WHAT DO SEPARABLE PHYSICS-INFORMED NEURAL NETWORKS NEED FROM A RECURRENT BLOCK?

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequence-model architectures combine interactions between tokens with nonlinear channel transformations, but the contributions of these components to physics-informed learning are difficult to distinguish. We study this question in separable physics-informed neural networks, where independent coordinate branches form a multidimensional field through a sum-of-products representation. We introduce Separable RWKV (S-RWKV), whose branches apply a smooth matrix-memory update to groups of learned Fourier features. Recurrence is confined to feature groups at a single coordinate and does not propagate between physical points. This construction allows us to remove the memory path while retaining the input encoding, channel transformations, field assembly, and training protocol. Across six benchmarks with repeated runs, the memory-free variant has lower mean errors than the full model on most scalar problems and requires fewer parameters and forward operations. Direct projection of the Fourier features, without channel transformations, produces larger errors and greater variation across runs on Convection and Wave. Comparisons with a separable multilayer perceptron show that the branch body remains important under matched input features and training conditions. Stage-wise results show that the main accuracy differences arise before output-projection refinement. Under the shared protocol, the memory-free model also outperforms the compared attention- and state-space-based networks on the scalar benchmarks. These results support lightweight nonlinear processing of grouped spectral features as an effective design for separable physics-informed learning.

1 INTRODUCTION

Physics-informed neural networks (PINNs) fit a field by minimizing the residual of its governing equation together with errors in initial conditions, boundary conditions, and observations (Raissi et al., 2019). Because this objective involves coordinate derivatives, the architecture affects both the trial space and the optimization problem. Gradient imbalance, unequal convergence rates, and ill-conditioning can prevent training from reaching solutions the network can represent (Wang et al., 2021a; 2022a; Krishnapriyan et al., 2021; Rathore et al., 2024). Recent studies use sequence-model blocks to address these failures. PINNsFormer applies attention to a point and its temporal neighbours (Zhao et al., 2024), S-PFormer adds Fourier features to an attention decoder (Arni & Blanco, 2025), and PINNMamba propagates information with a selective state-space model (Xu et al., 2025a). All three report accuracy gains over multilayer perceptron (MLP) PINNs on convection, reaction, and wave benchmarks.

Two observations complicate the interpretation of these gains. First, these architectures combine token interactions with nonlinear channel transformations. In PINNsFormer and PINNMamba, the latter use a learnable sine–cosine activation. Comparing a complete architecture with an MLP does not isolate either component. In vision, MetaFormer found that the overall block structure can be more important than the choice of token mixer (Yu et al., 2022). Second, the benchmark failure modes have also been addressed with float64 arithmetic (Xu et al., 2025b) or regularization of a plain MLP (Andersen & Matsubara, 2026). These findings motivate architecture comparisons that control the training protocol.

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.

Separable PINNs offer a setting in which both issues can be controlled. They assemble a field as a sum of products of one-dimensional coordinate functions, each computed by its own small branch (Cho et al., 2023). The branch body can be exchanged while everything else stays fixed: the input encoding, the assembly, the optimizer schedule, the arithmetic, the collocation samples, and the rule that selects a model state. Each branch output defines a factor of a canonical polyadic (CP) representation. The same coordinate-wise output-projection refinement can therefore be applied across branch bodies.

We use this setting to ask what a separable PINN needs from a recurrent block. We construct S-RWKV, whose coordinate branches process groups of learned Fourier features with smooth blocks based on RWKV-7 (Peng et al., 2025). Each block passes a matrix memory across the feature groups of one coordinate and then applies channel mixing. We compare this model with a one-block variant, a memory-free variant, and direct projection without either path. We also replace the branch body with SPINN's $\tanh$ MLP while retaining the same input. All configurations share a task-specific float64 pipeline and are run ten times. A joint-coordinate control and an ablation over input, assembly, and branch body complete the comparison. We find:

- **Accurate approximation without recurrent memory.** The memory-free model has a lower mean error on Reaction, Convection, Wave, and Flow Mixing, although these differences are not statistically significant. On the first three problems, it uses 54% fewer parameters and 64% fewer forward FLOPs. On Navier–Stokes, it reduces peak GPU memory by 74%.
- **Channel mixing improves training reliability.** Directly projecting the Fourier features produces larger mean errors and greater variation across runs on Convection and Wave. With identical input, assembly, and training, and nearly identical forward cost, SPINN's MLP branches reach a relative error of 0.91 on Convection, compared with 6.3×10^{-6} for the memory-free model. This difference appears before refinement. In the input–assembly–body ablation, only the tested separable models combining Fourier input with channel mixing reach mean errors below 10^{-5} on Convection.
- **Lower error at reduced model size.** Under the shared pipeline, the memory-free model has more than three orders of magnitude lower mean error than PINNsFormer, S-PFormer, and PINNMamba on Reaction, Convection, and Wave. It uses approximately $25\,000$ parameters and requires 8–$34\times$ fewer forward operations. Its errors are also lower than those reported for PINNsFormer and PINNMamba under their original protocols.

The recurrence studied here runs over feature groups within a coordinate branch and is reset at every point. Interactions between physical points, implemented through attention in PINNsFormer and state-space recurrence in PINNMamba, are distinct from this mechanism.

2 Related Work

Architectures for PINN failure modes. Standard neural networks tend to fit low frequencies first (Rahaman et al., 2019), which motivates Fourier input mappings (Tancik et al., 2020; Wang et al., 2021b; Li et al., 2021), periodic activations (Sitzmann et al., 2020; Wong et al., 2024), multiplicative filters (Fathony et al., 2021), and residual designs that keep deep PINNs trainable (Wang et al., 2024). Sequence-model PINNs attribute their gains to modeling temporal dependence (Zhao et al., 2024; Arni & Blanco, 2025; Xu et al., 2025a). Recent analyses identify limited arithmetic precision as one source of these failure modes (Xu et al., 2025b) and residual overfitting at the collocation points as another (Andersen & Matsubara, 2026). Both studies report remedies that retain a plain MLP. We hold precision and training fixed and vary only the branch body.

Separable and spectral PINNs. Separable approximations underlie Proper Generalized Decomposition (Nouy, 2010) and tensor neural networks (Wang et al., 2022b). SPINN combines independent coordinate branches with a sum-of-products assembly and forward-mode differentiation on tensor-product grids (Cho et al., 2023). SV-SNN uses trainable trigonometric spatial factors with fully connected temporal factors (Xiong et al., 2025), and MS-SFNN fits the coefficients of separable cosine bases by least squares (Yang & He, 2026). We retain this combination of separability and spectral features and study the transformations used to learn the coordinate factors.

Memory and mixing. Linear attention can be written as a recurrence (Katharopoulos et al., 2020), and the delta rule corrects the key–value map stored in a matrix memory (Schlag et al., 2021). RWKV (Peng et al., 2023) was extended with matrix-valued states and dynamic recurrence (Peng et al., 2024). RWKV-7 adds controlled erasure and writing (Peng et al., 2025), while Gated Delta Networks combine gating with the delta rule (Yang et al., 2025). Outside PINNs, MetaFormer showed that the token mixer of a Transformer can be replaced by pooling with little loss (Yu et al., 2022), and simple linear models have outperformed Transformers on long-term forecasting benchmarks (Zeng et al., 2023). We ask the corresponding question for the recurrent block of a PINN branch.

Least-squares output layers. RFM (Chen et al., 2022) and PIELM (Dwivedi & Srinivasan, 2020) solve for the coefficients of fixed random features. Adaptive-basis training alternates gradient steps on the features with least-squares solves for the output layer (Cyr et al., 2020), and variable projection eliminates the linear coefficients altogether (Golub & Pereyra, 2003; Newman et al., 2021). Alternating least squares updates one tensor factor at a time (Kolda & Bader, 2009; Holtz et al., 2012), and energy natural gradients and Gauss–Newton methods precondition PINN training with function-space curvature (Müller & Zeinhofer, 2023; Jnini et al., 2024). Our scalar refinement alternates output-projection updates in learned coordinate bases (Appendix A.5).

3 Method

3.1 Separable Field Representation

Let $z = (z_1, \dots, z_d) \in \Omega \subset \mathbb{R}^d$ denote the spatial coordinates and, where applicable, time. For m output fields, each coordinate branch a returns mR components $F^{(a)}_{j,r}(z_a)$, where $j = 1, \dots, m$ and $r = 1, \dots, R$. Branches have independent parameters. The prediction is

$$\widehat{u}_j(z) = R^{-1/2} \sum_{r=1}^{R} \prod_{a=1}^{d} F^{(a)}_{j,r}(z_a) + b_j. \tag{1}$$

On a Cartesian grid, Eq. (1) is a CP decomposition of rank at most R of the tensor of field values after subtracting the bias (Kolda & Bader, 2009). For a scalar space–time field, this becomes $\widehat{u}(x,t) = R^{-1/2} \sum_r f_r(x) g_r(t) + b$. The fixed factor $R^{-1/2}$ rescales the sum without changing the representable class. We use $R = 32$ on Reaction, Convection, and Wave, and $R = 32$ per field on Navier–Stokes, where each branch outputs $3R$ components for (u, v, p). Conditions are imposed through the loss; no analytical form of the solution enters the assembly.

3.2 Coordinate Branches: A Memory Path and a Channel-Mixing Path

Each coordinate is mapped affinely to $[0, 1]$ and centered, $\xi = \widetilde{z} - 1/2$. The branch input consists of G groups of width w,

$$h^{(0)}_g(\xi) = \begin{bmatrix} \sin(2\pi\xi\omega_g + \varphi_g) \\ \cos(2\pi\xi\omega_g + \varphi_g) \end{bmatrix} + 0.1E(\xi), \qquad g = 1, \dots, G, \tag{2}$$

where $\omega_g, \varphi_g \in \mathbb{R}^{w/2}$ and the affine map $E : \mathbb{R} \to \mathbb{R}^w$ are learned. The width w is even, and E is shared across groups within a branch. On Reaction, Convection, and Wave, we use $G = 4$ groups of width $w = 32$ with frequency initialization scales $(0.5, 1, 2, 8)$; the scales are shared by the three problems and are not set from the exact solution (Appendix A).

The full branch applies $L = 2$ smooth RWKV-based blocks to the sequence of groups $h_1, \dots, h_G$. Each block has two paths. The *memory path* mixes each group with its predecessor, projects it into H heads, and updates a state $S_g \in \mathbb{R}^{w_h \times w_h}$, $w_h = w/H$, per head:

$$S_g = S_{g-1} \operatorname{diag}(\lambda_g) - (S_{g-1}\kappa_g)(a_g \odot \kappa_g)^\top + v_g k_g^\top, \qquad o_g = S_g q_g. \tag{3}$$

The three terms decay the stored state, erase a selected component, and write a key–value outer product. The vectors depend smoothly on the mixed group (Appendix A). Gated head outputs are added to the residual stream. The state starts from $S_0 = 0$ at every point and in every block, so it never passes between points and the ordering of points in a batch does not affect the prediction. The

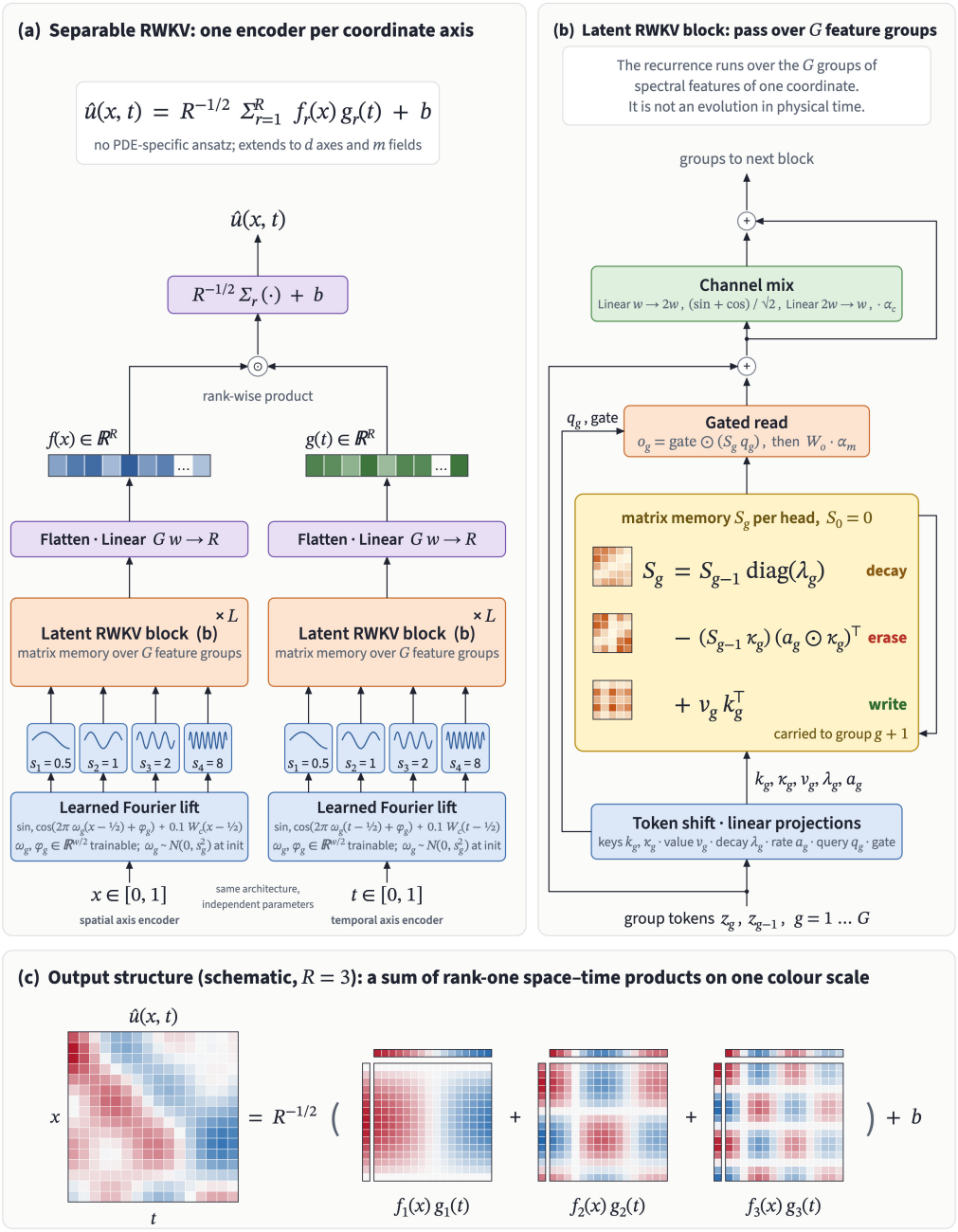


Figure 1: S-RWKV and its ablations. (a) Each coordinate has its own branch that maps groups of learned Fourier features to R field factors. (b) One RWKV block over the G feature groups of a coordinate. The memory path passes a matrix state S_g from group to group, and the channel-mixing path transforms each group on its own. No memory retains two channel-mixing layers and removes the memory path. No blocks removes both paths. (c) The field is a sum of rank-one products of coordinate factors, shown schematically for $R = 3$.

channel-mixing path then transforms each group independently. Let $\bar{h}_g^{(\ell)}$ denote the group after the memory residual update, or $h_g^{(\ell)}$ when the memory path is absent:

$$h_g^{(\ell+1)} = \bar{h}_g^{(\ell)} + \alpha_c^{(\ell)} \left[W_2^{(\ell)} \psi\left(W_1^{(\ell)} \bar{h}_g^{(\ell)} + b_1^{(\ell)} \right) + b_2^{(\ell)} \right], \qquad \psi(s) = \tfrac{1}{\sqrt{2}}(\sin s + \cos s), \tag{4}$$

with widths $w \to 2w \to w$. Layer parameters are shared across groups, but not across coordinate branches. The activation ψ is the sine–cosine activation of PINNsFormer with fixed, equal weights (Zhao et al., 2024). After the last block, the G groups are concatenated and projected to R factors per field. All operations are smooth, so the second-order coordinate derivatives in the residual are available by automatic differentiation.

3.3 Configurations and Controls

Four configurations share Eqs. (1)–(2) and the output projection. ***S-RWKV*** (full) uses two complete blocks. ***One block*** uses one. ***No memory*** applies two channel-mixing layers (4) and removes group mixing, the memory projections, and the memory readout, so no information passes between groups before the final projection. ***No blocks*** projects the Fourier features (2) directly. The memory ablation therefore removes the complete memory path, not only the state update.

Two controls compare the branch body and the field assembly. ***SPINN*** uses its original $\tanh$ MLP branches with the Fourier input (2), so SPINN and S-RWKV differ only in the branch body. ***Joint RWKV*** is non-separable: it computes Fourier features of the coordinate vector, processes them with three RWKV-based blocks over feature groups, and predicts the field with one output projection (Figure 3). Its width is 40 on Reaction, Convection, and Wave, with $54\,887$ parameters versus $55\,369$ for S-RWKV. On Navier–Stokes, its width is 56, with $107\,025$ parameters versus $107\,823$. An input–assembly–body ablation (Appendix D) also compares coordinate and Fourier inputs across these constructions.

3.4 Training Pipeline and Output-Projection Refinement

For $\mathcal{N}[u] = 0$ with condition operators $\mathcal{B}_k[u] = g_k$, the loss is

$$\mathcal{L}(\theta) = \lambda_r \,\mathrm{MSE}_{C_r}[\mathcal{N}[\widehat{u}_\theta]] + \sum_k \lambda_k \,\mathrm{MSE}_{C_k}[\mathcal{B}_k[\widehat{u}_\theta] - g_k], \tag{5}$$

where $\mathrm{MSE}_C[v] = |C|^{-1} \sum_{z \in C} \|v(z)\|_2^2$. Condition terms include initial and boundary constraints and, where available, observations. Weights are problem-specific and identical across architectures. Derivatives are taken with respect to physical coordinates.

On Reaction, Convection, and Wave, training of all parameters is followed by *output-projection refinement*. Internal features are frozen, and the coordinate projections are updated in alternation. With the other branches fixed, Eq. (1) is linear in the selected projection. Linear residuals and conditions therefore give a block least-squares problem; nonlinear residuals give a local Gauss–Newton subproblem. Both are solved by truncated singular value decomposition (SVD). A candidate is accepted only if it lowers physics-based criteria on both training and independent validation points (Appendices A.5 and C.1). Each separable update refits one coordinate projection and the global bias ($4\,129$ parameters for S-RWKV); a joint update refits one output layer (161 parameters for Joint RWKV). Navier–Stokes uses a separate Newton continuation with conjugate gradients (CG) for the output projections and biases (Appendix C.3). Refinement adds no parameters. Flow Mixing and Helmholtz use Adam only.

4 Experimental Setup

Benchmarks. Reaction, Convection ($\beta = 50$), Wave, and Navier–Stokes flow reconstruction come from the PINNsFormer suite (Zhao et al., 2024); Flow Mixing and Helmholtz come from SPINN (Cho et al., 2023). The equations and reference fields follow these sources (Appendix B). For Navier–Stokes, 2500 velocity observations are given, pressure is never observed, and errors are computed on the held-out slice $t = 10$ after aligning pressure by one constant. The reference solutions of Convection and Wave have CP rank two, and that of Helmholtz has rank one. The suite thus includes low-rank analytical fields as well as flow reconstruction from sparse observations.

Controlled pipeline. Within each problem, all architectures share the problem definition, residual weights, optimizer stages, float64 arithmetic, and model-selection rule; SPINN and the S-RWKV configurations also share the training script and samples. Reaction, Convection, and Wave use 1500

Table 1: Scalar problems: relative L_2 error, mean $\pm$ SD over ten runs; the lowest mean per column is in bold. Parameters and forward cost (GFLOPs for one field evaluation on 8192 interior points) refer to the Reaction, Convection, and Wave models; on Flow Mixing and Helmholtz, separable models have a third coordinate branch (costs in Appendix C.5). Reaction, Convection, and Wave use the refined multistage pipeline; Flow Mixing and Helmholtz use Adam only.

Model	Params	GFLOPs	Reaction	Convection	Wave	Flow Mixing	Helmholtz
PINNsFormer	453 561	11.751	4.04e-2 ± 1.01e-2	8.5e-3 ± 3.5e-4	6.9e-2 ± 9.4e-3	2.48e-2 ± 5.84e-3	5.22e-1 ± 9.76e-2
S-PFormer	370 991	9.615	6.33e-3 ± 2.64e-3	4.8e-2 ± 1.6e-2	7.7e-3 ± 6.3e-4	2.57e-2 ± 5.92e-3	4.71e-1 ± 1.19e-1
PINNMamba	285 763	40.721	2.14e-2 ± 8.14e-4	1.86e+0 ± 1.38e+0	6.8e-2 ± 1.7e-2	1.83e-2 ± 9.42e-3	1.83e-1 ± 4.70e-2
SPINN	75 841	1.221	6.95e-3 ± 7.08e-3	9.10e-1 ± 5.20e-3	3.54e-7 ± 1.86e-7	4.72e-3 ± 1.76e-3	3.00e-2 ± 1.08e-2
Joint RWKV	54 887	3.817	6.1e-2 ± 1.2e-2	1.00e+0 ± 5.81e-3	4.83e-1 ± 1.21e-1	5.50e-3 ± 1.08e-3	4.65e-2 ± 2.67e-2
S-RWKV (full)	55 369	3.386	5.43e-5 ± 7.58e-5	6.78e-6 ± 1.09e-5	5.79e-7 ± 8.60e-7	1.89e-3 ± 6.76e-4	2.22e-2 ± 7.47e-3
One block	32 005	1.761	8.02e-6 ± 1.05e-5	**3.63e-6 ± 1.16e-6**	**3.31e-7 ± 1.84e-7**	1.73e-3 ± 3.38e-4	2.53e-2 ± 7.45e-3
No memory	25 413	1.210	**3.25e-6 ± 2.81e-6**	6.26e-6 ± 9.86e-6	4.59e-7 ± 3.80e-7	**1.66e-3 ± 1.39e-4**	3.21e-2 ± 6.58e-3
No blocks	8 641	0.136	1.05e-4 ± 1.23e-4	1.54e-1 ± 3.42e-1	4.37e-2 ± 9.77e-2	1.11e-2 ± 3.27e-3	**2.20e-2 ± 5.14e-3**

Adam steps, 300 outer L-BFGS steps, and five refinement stages. Navier–Stokes adds SOAP (Vyas et al., 2024) and L-BFGS stages followed by Newton continuation stages that refine the output projections. Flow Mixing and Helmholtz use 50 000 Adam steps at learning rate 10^{-3} (Appendix C). Model states are selected by physics-based criteria on training and independent validation points; the reference solution is used only for evaluation.

Baselines and sampling. PINNsFormer, S-PFormer, and PINNMamba are retrained under this pipeline, which differs from their original protocols; we therefore also quote their published errors. On Flow Mixing and Helmholtz, separable models train on tensor-product grids of 256^3 and 64^3 points per step, and the other architectures on 1024 points per step (Appendix C.4). Comparisons with non-separable models on these two problems thus reflect both the architecture and its sampling budget. The comparison with SPINN matches the samples and training schedule.

Metric and statistics. We report the relative discrete L_2 error $\varepsilon_{\mathrm{rel}} = \|\widehat{u} - u_{\mathrm{ref}}\|_2 / \|u_{\mathrm{ref}}\|_2$, evaluated on a 201×201 grid for Reaction, Convection, and Wave. PINN objectives can be stiff and ill-conditioned (Wang et al., 2021a; Rathore et al., 2024), and training outcomes can depend strongly on numerical precision and initialization (Shao et al., 2026; Zhang et al., 2026). We use ten runs with distinct seeds for each main configuration and report the mean and sample standard deviation (SD). Full S-RWKV and No memory are compared with two-sided Welch tests at the 5% level per comparison. A non-significant result is not treated as evidence of equivalence.

5 Results

5.1 Accuracy without Recurrent Memory

Table 1 compares the four S-RWKV configurations with the baselines. On Reaction, Convection, Wave, and Flow Mixing, No memory has a lower mean error than the full model, although none of these Welch comparisons is significant. The result occurs both under the multistage pipeline and, for Flow Mixing, with Adam alone. On Reaction, Convection, and Wave, removing memory reduces parameters by 54% (25 413 versus 55 369) and forward FLOPs by 64% (1.210 versus 3.386 GFLOPs). One block gives the lowest mean errors on Convection and Wave.

Table 2: Navier–Stokes reconstruction: relative L_2 error on the held-out slice $t = 10$, mean ± SD over ten runs. Pressure is never observed.

Model	Params	u	v	p
PINNsFormer	454 106	9.93e-3 ± 3.35e-3	2.49e-2 ± 8.78e-3	7.31e-2 ± 3.03e-2
S-PFormer	369 488	2.22e-2 ± 1.57e-2	6.4e-2 ± 2.2e-2	8.8e-2 ± 8.9e-4
PINNMamba	286 308	**5.66e-3 ± 1.19e-3**	1.35e-2 ± 8.85e-3	5.3e-2 ± 2.3e-2
SPINN	138 531	1.06e-2 ± 6.72e-4	4.91e-2 ± 5.00e-3	6.36e-2 ± 1.38e-2
Joint RWKV	107 025	6.2e-3 ± 2.1e-3	**7.64e-3 ± 2.12e-3**	**1.36e-2 ± 2.04e-3**
S-RWKV (full)	107 823	7.0e-3 ± 1.1e-3	1.73e-2 ± 1.11e-3	1.90e-2 ± 1.05e-3
One block	72 777	6.7e-3 ± 1.0e-3	5.6e-2 ± 1.8e-2	2.14e-2 ± 2.77e-3
No memory	62 889	7.4e-3 ± 1.4e-3	1.67e-2 ± 1.33e-3	2.14e-2 ± 3.12e-3
No blocks	37 731	5.87e-3 ± 1.83e-3	2.67e-2 ± 1.11e-2	2.71e-2 ± 1.34e-2

Table 3: Convection: mean relative L_2 error over ten runs in the input–assembly–body ablation. The joint MLP with coordinate input is the classical PINN, the separable MLP is SPINN, the separable RWKV is S-RWKV, and separable channel mixing (Ch. mixing) is No memory; SDs, Reaction, and Wave are in Appendix D.

		Input lift	
Assembly	Body	Coord.	Fourier
Joint	MLP	7.5e-1	9.6e-1
Joint	RWKV	4.8e+0	1.00e+0
Separable	MLP	5.7e+0	9.10e-1
Separable	RWKV	7.3e-2	**6.78e-6**
Separable	Ch. mixing	6.2e-3	**6.26e-6**

The full model has lower mean errors in two comparisons. On Navier–Stokes, the full model reconstructs the unobserved pressure more accurately than No memory ($1.90 \cdot 10^{-2}$ versus $2.14 \cdot 10^{-2}$, significant) and has the lowest pressure error among the four configurations. Neither velocity comparison is significant (Table 2). On Helmholtz, the full model is significantly more accurate than No memory ($2.22 \cdot 10^{-2}$ versus $3.21 \cdot 10^{-2}$), but No blocks, which has neither memory nor channel mixing, reaches $2.20 \cdot 10^{-2}$, so memory is not required to reach this error level. The memory path accounts for a substantial part of the branch cost. On Navier–Stokes, removing it reduces the parameter count by 42% and the mean peak GPU allocation over the whole pipeline from 11.63 to 3.04 GiB (-74%; Appendix C.3).

5.2 Channel Mixing and Training Reliability

Removing the channel-mixing path as well (No blocks) projects the learned Fourier features directly onto the coordinate factors. On Convection and Wave, No blocks has mean errors of $1.5 \cdot 10^{-1}$ and $4.4 \cdot 10^{-2}$ with SDs more than twice as large, whereas the three S-RWKV configurations with channel mixing have mean errors below 10^{-5} and 10^{-6}, respectively (Table 1). Single No blocks runs can still be very accurate (Figures 5 and 6), which suggests that channel mixing improves training reliability rather than simply making these solutions representable. On Helmholtz and for the Navier–Stokes velocity u, No blocks has a slightly lower mean error than the full model. The required feature processing is therefore task-dependent.

The choice of nonlinear transformation also matters. SPINN uses the same Fourier input, assembly, and training pipeline, with nearly the same forward cost (1.221 versus 1.210 GFLOPs), but has a mean error of $9.10 \cdot 10^{-1}$ on Convection and $6.95 \cdot 10^{-3}$ on Reaction, where No memory reaches $6.26 \cdot 10^{-6}$ and $3.25 \cdot 10^{-6}$. On Wave, both reach errors of order 10^{-7}.

The input–assembly–body ablation (Table 3) shows that no single factor suffices on Convection. Fourier input alone does not yield accurate solutions for either MLP assembly or Joint RWKV under this protocol. Separable assembly alone is also insufficient for the MLP. Among the tested configurations, mean errors below 10^{-5} occur only when Fourier input, separable assembly, and channel mixing over feature groups are combined. Even with raw coordinate input, the two grouped separable bodies ($7.3 \cdot 10^{-2}$ and $6.2 \cdot 10^{-3}$) have lower mean Convection errors than every joint or MLP-bodied model with Fourier input. On Wave, by contrast, a joint MLP with Fourier input reaches $4.9 \cdot 10^{-7}$, so separability is not always required.

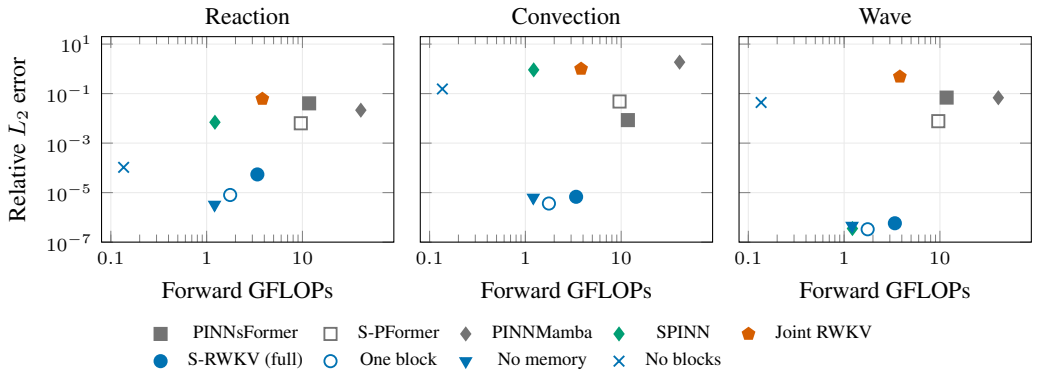


Figure 2: Accuracy against forward cost on the three benchmarks shared by the sequence-model PINNs, all trained with one float64 pipeline. Markers show mean relative L_2 errors over ten runs (SDs in Table 1); cost is one forward evaluation on the 8192 interior points of a training step. No memory lies more than three orders of magnitude below every sequence model at 8–34$\times$ lower cost.

The SPINN and S-RWKV bodies differ in three respects: activation ($\tanh$ versus ψ), weight sharing across feature groups, and scaled residual updates. These ingredients form the branch design compared here. PINNMamba also reports an increase of more than an order of magnitude in Convection error when its sine–cosine activation is replaced by $\tanh$ (Xu et al., 2025a).

The stage-wise errors (Appendix C.2) show where these differences arise. After the Adam stage, every architecture has mean errors between 3.6 and 7.6 on Reaction, Convection, and Wave. The L-BFGS stage separates them: the configurations with channel mixing reach $7 \cdot 10^{-5}$ or below, whereas SPINN and Joint RWKV remain at 1.2 on Convection. Refinement then reduces every architecture's mean error by a factor of approximately 1.1–1.5. It updates 4 129 coefficients per stage in the S-RWKV configurations, compared with 161 in Joint RWKV. The main differences between branch bodies are therefore already present before output-projection refinement.

5.3 Comparison with Sequence-Model PINNs and the Joint Control

Under the shared pipeline, both full S-RWKV and No memory have lower mean errors than PINNsFormer, S-PFormer, and PINNMamba on all five scalar problems (Table 1). On Reaction, Convection, and Wave, the benchmarks shared by all three original papers, No memory is more than three orders of magnitude more accurate than each of them, with 11–18$\times$ fewer parameters and 8–34$\times$ fewer FLOPs (Figure 2); the gap is already present before refinement (Appendix C.2).

For context, under its original protocol, PINNsFormer reports relative L_2 errors of $2.7 \cdot 10^{-2}$ on Convection and $3.0 \cdot 10^{-2}$ on Reaction (Zhao et al., 2024, Table 1). PINNMamba reports $2.0 \cdot 10^{-2}$, $2.2 \cdot 10^{-2}$, and $2.0 \cdot 10^{-2}$ on Convection, Reaction, and Wave (Xu et al., 2025a, Table 1). These reported errors are also more than three orders of magnitude larger than those of No memory. PINNMamba has a substantially higher Convection error under our pipeline (mean 1.86) than under its published protocol.

On Flow Mixing and Helmholtz, where the architecture-only comparison is with SPINN, No memory reduces SPINN's error by a factor of 2.8 on Flow Mixing, and full S-RWKV reduces it by 26% on Helmholtz, where No memory has a 7% higher mean error than SPINN.

The joint control has the same kind of body and a similar parameter count, but has larger mean errors than full S-RWKV on all five scalar problems. Navier–Stokes reverses this ordering (Table 2): Joint RWKV is more accurate than full S-RWKV and No memory on all three fields and gives the lowest errors for v and p, while PINNMamba gives the lowest error for u. Full S-RWKV and No memory both have lower mean errors than SPINN on all three fields.

6 Discussion and Limitations

The memory-free branch achieves high accuracy without a recurrent state on several tasks. This is consistent with MetaFormer's finding that block structure can remain effective when the token mixer is simplified (Yu et al., 2022). The stage-wise errors locate the main differences in gradient-based training. With the same Adam and L-BFGS schedule, channel mixing over grouped Fourier features reaches low errors on Reaction, Convection, and Wave. The MLP controls reach comparable errors on Wave, but not on Reaction or Convection, while output-projection refinement changes mean errors by at most a factor of 1.5. The memory-free branch therefore provides features that can be fitted accurately under the shared protocol. This architectural effect is distinct from changes to arithmetic precision or residual regularization (Xu et al., 2025b; Andersen & Matsubara, 2026).

Limitations. The conclusions concern the tested coordinate branches and task-specific protocols. The low-rank reference fields in Convection, Wave, and Helmholtz favor separable representations, while the joint control is more accurate on Navier–Stokes reconstruction. The comparison with SPINN evaluates the complete branch design, including activation, group-wise parameter sharing, and residual scaling. The results therefore compare architectures under a common protocol rather than establish a universal ranking of the methods.

7 Conclusion

For the separable PINNs studied here, lightweight channel mixing is an effective alternative to full recurrent processing. Removing the memory path yields lower mean errors on four benchmarks and reduces parameter count, forward cost, and GPU memory. Removing the channel transformations as well increases mean errors and variation across runs on Convection and Wave. The resulting memory-free model provides a compact design for learning separable coordinate factors, while the contribution of recurrent memory remains task-dependent.

AI use statement

Large language model tools were used to translate, draft, and edit the text, to search and check the literature, to typeset the paper and the summary plot, and to check the consistency of reported numbers. All experiments and numerical results were produced by the authors, who take full responsibility for the content.

Reproducibility statement

Appendix A specifies the architectures and the refinement stage, Appendix B the problems, and Appendix C the training protocols, errors after each training stage, sampling, precision, statistics, and cost measurements. Appendix D reports the input–assembly–body ablation.

References

Sokratis J. Anagnostopoulos, Juan Diego Toscano, Nikolaos Stergiopulos, and George Em Karniadakis. Residual-based attention in physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116805, 2024. doi: 10.1016/j.cma.2024.116805. URL https://doi.org/10.1016/j.cma.2024.116805.

Nigel T. Andersen and Takashi Matsubara. PINNs failure modes are overfitting. arXiv:2605.30910, 2026. URL https://arxiv.org/abs/2605.30910.

Rohan Arni and Carlos Blanco. Physics-informed neural networks with Fourier features and attention-driven decoding. In *NeurIPS 2025 AI4Science Workshop*, 2025. URL https://arxiv.org/abs/2510.05385.

Jingrun Chen, Xurong Chi, Weinan E, and Zhouwang Yang. Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method. *Journal of Machine Learning*, 1(3):268–298, 2022. doi: 10.4208/jml.220726. URL https://arxiv.org/abs/2207.13380.

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 23761–23788, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4af827e7d0b7bdae6097d44977e87534-Abstract-Conference.html.

Eric C. Cyr, Mamikon A. Gulian, Ravi G. Patel, Mauro Perego, and Nathaniel A. Trask. Robust Training and Initialization of Deep Neural Networks: An Adaptive Basis Viewpoint. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pp. 512–536. PMLR, 2020. URL https://proceedings.mlr.press/v107/cyr20a.html.

Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (PIELM)—a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391: 96–118, 2020. doi: 10.1016/j.neucom.2019.12.099. URL https://doi.org/10.1016/j.neucom.2019.12.099.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative Filter Networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=OmtmcPkkhT.

Gene Golub and Victor Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19(2):R1–R26, 2003. doi: 10.1088/0266-5611/19/2/201. URL https://doi.org/10.1088/0266-5611/19/2/201.

Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, 2012. doi: 10.1137/100818893. URL https://doi.org/10.1137/100818893.

Anas Jnini, Flavio Vella, and Marius Zeinhofer. Gauss-Newton Natural Gradient Descent for Physics-Informed Computational Fluid Dynamics. arXiv:2402.10680, 2024. URL https://arxiv.org/abs/2402.10680.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X. URL https://doi.org/10.1137/07070111X.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2109.01050.

Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable Fourier Features for Multi-Dimensional Spatial Positional Encoding. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/84c2d4860a0fc27bcf854c444fb8b400-Abstract.html.

Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023. doi: 10.1016/j.jcp.2022.111722. URL https://doi.org/10.1016/j.jcp.2022.111722.

Johannes Müller and Marius Zeinhofer. Achieving high accuracy with PINNs via energy natural gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25471–25485. PMLR, 2023. URL https://proceedings.mlr.press/v202/muller23b.html.

Elizabeth Newman, Lars Ruthotto, Joseph Hart, and Bart van Bloemen Waanders. Train Like a (Var)Pro: Efficient Training of Neural Networks with Variable Projection. *SIAM Journal on Mathematics of Data Science*, 3(4):1041–1066, 2021. doi: 10.1137/20M1359511. URL https://doi.org/10.1137/20M1359511.

Anthony Nouy. A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23–24):1603–1626, 2010. doi: 10.1016/j.cma.2010.01.009. URL https://doi.org/10.1016/j.cma.2010.01.009.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, 2023. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Xingjian Du, Teddy Ferdinan, Haowen Hou, Przemysław Kazienko, Kranthi Kiran GV, Jan Kocoń, Bartłomiej Koptyra, Satyapriya Krishna, Ronald McClelland, Jr., Jiaju Lin, Niklas Muennighoff, Fares Obeid, Atsushi Saito, Guangyu Song, Haoqin Tu, Cahya Wirawan, Stanisław Woźniak, Ruichong Zhang, Bingchen Zhao, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. Eagle and Finch: RWKV with Matrix-Valued States and Dynamic Recurrence. In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.05892.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. RWKV-7 "Goose" with expressive dynamic state evolution. arXiv:2503.14456, 2025. URL https://arxiv.org/abs/2503.14456.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045. URL https://doi.org/10.1016/j.jcp.2018.10.045.

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training PINNs: A loss landscape perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42159–42191, 2024. URL https://proceedings.mlr.press/v235/rathore24a.html.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Yingjie Shao, Ioannis N. Athanasiadis, George van Voorn, and Taniya Kapoor. Curvature-aware dynamic precision approach for physics-informed neural networks. *Neurocomputing*, 703: 134698, 2026. doi: 10.1016/j.neucom.2026.134698. URL https://arxiv.org/abs/2606.04736. Available as arXiv:2606.04736.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in*

Neural Information Processing Systems, volume 33, 2020. URL https://arxiv.org/abs/2006.09661.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/55053683268957697aa39fba6f231c68-Abstract.html.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam. arXiv:2409.11321, 2024. URL https://arxiv.org/abs/2409.11321.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021a. doi: 10.1137/20M1318043. URL https://epubs.siam.org/doi/10.1137/20M1318043.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021b. doi: 10.1016/j.cma.2021.113938. URL https://doi.org/10.1016/j.cma.2021.113938.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022a. doi: 10.1016/j.jcp.2021.110768. URL https://doi.org/10.1016/j.jcp.2021.110768.

Sifan Wang, Bowen Li, Yuhan Chen, and Paris Perdikaris. PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks. *Journal of Machine Learning Research*, 25(402):1–51, 2024. URL https://jmlr.org/papers/v25/24-0313.html.

Yifan Wang, Pengzhan Jin, and Hehu Xie. Tensor Neural Network and Its Numerical Integration. arXiv:2207.02754, 2022b. URL https://arxiv.org/abs/2207.02754.

Jian Cheng Wong, Chinchun Ooi, Abhishek Gupta, and Yew-Soon Ong. Learning in Sinusoidal Spaces With Physics-Informed Neural Networks. *IEEE Transactions on Artificial Intelligence*, 5(3):985–1000, 2024. doi: 10.1109/TAI.2022.3192362. URL https://doi.org/10.1109/TAI.2022.3192362.

Xiong Xiong, Zhuo Zhang, Rongchun Hu, Chen Gao, and Zichen Deng. Separated-variable spectral neural networks: A physics-informed learning approach for high-frequency PDEs. arXiv:2508.00628, 2025. URL https://arxiv.org/abs/2508.00628v1.

Chenhui Xu, Dancheng Liu, Yuting Hu, Jiajie Li, Ruiyang Qin, Qingxiao Zheng, and Jinjun Xiong. Sub-sequential physics-informed learning with state space model. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69507–69525, 2025a. URL https://proceedings.mlr.press/v267/xu25t.html.

Chenhui Xu, Dancheng Liu, Amir Nassereldine, and Jinjun Xiong. FP64 is all you need: Rethinking failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2505.10949.

Qihong Yang and Qiaolin He. Multi-scale separable Fourier neural networks for solving high-frequency PDEs. arXiv:2605.31027, 2026. URL https://arxiv.org/abs/2605.31027v4.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. MetaFormer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10819–10829, 2022. doi: 10.1109/CVPR52688.2022.01055.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Jinyuan Zhang, Peng He, He Hu, Yin Yuan, and ShengShuo Jiao. Neither precision nor architecture alone: Controlled tests of failure remedies for physics-informed neural networks. arXiv:2608.25327, 2026. URL https://arxiv.org/abs/2608.25327.

Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. PINNsFormer: A transformer-based framework for physics-informed neural networks. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.11833.

A Full Architecture Specification

A.1 Joint and Separable Parameterizations

Joint RWKV constructs a joint coordinate representation, while S-RWKV constructs independent coordinate factors (Figure 3). Figure 1 shows the full recurrent configuration; the No memory configuration retains the channel transformation in Eq. (4) and omits the state pathway. The notation z_g in the diagram corresponds to h_g in the formulas below. The panel with $R = 3$ illustrates the assembly structure schematically rather than the decomposition of a particular learned solution.

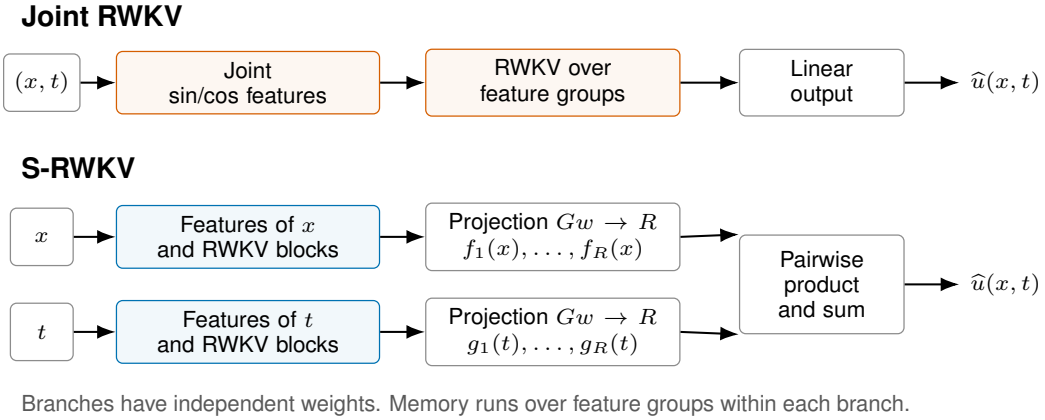


Figure 3: Joint RWKV and full S-RWKV. Joint RWKV processes coordinates jointly. In full S-RWKV, independent branches construct the factors of a separable field. In both configurations shown here, recurrence runs over feature groups; the state is not passed between physical points.

A.2 Normalization and Fourier Features

For a coordinate $z \in [z_{\min}, z_{\max}]$, we set $\widetilde{z} = (z - z_{\min})/(z_{\max} - z_{\min})$ and $\xi = \widetilde{z} - 1/2$. In each group, the frequency vector is initialized as $\omega_g \sim \mathcal{N}(0, s_g^2 I_{w/2})$, and the phase components are sampled independently and uniformly from $[-\pi, \pi]$. Sines and cosines are concatenated into a vector of width w. An affine mapping $E : \mathbb{R} \to \mathbb{R}^w$, shared by the groups within a branch, is added with a coefficient of 0.1. The frequencies, phases, and parameters of E are learned.

A.3 Group Mixing and Memory Heads

Let h_g be the input group of a block, and set $h_0 = h_1$. For $g = 1, \ldots, G$, we compute

$$\widetilde{h}_g = h_g + \sigma(\mu) \odot (h_{g-1} - h_g), \tag{6}$$

$$[\widetilde{q}_g, \widetilde{k}_g, \widetilde{v}_g, \widetilde{d}_g, \widetilde{a}_g, \widetilde{\gamma}_g] = W\widetilde{h}_g + c. \tag{7}$$

Here, σ is the logistic sigmoid and $\odot$ denotes elementwise multiplication. The projected vectors are split into $H = 4$ heads of width $w_h = w/H$. For one head,

$$q_g = \tanh(\widetilde{q}_g), \qquad \overline{k}_g = \tanh(\widetilde{k}_g), \qquad k_g = \overline{k}_g/\sqrt{w_h}, \tag{8}$$

$$\kappa_g = \frac{c_\kappa \odot \overline{k}_g}{\sqrt{1 + \|c_\kappa \odot \overline{k}_g\|_2^2}}, \qquad a_g = \sigma(\widetilde{a}_g), \qquad v_g = \widetilde{v}_g, \tag{9}$$

$$\lambda_g = \exp\left[-e^{-1/2}\sigma(\widetilde{d}_g + \beta_\lambda)\right], \qquad \gamma_g = \sigma(\widetilde{\gamma}_g). \tag{10}$$

The learnable scale c_κ is initialized to ones, the decay bias β_λ to -1, and the mixing parameter μ to zero. Memory S_g is updated by (3), with $S_0 = 0$ for each point and each block. The exact range of the decay is $\lambda_g \in (e^{-e^{-1/2}}, 1)$; its lower bound is approximately 0.545239.

The gated head outputs are concatenated before the output projection:

$$h_g^{\mathrm{mem}} = h_g + \alpha_m[W_o \,\mathrm{concat}_{\mathrm{heads}}(\gamma_g \odot (S_g q_g)) + b_o]\,, \tag{11}$$

$$h_g^{\mathrm{out}} = h_g^{\mathrm{mem}} + \alpha_c\big[W_2\psi(W_1 h_g^{\mathrm{mem}} + b_1) + b_2\big]\,, \tag{12}$$

$$\psi(s) = (\sin s + \cos s)/\sqrt{2}. \tag{13}$$

The channel-mixing path has dimensions $w \to 2w \to w$. The learnable scalars α_m, α_c are initialized to 0.1. No normalization layers or stochastic operations are used when evaluating the field. After the last block, the groups are concatenated into a vector of length Gw and projected to R components for each output field.

A.4 Derivatives of the Separable Field

For a scalar space–time solution, the derivatives are

$$\partial_t \widehat{u}(x,t) = R^{-1/2}\sum_r f_r(x) g_r'(t), \qquad \partial_{xx}\widehat{u}(x,t) = R^{-1/2}\sum_r f_r''(x) g_r(t). \tag{14}$$

Primes denote derivatives with respect to physical coordinates. If derivatives are first computed with respect to the normalized input $\widetilde{z} = (z - z_{\min})/(z_{\max} - z_{\min})$, conversion to a physical derivative of order k introduces the factor $(z_{\max} - z_{\min})^{-k}$. These factors are included in all physics residuals.

A.5 Output-Projection Refinement

For the scalar alternating refinement, write the output of the selected branch as $F_r^{(a)}(z_a) = W_r h^{(a)}(z_a) + c_r$. With the internal features and the other branches fixed, let $P_r(z_{-a}) = \prod_{b\neq a} F_r^{(b)}(z_b)$. Then

$$\widehat{u}(z) = R^{-1/2}\sum_r P_r(z_{-a})\big(W_r h^{(a)}(z_a) + c_r\big) + b = \Phi(z)^\top \alpha, \tag{15}$$

where α collects the selected W_r, c_r and the global bias. Linearity holds for one output projection with the other branches fixed. Joint variation of several coordinate projections is generally nonlinear. For a linear operator and linear conditions, the weighted residual vector is $A\alpha - y$, and refinement minimizes $\|A\alpha - y\|_2^2$. For a nonlinear residual $r(\alpha)$, we use the local model $r(\alpha + \Delta\alpha) \approx r(\alpha) + J(\alpha)\Delta\alpha$. Alternating projection refinement changes the coefficients of the learned field while preserving its structure and parameter count.

Fixed random features and learned bases. RFM (Chen et al., 2022) and PIELM (Dwivedi & Srinivasan, 2020) determine coefficients in trial spaces constructed from randomized features. PIELM uses a pseudoinverse of the collocation system for linear PDEs. MS-SFNN (Yang & He, 2026) couples separable cosine bases and coordinate-wise frequency scaling with least-squares fitting of the final combination coefficients. Our refinement also operates in a fixed feature space, but only after frequencies, phases, and internal transformations have been trained jointly. Moreover, it updates a selected coordinate projection with the other factors fixed, rather than only the final coefficients of a fully fixed product basis. This connects it to adaptive-basis training (Cyr et al., 2020) and block-coordinate least-squares methods. No blocks also learns its frequencies and phases before refinement.

Alternating solves and variable projection. With the other branches fixed, updating one output projection changes one coordinate factor within its learned basis. For linear equations and conditions, this is an ALS-type block least-squares step. The same block-coordinate principle is used in CP approximation (Kolda & Bader, 2009) and in alternating linear schemes for tensor-train solvers (Holtz et al., 2012); the latter update tensor-train cores rather than neural coordinate factors. For nonlinear residuals, the scalar stages use a locally linearized block problem. Variable projection instead eliminates linear variables while optimizing the remaining nonlinear parameters (Golub & Pereyra, 2003; Newman et al., 2021). Here, internal features stay fixed throughout refinement, and output coefficients are updated in alternation without reduced optimization of the internal parameters.

Energy natural gradients. ENGD (Müller & Zeinhofer, 2023) uses the Hessian of the objective in function space to define a metric on parameter-induced field variations. For quadratic residual energies of linear PDEs, its Gram matrix pairs the residual operators applied to parameter derivatives of the field. Restricting these variations to one output projection gives the geometry of a block least-squares problem. Our scalar stages restrict updates to these coefficients, solve sampled systems by truncated SVD, and accept candidates using independent physics-based checks. For nonlinear residuals, they use a local Gauss–Newton model rather than the full function-space Hessian.

A.6 Configurations

For Reaction, Convection, and Wave, the coordinate branch of full S-RWKV uses $G = 4$, $w = 32$, $H = 4$, $R = 32$, and $L = 2$; on Navier–Stokes, each branch outputs $3R$ components, R per field. No memory retains two channel-mixing layers but removes group mixing, the memory projections, and the memory readout. One block means $L = 1$ with a complete block retained. In No blocks, the output projection receives the learnable Fourier features directly. Parameter counts and forward costs of all configurations are listed in Tables 1, 2, and 8.

B Physical Problems

This appendix specifies the six main benchmark problems. The first four are taken from PINNsFormer (Zhao et al., 2024); Flow Mixing and Helmholtz follow the SPINN formulations (Cho et al., 2023). The training protocol is described in Appendix C. The supplementary Klein–Gordon ablation and an exploratory Allen–Cahn run are reported in Appendices E and F.

Fields and error distributions. Figures 4–11 show predicted fields and error maps for the six main problems. Model names are given above the panels; No memory, One block, and No blocks denote the corresponding S-RWKV configurations. MLP denotes the classical PINN network of the PINNsFormer benchmark (four linear layers of hidden width 512 with $\tanh$) with Fourier input features; on Reaction, Convection, and Wave it is the Fourier-input joint MLP of the input–assembly–body ablation (Appendix D), and on Navier–Stokes it is shown for reference only. Prediction rows alternate with error rows, and color scales appear on the right. For Flow Mixing, the error maps and panel L_2 values are evaluated at $t = 4$, without aggregation over time. For Helmholtz, the error maps are aggregated over z. Each panel shows a single run. The panel values are individual-run errors, whereas the tables report ten-run summaries.

B.1 Reaction

On $x \in [0, 2\pi]$, $t \in [0, 1]$, we solve

$$u_t - \rho u(1-u) = 0, \qquad \rho = 5, \qquad u(x,0) = u_0(x) = \exp\left[-\frac{(x-\pi)^2}{2(\pi/4)^2}\right]. \tag{16}$$

Periodic equality of field values is imposed at the ends of the spatial interval. The reference field is

$$u(x,t) = \frac{u_0(x)e^{\rho t}}{1 - u_0(x) + u_0(x)e^{\rho t}}. \tag{17}$$

The reference is used to evaluate error. The model state is selected using physics-based criteria.

B.2 Convection

On the same domain, we consider

$$u_t + \beta u_x = 0, \qquad \beta = 50, \qquad u(x,0) = \sin x \tag{18}$$

with periodic boundary values. The reference $u(x,t) = \sin(x - \beta t)$ can be expressed as a sum of two separable components. The model's coordinate factors are learned rather than set to the reference functions.

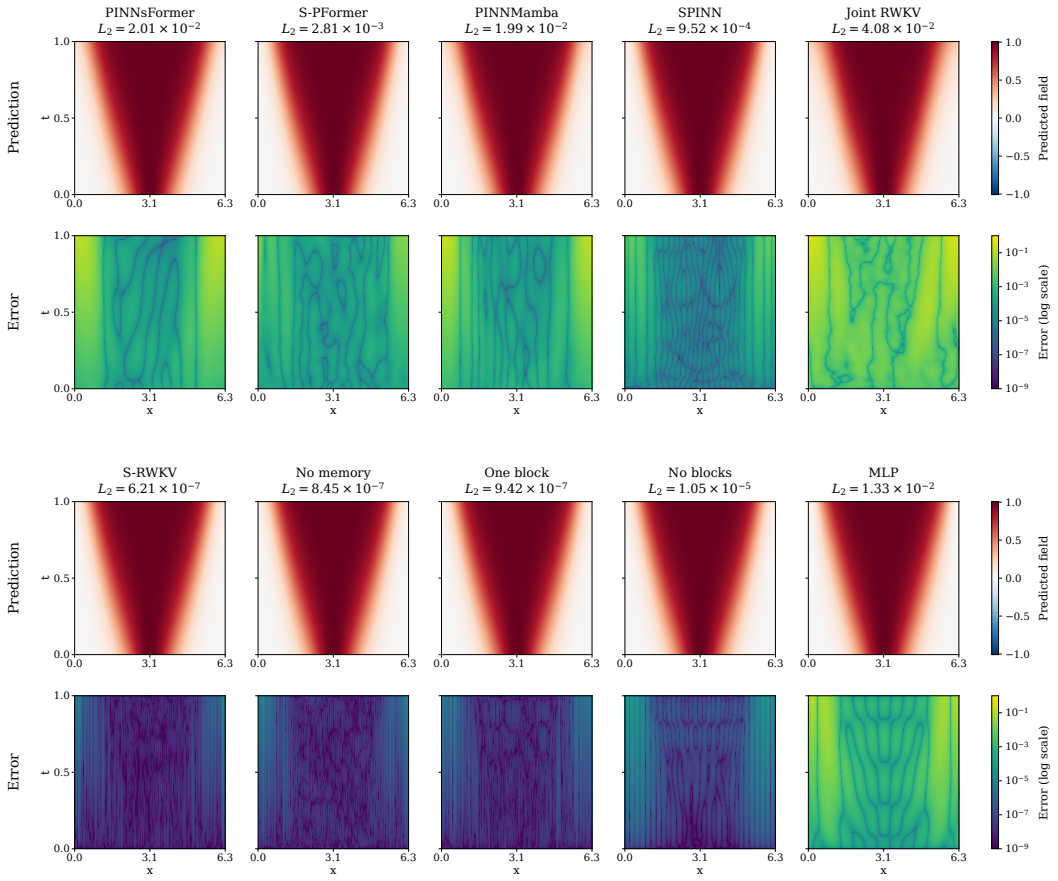


Figure 4: Reaction. Predicted fields (upper rows) and error maps (lower rows). The error color scale is logarithmic; the L_2 values refer to individual solutions.

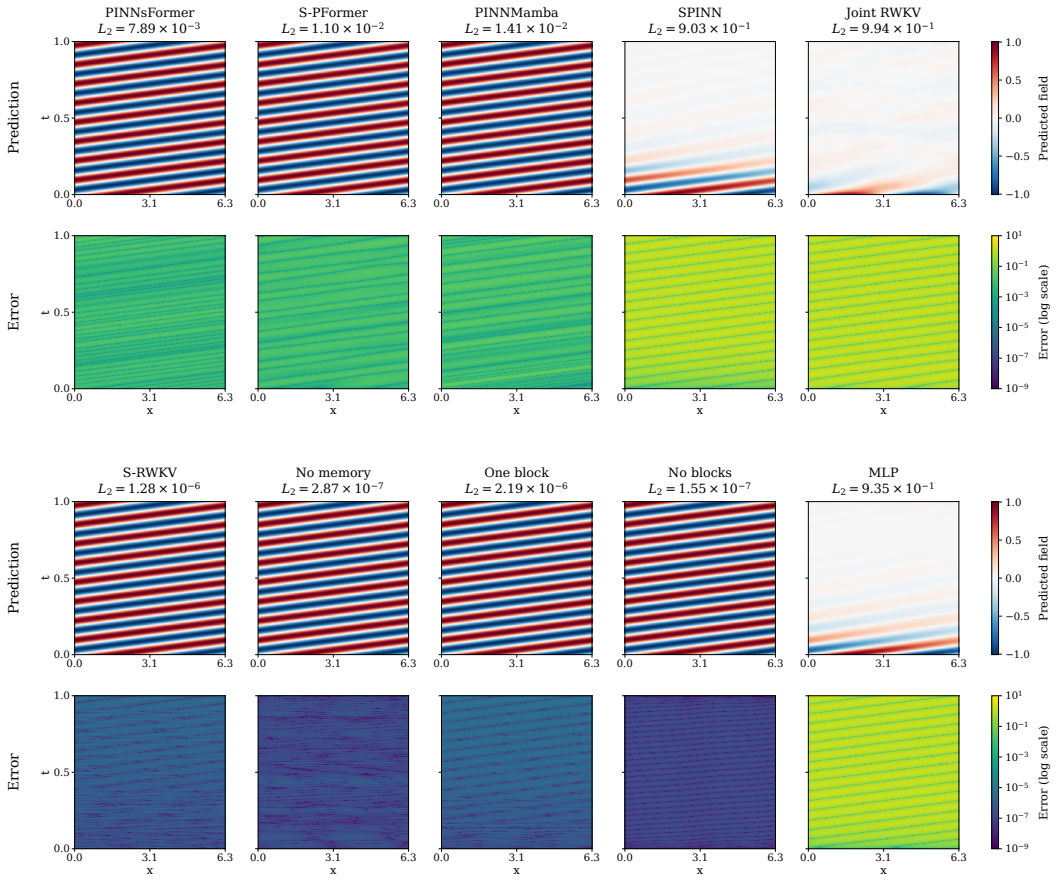


Figure 5: Convection. Predicted fields (upper rows) and error maps (lower rows). The error color scale is logarithmic; the L_2 values refer to individual solutions.

B.3 WAVE

On $(x, t) \in [0, 1]^2$, we use

$$u_{tt} - 4u_{xx} = 0, \tag{19}$$

$$u(x, 0) = \sin(\pi x) + \tfrac{1}{2}\sin(3\pi x), \qquad u_t(x, 0) = 0, \tag{20}$$

$$u(0, t) = u(1, t) = 0. \tag{21}$$

The reference field

$$u(x, t) = \sin(\pi x)\cos(2\pi t) + \tfrac{1}{2}\sin(3\pi x)\cos(6\pi t) \tag{22}$$

is also a sum of two separable components. The reference frequencies are not used to initialize the network.

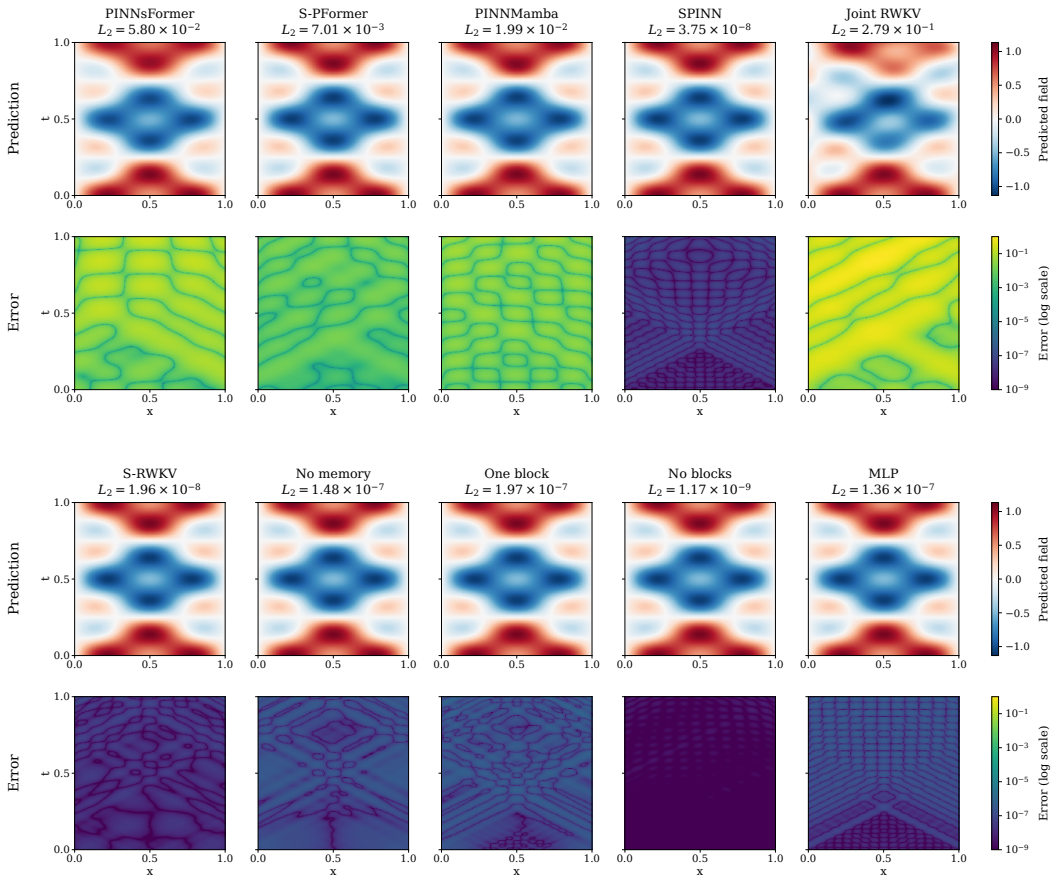


Figure 6: Wave. Predicted fields (upper rows) and error maps (lower rows). The error color scale is logarithmic; the L_2 values refer to individual solutions.

B.4 NAVIER–STOKES EQUATIONS

We use the public cylinder-wake dataset from the PINNs project (Raissi et al., 2019). On the domain $x \in [1, 8]$, $y \in [-2, 2]$, $t \in [0, 19.9]$, the equations are

$$u_t + uu_x + vu_y + p_x - \nu(u_{xx} + u_{yy}) = 0, \tag{23}$$

$$v_t + uv_x + vv_y + p_y - \nu(v_{xx} + v_{yy}) = 0, \tag{24}$$

$$u_x + v_y = 0, \qquad \nu = 0.01. \tag{25}$$

Training uses 2500 observations of the two velocity components: 1209 before $t = 10$ and 1291 after it. Evaluation uses 5000 spatial points on the $t = 10$ slice, which is excluded from the observations.

No separate initial or boundary velocity values are imposed. Viscosity is fixed, and pressure is reconstructed without observations. Its level is fixed by $p(1, -2, t) = 0$ at 65 time points.

Before evaluating pressure on the $t = 10$ slice, we compute a single constant offset

$$\delta_p = N^{-1} \sum_{i=1}^{N} (p_i^{\text{ref}} - \widehat{p}_i), \tag{26}$$

and then compare $\widehat{p} + \delta_p$ with p^{ref}. This procedure is identical for all models and does not change the spatial derivatives of pressure.

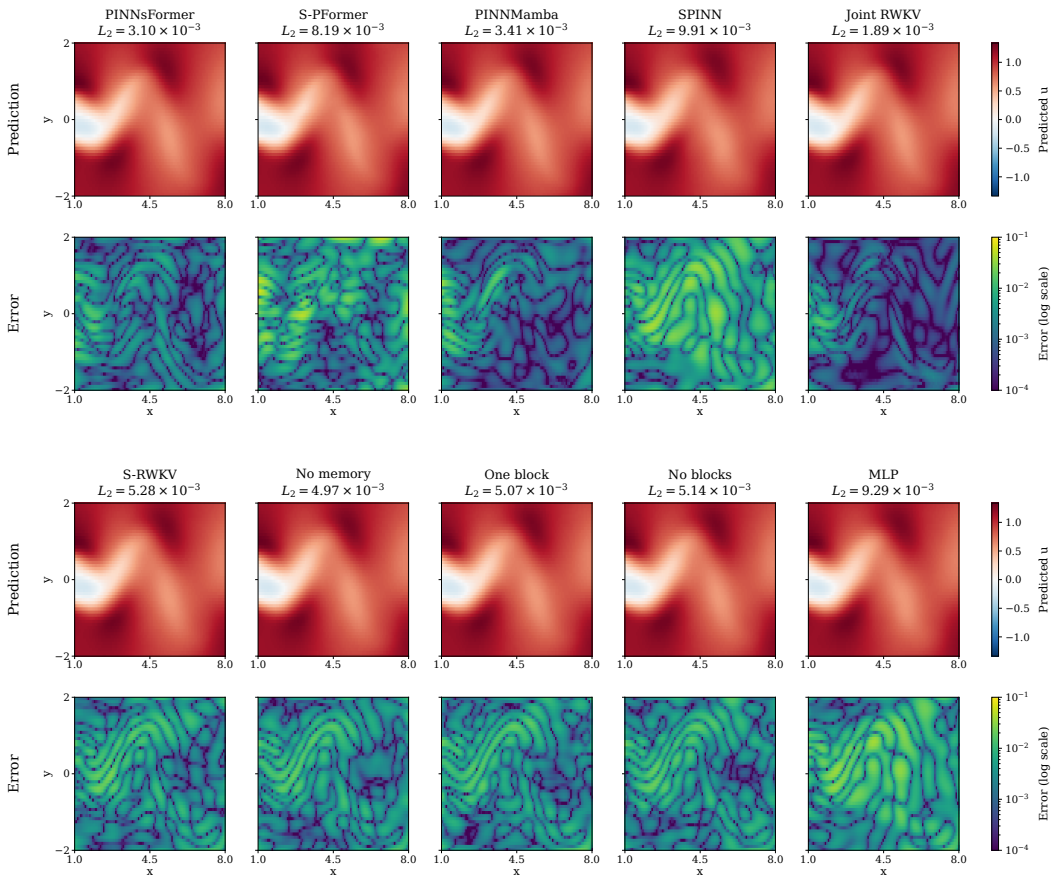


Figure 7: Navier–Stokes: velocity u on the $t = 10$ slice. Predicted fields and error maps with a logarithmic error color scale.

B.5 Flow Mixing

The SPINN formulation (Cho et al., 2023) describes transport in a nonuniform velocity field:

$$u_t + a(x, y)u_x + b(x, y)u_y = 0, \qquad (x, y, t) \in [-4, 4]^2 \times [0, 4]. \tag{27}$$

The velocity field is defined by

$$r = \sqrt{x^2 + y^2}, \quad \omega(r) = \frac{\operatorname{sech}^2(r) \tanh(r)}{0.385\, r}, \quad a = -\omega(r)y, \quad b = \omega(r)x. \tag{28}$$

We use the continuous extension $\omega(0) = 1/0.385$. The reference

$$u_{\text{ref}}(x, y, t) = -\tanh\left[\tfrac{1}{2}\big(y\cos(\omega t) - x\sin(\omega t)\big)\right] \tag{29}$$

provides the initial and boundary values.

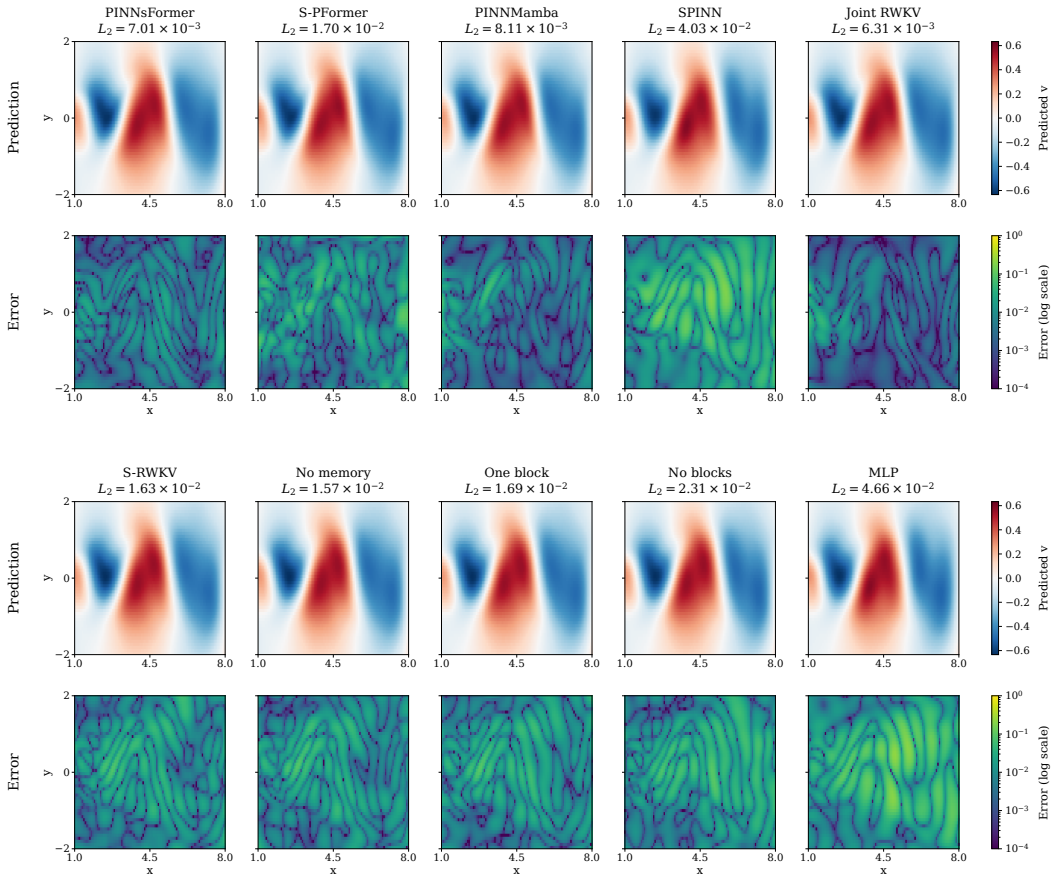


Figure 8: Navier–Stokes: velocity v on the $t = 10$ slice. Predicted fields and error maps with a logarithmic error color scale.

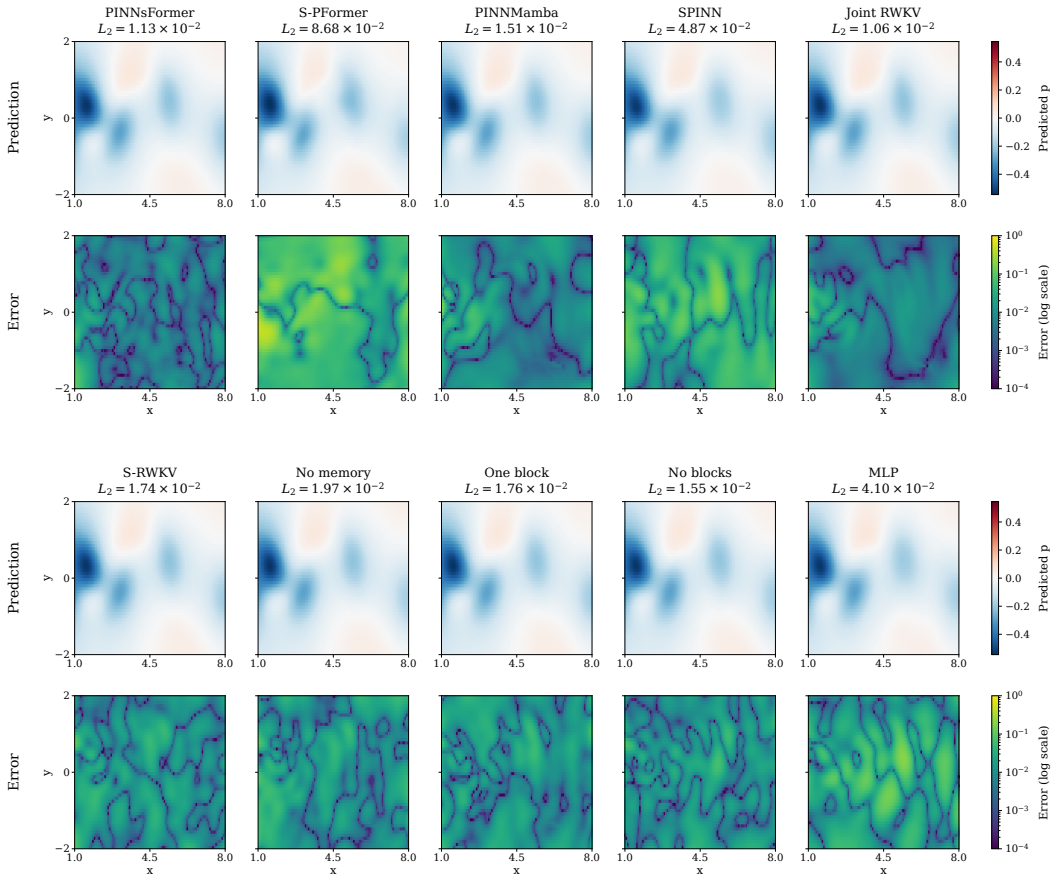


Figure 9: Navier–Stokes: pressure p on the $t = 10$ slice. Predicted fields and error maps with a logarithmic error color scale.

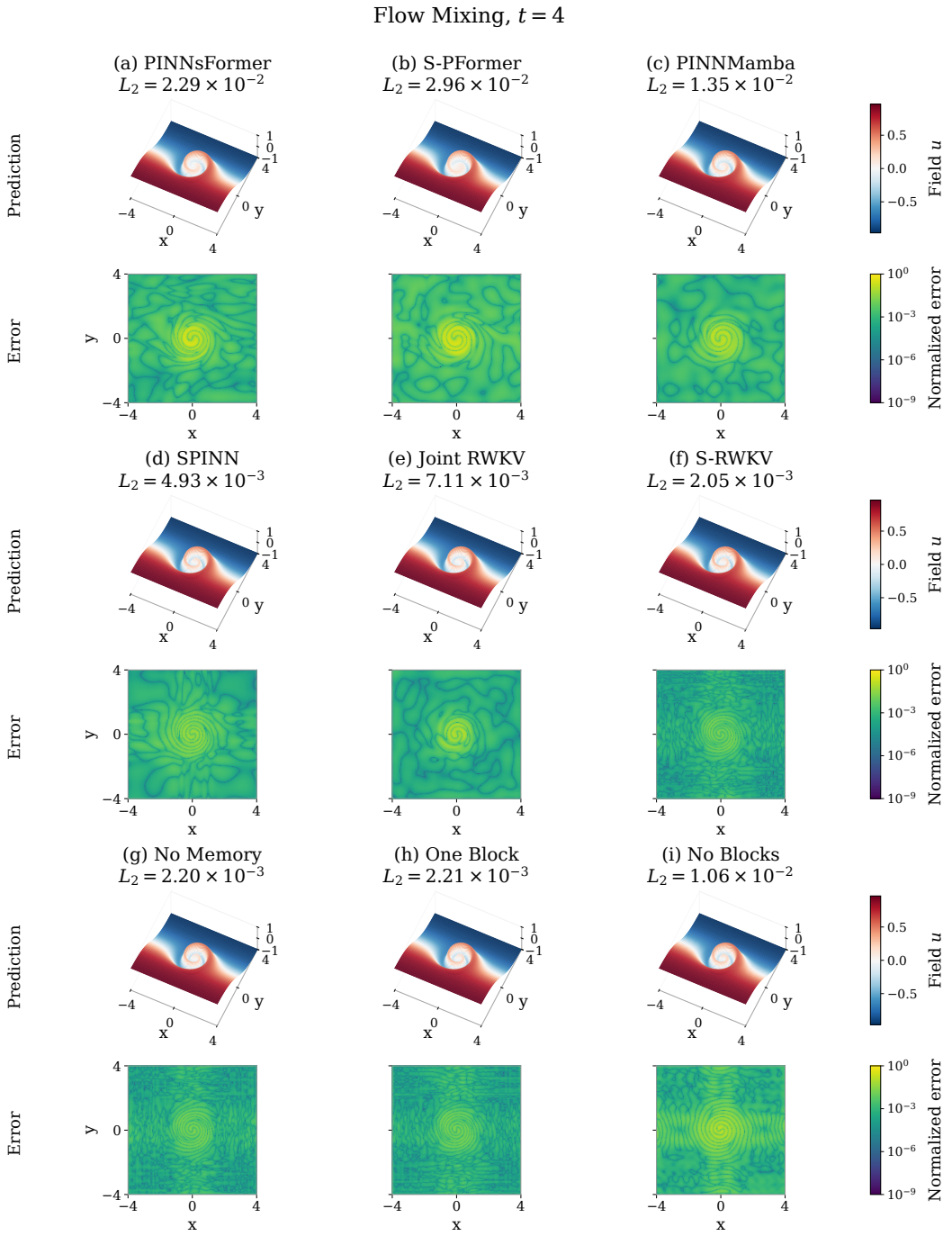


Figure 10: Flow Mixing at $t = 4$. Predicted fields and normalized error maps with a logarithmic error color scale. Both the error maps and the relative L_2 values above the panels are computed over the spatial (x, y) slice at $t = 4$, not over the full space–time domain.

B.6 HELMHOLTZ EQUATION

The SPINN formulation (Cho et al., 2023) is

$$\Delta u + k^2 u = q, \qquad x \in \Omega = [-1, 1]^3, \qquad u|_{\partial\Omega} = 0, \tag{30}$$

where we set $k = 1$ and $(a_1, a_2, a_3) = (4, 4, 3)$. The reference solution and consistent right-hand side are

$$u_{\mathrm{ref}}(x) = \prod_{j=1}^{3} \sin(a_j \pi x_j), \qquad q(x) = \Big(k^2 - \pi^2 \sum_{j=1}^{3} a_j^2\Big) u_{\mathrm{ref}}(x) = (1 - 41\pi^2) u_{\mathrm{ref}}(x). \tag{31}$$

The reference solution is a single product of coordinate functions. The coefficients a_j specify its spatial frequencies, while k is the Helmholtz wavenumber. The network's coordinate factors are learned independently of the analytical form of the reference.

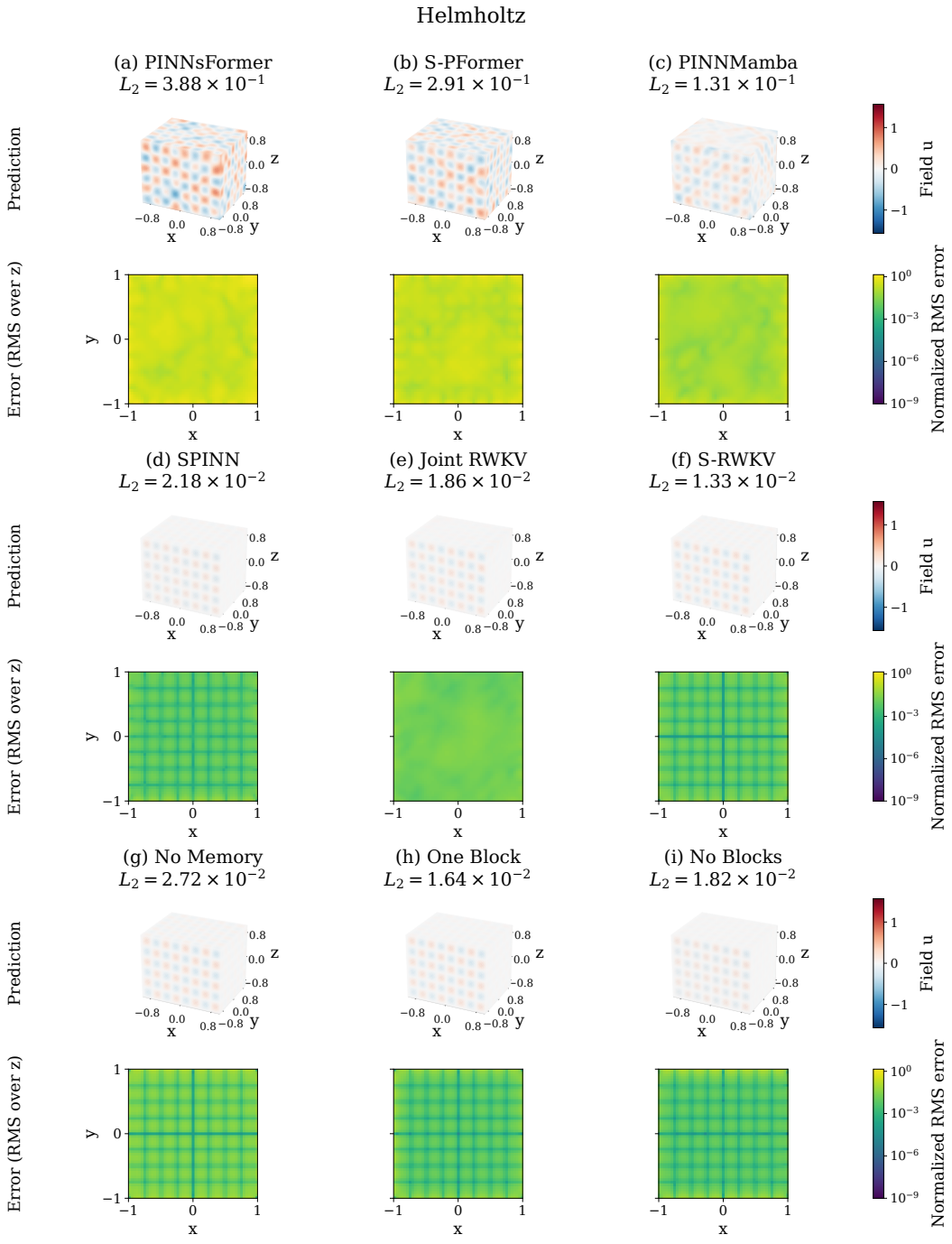


Figure 11: Helmholtz. Predicted three-dimensional fields and maps of normalized root-mean-square error over z. The error color scale is logarithmic.

C Training Protocols and Statistics

SPINN, full S-RWKV, and all three S-RWKV ablations use the same training script on every problem. Within each problem, the loss terms and weights, sampling, optimizer stages, learning-rate schedule, and state-selection rule are held fixed when the architecture is changed. Reaction, Convection, Wave, and Navier–Stokes use a multistage protocol shared with the other compared architectures. Flow Mixing, Helmholtz, and Klein–Gordon use $50\,000$ Adam steps; the separate point sets for the non-separable models are specified in Section C.4. The mean and sample SD are computed over ten independent runs.

C.1 Reaction, Convection, and Wave

Training begins with 1500 Adam steps at an initial learning rate of 10^{-3} and 300 outer L-BFGS steps. We use 8192 interior points and 257 points for each set of conditions. Independent physics-based validation uses 4096 interior points and 1025 points for each set of conditions. All architectures, including the sequence-model baselines, are trained and evaluated in `float64`.

The squared-residual weights are $1/26$ for Reaction, $1/2501$ for Convection, and $1/17$ for Wave. The initial-field weight is 100, the boundary-term weight is 10, and the initial-velocity weight for Wave is 1. For Reaction and Convection, the boundary criterion is the mean squared difference between values at the ends of the spatial interval. For Wave, it is half the sum of the mean squared errors at the two boundaries.

The next five stages refine the output projections with internal features held fixed. In the separable scalar network, the coordinate projections alternate in the order t, x, t, x, t, and the global bias is updated at each stage. In the joint model, a single output projection is refined. Each stage uses 16384 interior points and 1025 points for each set of conditions. A candidate is checked on a separate set of 16384 interior points and 4097 points for each set of conditions. The samples, weights, number of attempts, SVD thresholds, and update scales are identical across the compared models.

For the selected output parameters α and weighted residual vector $r(\alpha)$, we solve the local least-squares subproblem

$$\min_{\Delta\alpha} \left\| r(\alpha) + J(\alpha)\Delta\alpha \right\|_2^2. \tag{32}$$

After column normalization, SVD is applied with relative thresholds $10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$. Step scales $1, 1/2, 1/4, 1/8, 1/16$ are tested, with at most five attempts per stage. A candidate is accepted if it reduces the physics-based criteria on both training and independent validation points and passes a numerical consistency check. Otherwise, the previous state is retained.

The consistency check compares the calculation through the isolated projection with the actual network output. Absolute tolerances for RMS and maximum discrepancy are 10^{-9} and 10^{-8}, with an additional relative term. For an already accepted state, the tolerances are ten times larger. The separable network is also checked through the other axis parameterization. Reference-solution error is not used to select the state.

C.2 Errors after Each Training Stage

Tables 4–6 report the relative L_2 error of every architecture after the Adam stage, after the L-BFGS stage, and after output-projection refinement, over the same ten runs as Table 1; the lowest mean in each column is shown in bold. After 1500 Adam steps, all mean errors lie between 3.6 and 7.6. The L-BFGS stage separates the architectures. The three configurations with channel mixing reach mean errors of $7 \cdot 10^{-5}$ or below on all three problems, whereas SPINN and Joint RWKV remain at 1.2 on Convection. Refinement then lowers the mean error of every architecture on every problem by a factor between 1.1 and 1.5 and preserves the main accuracy differences, although it refits a whole coordinate factor in the S-RWKV configurations ($4\,129$ parameters per stage) and a single output layer in Joint RWKV (161 parameters). The main differences reported in Section 5 are therefore already present before least-squares refinement.

Table 4: Reaction: relative L_2 error after Adam, after L-BFGS, and after output-projection refinement, mean $\pm$ sample SD.

Architecture	After Adam	After L-BFGS	After refinement
PINNsFormer	5.1e+0 $\pm$ 2.2e+0	5.1e-2 $\pm$ 1.3e-2	4.04e-2 $\pm$ 1.01e-2
S-PFormer	6.2e+0 $\pm$ 2.1e+0	7.6e-3 $\pm$ 3.2e-3	6.33e-3 $\pm$ 2.64e-3
PINNMamba	4.8e+0 $\pm$ 1.6e+0	3.0e-2 $\pm$ 1.2e-3	2.14e-2 $\pm$ 8.14e-4
SPINN	5.4e+0 $\pm$ 1.2e+0	8.1e-3 $\pm$ 8.2e-3	6.95e-3 $\pm$ 7.08e-3
Joint RWKV	5.9e+0 $\pm$ 2.0e+0	8.3e-2 $\pm$ 1.6e-2	6.1e-2 $\pm$ 1.2e-2
S-RWKV	6.8e+0 $\pm$ 1.3e+0	7.0e-5 $\pm$ 9.8e-5	5.43e-5 $\pm$ 7.58e-5
No memory	**4.5e+0 $\pm$ 2.1e+0**	**3.8e-6 $\pm$ 3.2e-6**	**3.25e-6 $\pm$ 2.81e-6**
One block	7.6e+0 $\pm$ 1.1e+0	1.1e-5 $\pm$ 1.4e-5	8.02e-6 $\pm$ 1.05e-5
No blocks	6.7e+0 $\pm$ 1.8e+0	1.2e-4 $\pm$ 1.4e-4	1.05e-4 $\pm$ 1.23e-4

Table 5: Convection: relative L_2 error after Adam, after L-BFGS, and after output-projection refinement, mean $\pm$ sample SD.

Architecture	After Adam	After L-BFGS	After refinement
PINNsFormer	5.1e+0 $\pm$ 1.8e+0	1.0e-2 $\pm$ 4.2e-4	8.5e-3 $\pm$ 3.5e-4
S-PFormer	5.1e+0 $\pm$ 1.6e+0	6.9e-2 $\pm$ 2.3e-2	4.8e-2 $\pm$ 1.6e-2
PINNMamba	5.9e+0 $\pm$ 2.5e+0	2.6e+0 $\pm$ 1.9e+0	1.86e+0 $\pm$ 1.38e+0
SPINN	6.7e+0 $\pm$ 1.6e+0	1.2e+0 $\pm$ 7.0e-3	9.10e-1 $\pm$ 5.20e-3
Joint RWKV	5.3e+0 $\pm$ 2.6e+0	1.2e+0 $\pm$ 7.1e-3	1.00e+0 $\pm$ 5.81e-3
S-RWKV	6.6e+0 $\pm$ 2.4e+0	1.0e-5 $\pm$ 1.6e-5	6.78e-6 $\pm$ 1.09e-5
No memory	4.3e+0 $\pm$ 2.3e+0	9.3e-6 $\pm$ 1.5e-5	6.26e-6 $\pm$ 9.86e-6
One block	6.0e+0 $\pm$ 1.7e+0	**4.9e-6 $\pm$ 1.6e-6**	**3.63e-6 $\pm$ 1.16e-6**
No blocks	**4.3e+0 $\pm$ 1.7e+0**	2.1e-1 $\pm$ 4.6e-1	1.54e-1 $\pm$ 3.42e-1

Table 6: Wave: relative L_2 error after Adam, after L-BFGS, and after output-projection refinement, mean ± sample SD.

Architecture	After Adam	After L-BFGS	After refinement
PINNsFormer	**3.6e+0 ± 2.1e+0**	9.0e-2 ± 1.2e-2	6.9e-2 ± 9.4e-3
S-PFormer	5.5e+0 ± 1.1e+0	1.1e-2 ± 8.6e-4	7.7e-3 ± 6.3e-4
PINNMamba	5.2e+0 ± 2.4e+0	1.0e-1 ± 2.6e-2	6.8e-2 ± 1.7e-2
SPINN	5.8e+0 ± 2.1e+0	4.3e-7 ± 2.2e-7	3.54e-7 ± 1.86e-7
Joint RWKV	5.5e+0 ± 1.8e+0	5.9e-1 ± 1.5e-1	4.83e-1 ± 1.21e-1
S-RWKV	6.6e+0 ± 1.9e+0	6.6e-7 ± 9.8e-7	5.79e-7 ± 8.60e-7
No memory	5.3e+0 ± 2.5e+0	5.4e-7 ± 4.5e-7	4.59e-7 ± 3.80e-7
One block	6.2e+0 ± 1.1e+0	**3.8e-7 ± 2.1e-7**	**3.31e-7 ± 1.84e-7**
No blocks	4.5e+0 ± 1.7e+0	5.2e-2 ± 1.2e-1	4.37e-2 ± 9.77e-2

C.3 Navier–Stokes Equations

Function-space Gauss–Newton and the corresponding natural gradient have been applied to physics-informed solutions of the Navier–Stokes equations (Jnini et al., 2024). Below we describe the multistage protocol used in our experiments.

The initial stage consists of 1500 Adam steps and 300 outer L-BFGS steps on 8192 physics points. This is followed by 4000 SOAP steps (Vyas et al., 2024) with an initial learning rate of 3×10^{-4} and a final rate of 3×10^{-6}, then 1000 L-BFGS steps on 16384 points. Four Newton continuation stages with conjugate gradients follow: 500, 100, 200, and 100 outer steps on 65536 points, with limits of 100, 500, 500, and 1000 inner iterations. The initial Adam/L-BFGS block and the subsequent SOAP/L-BFGS block update all parameters. The four Newton continuation stages update the output projections and biases with the internal features fixed.

The velocity-observation weight is 10, the pressure-level condition has weight 0.01, and the physics residuals have unit weight. At the final stage, the multiplier for the observed v component increases to 2, while the state-selection rule remains unchanged. Independent physics-based validation uses 8192 points. The fixed sampling sequences are identical across the compared models.

The refinement stages update $37\,155$ parameters in the separable architectures and 675 parameters in Joint RWKV. For Joint RWKV, gradients in the SOAP/L-BFGS stage are computed in chunks of 8192 points rather than 12288 because of memory constraints. The full gradient uses all 16384 points with a common normalization; the step count and physics sample are unchanged.

Computational cost measurement. Peak GPU tensor-memory allocation is measured across all training stages, including refinement attempts and internal checks. Final evaluation against the reference is excluded. GPU memory is measured separately from the process's system memory. Because GPU load differed between runs, training time is not used to compare the speed of the architectures.

C.4 Adam Training for Main and Supplementary Problems

This subsection specifies training for the main Flow Mixing and Helmholtz benchmarks and the supplementary Klein–Gordon ablation in Appendix E. The latter is not included among the six main benchmark problems.

Optimization and numerical precision. Each of the three problems uses $50\,000$ Adam steps with an initial learning rate of 10^{-3}, updating all network parameters. Computations use PyTorch on an NVIDIA V100 in `float64` for all architectures. Training uses only Adam, without L-BFGS or separate output-projection refinement.

Samples for separable models. SPINN, full S-RWKV, and the No memory, One block, and No blocks configurations use Cartesian products of independently sampled uniform coordinates. The domain is $(x, y, t) \in [-4, 4]^2 \times [0, 4]$ for Flow Mixing, $(x, y, z) \in [-1, 1]^3$ for Helmholtz, and $(x, y, t) \in [-1, 1]^2 \times [0, 10]$ for Klein–Gordon. Sample sizes are given in Table 7. Sets are refreshed every 100 steps. In corresponding runs, the separable models receive identical coordinates; sample generation is independent of random network initialization.

Table 7: Training samples for SPINN, S-RWKV, and its ablations. Flow Mixing and Helmholtz are main benchmarks; Klein–Gordon is supplementary only. The boundary-point count is given per spatial face. The stationary Helmholtz problem has no initial condition.

Problem	Interior	Initial	Per face	Faces
Flow Mixing	256^3	256^2	256^2	4
Helmholtz	64^3	—	64^2	6
Klein–Gordon	64^3	64^2	64^2	4

Loss function. The base physics-informed loss combines the mean squared residuals of the equation, initial condition, and boundary conditions:

$$\mathcal{L}(\theta) = \lambda_r \,\mathrm{MSE}_{C_r}[\mathcal{R}_\theta] + \lambda_0 \,\mathrm{MSE}_{C_0}[\widehat{u}_\theta - u_0] + \sum_{j=1}^{N_\Gamma} \mathrm{MSE}_{C_{\Gamma_j}}[\widehat{u}_\theta - g_j]. \tag{33}$$

Here, $\mathcal{R}_\theta$ is the equation residual. The finite sets C_r, C_0, and C_{Γ_j} sample the interior, initial surface, and spatial face j, respectively; N_Γ is the number of faces. Each MSE is the empirical average defined after Eq. (5). The weight λ_r is 10 for Flow Mixing and 1 for Helmholtz and Klein–Gordon. The initial condition has unit weight; this term is absent for the stationary Helmholtz problem. The boundary loss is the sum of the mean squared errors over faces, without division by their number. Helmholtz uses homogeneous Dirichlet conditions. For Flow Mixing and Klein–Gordon, initial and boundary values are computed from the analytical solution. On Klein–Gordon, the initial condition constrains only u; no separate initial-velocity penalty is imposed, as in the benchmark implementation used here.

Adam learning-rate schedule. The following schedules are shared by SPINN, full S-RWKV, and all three ablations. On Flow Mixing and Helmholtz, the learning rate remains 10^{-3}. On Klein–Gordon, these five separable models use $40\,000$ updates at a learning rate of 10^{-3}, followed by cosine decay to 10^{-5}:

$$\eta_k = \begin{cases} 10^{-3}, & k \le 40\,000, \\ 10^{-5} + \dfrac{10^{-3} - 10^{-5}}{2}\left[1 + \cos\left(\pi \dfrac{k - 40\,000}{10\,000}\right)\right], & 40\,000 < k \le 50\,000. \end{cases} \tag{34}$$

The learning rate is determined by the cumulative update index, including when training resumes from a saved state. The schedule is task-specific, not architecture-specific within this group.

Samples for the other architectures. PINNsFormer, S-PFormer, PINNMamba, and Joint RWKV use 1024 base interior points, 1024 initial-condition points for time-dependent problems, and 256 points per spatial face. Points are sampled uniformly and refreshed every 100 steps. In corresponding runs, the four models receive identical base points. Each model is trained for $50\,000$ Adam steps at a constant learning rate of 10^{-3}. The update count and problem definitions match those of the separable models, while training sample sizes differ. All architectures are evaluated after Adam here; the L-BFGS stages from the original sequence-model papers are not part of this protocol.

Sequence representations. PINNsFormer and S-PFormer use sequences of five points with a physical spacing of 10^{-4}, while PINNMamba uses seven points with a spacing of 10^{-2}. Sequences are constructed along time for Flow Mixing and Klein–Gordon, and along z for Helmholtz. Derivatives are computed with respect to the base-point coordinate while shifting the entire sequence. Points outside the physical domain are excluded from the losses; initial and boundary constraints apply only to points on the corresponding surface. PINNMamba additionally enforces prediction consistency in overlapping windows with unit weight. Sequence models are evaluated using the first sequence element. Joint RWKV processes individual points with coordinates normalized to $[0, 1]$.

C.5 Forward-Pass Cost

Table 8 reports the number of floating-point operations in one forward evaluation of the predicted field on the interior collocation points of one training step. The interior samples are those of Appendices C.1 and C.4: 8192 points for Reaction, Convection, and Wave; Cartesian products of 256^3 and 64^3 points for the separable models on Flow Mixing and Helmholtz; and 1024 base points for the other architectures on these two problems.

Table 8: Forward-pass cost in GFLOPs: one evaluation of the predicted field on the interior collocation points of one training step.

Architecture	Reaction, Convection, and Wave	Flow Mixing	Helmholtz
PINNsFormer	11.751	4.627	4.627
S-PFormer	9.615	3.786	3.786
PINNMamba	40.721	4.088	4.088
Joint RWKV	3.817	0.935	0.935
SPINN	1.221	4.369	0.023
S-RWKV	3.386	4.514	0.057
No memory	1.210	4.412	0.032
One block	1.761	4.438	0.038
No blocks	0.136	4.362	0.019

On Flow Mixing and Helmholtz, the separable models evaluate Cartesian products of per-axis samples, whereas the other architectures use 1024 base points with a three-dimensional input on both problems, so their two columns coincide. Klein–Gordon uses the same interior sample sizes as Helmholtz and gives identical counts at the reported precision. Counts are floating-point operations, not multiply–accumulate operations.

Coordinate-branch configurations. On Reaction, Convection, and Wave, removing the recurrent pathway reduces the forward cost from 3.386 to 1.210 GFLOPs, approximately 64%; One block requires 1.761 GFLOPs. The relative reduction exceeds the 54% reduction in parameters because the weights of each block are shared by the G feature groups: the operation count grows with G, whereas the parameter count does not. No blocks requires 0.136 GFLOPs, 25 times less than the full configuration; Section 5.2 discusses its larger spread across runs.

Equal forward cost. SPINN and No memory have nearly the same forward cost on Reaction, Convection, and Wave, 1.221 and 1.210 GFLOPs, although SPINN has about three times as many parameters. At this cost, No memory has mean errors lower by more than three orders of magnitude on Reaction and Convection, and both configurations reach errors of order 10^{-7} on Wave (Table 1). Both use the Fourier input and the separable assembly, so at equal forward cost the difference comes from the branch body.

Joint and sequence models. On Reaction, Convection, and Wave, Joint RWKV and full S-RWKV are close in both parameters ($54\,887$ and $55\,369$) and forward cost (3.817 and 3.386 GFLOPs). PINNsFormer, S-PFormer, and PINNMamba require 11.751, 9.615, and 40.721 GFLOPs, which is 2.8–12 times the cost of full S-RWKV and 8–34 times that of No memory; their mean errors are higher on all three problems (Table 1).

Flow Mixing. The separable configurations require 4.362–4.514 GFLOPs, and the three sequence models require 3.786–4.627 GFLOPs. At comparable forward cost, the separable models therefore evaluate 256^3 interior points, whereas the sequence models evaluate 1024 base points. The costs of the separable configurations differ by less than 4%, compared with a factor of 25 between full S-RWKV and No blocks on Reaction, Convection, and Wave. On the 256^3 grid, the cost is dominated by the sum-of-products assembly shared by all separable configurations rather than by the coordinate branches. Joint RWKV requires 0.935 GFLOPs, about one fifth of the cost of full S-RWKV, so the separable–joint comparison on this problem is not matched in forward cost.

Helmholtz. The separable configurations require 0.019–0.057 GFLOPs on 64^3 interior points, whereas the non-separable models require 0.935–4.627 GFLOPs on 1024 base points, from 16 to 244 times more. No blocks and full S-RWKV have the lowest mean errors on this problem in Table 1 ($2.20 \cdot 10^{-2}$ and $2.22 \cdot 10^{-2}$), No blocks at one third of the forward cost.

These counts describe a single evaluation of the field. They complement the peak GPU allocation reported for Navier–Stokes in Section 5.1 and are not a measure of total training cost.

D Ablation of Input, Assembly, and Branch Body

D.1 Factors and Protocol

The comparisons in Section 5 do not show how much of the accuracy comes from the Fourier input itself, because SPINN and all S-RWKV configurations use it. This ablation varies three factors separately on Reaction, Convection, and Wave: the input lift (coordinate or Fourier), the field assembly (joint or separable), and the branch body (MLP, RWKV, or channel-only). The channel-only body is the No memory configuration and is defined only for the separable assembly, which gives ten configurations.

The Fourier input is the learned lift of Eq. (2) with $G = 4$ groups and the default scales $(0.5, 1, 2, 8)$; the separable branches use group width $w = 32$. Separable models apply the lift to each coordinate. Joint models apply it to the coordinate vector, with frequency vectors $\omega \in \mathbb{R}^d$; such features can represent oblique oscillations in the joint coordinate space. For MLP bodies, the coordinate input is the coordinate itself. For the RWKV and channel-only bodies, it is a group-specific affine map

$$h_g^{(0)}(\xi) = a_g\, \xi + c_g, \qquad a_g, c_g \in \mathbb{R}^w, \quad g = 1, \dots, G, \tag{35}$$

because the affine term in Eq. (2) is shared by all groups and would otherwise produce G identical tokens; in Joint RWKV, a_g is replaced by a matrix acting on the coordinate vector. MLP bodies are $\tanh$ networks: the SPINN branch in the separable assembly, which with Fourier input is the SPINN baseline of Table 1, and a network of four linear layers with hidden width 512 in the joint one, the PINN configuration of Zhao et al. (2024). The joint MLP has about ten times as many parameters as full S-RWKV.

Classical MLP PINN. The joint MLP with coordinate input is the classical PINN baseline used in PINNsFormer (Zhao et al., 2024): a plain $\tanh$ network on the raw coordinates. It uses the same float64 pipeline, samples, and refinement as the other models in this ablation. Table 9 reports its results on all three problems; Table 3 summarizes Convection in the main text. Under this protocol, its mean errors are $9.6 \cdot 10^{-1}$ on Reaction, $7.5 \cdot 10^{-1}$ on Convection, and $6.8 \cdot 10^{-1}$ on Wave. Plain MLPs in float64 have been reported to solve these benchmarks under other schedules (Xu et al., 2025b).

All configurations use the protocol of Appendix C.1 with the same samples, loss weights, float64 arithmetic, refinement settings, and physics-based state selection. Run k of every configuration shares the sampling stream, and errors are evaluated after the five projection stages.

D.2 Results

Table 9 reports final errors; Table 3 in the main text summarizes the Convection column. Ten independent runs are used for each configuration.

Table 9: Input–assembly–body ablation on Reaction, Convection, and Wave: final relative L_2 error after output-projection refinement, mean $\pm$ sample SD over ten independent runs. Adjacent rows differ only in the input lift.

Architecture	Input	Parameters	Reaction	Convection	Wave
MLP	coord.	527 361	9.6e-1 $\pm$ 2.9e-2	7.5e-1 $\pm$ 6.1e-2	6.8e-1 $\pm$ 1.5e-1
MLP	Fourier	592 161	3.8e-2 $\pm$ 2.0e-2	9.6e-1 $\pm$ 2.0e-2	4.9e-7 $\pm$ 2.2e-7
SPINN	coord.	42 945	4.2e-2 $\pm$ 1.9e-2	5.7e+0 $\pm$ 2.1e+0	6.8e-2 $\pm$ 1.2e-2
SPINN	Fourier	75 841	6.95e-3 $\pm$ 7.08e-3	9.10e-1 $\pm$ 5.20e-3	3.54e-7 $\pm$ 1.86e-7
Joint RWKV	coord.	55 007	5.7e+0 $\pm$ 1.5e+0	4.8e+0 $\pm$ 1.7e+0	6.4e-1 $\pm$ 9.9e-2
Joint RWKV	Fourier	54 887	6.1e-2 $\pm$ 1.2e-2	1.00e+0 $\pm$ 5.81e-3	4.83e-1 $\pm$ 1.21e-1
S-RWKV	coord.	55 497	5.7e-3 $\pm$ 2.7e-3	7.3e-2 $\pm$ 6.9e-3	8.6e-3 $\pm$ 1.9e-4
S-RWKV	Fourier	55 369	5.43e-5 $\pm$ 7.58e-5	6.78e-6 $\pm$ 1.09e-5	5.79e-7 $\pm$ 8.60e-7
No memory	coord.	25 541	6.1e-3 $\pm$ 1.6e-3	6.2e-3 $\pm$ 1.9e-3	7.3e-3 $\pm$ 8.3e-4
No memory	Fourier	25 413	3.25e-6 $\pm$ 2.81e-6	6.26e-6 $\pm$ 9.86e-6	4.59e-7 $\pm$ 3.80e-7

MLP and Joint RWKV use the joint assembly; SPINN, S-RWKV, and No memory are separable. Coord.: the coordinate itself for MLP bodies and Eq. (35) for grouped bodies. Fourier: the lift of Eq. (2) with scales $(0.5, 1, 2, 8)$, applied to each coordinate in separable models and to the coordinate vector in joint models. The Fourier-input rows of SPINN, Joint RWKV, S-RWKV, and No memory repeat Table 1.

With Fourier input, the configurations differ most on Convection: SPINN, the joint MLP, and Joint RWKV have mean errors of $9.10 \cdot 10^{-1}$, $9.6 \cdot 10^{-1}$, and 1.00, whereas full S-RWKV and No memory reach $6.78 \cdot 10^{-6}$ and $6.26 \cdot 10^{-6}$. On Reaction, No memory is more accurate than SPINN by more than three orders of magnitude ($3.25 \cdot 10^{-6}$ versus $6.95 \cdot 10^{-3}$). Under this protocol, Fourier input alone is insufficient, but substantially improves the grouped bodies: with coordinate input, S-RWKV and No memory have mean errors between $5.7 \cdot 10^{-3}$ and $7.3 \cdot 10^{-2}$, roughly two to four orders of magnitude above their Fourier-input counterparts. With coordinate input, the two grouped configurations are close on Reaction and Wave, and No memory has a lower mean on Convection ($6.2 \cdot 10^{-3}$ versus $7.3 \cdot 10^{-2}$). On Wave, the joint MLP with Fourier input reaches $4.9 \cdot 10^{-7}$, of the same order as the separable configurations, so separability is not required for this problem. Joint RWKV has mean errors above $4 \cdot 10^{-1}$ on Convection and Wave with both inputs.

These comparisons concern three equations with smooth reference solutions under a single training protocol. Two of the reference solutions are sums of two separable components.

E Additional Ablation: Inherited Klein–Gordon Benchmark

E.1 Source of the Incomplete Initial Conditions

This auxiliary experiment follows the displacement-only Klein–Gordon benchmark in SPINN (Cho et al., 2023, Appendix D.3, Eqs. (57)–(59)). The published conditions and the initial-condition losses in `apply_model_spinn` and `apply_model_pinn` of `klein_gordon3d.py` constrain the initial field, but not its time derivative.[1] We keep these constraints unchanged for every architecture. Because they do not specify complete initial data for a problem that is second order in time, Klein–Gordon is excluded from the main benchmarks, tables, and conclusions; the results below are an additional architecture ablation against a prescribed reference field.

E.2 Inherited Formulation and Missing Condition

The benchmark is

$$u_{tt} - u_{xx} - u_{yy} + u^2 = f(x, y, t), \qquad (x, y, t) \in [-1, 1]^2 \times [0, 10], \tag{36}$$

with the manufactured reference

$$u_{\mathrm{ref}}(x, y, t) = (x + y)\cos(2t) + xy\sin(2t), \qquad f = -4u_{\mathrm{ref}} + u_{\mathrm{ref}}^2. \tag{37}$$

As in SPINN, training enforces $u(x, y, 0) = x + y$ and reference Dirichlet values on the four spatial faces. The differential equation, forcing, and reference field are mutually consistent. The standard initial-boundary-value formulation also requires an initial-velocity condition, which for this reference is

$$u_t(x, y, 0) = 2xy. \tag{38}$$

This condition is absent from both the source benchmark and the physical loss used for every model in this supplementary experiment. Thus, the inherited objective does not impose the complete Cauchy data associated with the reference. We make no uniqueness claim for the displacement-only formulation and interpret the reported errors as reference-approximation errors, not as evidence of solving a fully specified initial-boundary-value problem.

E.3 Coordinate-Branch Ablation and Supplementary Results

Tables 10 and 11 report the comparison-model and coordinate-branch results for this inherited objective. The ablation varies the matrix-memory mechanism and branch depth while retaining the same sum-of-products field representation and physical constraints. Training settings are given in Appendix C.4; ten independent runs are used for each model.

Full S-RWKV has a mean reference error and sample SD of $7.10 \cdot 10^{-5} \pm 3.93 \cdot 10^{-5}$, compared with $7.31 \cdot 10^{-5} \pm 3.87 \cdot 10^{-5}$ for No memory, and is sixteen times more accurate than SPINN. The one-block variant has a mean reference error of $8.85 \cdot 10^{-5}$; removing all blocks increases it to $2.64 \cdot 10^{-4}$ (Table 11). These observations describe sensitivity to architectural choices under the inherited objective only; they are not used to support the main accuracy claims.

[1] SPINN reference implementation: `klein_gordon3d.py`, accessed 24 September 2026.

Table 10: Supplementary comparison on the inherited Klein–Gordon benchmark: relative L_2 error against the prescribed analytical reference, mean $\pm$ SD over ten independent runs. These results are not used in the main comparison or conclusions.

Architecture	Relative L_2 error
PINNsFormer	7.99e-2 $\pm$ 1.14e-2
S-PFormer	4.41e-1 $\pm$ 1.19e-1
PINNMamba	1.54e-2 $\pm$ 7.96e-3
SPINN	1.13e-3 $\pm$ 7.25e-4
Joint RWKV	2.86e-2 $\pm$ 3.14e-2
S-RWKV (full)	7.10e-5 $\pm$ 3.93e-5

Table 11: Supplementary coordinate-branch ablation on the inherited Klein–Gordon benchmark: relative L_2 error against the prescribed analytical reference, mean $\pm$ sample SD over ten independent runs. All variants use the same displacement-only physical constraints. The lowest mean among the four configurations is shown in bold.

Architecture	Relative L_2 error
S-RWKV (full)	**7.10e-5 $\pm$ 3.93e-5**
No memory	7.31e-5 $\pm$ 3.87e-5
One block	8.85e-5 $\pm$ 3.72e-5
No blocks	2.64e-4 $\pm$ 1.70e-4

The full S-RWKV result is repeated from Table 10. All variants use the same training and evaluation protocol.

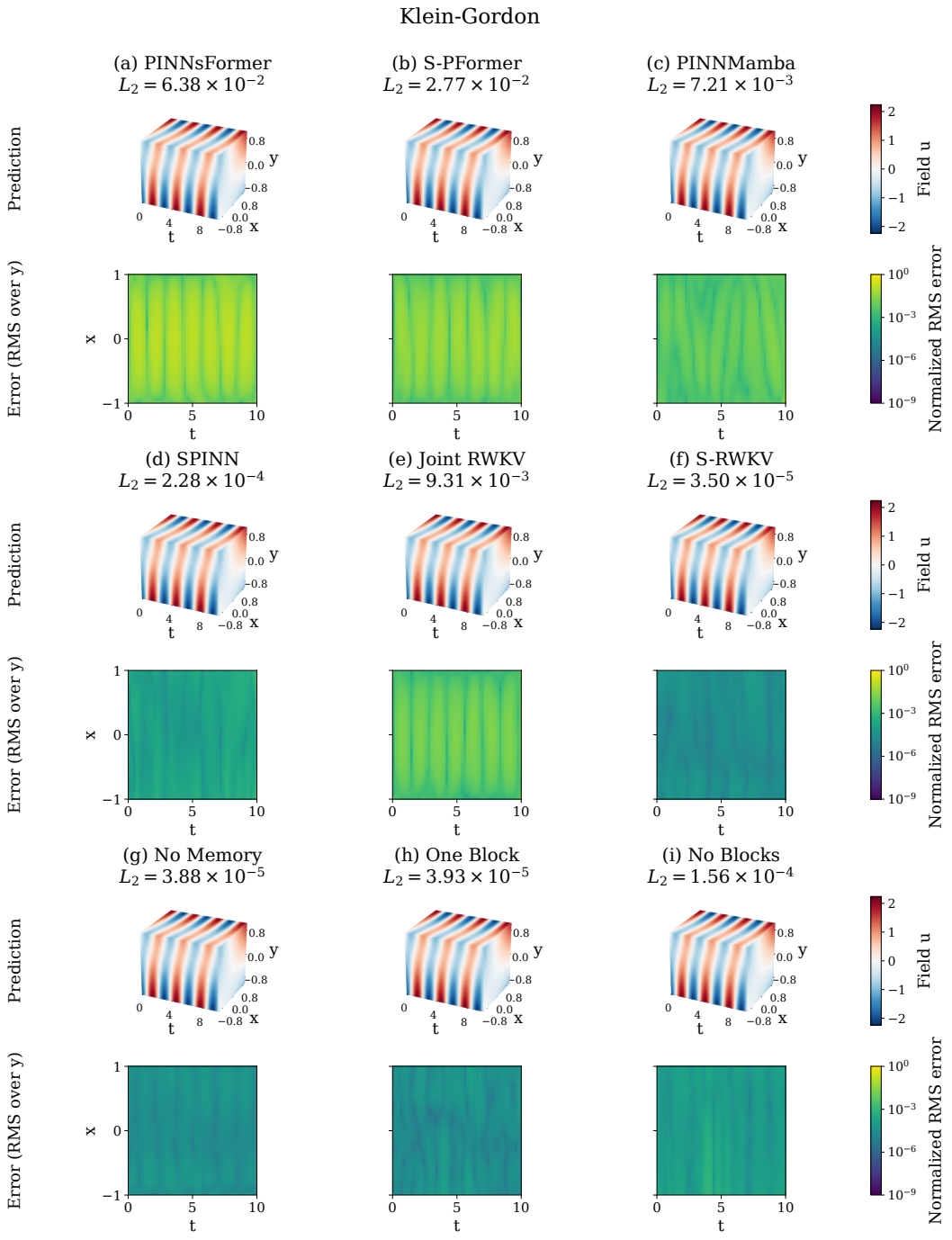


Figure 12: Supplementary Klein–Gordon experiment using the displacement-only benchmark inherited from SPINN. Predicted fields in (t, x, y) coordinates and maps of normalized root-mean-square deviation from the prescribed reference over y. The error color scale is logarithmic. This experiment is excluded from the main benchmark comparison.

F EXPLORATORY SINGLE-RUN EXPERIMENT: ALLEN–CAHN

We consider the one-dimensional Allen–Cahn equation

$$u_t - 10^{-4} u_{xx} + 5(u^3 - u) = 0, \qquad (x, t) \in [-1, 1] \times [0, 1], \tag{39}$$

with $u(x, 0) = x^2 \cos(\pi x)$ and periodic value and derivative conditions at $x = \pm 1$. We evaluate all models on the same 512×201 reference grid.

This supplementary experiment uses a separate training protocol and one run per model. The models are trained with the self-adaptive (SA) loss weighting of SA-PINN (McClenny & Braga-Neto, 2023), in which pointwise loss weights are trained by gradient ascent together with the network (rows marked "+ SA" in Table 12). Without SA weights, none of the architectures learned the solution in our runs. Allen–Cahn therefore lies outside the fixed pipeline of the main benchmark, and every number below comes from a single run. With SA weights, S-RWKV reaches a relative L_2 error of 8.68×10^{-3}, improving on the SA-PINN MLP (1.28×10^{-2}), and the memory-free configuration reaches 1.14×10^{-2}. Residual-based attention (RBA) reports 5.8×10^{-5} on this problem (Anagnostopoulos et al., 2024), which is substantially lower than the errors in this single-run comparison. The PINNMamba panel uses intermediate weights at step 13,800.

Table 12: Allen–Cahn relative L_2 error, one run per model. "+ SA" marks training with the self-adaptive weights of SA-PINN, and SA-PINN MLP is an MLP trained with them (McClenny & Braga-Neto, 2023). The RBA value is the five-run mean reported by Anagnostopoulos et al. (2024); PINNMamba uses the checkpoint at step 13,800.

Architecture	Relative L_2
RBA (literature)	5.8×10^{-5}
S-RWKV (full) + SA	8.68×10^{-3}
No memory + SA	1.14×10^{-2}
SA-PINN MLP	1.28×10^{-2}
Joint RWKV + SA	2.16×10^{-2}
SPINN + SA	5.92×10^{-2}
PINNMamba	7.85×10^{-1}
PINNsFormer + SA	8.46×10^{-1}
S-PFormer + SA	9.91×10^{-1}

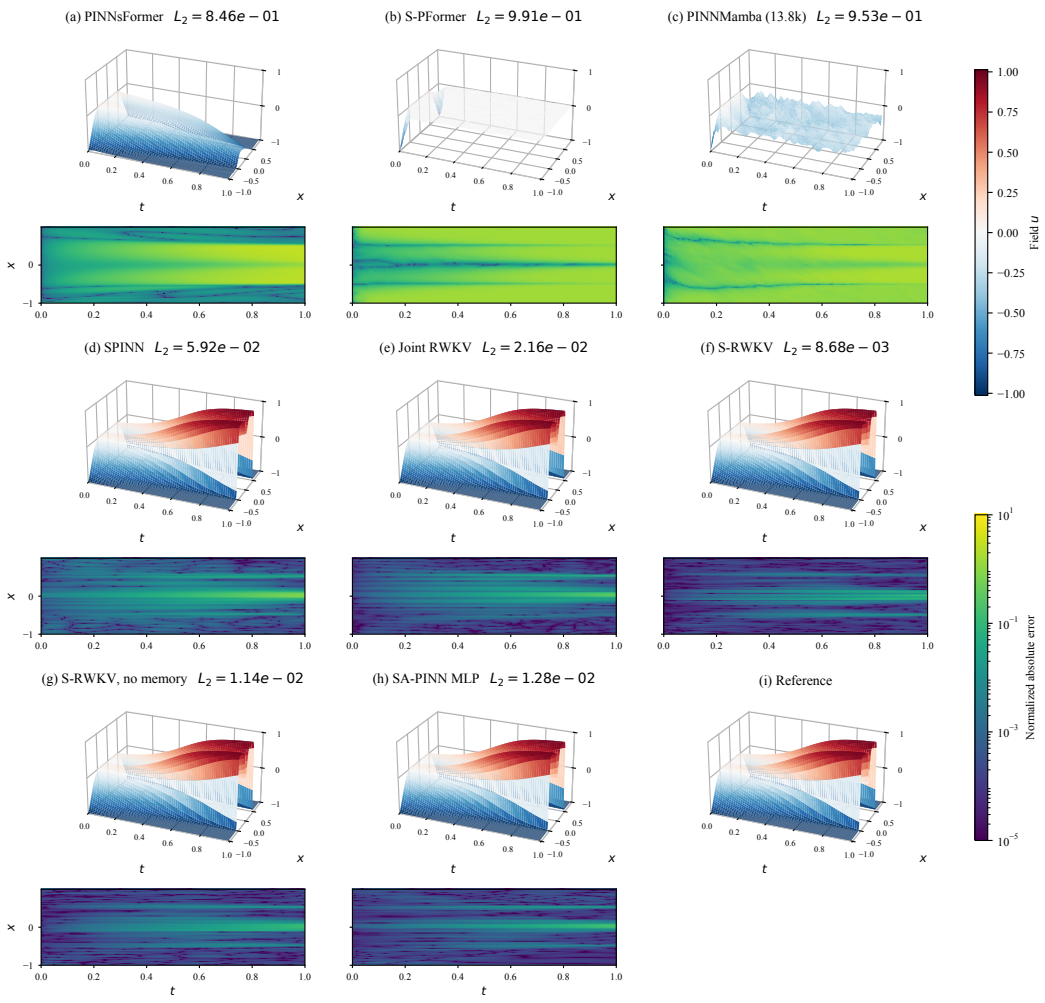


Figure 13: Allen–Cahn predictions and pointwise absolute errors normalized by the reference RMS. Field scales are shared across models; the error scale is logarithmic.