6.42–7.66)	0.14 (8.43–8.78)	0.37 (7.63–8.55)
OFormer	0.03 (0.90–0.96)	0.11 (3.23–3.52)	0.12 (2.95–3.27)
DeepONet*	0.20 (6.28–6.75)	0.37 (14.54–15.49)	0.44 (11.44–12.47)
POD-DeepONet	0.13 (15.41–15.75)	0.61 (24.45–25.92)	0.41 (18.66–19.73)
FNO-T	0.20 (4.08–4.57)	0.17 (5.04–5.48)	0.25 (4.86–5.47)
POD+MLP	0.10 (6.93–7.19)	0.35 (13.07–14.06)	0.76 (13.18–15.04)

Variant	H1 Q	H2 Q	H1 Mean	H2 Mean
SAO (ours)	0.005 (0.155–0.167)	0.007 (0.118–0.136)	0.011 (0.082–0.111)	0.012 (0.094–0.122)
SAO−noFiLM	0.043 (0.970–1.067)	0.015 (0.439–0.473)	0.013 (0.528–0.556)	0.007 (0.358–0.376)
SAO−noAttn	0.009 (0.383–0.408)	0.012 (0.544–0.579)	0.005 (0.199–0.211)	0.004 (0.294–0.304)
SAO−fixedModes	0.018 (0.178–0.220)	0.007 (0.143–0.161)	0.011 (0.092–0.120)	0.008 (0.101–0.122)
SAO−MLP	0.014 (0.385–0.419)	0.019 (0.958–1.002)	0.016 (0.210–0.249)	0.011 (0.486–0.514)
OFormer	0.199 (0.984–1.486)	0.133 (0.993–1.302)	0.079 (0.478–0.677)	0.061 (0.463–0.612)
DeepONet*	0.005 (0.334–0.346)	0.008 (0.232–0.253)	0.003 (0.184–0.191)	0.005 (0.150–0.161)
POD-DeepONet	0.008 (0.355–0.372)	0.007 (0.288–0.306)	0.003 (0.233–0.240)	0.004 (0.214–0.224)
FNO-T	0.011 (0.688–0.713)	0.037 (0.791–0.888)	0.005 (0.343–0.354)	0.013 (0.451–0.484)
POD+MLP	0.001 (0.468–0.472)	0.027 (0.651–0.721)	0.002 (0.311–0.315)	0.015 (0.451–0.489)

H Evaluation protocols and additional analyses

The following sections describe the baselines, additional controls, physics-based diagnostics, and parameter-extrapolation tests referenced in §4.2.

H.1 Baseline interface adaptation

All five external baselines are re-implemented behind SAO's own interface, $(\mathbf{g}, c) \mapsto \widehat{y} \in \mathbb{R}^{V \times N_t}$, where $\mathbf{g} \in \mathbb{R}^{N_t}$ is the z-scored inlet waveform and $c \in \mathbb{R}^{d_c}$ is the z-scored query vector: the sinusoidal position encoding $\mathbf{p}(x)$ of Eq. (5) for B1–B3 ($d_c = 17$); $[\mathbf{p}(x), E, R]$ for H1 ($d_c = 19$); and a three-way vessel one-hot concatenated with $\mathbf{p}(x)$ and the three per-vessel E and R values for H2 ($d_c = 26$). On Burgers V=1; on H1 and H2 V=3 and the baselines predict the same (P, Q, A) triple as SAO. The adaptation is the minimal one in every case: DeepONet's branch and trunk emit $p \cdot V$ features instead of p, FNO's final 1×1 convolution emits V channels instead of one, OFormer's output head becomes $\mathrm{Linear}(d, V)$, and the two POD models fit one basis per variable. The per-variable normalization of Appendix F.1 is reused verbatim. Every model therefore consumes the same (g, x, μ) triple and emits the full response time series in a single forward pass

with no temporal rollout; the models differ only in how the boundary signal and the query are combined. The evaluation grid is built by a shared routine that mirrors each case's SAO test script exactly, so baseline and SAO dense RL2 are computed on identical query sets.

POD+MLP. A fixed temporal basis $\Phi \in \mathbb{R}^{r \times N_t}$ and mean $\mu_\Phi \in \mathbb{R}^{N_t}$ are computed once by a truncated SVD of the z-scored training responses, retaining the smallest rank that reaches 99.99% cumulative energy (capped at $r \leq 64$, so the fixed-basis ceiling is not artificially low). A depth-3 GELU MLP maps the concatenation $[\mathbf{g} \,|\, c] \in \mathbb{R}^{N_t + d_c}$ to r coefficients, and the prediction is $\widehat{y} = \text{coeff} \cdot \Phi + \mu_\Phi$ (one basis, and one coefficient block, per output variable). The basis is frozen after the SVD, so the inlet waveform and the query influence the response only through the coefficients.

DeepONet (inlet-to-basis). Given SAO's input assignment rather than the published one: a depth-3 branch MLP maps the query vector c to p coefficients per variable, and a depth-3 trunk MLP maps, at each of the N_t time points, the concatenation of a 17-dimensional sinusoidal encoding of $t \in [0, 1]$ with the raw inlet vector $\mathbf{g} \in \mathbb{R}^{N_t}$ to the basis, combined by the inner product over the p features plus a learned bias in $\mathbb{R}^{V \times N_t}$. The inlet therefore generates the temporal dictionary, as in SAO, and the query selects from it. The published assignment, with the inlet on the branch and the output coordinate on the trunk, is not reported here. Note that the trunk must re-read the whole inlet at each of the N_t output times, which is why this model's per-query inference cost in Table G.3 is two orders of magnitude above the others'.

POD-DeepONet. A branch MLP on $\mathbf{g}$ and a trunk MLP on c (no time input) each produce a p-vector; these are multiplied elementwise in the standard DeepONet product, and a linear layer maps the result to r coefficients over the same fixed POD basis used by POD+MLP. All temporal structure resides in Φ, and the network predicts only how to weight it.

Temporally reformulated FNO (FNO-T). The query is broadcast along the time axis and concatenated to the inlet as extra channels, giving a $(1 + d_c)$-channel signal of length N_t. A 1×1 convolution lifts this signal to width C. Four blocks apply a spectral convolution along time (real FFT, complex weights on the first 32 retained modes, inverse FFT) in parallel with a 1×1 convolution, followed by a GELU. A two-layer 1×1-convolution head projects the result back to V channels. The boundary-to-probe map is realized through channel concatenation. FNO-T serves as a temporally reformulated spectral control, while Spatial FNO-BVP in Appendix H.5 provides the native spatial comparison.

OFormer. The encoder tokenizes the inlet as N_t tokens $[\, g(t_k) \,|\, \text{encoding of } t_k]$ and applies four Galerkin blocks (softmax-free linear attention with LayerNorm applied to the key and value heads) each followed by a feed-forward sublayer. The decoder forms N_t query tokens $[\,\text{encoding of } t_k \,|\, c\,]$ and applies two cross-attention blocks against the encoder memory, after which a LayerNorm/MLP head emits the V predicted values at each time. The query location and physical parameters enter only through the decoder query tokens.

H.2 Capacity matching

For each case the parameter target is the trained SAO model's own parameter count, read from the SAO metrics record for H1 and H2 and reconstructed from the case configuration and counted directly for B1–B3. Each baseline exposes exactly one integer size knob in which the parameter count is monotone: the hidden width for POD+MLP (depth fixed at 3); the shared branch/trunk width, which also sets p, for DeepONet and POD-DeepONet; the channel width for FNO (32 modes and 4 layers fixed); and the model dimension for OFormer (rounded down to a multiple of the 4 attention heads, depth 4 fixed). A binary search over that knob selects the value minimizing $|\text{params} - \text{target}|$, and the builder asserts a relative gap of at most 5% before any training begins; a separate audit script rebuilds every (model, case) pair and reports the worst gap over the whole grid. The targets are the three-field SAO models' own counts, $3{,}672{,}620$ on H1 and $3{,}678{,}676$ on H2, and $3{,}669{,}784$ on B1–B3 (the B2 count, used as the common Burgers target; the B1 and B3 models have $3{,}668{,}752$ and $3{,}672{,}364$ parameters, within 0.1% of it). The realized gaps are far inside the contract: on H1 the five baselines land between $3{,}662{,}755$ and $3{,}674{,}299$ parameters and on H2 between $3{,}664{,}467$ and $3{,}680{,}621$. The worst relative gap over all 5×5 (model, case) pairs

is 0.39% (FNO-T on H2), and the worst on the hemodynamics cases alone is the same 0.39%; this is the $\sim 0.4\%$ figure quoted in Tables 4.1 and 4.2. Only this size knob is model-specific; no other architectural hyperparameter was searched. `SAO-MLP` (the ablation with an MLP in place of the spectral Dynamic Net) is deliberately not capacity-matched: a capacity-matched version (3.68M parameters, three seeds) overfits on H1, its best validation epoch falling from $\approx 2{,}850$ to ≈ 100 and its three-field test RL2 rising from 0.22% to 0.40%, while H2 is unchanged (0.48% against 0.50%), so the ablation tables use the smaller model.

H.3 Baseline training protocol

The baselines inherit SAO's optimizer and schedule verbatim; they were not tuned per model. The baseline trainer uses the same hyperparameter block as the SAO trainer: batch size 64, the same per-case epoch budget as SAO (2000 on B1–B3, 3000 on H1, 4500 on H2; Table F.1), AdamW with learning rate 3×10^{-4}, $\beta = (0.9, 0.95)$, weight decay 10^{-4}, 500 warmup steps converted to whole epochs followed by cosine annealing to $\eta_{\min} = 10^{-6}$, gradient clipping at 2.0, and early stopping with patience 500, a 200-epoch minimum, and a 10^{-3} relative-improvement threshold on the 10-probe validation RL2. It also reuses SAO's data pipeline, normalization, simulation-level split, and checkpoint-on-best-validation-RL2 rule, and on H1 and H2 it uses SAO's three-field objective, the mean of the per-variable raw-space relative L^2 errors of Eq. (33). The only deliberate differences are that the gate-sparsity term is dropped (it has no counterpart in models without a mode gate; on Burgers this leaves plain MSE) and the width knob of §H.2. The sweep script enumerates a fixed grid of models $\times$ cases $\times$ seeds and performs no per-model hyperparameter search.

Every model in Tables 4.1 and 4.2, SAO included, was then evaluated in one uniform single-GPU pass with a shared evaluation routine, so no cell differs from another by device or code path; SAO and baseline prediction arrays were verified to have identical shapes and bit-identical ground truth. The baseline numbers in Tables 4.1 and 4.2 are the accuracies these architectures reach under one common training recipe at matched capacity, not per-model-tuned lower bounds. A model-specific learning rate, schedule, or depth/width search could improve any of them, and the comparison should be read as matched-budget and matched-recipe rather than as each baseline's best attainable accuracy.

H.4 Learning rate sensitivity

The shared recipe fixes one peak learning rate, 3×10^{-4}, for every model. To test whether that choice drives the gaps in Tables 4.1 and 4.2, we searched eight model–case pairs: FNO-T on B2 and H2, OFormer on B2, DeepONet* on H2, SAO on B2 and H2, and OFormer and SAO on B1. The B1 comparison was added because it is the only benchmark where a baseline matches SAO under the shared recipe. The grid was $\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$ with three seeds per arm and every other setting unchanged, including epoch caps, patience, and capacity. The rate was selected by mean validation RL2 only. When the selected rate sat on a grid edge, one arm one half-decade further out was added and the selection repeated, at most twice, after which the search stopped regardless of the outcome. SAO was searched with the identical grid. In total 87 runs used ≈ 140 GPU-hours.

Table H.1 lists the shared-recipe value next to the validation-selected arm. Tuning preserves the pairwise ordering on B2 and H2 but moves OFormer ahead of SAO on B1. FNO-T on B2 improves from 5.19 to 4.15%; the two H2 baselines move by at most 3% of their value; OFormer on B2 is unchanged on average and becomes seed-unstable at the rates validation prefers ($3.25 \pm 0.49\%$ at the selected 10^{-2}, $4.44 \pm 1.26\%$ at 10^{-3}). SAO gains most: on H2 the selected rate 10^{-5} gives $0.072 \pm 0.012\%$ against 0.107%, and on B2 $1.00 \pm 0.01\%$ against 1.03%. On B1, validation selects 3×10^{-5} for OFormer, which reaches $0.88 \pm 0.03\%$ against SAO's 0.93%; its 10^{-4} arm, a near tie on validation, reaches $0.80 \pm 0.05\%$. OFormer therefore edges ahead of SAO on B1 after tuning, while SAO's own validation-selected arm on B1 ($1.00 \pm 0.05\%$) is no better than its shared-recipe value. Three selections ended on a grid edge after the permitted extensions (SAO on H2, OFormer on B2, OFormer on B1), and the SAO H2 arm at 10^{-5} also reaches the epoch cap on every seed, so lower rates would need longer schedules. The audit narrows the caveat of Tables 4.1 and 4.2 to schedule and architecture search: on the pairs examined, the model ordering is stable on B2 and H2. On B1, where OFormer and SAO were initially level, tuning moves OFormer ahead.

Table H.1: Learning rate sensitivity for eight model–case pairs: test RL2 (%) under the shared recipe (five seeds, peak rate 3×10^{-4}) and at the validation-selected rate (three seeds, mean $\pm$ sd), with the selected rate and whether it lies on the edge of the searched range. Tuning preserves the pairwise ordering on B2 and H2 but moves OFormer ahead of SAO on B1.

Case	Model	Shared recipe	Tuned	Selected rate	Edge
B2	FNO-T	5.19	4.15±0.28	10^{-4}	no
B2	OFormer	3.39	3.25±0.49	10^{-2}	yes
B2	SAO	1.03	1.00±0.01	10^{-4}	no
H2	FNO-T	0.473	0.471±0.018	3×10^{-4}	no
H2	DeepONet*	0.154	0.150±0.000	10^{-4}	no
H2	SAO	0.107	0.072±0.012	10^{-5}	yes
B1	OFormer	0.92	0.88±0.03	3×10^{-5}	yes
B1	SAO	0.93	1.00±0.05	10^{-4}	no

H.5 A SPATIAL FNO CONTROL

FNO-T is the control for the time–space reformulation: it is an FNO applied along time, with the query broadcast as extra channels. Its gap to SAO may arise from this interface because an FNO ordinarily receives the whole (x, t) field. We therefore also trained *Spatial FNO-BVP*, a two-dimensional FNO that takes the inlet broadcast along x together with coordinate and parameter channels and emits the entire field at once, with spectral convolutions over both axes (16 modes in x, 32 in t).

Every model in this paper is trained under the random-probe protocol of Appendix D.2, on 10 probed locations per simulation, whereas a field-to-field model would ordinarily consume all 100 interior locations, a tenfold advantage that has nothing to do with architecture. We therefore mask the training loss to exactly the probed rows of each simulation, read from the same stored assignment the other models use, and confirmed from the training logs that 9.90% of the rows carry gradient. The recipe, splits, epoch caps and dense evaluation are otherwise unchanged. Capacity matching is slightly looser than for the five main baselines because the width knob of a two-dimensional FNO is coarser: 3,694,337 parameters on B2 ($+0.67\%$ of SAO's count) and 3,694,865 on H2 ($+0.44\%$), against at most 0.39% for the others.

Table H.2: Spatial FNO control against SAO and FNO-T: dense test RL2 (%), mean $\pm$ sd over three seeds for the spatial FNO and five for the other rows; H2 values average P, Q and A. All three share inlet, query, parameters, split, recipe and probe supervision.

Model	B2	H2 (three-field mean)
SAO (ours)	1.03±0.04	0.107±0.012
Spatial FNO-BVP	2.46±0.04	0.147±0.005
FNO-T	5.19±0.17	0.473±0.013

Giving FNO the complete (x, t) field recovers much of the gap (Table H.2). Spatial FNO-BVP improves over FNO-T by $2.1\times$ on B2 and $3.2\times$ on H2, but SAO remains $2.4\times$ and $1.4\times$ more accurate, respectively. On H2, Spatial FNO-BVP matches SAO on flow rate (0.132% versus 0.129%), while its pressure and area errors are about $1.6\times$ larger. Thus, SAO's advantage over FNO-T is not explained by the temporal input representation alone: SAO also outperforms a spatial FNO that receives the complete field under the same supervision and parameter budget. This control covers three seeds and two benchmarks and is not included in Tables 4.1 and 4.2.

H.6 SHARED AND FIELD-SPECIFIC TEMPORAL BASES

Both SAO and the inlet-to-basis DeepONet read P, Q and A out of a temporal dictionary generated from the inlet, and both must decide whether that dictionary is shared across the three fields or split per field. They answer differently by default: SAO shares one basis of rank 10 or 20 and gives each field its own coefficients, while the DeepONet gives each field its own block of 662 functions. This subsection tests the choice in both directions, at matched capacity: every variant is re-fitted to within

0.1% of its reference parameter count and trained with the same recipe, seeds and evaluation grid (Table H.3).

Table H.3: Sharing study on hemodynamics: dense test RL2 (%), five seeds, mean $\pm$ sd. `SAO-splitBasis` adds $K/2$ private functions per field to SAO's shared basis; the last row constrains the inlet-to-basis DeepONet to one basis for all fields. No cell is bolded because most differences are not separable across seeds (paired tests in the text).

	H1 – single artery (K=10)				**H2 – Y-bifurcation** (K=20)			
Model	**P**	**Q**	**A**	**Mean**	**P**	**Q**	**A**	**Mean**
SAO (shared basis)	0.088	0.161	0.041	0.097	0.139	0.129	0.052	0.107
`SAO-splitBasis` (shared + private basis)	0.093	0.164	0.045	0.100	0.123	0.116	0.046	0.095
DeepONet* (inlet-to-basis assignment)	0.151	0.341	0.070	0.187	0.171	0.241	0.051	0.154
DeepONet* (shared basis)	0.146	0.341	0.068	0.185	0.163	0.240	0.048	0.150

Sharing one basis does not significantly change the three-field mean in either architecture. For DeepONet, the mean changes from 0.187 to 0.185 on H1 (p=0.15) and from 0.154 to 0.150 on H2 (p=0.37). Adding private functions to SAO changes the mean from 0.097 to 0.100 on H1 (p=0.43) and from 0.107 to 0.095 on H2 (p=0.15). Only H2 pressure improves significantly (0.139 to 0.123, p=0.039), and this effect does not recur on H1. The model-free analysis supports sharing: at equal total dimension, one basis fitted to the stacked fields reconstructs them better than three field-specific bases (Appendix I.4). SAO therefore retains the simpler shared basis.

The gap between the two architectures remains after changing how the fields share a basis: SAO stays ahead of the DeepONet by 1.8–$1.9\times$ on the H1 three-field mean and 1.4–$1.6\times$ on H2 across the four sharing combinations. The remaining structural differences are the rank, 10 or 20 against 662, and the way the query reaches the basis, through FiLM's per-channel modulation in SAO and not at all in the DeepONet, whose basis depends only on the inlet and time. The Burgers cases point the same way and more sharply, where SAO is 7–$15\times$ ahead.

H.7 Physics-based error measures

Beyond the aggregate RL2 of Tables 4.1 and 4.2, we compute physics-based error measures post hoc from the saved dense predictions. The analysis includes three Burgers measures and three hemodynamic measures for every model; the external-baseline values are computed separately from their saved predictions. Let $\hat{u}, u$ denote prediction and reference on the dense (x, t) grid of one test simulation.

- **Steep-gradient-window RL2.** $\|\hat{u} - u\|_2/\|u\|_2$ restricted to the mask $\{|\partial_x u| \geq q_{0.90}(|\partial_x u|)\}$, i.e. the top-$10\%$ steepest-gradient cells of the reference field. It reports whether the aggregate L^2 error hides a concentrated error at the shock.
- **Shock-location error.** At the time index whose maximum $|\partial_x u|$ is largest, the distance between the $\arg\max_x |\partial_x u|$ of the reference and of the prediction, expressed as a percentage of the domain length.
- **Relative L_∞.** $\max|\hat{u} - u|/\max|u|$ per simulation.
- **Junction mass residual (H2).** With Q_p the parent outlet and Q_{d1}, Q_{d2} the daughter inlets, the residual $r(t) = Q_p(t) - Q_{d1}(t) - Q_{d2}(t)$ is reported as the cycle-L_2 ratio $\|r\|_2/\|Q_p\|_2$. Nothing in the loss enforces this balance, so it measures learned rather than imposed conservation.

Each metric is aggregated within a seed over that seed's test simulations and then averaged across the 5 seeds; the quoted spread is across seeds.

Burgers. On the Burgers benchmarks (Table 4.4) SAO has the lowest steep-gradient-window error of any variant tested. The steep-window RL2 tracks the overall RL2 closely on all three Burgers

cases (0.93–2.33% against 0.90–1.03%), so the aggregate metric is not concealing a localized shock error. Shock localization is quantized by the $\Delta x = 0.01$ output grid, so a score of 0.000 means the predicted $\arg\max_x |\partial_x u|$ fell on the reference grid point in every test simulation: SAO achieves this throughout B2 and B3, and on B1 misses by a single cell in one of 50 test simulations for four of the five seeds, the fifth being exact on all 50. Removing FiLM destroys shock localization on B1 (45.2% of the domain), and the two fixed-basis baselines mislocate the front by 5.1% (DeepONet*) and 18.1% (POD-DeepONet) on B1.

Hemodynamics. The junction residual of the three-field hemodynamic models (Table 4.3) is lowest for SAO among its ablations, with a margin over `SAO-fixedModes` (all Fourier modes kept) within seed variation, and lower still for DeepONet* and POD-DeepONet. A value below the residual measured from the discretized reference is not automatically more physically accurate, since smoothing or bias can also reduce this measure. We therefore interpret it as descriptive agreement rather than evidence of improved conservation. Table 4.3 reports three quantities that the training objective never enforces: the models are trained on P, Q and A alone, so each measures how far a model's three outputs stay mutually consistent with the governing relations. *Derived* U is the relative L^2 error of the velocity $\hat{Q}/\hat{A}$ formed from two predicted fields, against the solver's velocity field. *Tube-law consistency* is $\|\hat{P} - (a\sqrt{\hat{A}} + b)\|/\|\hat{P}\|$, where (a, b) are fitted per simulation and vessel to the ground-truth (P, A) pair, so it measures whether the two predicted heads remain mutually compatible with the elastic tube law of Appendix B.3. *Junction mass* is the H2 residual $\|\hat{Q}_p - \hat{Q}_{d1} - \hat{Q}_{d2}\|/\|\hat{Q}_p\|$ at the bifurcation (defined above). Among SAO and its ablations, SAO has the lowest value in each column, not always separably. The H2 tube-law margin over `SAO-fixedModes` is significant (p=0.034); the junction margin over `SAO-fixedModes` (0.192% against 0.215%) is not (p=0.22, mixed signs), so the defensible statement about the junction is the comparison against the other three ablations (0.529–0.921%) and against the true-Q reference. The same residual evaluated on the ground-truth waveforms is 0.206%, nonzero at the discrete level, and SAO's predictions sit below it.

The external baselines write none of these quantities during training, so their rows were computed afterwards from the saved dense test predictions with the same formulas and array conventions as the SAO harness; run on SAO's own saved predictions, that code reproduces the stored SAO values to 3×10^{-10}. DeepONet* has the lowest H2 tube-law and junction residuals of all models (0.113% and 0.146%) and POD-DeepONet the second-lowest junction residual (0.156%); both sit below the 0.206% floor, as does SAO. Because the floor is the reference's own discretization residual, values below it do not order the models, and the two baselines carry 1.4–$2.1\times$ SAO's three-field error and 1.6–$2.2\times$ its derived velocity error on H2 (Tables 4.2 and 4.3). On H1, SAO has the lowest derived velocity and tube-law errors of all ten models, with `SAO-fixedModes` second on the velocity and DeepONet* second on the tube law.

H.8 Reference-solver refinement on H2

All errors in this paper are measured against reference solutions on a fixed grid, so part of any reported error is the reference's own discretization error. To measure that floor on H2 we re-ran the first 20 test simulations in openBF with twice the number of nodes per vessel (399 instead of 199; the time step halves with the grid under the same CFL number of 0.7) and compared the original and refined solutions on the 101 saved points over the full time grid, using the RL2 of §4.1 pooled over the three vessels. Table H.4 reports that difference next to SAO's error on the same simulations (seed 42), measured both against the original reference, as everywhere else in the paper, and against the refined solution.

On pressure and area the reference's grid error is five to eight times smaller than SAO's error, and SAO's error is the same whichever solution it is measured against, so those errors are model error. On flow rate the grid error, 0.118%, is 86% of SAO's error against the original reference, and SAO's error against the refined solution rises to 0.182%, close to the root-sum-square of the two. The H2 flow-rate comparison is therefore sensitive to reference discretization, and differences between models below $\approx 0.1\%$ are on the scale of the measured grid discrepancy. At 99 nodes, the solver takes 5.5 min per simulation, $2.2\times$ less than the reference, but its errors against the 399-node solution are 0.074, 0.346 and 0.019% on P, Q and A, above SAO's on flow rate. Re-running one stored

H2 simulation reproduced it to machine precision, which also fixes the reference provenance: the generator uses a CFL number of 0.7, not the 0.9 recorded in the stored configuration files.

The Burgers references were audited the same way. Re-running the stored simulations with the Clawpack solver reproduces them bit-exactly on all three cases, and re-solving 20 test simulations at twice and four times the fine grid (N_x=2048 and 4096 instead of 1024) moves the reference by 0.50 and 0.64% on B1, 1.12 and 1.55% on B2 and 1.81 and 2.57% on B3, against SAO's 1.01, 1.08 and 0.83% on the same simulations (seed 42). On B2 and B3 the reference's own grid error is therefore as large as or larger than SAO's error, as on the H2 flow rate.

Table H.4: Reference-solver refinement on H2, 20 test simulations: RL2 (%) mean ± sd over simulations, pooled over vessels. Grid error is the original (199-node) against the refined (399-node) solution; the SAO columns are seed 42 against each.

Field	Grid error	SAO vs. original	SAO vs. refined
P	0.024±0.003	0.136±0.168	0.140±0.164
Q	0.118±0.007	0.138±0.081	0.182±0.088
A	0.006±0.001	0.046±0.044	0.046±0.043

H.9 x-INDEPENDENT CONTROL

The control of §4.2 is the three-field SAO of Table 4.2 (H1 K=10, H2 K=20, same recipe and epoch caps) with the 17 sinusoidal positional-encoding dimensions zeroed in the static input at both training and test time; the vessel one-hot (H2) and the physical parameters E, R are retained, so only spatial information is withheld. One seed (42) suffices: with no spatial structure left to learn, training early-stops almost immediately (best epochs 13 on H1 and 25 on H2). The analytic bound quoted in the main text is the error of the exact per-simulation (per-vessel on H2) spatial-mean predictor, the L^2-optimal x-independent prediction, computed from the test ground truth, and $\mathrm{Var}_x/\mathrm{Var}_t$ is the ratio of pooled variances on the same data. Table H.5 lists both quantities, the control's dense-test error and SAO's for each field. On the flow rate the control sits just above its bound on both cases, so training recovers everything an x-independent predictor can reach and no more; on the H2 pressure and area fields, whose bounds are tiny, it stops short of them, because early stopping is driven by the three-field mean that Q dominates. The bound itself is set by how much each field varies along the vessel: the flow rate carries the wave-reflection structure, whereas pressure and area follow the near-linear tube law and are far more uniform along x, which is why the H2 area bound is already at SAO's level. The control is not a row of Table 4.2: it is a single seed with no spread.

Table H.5: x-independent control on hemodynamics: dense test RL2 (%) per field, H2 averaged over vessels. Bound: the per-simulation spatial-mean predictor on the test ground truth; control: the SAO of Table 4.2 with the positional encoding zeroed (seed 42); SAO: five-seed mean.

Case	Field	$\mathrm{Var}_x/\mathrm{Var}_t$	x-indep. bound	Control	SAO
H1	P	0.022	2.30	2.36	0.088
H1	Q	0.031	14.73	14.86	0.161
H1	A	0.0018	1.04	1.08	0.041
H2	P	1.4×10^{-4}	0.22	0.37	0.139
H2	Q	0.012	9.86	10.16	0.129
H2	A	5×10^{-6}	0.05	0.23	0.052

H.10 OUT-OF-DISTRIBUTION PARAMETER EXTRAPOLATION

The study re-trains SAO from scratch on a restricted range of one material parameter and evaluates on the held-out extremes. Simulations are sorted by the parameter of interest (E or R on H1, and the network mean over the three vessels on H2, whose per-vessel values are sampled independently) and the lowest and highest 10% become the OOD test set; the middle 80% is shuffled with a fixed splitting seed, identical across training seeds, into 100 validation, 100 in-distribution test and 600 training simulations. Training uses the three-field configuration of Table 4.2 (K=10 on H1, K=20

on H2, the per-case epoch caps and the objective of Eq. (33)); only the model initialization and the training order vary with the seed. Both sets are dense-evaluated at all 101 grid points, all three vessels on H2, with the metric of §4.1, so the columns of Table H.6 are directly comparable. Because the training set is smaller (600 rather than 700 simulations) and covers only the middle 80% of the parameter range, the levels are not comparable with the hemodynamics columns of Table 4.2; the in-distribution-to-tail ratio within a row is the meaningful quantity. All cells use the three training seeds $\{42, 123, 7\}$.

Table H.6: Parameter extrapolation of the three-field models: dense test RL2 (%) averaged over P, Q and A, inside the training range and on the held-out 10% tails, mean $\pm$ sd over three seeds; the tail column pools both tails, the next gives them separately, and degradation is tail over in-distribution. The last row repeats the H2 radius axis without seed 42, whose small-radius tail error is 1.03% against 0.19% for the other two seeds.

Case	Axis	In-dist.	Tail	Lower / upper tail	Degradation
H1	E	0.119±0.008	0.156±0.007	0.174 / 0.138	1.31×
H1	R	0.115±0.004	0.178±0.009	0.169 / 0.188	1.55×
H2	E	0.139±0.012	0.150±0.006	0.158 / 0.142	1.08×
H2	R	0.126±0.010	0.340±0.280	0.469 / 0.210	2.70×
H2, R, seeds 123 and 7		0.120	0.178	0.191 / 0.166	1.48×

Three of the four case/axis pairs degrade mildly and consistently, from $1.08\times$ to $1.55\times$. On the H2 radius axis, seed 42 instead reaches 1.03% on the small-radius tail against 0.19% for the other two seeds, which lifts the pooled mean to $2.70\times$ and inflates the spread beyond the mean itself. That seed also stopped earlier (epoch 2018 against 4027 and 3334), but three seeds do not distinguish an optimization failure from unstable extrapolation. We therefore report both the three-seed mean and the two-seed value excluding the outlier. The radius is the harder axis on both topologies ($1.55\times$ against $1.31\times$ on H1) and the small-radius tail is harder than the large-radius one, which is consistent with the wave speed and characteristic impedance rising as the lumen narrows. On the stiffness axis the low-E, most compliant tail is likewise the harder of the two on both cases. Per field, the pressure and area errors degrade more than the flow rate does on H1 ($1.39\times$ and $1.43\times$ on the stiffness axis against $1.22\times$ for Q; $1.91\times$ and $1.86\times$ against $1.29\times$ on the radius axis), so a study of the flow rate alone understates how far these models are extrapolating. This study covers extrapolation in the material parameters only; new inlet-frequency families, off-grid queries, resolution transfer, and geometry or topology change are not tested and remain out of scope.

I Separability floors and structured alternatives

Because FiLM allows the conditioning input to modulate the dynamic basis, (9) does not impose a strict separation of variables or a rank-K bound on the assembled field. We compare this formulation with strict factorization. First, we compute a ground-truth accuracy floor for strictly factorized models (§I.1). We then evaluate a strictly factorized SAO on the hemodynamics benchmarks (§I.2) and structured alternatives on Burgers (§I.3). All errors are dense test RL2 (%) under the metric and protocol of §4.1. The floors include every field of the three-field benchmarks, and the hemodynamics models use the same three-field configuration as the H1 and H2 results in Table 4.2.

I.1 The separability floor

Fix one simulation and collect its response on the evaluation grid into $Y \in \mathbb{R}^{N_x \times N_t}$ with $Y_{ij} = y(x_i, t_j)$. Any model whose synthesis is a strict factorization

$$\hat{y}(x,t) \;=\; \sum_{k=1}^{K} \alpha_k(x)\, S_k(t), \tag{34}$$

in which the temporal basis $\{S_k\}$ does not depend on the query x, emits a matrix of rank at most K, however α_k and S_k are parameterized. By the Eckart–Young theorem its relative ℓ^2 error on

that simulation is therefore at least $\|Y - Y_K\|_F / \|Y\|_F$, where Y_K is the rank-K truncated SVD of Y. Averaging that residual over the test simulations, the way RL2 is aggregated (§4.1), gives a *separability floor*: a lower bound on the dense test RL2 of every model of the form (34), obtained from the test ground truth alone and hence independent of architecture, capacity, optimizer, and seed. The bound is generous, since it grants each simulation the optimal basis rather than a learned one. It is also the right bound for `SAO-noFiLM`, whose basis is a function of the inlet waveform alone and therefore constant in x within a simulation.

Table I.1: Separability floors: the lowest dense test RL2 (%) attainable by any model of the form (34), computed from the test ground truth by SVD (inlet row excluded; H2 averaged over vessels). K is each case's model rank; the last columns give the smallest rank at which the floor falls below 1% and 0.1%.

Case	**Field**	K	**Floor at K (%)**	K **for** 1%	K **for** 0.1%
B1	u	6	18.66	60	93
B2	u	10	4.99	21	44
B3	u	20	0.21	13	25
H1	P	10	0.0027	2	4
H1	Q	10	0.018	4	6
H1	A	10	0.0012	2	3
H2	P	20	7×10^{-6}	1	2
H2	Q	20	1.3×10^{-4}	2	3
H2	A	20	3×10^{-6}	1	1

The two families sit in qualitatively different regimes (Table I.1). For the hemodynamic flow field the floor is numerically negligible (0.018% on H1 at $K{=}10$ and 0.00013% on H2 at $K{=}20$, respectively $\approx 9\times$ and $\approx 1000\times$ below SAO's own flow-rate error, and the floors on pressure and area are smaller still) and it is insensitive to the exact rank, already below 0.1% at $K{=}6$ on H1 and at $K{=}3$ on H2. Strict separability is simply not the binding constraint there. On B1 the floor is 18.66% at the model's rank $K{=}6$, and $K{=}60$ is needed before it reaches 1%: the response is a travelling, steepening front whose rank-K approximation error decays slowly: the Kolmogorov n-width barrier that makes linear reduced models inefficient for transport-dominated problems (Greif & Urban, 2019). Consistently with this, the trained `SAO-noFiLM` ablation reaches $22.56 \pm 0.09\%$ on B1, close to the floor of the class it belongs to, so its gap to SAO is largely a property of the hypothesis class rather than of its optimization.

This behavior is consistent with known limitations of linear reconstruction for transport-dominated solution families. Lanthaler et al. (2023) show that operator-learning architectures with a *linear* reconstruction step (a fixed, input-independent output basis, as in DeepONet and PCA-Net) obey a lower bound on the approximation error for PDEs with discontinuities that decays only algebraically in the number of basis functions, whereas architectures whose reconstruction may shift the output nonlinearly with the input escape that bound. Their theorem does not directly establish the per-simulation, query-separability floor studied here because SAO's temporal functions already depend on the inlet waveform. Table I.1 instead provides the relevant empirical bound for our stricter within-simulation factorization and motivates asking how the synthesis should depend on the query.

I.2 Strict factorization in hemodynamics

`SAO-muFiLM` is an SAO whose Static Net still produces the coefficients $\alpha_\theta(c)$ from the full conditioning input $c = (x, \mu)$, but whose FiLM parameters $(\gamma_\ell, \beta_\ell)$ of (6) are computed from the physical parameters μ alone and never from the query x. Within one simulation the basis is then constant in x and the synthesis is exactly (34). This is a structural guarantee, and we verified it directly: over a sweep of query locations at fixed μ the `SAO-muFiLM` gains and shifts are bit-identical, while the same sweep moves the full model's by 1.96 and 1.11. Everything else (architecture, hyperparameters, seeds, data, epoch caps and the three-field objective) matches the SAO it is compared with, and the parameter count is within $+0.04\%$.

Table I.2: Strictly factorized (SAO−muFiLM) against full-FiLM SAO on hemodynamics: dense test RL2 (%) averaged over the three fields, mean ± sd over five seeds, same recipe and epoch caps. The H2 SAO−muFiLM spread comes from one seed at 0.199 against 0.091–0.107.

Case	SAO−muFiLM (strictly factorized)	SAO (full FiLM)
H1	0.149±0.009	0.097±0.011
H2	0.119±0.045	0.107±0.012

Strict factorization raises H1 error but has no resolved effect on H2 (Table I.2). On H1 the strict factorization costs $1.54\times$ on the three-field mean, 0.149 against 0.097, in the same direction on all five seeds and significant under a paired test (p=0.001); the penalty is largest on area ($1.96\times$, p=0.0006) and flow rate ($1.57\times$, p=0.0003) and is not separable on pressure ($1.29\times$, p=0.09). It also converges into that worse solution much sooner, with best epochs of 508 to 911 against 2123 to 2617 for the full model. On H2 the two are indistinguishable: the ratio is 1.12 with mixed per-seed signs and p=0.63, the medians are level (0.105 against 0.107), and the mean and spread are set by one seed that lands at 0.199 while the other four sit at 0.091 to 0.107, at or below the full model. The one place H2 does separate is the junction: SAO−muFiLM leaves a mass residual of 0.303% against SAO's 0.192%, above the reference data's own 0.206%.

The difference between the two cases is rank against demand. Both read three fields out of one shared basis, but H1 does so at K=10 and H2 at K=20, and the H2 responses are already separable to $10^{-4}\%$ at that rank (Table I.1). On H1 the basis is the scarce resource, so freezing it across queries forces all spatial variation into ten coefficients per field and costs accuracy; on H2 the coefficients have room to absorb it. Query-dependent basis functions improve accuracy on H1, where the shared basis is restrictive, but their effect is not resolved on H2 at the tested rank.

I.3 Burgers: a ladder of structured syntheses

B1 is the opposite regime: the floor forbids any strict factorization from going below 18.66%, so the question is how much structure the query-dependence needs. We therefore built a ladder of syntheses between SAO−noFiLM and SAO, each rung adding one explicitly parameterized mechanism, and trained each under the paper protocol on the same five seeds (Table I.3).

Shifted synthesis. The first rung replaces (34) with a per-basis, condition-dependent time shift,

$$\hat{u}(x,t) \;=\; \sum_{k=1}^{K} \alpha_k(\cdot)\; S_k(t-\tau_k(\cdot))\,, \tag{35}$$

the neural-operator analogue of the shifted proper orthogonal decomposition of Reiss et al. (2018), in which each mode carries its own shift so that a travelling feature is represented by a fixed profile in a moving frame rather than by many stationary modes. In SAO−shift the delays τ_k are a function of the query location only; the basis S_k itself is still produced by the spectral Dynamic Net from the inlet waveform alone, which we verified is bit-identical across queries. Shifts are applied in the Fourier domain on a zero-padded signal, so the translation is linear rather than circular, and τ_k is bounded to $\pm T/2$ by a $\tanh$. The mechanism adds 774 parameters over SAO−noFiLM. SAO−shiftInlet keeps the same synthesis but conditions the delays on the inlet as well (on the query location *and* an 8-dimensional linear embedding of the waveform) for $3{,}414$ parameters over SAO−noFiLM.

Table I.3: Ladder of syntheses on B1 from strict factorization to full FiLM: dense test RL2 (%), mean ± sd over five seeds; the floor row is the bound of Table I.1; parameter deltas are relative to SAO−noFiLM.

Model	Synthesis	B1 RL2 (%)	Δ params
Separability floor	any strict factorization, K=6	18.66	—
SAO−noFiLM	strict factorization, learned	22.56±0.09	(ref.)
SAO−shift	+ delays $\tau_k(x)$	11.00±0.70	+774
SAO−shiftInlet	+ delays $\tau_k(x,g)$	3.65±0.35	+3,414
SAO	full FiLM	0.93±0.03	—

Conditioning the shifts on location reduces the error from 22.56% to 11.00% with fewer than 800 additional parameters. Conditioning them on both location and the inlet reduces the error further to 3.65%, below the 18.66% floor for strict factorization. This result is consistent with the Burgers characteristics: because the propagation speed is u, the arrival time at a given location depends on the inlet amplitude. The waveform-conditioned delay can represent this dependence, whereas the location-only delay cannot. A waveform-conditioned complex filter (`SAO-filterInlet`) obtained 8.69% for seed 42; we did not evaluate it on the remaining seeds.

Rigid shifts cannot reproduce the waveform deformation caused by nonlinear steepening. They translate each basis function without changing its shape, leaving a gap between `SAO-shiftInlet` at 3.65% and full FiLM at 0.93%. Of the 21.6 percentage-point difference between `SAO-noFiLM` and SAO on B1, waveform-conditioned shifts recover 18.9 points, and full FiLM supplies the remaining 2.7.

I.4 Geometry of the solution set and of the learned trial space

We use post hoc measurements on the trained checkpoints of Tables 4.1 and 4.2 to relate the separability floors to solution-set geometry and variation in SAO's trial space.

Solution-set curvature. Treat each response time series $y(x_i, \cdot) \in \mathbb{R}^{N_t}$ as one sample. Let $E_{\text{global}}(K)$ be the mean relative ℓ^2 error of reconstructing a held-out test series in the rank-K POD basis of all training series, and $E_{\text{local}}(K)$ the same quantity in a rank-K basis fitted to that series' 200 nearest training conditions, with distance the sum of normalized inlet L^2, query, and parameter distances. Their ratio $R_{\text{local}}(K) = E_{\text{global}}/E_{\text{local}}$ is the usual measure of a set that is curved but locally low-dimensional: a large ratio means a basis fitted nearby is worth much more than one fitted globally. Figure I.1(a) and Table I.4 give it. The Burgers responses are strongly curved, and increasingly so with wave count (3.3, 5.3, 10.1 at the models' own ranks); the hemodynamic responses are not (1.13 on H1, and below one on H2). The H2 value should not be read as negative curvature: the global basis there already reaches a floor of $\sim 10^{-4}\%$ (Table I.1), so at rank 20 there is no curvature left to exploit and the locally fitted basis, estimated from 200 neighbours rather than 210,000 series, is simply the noisier of the two. At rank two, where estimation noise is negligible, both hemodynamic cases sit modestly above one (1.3–1.9). This is the geometric statement behind §I.2 and §I.3: strict factorization is nearly free where the solution set is flat and prohibitive where it is curved.

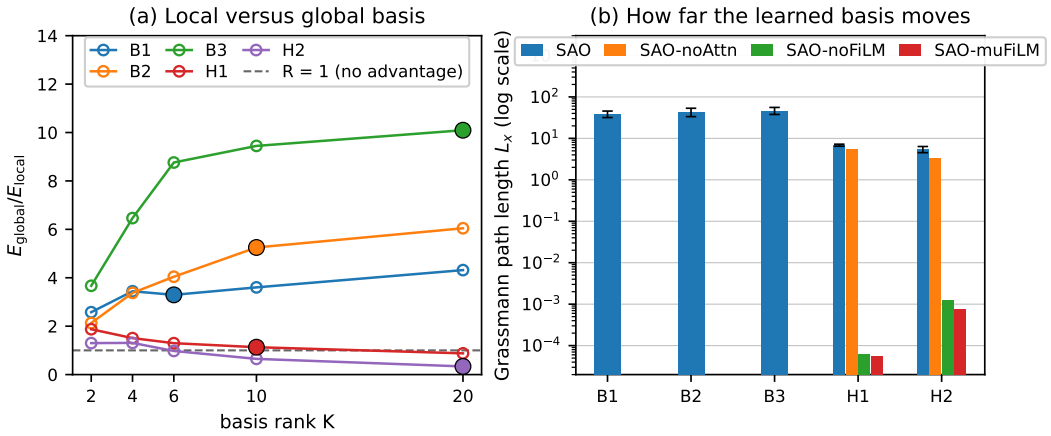

Figure I.1: **Solution-set curvature and motion of the learned trial space. (a)** Ratio of the rank-K reconstruction error in a global POD basis to that in a basis fitted to a query's 200 nearest training conditions; values above 1 mean locality pays, and filled markers are each case's own rank. **(b)** Grassmann path length of SAO's basis along the query grid (log scale), with `SAO-noAttn` and the two query-independent models (`SAO-noFiLM`, `SAO-muFiLM`) at floating-point noise.

Table I.4: Geometry of the solution set and of the learned trial space at each case's rank K: R_{local} the global-to-local POD error ratio (hemodynamics averaged over fields); L_x the Grassmann path length along the query grid (seed 42, mean $\pm$ sd over test simulations; per vessel on H2); A_r the fraction of the local data tangent captured by a rank-K basis (SAO, a global POD basis of the same rank, and `SAO-noFiLM`). In the path-length columns `SAO-noFiLM` and `SAO-muFiLM` are structural zeros and the Burgers ablations were not re-extracted.

		Grassmann path length L_x				**Tangent A_r**		
Case	R_{local}	**SAO**	`SAO-noAttn`	`SAO-noFiLM`	`SAO-muFiLM`	**SAO**	**POD**	`SAO-noFiLM`
B1 (K=6)	3.3	38.7±6.9	—	—	—	0.42	0.41	0.39
B2 (K=10)	5.3	43.4±9.9	—	—	—	0.47	0.62	0.52
B3 (K=20)	10.1	46.8±9.0	—	—	—	0.57	0.68	0.56
H1 (K=10)	1.13	6.9±0.4	5.4±0.5	6×10^{-5}	6×10^{-5}	0.59	0.81	0.60
H2 (K=20)	0.34	5.4±0.9	3.3±0.4	1×10^{-3}	8×10^{-4}	0.63	0.82	0.64

Trial-space variation. For each query we extract the basis the synthesis actually uses, orthonormalize it by QR, and measure the Grassmann distance $d(\mathcal{V}_i, \mathcal{V}_j) = (\sum_\ell \theta_\ell^2)^{1/2}$ between consecutive query locations, with $\cos\theta_\ell$ the singular values of $Q_i^\top Q_j$ (Edelman et al., 1998; Amsallem & Farhat, 2008). Summing along the dense grid gives a path length L_x per simulation. The extraction was checked against each model's own forward pass on all fifteen checkpoints and reproduces it exactly. The measurement separates the two families by an order of magnitude (Figure I.1(b)): SAO's basis traverses 38.7 to 46.8 along a Burgers domain but only 6.9 along the H1 artery and 5.4 along an H2 vessel, in the same direction as the curvature above. Two models act as structural controls and behave exactly as designed: `SAO-noFiLM`, whose basis sees only the inlet, and `SAO-muFiLM`, whose FiLM parameters are computed from μ alone, both give L_x at the level of floating-point noise (10^{-5} to 10^{-3}). Removing attention lowers the path length without eliminating it (5.4 on H1 and 3.3 on H2 for `SAO-noAttn`), so cross-mode attention accounts for part but not all of the query dependence. On H2 the jump between the parent's basis and a daughter's at the bifurcation is 3.33, comparable to a whole vessel's own path length; it is present in `SAO-muFiLM` as well (3.48), which places it with the vessel identifier in the conditioning input rather than with the spatial coordinate. Finally, the path length tracks the inlet amplitude closely (Spearman 0.95 on B1, 0.82 and 0.88 on B2 and B3, so the loudest waveforms bend the basis most) but is essentially uncorrelated with a simulation's own error (Spearman 0.09 to 0.22): basis motion is a property of the input, not a predictor of accuracy.

Shared temporal basis. The shared-basis choice of §3.4 can be tested without any model. Fitting one POD basis of rank $3r$ to the stacked (P, Q, A) data and three private bases of rank r to the fields separately spends the same total dimension; the shared basis reconstructs the held-out test data $4.7\times$ better on H1 and $7.3\times$ better on H2 at r=10, and the advantage grows with r from $2.0\times$ at r=2. The principal angles between the field subspaces shrink accordingly, from 28–70° at rank two to 14–23° at rank ten on H1 and 4–15° on H2, where the pressure and area spaces are nearly coincident ($4.3°$), as the elastic tube law makes P close to an affine function of $\sqrt{A}$. The three fields do share temporal structure, and sharing one basis is the cheaper way to represent them.

Reconstruction accuracy and local subspace alignment. The captured fraction $A_r = \|Q^\top T_r\|_F^2/r$, where T_r contains the leading local-PCA directions in a query's neighbourhood, measures how well a basis spans variation across nearby inlets. SAO scores below a global POD basis of the same rank on four benchmarks and is comparable to `SAO-noFiLM` throughout (Table I.4). Yet SAO reconstructs individual responses two orders of magnitude more accurately than the global POD basis. Local subspace alignment therefore does not explain SAO's reconstruction accuracy. The measurements instead show that SAO's trial space varies more on the benchmark family with greater solution-set curvature, that FiLM and attention contribute to this variation, and that the three hemodynamic fields share temporal structure.