MULTIPHYSICS POST-TRAINING: FROM DECOUPLED MODULES TO COUPLED EQUILIBRIA

Anonymous authors
Paper under double-blind review

ABSTRACT

Multiphysics simulation is essential for systems governed by interacting physical processes, classically relying on partitioned analysis to couple interdisciplinary solvers through interface exchange. Neural simulation can inherit this modular architecture by assembling independently pretrained physical modules. However, directly assembling modules pretrained to learn isolated single physics introduces an objective mismatch, as they are never optimized for the target coupled equilibrium. To address this, we derive module-to-system error bounds, proving that coupled accuracy is governed by cross-interface operator coupling rather than module fidelity on isolated single physics alone. This motivates multiphysics post-training, a learning paradigm that adapts pretrained modules via a system-level loss on their joint predictions while keeping each module's inputs, outputs, and exchange structure unchanged. Under this paradigm, we provide two complementary implementations, Finite Unrolling (Unrolled-K) for low-latency inference and Equilibrium Post-Training (EquiPT), which achieves $\mathcal{O}(1)$ activation memory and high fidelity via an implicit adjoint. Across neutron–thermal, reaction–diffusion, and fluid–structure interaction benchmarks, EquiPT reduces system error by 48.7%–96.1% over frozen assembly. These gains stem from training at the coupled equilibrium rather than from coupled data alone, as EquiPT also reduces system error by 34.2%–74.1% relative to module-wise continued training on the same coupled data. Pretrained initialization further improves accuracy under the same post-training budget. Furthermore, post-training shifts the measured coupling spectral radii toward lower values, lowering the ceiling on interface error amplification along each eigenmode.

1 INTRODUCTION

Many real-world scientific and engineering systems are governed by complex interactions among distinct physical processes, such as thermal–hydraulic feedback dictating reactor safety in nuclear engineering (Ivanov & Avramova, 2007), or fluid–structure interactions determining prosthetic heart valve design in cardiovascular biomechanics (Hsu et al., 2015). Multiphysics simulation is therefore vital to diverse disciplines. Unlike single-physics modeling, coupled systems exhibit bidirectional dependencies across physical interfaces. This interdependence makes predictive modeling intrinsically challenging, as constituent processes typically span vastly disparate spatiotemporal scales and interact through strong nonlinear cross-field couplings.

To tackle this complexity, classical partitioned analysis preserves explicit physical boundaries, thereby decoupling multidisciplinary constraints, accommodating heterogeneous scales, and enabling independent development of specialized domain solvers (Küttler & Wall, 2008; Degroote et al., 2009). In practice, partitioned simulation assigns each physics a specialized solver and iteratively exchanges interface information until reaching mutual consistency (Martins & Hwang, 2013). To reduce the high computational cost of repeated numerical solves, learned neural surrogates have emerged as efficient alternatives (Li et al., 2020; Herde et al., 2024; Hao et al., 2024). Extending this success to multiphysics, recent modular approaches independently pretrain specialized neural modules on isolated single physics and assemble them via iterative field exchange at deployment (Cai et al., 2021; Mao et al., 2024), often using classical relaxation or acceleration schemes (e.g., Anderson acceleration) to stabilize coupling iterations (Küttler & Wall, 2008; Walker & Ni, 2011).

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

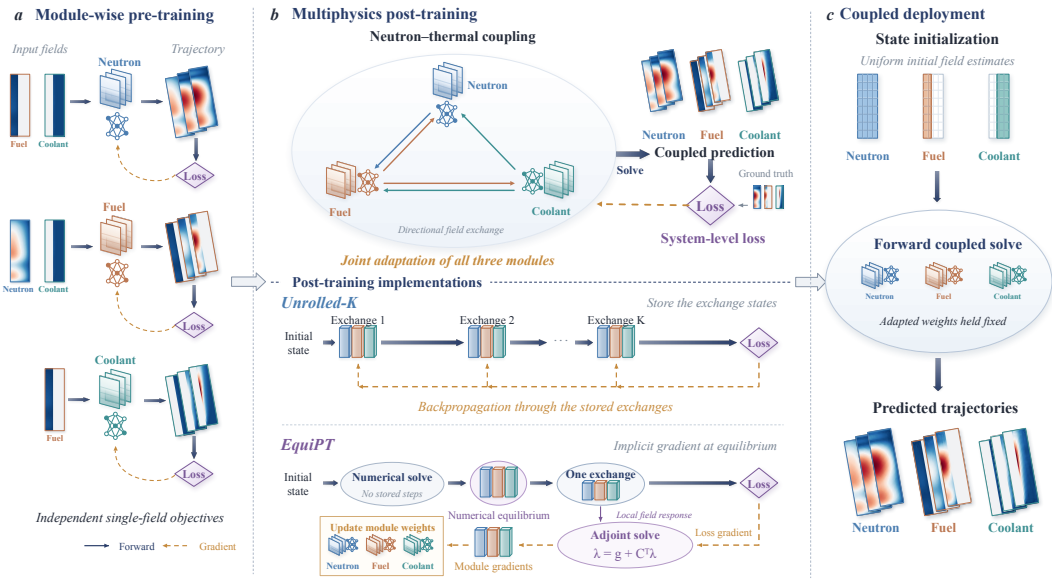

Figure 1: **Overview of multiphysics post-training.** Illustrated on neutron–thermal coupling (NT), Stage I uses module-wise pre-training to learn isolated single physics, Stage II jointly adapts the modules with a system-level loss on the state after K stored exchanges (Unrolled-K) or on the numerical equilibrium via an implicit adjoint (EquiPT), and coupled deployment uses the adapted modules for trajectory prediction through iterative field exchange. An exchange is one solver iteration. Stacked field images schematically indicate trajectories.

Nevertheless, existing approaches suffer from an objective mismatch between module-wise pre-training and coupled system deployment. During module-wise pre-training, each module predicts its own field from reference partner fields to learn isolated single physics, analogous to teacher forcing. The modules are thus optimized only for isolated single physics, leaving them blind to system-level interactions. At deployment, however, no reference partner fields are available, and modules iteratively exchange their own predictions towards coupled equilibrium with frozen parameters, so they cannot coordinate or adapt to mutual errors, allowing interface discrepancies to amplify and even break down. To formalize this vulnerability, we establish *module-to-system error bounds*, proving that coupled prediction error depends both on module accuracy on isolated single physics and on error amplification through cross-interface operator coupling. Inference-time strategies with frozen module weights can improve numerical convergence, but they do not directly optimize the learned equilibrium's prediction accuracy.

To bridge this gap, we introduce *multiphysics post-training*, a learning paradigm that adapts pre-trained physical modules through a system-level loss on their joint predictions (Fig. 1). Our key insight is that the fixed-point structure of partitioned coupling can serve as an effective architectural inductive bias, enabling multiphysics post-training while preserving each module's physical interfaces. To our knowledge, this is the first post-training framework for partitioned multiphysics simulation, jointly adapting separately pretrained modules through their assembled coupled system. To balance resources and inference performance, our paradigm admits two complementary implementations. (i) *Equilibrium Post-Training (EquiPT)* targets the coupled numerical equilibrium via an implicit adjoint (Christianson, 1994; Bai et al., 2019) and achieves $\mathcal{O}(1)$ activation memory and high fidelity. (ii) *Finite Unrolling (Unrolled-K)* backpropagates through K truncated exchanges to minimize latency under fixed iteration budgets.

We evaluate our framework across three physical benchmarks, neutron–thermal coupling (NT) in nuclear engineering, Gray–Scott reaction–diffusion (GS) in chemistry, and fluid–structure interaction (FSI) in mechanics. Across all domains, multiphysics post-training outperforms frozen modular assembly, with EquiPT reducing system prediction error by 48.7%–96.1%, exhibiting its most pronounced gains in strongly coupled regimes. Controlled ablations trace these gains to training at the coupled equilibrium rather than to coupled data alone, as module-wise continued training on the same coupled data (MW-CT) has 1.5 to 3.9 times EquiPT's system error. Pretrained initialization

further improves accuracy (2.2 to 9.8 times the error without it). Our spectral diagnostics offer a mechanistic interpretation of these gains. Theoretically, the coupling Jacobian's spectral radius sets an error amplification ceiling for each eigenmode. Empirically, the reduced radius after post-training accompanies the lower system error.

Our main contributions are as follows:

1. **Formulation and error analysis of learned coupling.** We formulate modular surrogate inference as an operator fixed-point problem and derive module-to-system error bounds, revealing cross-field error amplification and the eigenmode ceiling set by the coupling spectral radius.
2. **A multiphysics post-training paradigm.** We propose multiphysics post-training, an interface-preserving learning paradigm for assembling separately pretrained physical modules into coupled prediction. It uses the fixed-point structure of partitioned coupling as an architectural inductive bias and optimizes the pretrained modules via a system-level loss on their coupled equilibrium.
3. **Concrete implementations and evaluation.** We provide complementary EquiPT and Unrolled-K implementations, demonstrate substantial error reductions across three physical benchmarks, and analyze their trade-offs and underlying mechanisms.

2 Related Work

Fixed-Point Iteration and Implicit Differentiation. Fixed-point theory formalizes coupled nonlinear systems via equilibria $x^* = \mathcal{F}(x^*)$ (Ortega & Rheinboldt, 1970). While standard Picard iteration requires contractivity, practical coupled problems are often non-contractive or ill-conditioned. Relaxation, Anderson acceleration, and interface quasi-Newton methods therefore stabilize divergent iterations and accelerate convergence (Anderson, 1965; Küttler & Wall, 2008; Degroote et al., 2009; Walker & Ni, 2011). To differentiate through equilibria, implicit differentiation applies the Implicit Function Theorem (IFT) to the fixed-point equation, computing exact adjoint gradients with constant $\mathcal{O}(1)$ activation memory (Christianson, 1994; Griewank & Walther, 2008; Martins & Hwang, 2013; Gray et al., 2019). Deep Equilibrium Models (DEQs) embed this root finding in neural networks (Bai et al., 2019; 2020; Marwah et al., 2023), motivating well-posed parameterizations and gradient approximations (Winston & Kolter, 2020; Fung et al., 2022; Blondel et al., 2022). These fixed-point and adjoint foundations underpin our multiphysics post-training framework.

Data-Driven Multiphysics Simulation. Data-driven multiphysics modeling spans monolithic learning and partitioned modular assembly. Monolithic models predict all interacting fields in one architecture (Shukla et al., 2020; Xiao et al., 2023; Subramanian et al., 2023). However, this approach sacrifices modular flexibility, scales poorly across disparate physical stiffness, struggles with moving meshes, and requires abundant, often unavailable coupled data across all fields (Hou et al., 2012; Keyes et al., 2013). We therefore focus exclusively on partitioned schemes, which preserve modularity through iterative boundary exchange between specialized neural operators (Cai et al., 2021; Totounferoush et al., 2021; Moseley et al., 2023; Zerihun & Lee, 2026), while generative composition coordinates independent diffusion priors during inference (Du et al., 2020; Li et al., 2022; Zhang et al., 2024; Gao et al., 2026). However, existing modular pipelines freeze operators at deployment, leaving this coupling mismatch unresolved.

System-Level Adaptation in Scientific Machine Learning. Bridging isolated training and downstream deployment has motivated several system-level adaptation strategies. Temporally, multi-step autoregressive fine-tuning curbs rollout drift via unrolled trajectories (Brandstetter et al., 2022; Lam et al., 2023; Bi et al., 2023; Duncan et al., 2026). In hybrid architectures, solver-in-the-loop frameworks embed neural closures in numerical solvers, differentiating through numerical steps to enforce macroscopic conservation (Um et al., 2020; Kochkov et al., 2021; 2024; Fan & Wang, 2024). For operator composition, CompNO (Hmida et al., 2026) trains auxiliary aggregators over frozen libraries. Unlike these methods, which adapt a time integrator, a hybrid correction, or operator mixing, we jointly adapt pretrained modules with a system-level loss at coupled equilibrium.

3 LIMITATIONS OF MODULE-WISE PRE-TRAINING

We build on a multiphysics coupling workflow (Gupta & Jaiman, 2022; Bochev et al., 2024) based on module-wise pre-training and coupled inference. Section 3.1 formalizes this workflow, and Section 3.2 analyzes why pretrained modules can lose accuracy when coupled.

3.1 MODULE-WISE PRE-TRAINING AND COUPLED INFERENCE

Physical prediction problem. Consider M interacting physical processes governed by

$$\partial_t u_i = f_i(\mathbf{u}; \mathbf{c}) \qquad i = 1, \ldots, M.$$

Here $u_i(\cdot, t)$ is an $\mathbb{R}^{q_i}$-valued function on spatial domain Ω_i, where q_i is its component count, and $\mathbf{u} = (u_1, \ldots, u_M)$ collects these fields. The given conditions $\mathbf{c}$ include geometry, parameters, forcing, and initial or boundary conditions. After discretization, $u_i^n \in \mathbb{R}^{n_i}$ denotes field values at time t_n, with n_i degrees of freedom. An implicit time-stepping rule gives $u_i^{n+1} = u_i^n + \Delta t\, f_i(\mathbf{u}^{n+1}; \mathbf{c})$, where n denotes the time steps and Δt is the step size. Given $\mathbf{u}^n$ and the other fields u_j^{n+1} for all $j \neq i$, solving this rule for u_i^{n+1} defines the isolated single-physics solution operator $\mathcal{M}_i$,

$$u_i^{n+1} = \mathcal{M}_i\big(u_1^{n+1}, \ldots, u_{i-1}^{n+1}, u_{i+1}^{n+1}, \ldots, u_M^{n+1}; \mathbf{u}^n, \Delta t, \mathbf{c}\big).$$

To solve all fields jointly, each operator's new-time inputs must agree with the other operators' outputs. Writing the unknown fields as $\mathbf{u}$ and absorbing the fixed inputs $\mathbf{u}^n$, Δt, and $\mathbf{c}$ into $\mathcal{M}_i$ gives the abstract form $u_i = \mathcal{M}_i(u_1, \ldots, u_M)$ for $i = 1, \ldots, M$. In this example, $\mathcal{M}_i$ uses only the other fields, while the general formulation also allows dependence on u_i. Stacking these operators into $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_M)$ gives the fixed-point equation

$$\mathbf{u} = \mathcal{M}(\mathbf{u}). \tag{1}$$

For a candidate state $\mathbf{u}$, the residual $R_{\text{phys}}(\mathbf{u}) = \mathbf{u} - \mathcal{M}(\mathbf{u})$ vanishes at coupled solutions. We collect fixed problem data into $\mathbf{x} = (\mathbf{u}^n, \Delta t, \mathbf{c})$ and write $\mathcal{M}_i(\mathbf{u}; \mathbf{x})$.

Module-wise pre-training. A physical module is a learned single-field model, typically pretrained on input–output pairs of the isolated single-physics response $\mathcal{M}_i$. Each sample comprises problem data $\mathbf{x}$, reference partner fields $\tilde{\mathbf{u}}$ that operator i takes as input, and the corresponding reference output u_i^{ref}. These pairs satisfy $u_i^{\text{ref}} = \mathcal{M}_i(\tilde{\mathbf{u}}; \mathbf{x})$, and we use them to train $\mathcal{M}_{\theta_i, i}$, with parameters θ_i, to approximate $\mathcal{M}_i$. For fixed $\mathbf{x}$, we again use the shorthand $\mathcal{M}_{\theta_i,i}(\mathbf{u})$. The model takes the physical operator's inputs, and training minimizes the prediction error over the samples,

$$\ell_{\text{pretrain},i} = \mathbb{E}_{\text{data}}\left[\left\|u_i^{\text{ref}} - \mathcal{M}_{\theta_i,i}(\tilde{\mathbf{u}}; \mathbf{x})\right\|_2^2\right]. \tag{2}$$

Coupled inference. At inference, $\mathbf{x}$ is given and the fields $\mathbf{u}$ are unknown. Replacing the physical operators in (1) with the pretrained models gives $\mathbf{u}_\theta^\star = \mathcal{M}_\theta(\mathbf{u}_\theta^\star; \mathbf{x})$, where $\mathcal{M}_\theta$ stacks the outputs of the modules $\mathcal{M}_{\theta_i,i}$ and $\theta = (\theta_1, \ldots, \theta_M)$. At this solution, each model's field inputs match the corresponding predictions. The learned equations have residual $R(\mathbf{u}, \theta; \mathbf{x}) = \mathbf{u} - \mathcal{M}_\theta(\mathbf{u}; \mathbf{x})$. Iterative methods such as Picard iteration or Anderson acceleration solve the coupled system. From initial field estimates with $\mathbf{x}$ fixed, the solver performs one complete field exchange at a time. We denote this exchange by G_θ, its field Jacobian by $J_G = \partial_{\mathbf{u}} G_\theta$, and its exchange residual by $R_G = \mathbf{u} - G_\theta(\mathbf{u})$. The exchange evaluates the coupled model, so G_θ and J_G correspond to $\mathcal{M}_\theta$ and its Jacobian. Appendix B.1 details the update rules and this correspondence.

3.2 ERROR PROPAGATION ACROSS COUPLED FIELDS

During coupled inference, each model receives predicted fields, so its output reflects both its approximation error and input errors. We analyze how input dependencies propagate these single-physics errors across coupled fields.

We describe input sensitivity using the coupling Jacobian $C = C(\mathbf{u}_\theta^\star)$ at the learned coupled solution, where $C(\mathbf{u}) = \partial_{\mathbf{u}} \mathcal{M}_\theta(\mathbf{u})$. Its block $C_{ij} = \partial_{u_j} \mathcal{M}_{\theta_i,i}|_{\mathbf{u}_\theta^\star}$ maps a small change in input field j to the resulting first-order change in model i's output. For modules without self-dependence, $C_{ii} = 0$.

The spectral radius $\rho(C)$ is the largest modulus of its eigenvalues, and we refer to it as the operator coupling intensity of the learned modules. We state the main conclusions informally below. The complete assumptions, quantitative bounds, and proofs are given in Appendix B.2, using standard perturbation arguments for fixed-point equations (Ortega & Rheinboldt, 1970).

Theorem 1 (Local module-to-system error bounds; informal)**.** *For fixed problem data* $\mathbf{x}$*, let* $\mathbf{u}^\star_{\mathrm{true}}$ *and* $\mathbf{u}^\star_\theta$ *denote the physical and learned coupled solutions. Define the module bias* $\mathbf{b} = \mathcal{M}_\theta(\mathbf{u}^\star_{\mathrm{true}}) - \mathbf{u}^\star_{\mathrm{true}}$ *and the coupled error* $\mathbf{h} = \mathbf{u}^\star_\theta - \mathbf{u}^\star_{\mathrm{true}}$*. Under the smoothness, locality, and spectral assumptions in Appendix B.2, with* $0 < \rho(C) < 1$*, define the resolvent* $A = (I - C)^{-1}$ *at the learned equilibrium. Then*

$$\begin{aligned} \mathbf{h} &= A\mathbf{b} + \mathcal{O}(\|\mathbf{b}\|_2^2), \\ \frac{\|A\mathbf{b}\|_2}{1+\eta} \le \|\mathbf{h}\|_2 &\le \frac{\|A\mathbf{b}\|_2}{1-\eta}, \end{aligned} \tag{3}$$

Here $\eta = \frac{1}{2} L_C \|A\|_2 \|\mathbf{h}\|_2 \le \frac{1}{2}$ *bounds the nonlinear correction relative to* $\|\mathbf{h}\|_2$*, where* L_C *is a Lipschitz constant of the coupling Jacobian on the convex region containing both equilibria. The remainder constant depends on* L_C *and* $\|A\|_2$*, as detailed in Appendix B.2.*

Module-wise pre-training optimizes accuracy under reference partner inputs, while deployment accuracy depends on the assembled system's response. During coupled inference, each module's errors perturb its partners' inputs. Theorem 1 separates the module bias $\mathbf{b}$ from its propagation through the resolvent $(I - C)^{-1}$ and bounds the resulting equilibrium error from both sides. This motivates post-training with a system-level loss that directly supervises the coupled predictions.

The spectrum of C characterizes how the resolvent acts on individual modal coefficients of the bias. For a left eigenvector $\boldsymbol{\psi}_\lambda$ associated with eigenvalue λ, the scalar $\boldsymbol{\psi}_\lambda^* \mathbf{b}$ measures the bias in that mode, while $\boldsymbol{\psi}_\lambda^* A\mathbf{b}$ is its coefficient in the linear response, with * denoting conjugate transpose. Their relation makes the spectral dependence explicit.

Corollary 2 (Modal gain and spectral ceiling; informal)**.** *Let* $\rho = \rho(C) < 1$ *and* $A = (I - C)^{-1}$*. For each eigenvalue* λ *and nonzero left eigenvector satisfying* $\boldsymbol{\psi}_\lambda^* C = \lambda \boldsymbol{\psi}_\lambda^*$*, let* g_λ *denote the magnitude of the scalar multiplier by which* A *propagates that modal coefficient. Then*

$$\boldsymbol{\psi}_\lambda^* A\mathbf{b} = \frac{\boldsymbol{\psi}_\lambda^* \mathbf{b}}{1-\lambda}, \qquad g_\lambda := \frac{1}{|1-\lambda|} \le \frac{1}{1-\rho}. \tag{4}$$

Thus, $(1-\rho)^{-1}$ *is a common ceiling on individual modal gains, attained exactly when* $\lambda = \rho$*.*

The modal gain depends on an eigenvalue's distance from $+1$, while the spectral radius supplies a common ceiling when $\rho(C) < 1$. Reducing $\rho(C)$ lowers this ceiling, though realized error also depends on which modes the bias excites, modal nonorthogonality, and nonlinear corrections. We therefore interpret spectral shifts alongside measured prediction errors, using this resolvent analysis to diagnose potential error amplification in subsequent experiments.

4 Multiphysics Post-Training

We propose multiphysics post-training, a fine-tuning framework for pretrained physical modules. The dataset comprises module-wise pre-training pairs with reference partner fields as inputs, coupled training data $(\mathbf{x}^{(m)}, \mathbf{u}^{(m)}_{\mathrm{gt}})$ for post-training, and held-out coupled data, randomly halved into a validation set to select the best checkpoint during training and a test set. Post-training minimizes the empirical loss on the coupled training data:

$$\mathcal{L}_{\mathrm{sys}}(\theta) = \mathbb{E}_m \ell\big(\mathbf{u}^{(m)}_{\mathrm{out}}(\theta), \mathbf{u}^{(m)}_{\mathrm{gt}}\big), \tag{5}$$

where $\mathbf{u}^{(m)}_{\mathrm{out}}$ is the joint prediction, $\mathbf{u}^{(m)}_{\mathrm{gt}}$ is the ground truth, and ℓ is the per-example loss. The gradient of this loss adapts the assembled modules while preserving their physical inputs, outputs, and exchange structure. Both implementations supervise numerical estimates of the coupled equilibrium. Unrolled-K uses the state after K field exchanges, whereas EquiPT uses the state returned by a numerical equilibrium solve.

Finite Unrolling. Unrolled-K performs K field exchanges, $\mathbf{u}_{k+1} = H_{\theta,d}(\mathbf{u}_k)$, where $H_{\theta,d}(\mathbf{u}) = (1-d)\mathbf{u} + d\, G_\theta(\mathbf{u}; \mathbf{x})$ with damping factor $d \in (0, 1]$, setting $\mathbf{u}_{\mathrm{out}} = \mathbf{u}_K$ for finite-depth objective

$\mathcal{L}_K$. Backpropagation yields exact gradients of this finite computation, storing $\mathcal{O}(K)$ activation states or using checkpointing to recompute intermediate steps (Griewank & Walther, 2008).

Equilibrium Post-Training. EquiPT first solves $R(\mathbf{u}, \theta; \mathbf{x}) = 0$ numerically for the coupled fixed point $\mathbf{u}_\theta^\star$ and minimizes the equilibrium loss

$$\mathcal{L}(\theta) = \mathbb{E}_m \ell(\mathbf{u}_\theta^\star(\mathbf{x}^{(m)}), \mathbf{u}_{\mathrm{gt}}^{(m)}). \tag{6}$$

Backpropagating through a long fixed-point solve requires substantial memory and computation. Following deep equilibrium models (DEQ) (Bai et al., 2019), we implicitly differentiate the fixed point, here the coupled state of pretrained physical modules, to avoid storing the solver's computation graph and potentially reduce backward computation. For one example with fixed point $\mathbf{u}^\star = \mathbf{u}_\theta^\star(\mathbf{x})$, let $C = C(\mathbf{u}^\star)$ and $P = \partial_\theta \mathcal{M}_\theta|_{\mathbf{u}^\star}$ be the coupling and parameter Jacobians, and $\mathbf{g} = \nabla_{\mathbf{u}} \ell(\mathbf{u}, \mathbf{u}_{\mathrm{gt}})|_{\mathbf{u}^\star}$ the loss gradient with respect to the joint state. When $I - C$ is invertible, the adjoint vector $\boldsymbol{\lambda} \in \mathbb{R}^N$ gives the per-example loss gradient through

$$(I - C^\top)\boldsymbol{\lambda} = \mathbf{g}, \qquad \nabla_\theta \ell = P^\top \boldsymbol{\lambda}, \tag{7}$$

where ∇_θ denotes the total gradient through $\mathbf{u}_\theta^\star$. Averaging these gradients gives $\nabla_\theta \mathcal{L}$.

The coupled equations determine the equilibrium gradient. Provided that solvers reach the same exact equilibrium and implicit differentiation conditions hold, this gradient is solver-independent (Proposition 7). In practice, finite forward and approximate adjoint solves estimate this gradient. Appendix C.3 gives full implementation details and training pseudocode (Algorithm 1).

Relation between unrolled and implicit gradients. Unrolled-K minimizes prediction error after K field exchanges, whereas EquiPT does so at the coupled fixed point. At fixed parameters, if a smooth iteration converges to that point and is locally contractive there, the unrolled gradient converges to EquiPT's implicit gradient as $K \to \infty$ (Corollary 10). EquiPT obtains this gradient from the fixed-point equation without backpropagating through solver iterations. Appendix B.3 and Appendix B.4 give the derivations and solver-independence result.

5 Experiments

Our experiments evaluate multiphysics post-training in three parts. First, we assess prediction accuracy across benchmarks and isolate the sources of improvement with controlled ablations (§5.2). Second, we compare finite unrolling and Equilibrium Post-Training across deployment depth, solver choice, and computational costs (§5.3). Third, we analyze performance across coupling regimes and relate observed gains to spectral diagnostics and theoretical modal ceilings (§5.4).

5.1 Experimental Setup

Benchmarks. We evaluate three distinct multiphysics systems, namely neutron–thermal transport (NT) modeling three-way cyclic thermal-hydraulic feedback, Gray–Scott reaction–diffusion (GS) capturing stiff autocatalytic reaction kinetics, and fluid–structure interaction (FSI) simulating bidirectional momentum and boundary exchange between fluid and a flexible beam behind a cylinder (Fig. 5, Appendix C.1). Pre-training pairs come from coupled GS and FSI simulations but decoupled NT solves. Coupled NT inference therefore also shifts the input distribution (Appendix C.1).

Training configurations. We compare post-training against three baselines and two controlled ablations (Table 4). All methods assemble one module per field. *Frozen assembly* evaluates unadapted Stage I modules at inference. *GenCP* and *M2PDE* provide frozen generative composition baselines. *Unrolled-K* and *EquiPT* implement our paradigm via finite unrolling and implicit equilibrium differentiation, respectively. To separate the contributions of pre-training and coupled supervision, *EquiPT w/o pre-training* trains from random initialization, while *module-wise continued training* (MW-CT) continues Stage I objectives on Stage II coupled data. All fine-tuned models share identical pretrained checkpoints, loss functions, and optimization budgets (Appendices C.2 and C.3).

Evaluation. For predictive accuracy, system error is the unweighted mean of field-wise relative L^2 test errors. At deployment, Unrolled-K predicts at its supervised training depth, whereas EquiPT solves to numerical equilibrium. Appendix C.4 documents deployment settings.

Table 1: **Relative L^2 prediction errors across multiphysics benchmarks.** Evaluation of baselines, training controls, and multiphysics post-training methods on NT, GS, and FSI. Errors are means over three training runs, with relative sample standard deviations in parentheses. GenCP and M2PDE report single runs. Bold and underlined values mark the lowest and second-lowest errors before rounding.

Method	NT				GS			FSI		
	System	Neutron	Fuel	Coolant	System	u	v	System	Structure	Fluid
Baselines										
Frozen (Martins & Hwang, 2013)	0.2056 (±7.23%)	0.0069 (±14.81%)	0.0860 (±10.99%)	0.5241 (±7.26%)	0.0313 (±22.70%)	0.0195 (±25.69%)	0.0431 (±21.34%)	0.1728 (±13.34%)	0.3126 (±13.93%)	0.0330 (±8.22%)
GenCP (Gao et al., 2026)	0.0315	**0.0029**	0.0372	0.0544	0.0373	0.0198	0.0548	0.0939	0.1485	0.0393
M2PDE (Zhang et al., 2024)	0.1120	0.0146	0.2298	0.0916	0.4555	0.2605	0.6505	0.1334	0.2208	0.0459
Training controls										
EquiPT w/o pre-training	0.0787 (±11.56%)	0.0231 (±9.63%)	0.1003 (±14.97%)	0.1127 (±8.92%)	0.0353 (±22.81%)	0.0203 (±19.78%)	0.0503 (±24.02%)	0.1051 (±2.65%)	0.1215 (±4.54%)	0.0887 (±0.20%)
MW-CT	0.0123 (±6.28%)	0.0041 (±5.19%)	0.0112 (±12.25%)	0.0215 (±7.34%)	0.0350 (±40.14%)	0.0219 (±44.33%)	0.0481 (±38.22%)	0.1650 (±14.08%)	0.2980 (±14.69%)	0.0321 (±8.56%)
Multiphysics post-training (ours)										
Unrolled-1	0.0704 (±0.39%)	0.0048 (±7.40%)	0.1769 (±0.05%)	0.0294 (±2.08%)	0.0201 (±1.76%)	0.0150 (±5.85%)	0.0252 (±2.83%)	0.0542 (±3.76%)	0.0684 (±5.89%)	0.0400 (±2.16%)
Unrolled-15	0.0094 (±5.14%)	0.0040 (±5.40%)	0.0107 (±10.48%)	0.0134 (±7.65%)	0.0196 (±15.71%)	0.0110 (±13.84%)	0.0282 (±16.46%)	**0.0417** (±5.06%)	**0.0602** (±6.45%)	0.0232 (±2.45%)
EquiPT	**0.0081** (±4.46%)	0.0040 (±5.68%)	**0.0086** (±3.23%)	**0.0116** (±8.08%)	**0.0161** (±3.04%)	**0.0092** (±3.32%)	**0.0229** (±2.88%)	0.0428 (±6.12%)	0.0639 (±7.65%)	**0.0216** (±2.29%)

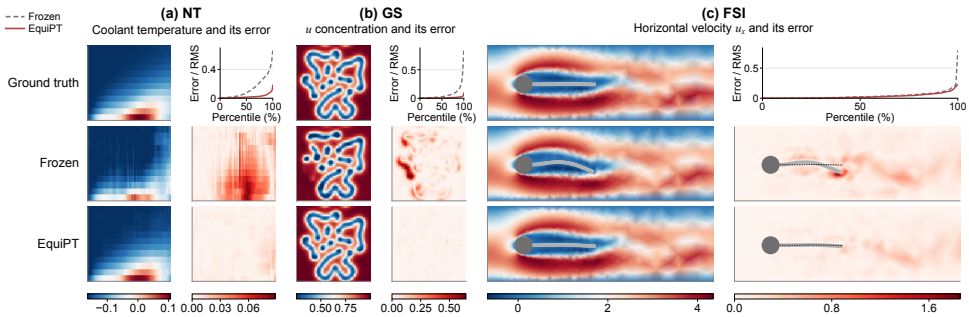

Figure 2: **Qualitative physical field predictions across benchmarks.** Test predictions of Frozen assembly and EquiPT. FSI fields are rendered on each method's predicted geometry and dashed lines mark the ground truth beam centerline.

5.2 Prediction accuracy and controlled ablations

As reported in Table 1, multiphysics post-training outperforms frozen assembly across all benchmarks. EquiPT reduces system prediction error by 96.1% on NT, 48.7% on GS, and 75.3% on FSI. Finite unrolling also yields consistent improvements with both Unrolled-1 and Unrolled-15. Qualitatively, Fig. 2 illustrates the physical consequences of this adaptation. Compared with frozen assembly, EquiPT resolves sharp NT coolant temperature profiles, closely follows the ground truth GS reaction pattern, and largely removes FSI beam distortion and near-wall velocity drift (fuller multi-field visualizations in Appendix D.9).

MW-CT starts from the same pretrained checkpoints as EquiPT and uses the same Stage II coupled data and optimization budget, but continues the module-wise objectives. Relative to MW-CT, EquiPT reduces system error by 34.2%, 54.2%, and 74.1% on NT, GS, and FSI, respectively. These comparisons show that coupled-data exposure alone does not explain the gains and point to system-level adaptation of the assembled modules as their source. Theorem 1 is consistent with this result. The equilibrium error is the module bias propagated through $(I - C)^{-1}$ up to a controlled nonlinear remainder. Module-wise training targets module prediction errors under reference partner inputs, whereas EquiPT directly supervises coupled predictions with the system-level loss.

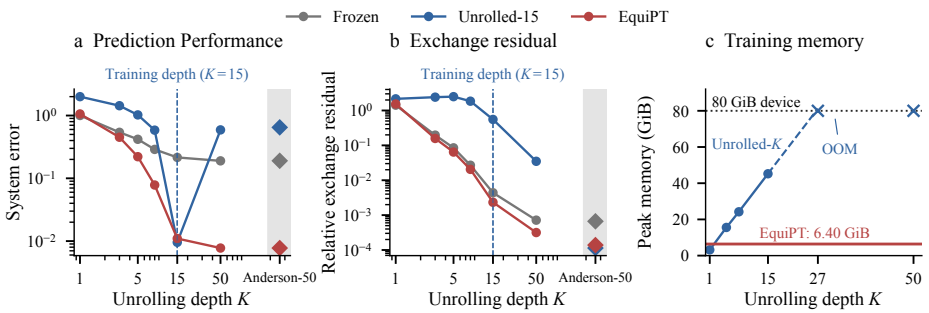

Figure 3: **Deployment behavior and training memory on NT.** (a) System error and (b) relative exchange residual versus deployment unrolling depth K for Frozen assembly, Unrolled-15, and EquiPT. The dashed vertical line marks the supervised training depth ($K = 15$). Diamond markers denote terminal Anderson-50 equilibrium solutions. (c) Peak training memory versus unrolling depth, where EquiPT preserves $\mathcal{O}(1)$ activation memory while Unrolled-K memory scales linearly.

Pretrained initialization further improves accuracy under the same post-training budget. EquiPT trained from random initialization has 9.8, 2.2, and 2.5 times the system error of pretrained EquiPT on NT, GS, and FSI, respectively.

Published generative composition frameworks, which compose frozen single-physics generative models, also fall short of EquiPT (Table 1). GenCP has 3.9, 2.3, and 2.2 times EquiPT's system error on NT, GS, and FSI, respectively, and M2PDE has 13.9, 28.4, and 3.1 times (Appendix D.3).

5.3 Deployment and Computational Trade-offs

Deployment depth and solver independence. EquiPT achieves lower system error than Unrolled-15 on NT and GS, while their FSI errors are comparable at 0.0428 and 0.0417, respectively (Table 1). These errors are measured at each model's prescribed deployment setting, so they do not show how the two implementations behave when the deployment setting changes. Taking NT as an example, the two respond differently to deployment depth (Fig. 3a,b). Unrolled-15 reaches an error of 0.0096 with a relative exchange residual of 0.55 at its supervised training depth, but solving the same post-trained modules to a tighter 1.1×10^{-4} residual with Anderson acceleration sharply raises the error to 0.6477. Appendix D.4 shows the same pattern across all three benchmarks and four training depths, so each unrolled model is tied to its training depth. Driving the residual lower degrades accuracy because training optimizes only the finite-depth output, whereas root-solving deployment evaluates the learned equilibrium. In contrast, EquiPT supervises the coupled fixed point directly and attains low error at numerical equilibrium.

Besides, to examine whether this degradation depends on solver choice, we evaluate the same learned modules with different root solvers. Once root solvers reach the same regular solution, the learned equations determine a solver-independent equilibrium (Proposition 7). On NT, EquiPT yields the identical 0.0077 system error under Picard, Aitken, Anderson, and Newton–Krylov, showing deployment robustness across solvers. Likewise, Picard, Aitken, and Anderson all drive Unrolled-15 to the same 0.6477 system error. The persistence of this error across the three solvers supports the distinction between residual convergence and prediction accuracy, as root solvers reduce residuals but cannot change the learned equilibrium, whose accuracy is set during training (Appendix D.5).

Computational cost and efficiency. As Fig. 3c shows, peak NT GPU training memory grows linearly with unrolling depth, reaching 45.22 GiB at $K = 15$ and running out of memory (OOM) at $K = 27$. In contrast, EquiPT's implicit adjoint stores no forward exchanges, so its activation memory is $\mathcal{O}(1)$ and its peak memory stays at 6.40 GiB. At inference, runtime depends on the solver and exchange count. Unrolled-15 requires $\approx 30\%$ of EquiPT's runtime on NT and GS, whereas Unrolled-1 requires $\approx 2\%$. FSI is an exception. With an easier fixed point where Anderson converges in a few iterations, Unrolled-15 still executes 15 exchanges, making EquiPT 2.3$\times$ faster at matched accuracy. In practice, finite unrolling suits latency-critical deployment within memory budgets, whereas EquiPT is well suited when unrolled activation storage exceeds training budgets or strict equilibrium fidelity is required (Table 9).

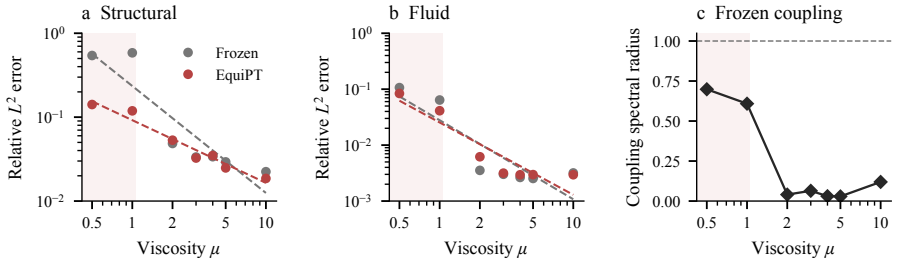

Figure 4: **FSI prediction error and interface sensitivity across fluid viscosities μ.** (a) Structural prediction error and (b) fluid prediction error for Frozen assembly and EquiPT. (c) Coupling spectral radius $\rho(C)$ of the Frozen assembly. The shaded region marks the strongly coupled regime ($\mu \leq 1$).

5.4 Coupling sensitivity and mechanism analysis

On FSI, Fig. 4 relates prediction error to coupling intensity across fluid viscosities μ, where lower viscosity amplifies interface momentum transfer and intensifies fluid–structure interaction. In the strongly coupled regime ($\mu \leq 1$), EquiPT gains the most and reduces structural error by 76.8%. As viscosity rises and physical coupling weakens, Frozen's errors decrease and both methods reach comparable accuracy, indicating that post-training primarily benefits strongly coupled regimes.

The modal-ceiling analysis below applies to diagnostic states with $\rho(C) < 1$. By Corollary 2, the component of the module bias along each left eigenmode of the coupling Jacobian is amplified at equilibrium by a linearized gain of at most $(1 - \rho(C))^{-1}$, a ceiling that rises sharply as $\rho(C) \to 1$. Across fluid viscosities (Fig. 4c), $\rho(C)$ of the frozen FSI assembly is elevated in the strongly coupled regime ($\mu \leq 1$), with many estimates near one (Fig. 6). The modal gain ceiling is therefore high in strongly coupled regimes, which coincides with the large unadapted errors.

Across all three benchmarks, post-training systematically shifts the diagnostic $\rho(C)$ distributions toward lower values than Frozen (Fig. 6), reducing mean spectral radii by 27% on NT, 12% on GS, and 18% on FSI. These shifts indicate lower estimated modal-gain ceilings and accompany EquiPT's improved prediction accuracy. Unrolled-15 also lowers these radii, yet its equilibrium remains highly inaccurate, since a lower ceiling alone does not determine the error and its 15-exchange objective never supervises the fixed point, which causes a training–deployment mismatch.

6 Discussion

Post-training preserves modularity and optimizes joint predictions, leaving room for three extensions. First, because our formulation requires only differentiability across exchanged boundary quantities, modules can incorporate adjoint-capable numerical solvers (e.g., differentiable FEM), enabling hybrid differentiable physics. Second, while our data provide dense reference fields, engineering systems often feature sparse sensors or interface measurements, and restricting the system-level loss to observed quantities directly accommodates partial observations. Third, scaling from 2D benchmarks to 3D geometries introduces higher-dimensional states where implicit solves become substantially more demanding, motivating faster forward and adjoint solvers alongside 3D applications such as prosthetic heart valves.

7 Conclusion

This work demonstrates that pretrained physical modules coupled along their interfaces should be trained on their joint predictions. Our module-to-system error bounds show that local module errors propagate through the coupling Jacobian, whose spectral radius bounds modal error amplification. Multiphysics post-training mitigates this propagated error while keeping modular interfaces. Across NT, GS, and FSI, EquiPT significantly outperforms frozen assembly, and its advantage over MW-CT on the same coupled data shows that the gains come from training at the coupled equilibrium rather than from coupled data alone. Pretrained initialization further improves accuracy under the same budget. The two implementations meet different needs, with Unrolled-K offering low latency and EquiPT targeting converged equilibria with $\mathcal{O}(1)$ memory. Finally, post-training reduces the coupling spectral radius, which lowers the modal error ceiling and accompanies the accuracy gains.

AI USE STATEMENT

In this work, generative AI tools (large language models) were used to assist with manuscript polishing and visualization scripts. Generative AI was not used for scientific ideation, methodology development, or experimental data collection. The authors have reviewed all AI-assisted content and take full responsibility for the final manuscript.

ETHICS STATEMENT

This work focuses on scientific machine learning for partitioned multiphysics simulation using synthetic numerical benchmarks. It involves no human participants, personal data, or sensitive applications. As with surrogate models in engineering simulation, real-world deployment in safety-critical domains requires rigorous numerical verification to prevent unmonitored out-of-distribution errors and numerical instability.

REPRODUCIBILITY STATEMENT

To ensure full reproducibility, Section 4 and Appendix C.3 (including Algorithm 1) detail the training objectives, implicit adjoint backward rules, and optimization workflows. Complete mathematical proofs for Theorem 1, Lemmas 4 and 8, Propositions 3, 6, 7, and 9, and Corollaries 2, 5, and 10 are provided in Appendix B. Experimental details, including modular neural architectures, coupling exchange schemes, solver tolerances, and training configurations across all three physical systems, are thoroughly documented in Appendix C, with key mathematical notation summarized in Appendix A. Anonymized source code, training configurations, and reproduction instructions are available at `https://anonymous.4open.science/r/Multiphysics-Posttraining-Anonymous-493D`.

REFERENCES

Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

Daniel Andrés Arcones, Rishith Ellath Meethal, Birgit Obst, and Roland Wüchner. Neural network-based surrogate models applied to fluid–structure interaction problems. *Collection of Papers Presented at the WCCM-APCOM*, 2022.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Multiscale deep equilibrium models. *Advances in neural information processing systems*, 33:5238–5250, 2020.

Aslak W Bergersen, Andreas Slyngstad, Sebastian Gjertsen, Alban Souche, and Kristian Valen-Sendstad. turtlefsi: A robust and monolithic fenics-based fluid-structure interaction solver. *Journal of Open Source Software*, 5(50):2089, 2020.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35:5230–5242, 2022.

Pavel Bochev, Justin Owen, Paul Kuberry, and Jeffrey Connors. Dynamic flux surrogate-based partitioned methods for interface problems. *Computer Methods in Applied Mechanics and Engineering*, 429:117115, 2024.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Shengze Cai, Zhicheng Wang, Lu Lu, Tamer A Zaki, and George Em Karniadakis. Deepm&mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *Journal of Computational Physics*, 436:110296, 2021.

Bruce Christianson. Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326, 1994.

Joris Degroote, Klaus-Jürgen Bathe, and Jan Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. *Computers & Structures*, 87 (11-12):793–801, 2009.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

S Dubreuil, Nathalie Bartoli, Christian Gogu, and T Lefebvre. Propagation of modeling uncertainty by polynomial chaos expansion in multidisciplinary analysis. *Journal of Mechanical Design*, 138 (11):111411, 2016.

James PC Duncan, Elynn Wu, Surya Dheeshjith, Adam Subel, Troy Arcomano, Spencer K Clark, Brian Henn, Anna Kwa, Jeremy McGibbon, W Andre Perkins, et al. Samudrace: Fast and accurate coupled climate modeling with 3d ocean and atmosphere emulators. *Geophysical Research Letters*, 53(11):e2025GL119340, 2026.

Xiantao Fan and Jian-Xun Wang. Differentiable hybrid neural modeling for fluid-structure interaction. *Journal of Computational Physics*, 496:112584, 2024.

Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. Jfb: Jacobian-free backpropagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6648–6656, 2022.

Tianrun Gao, Haoren Zheng, Wenhao Deng, Haodong Feng, Tao Zhang, Ruiqi Feng, Qianyi Chen, and Tailin Wu. Gencp: Towards generative modeling paradigm of coupled physics. *arXiv preprint arXiv:2601.19541*, 2026.

Zhengyang Geng and J Zico Kolter. Torchdeq: A library for deep equilibrium models. *arXiv preprint arXiv:2310.18605*, 2023.

Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Jean Charles Gilbert. Automatic differentiation and iterative processes. *Optimization methods and software*, 1(1):13–21, 1992.

Justin S Gray, John T Hwang, Joaquim RRA Martins, Kenneth T Moore, and Bret A Naylor. Openmdao: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, 2019.

Andreas Griewank and Christele Faure. Reduced functions, gradients and hessians from fixed-point iterations for state equations. *Numerical Algorithms*, 30(2):113–139, 2002.

Andreas Griewank and Andrea Walther. *Evaluating derivatives - principles and techniques of algorithmic differentiation, Second Edition*. SIAM, 2008. ISBN 978-0-89871-659-7. doi: 10.1137/1.9780898717761. URL https://doi.org/10.1137/1.9780898717761.

Rachit Gupta and Rajeev Jaiman. A hybrid partitioned deep learning methodology for moving interface and fluid–structure interaction. *Computers & Fluids*, 233:105239, 2022.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. *arXiv preprint arXiv:2403.03542*, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Hamda Hmida, Hsiu-Wen Chang Joly, and Youssef Mesri. Compno: A novel foundation model approach for solving partial differential equations. *Applied Sciences*, 16(2):972, 2026.

Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction — a review. *Communications in Computational Physics*, 12(2):337–377, 2012.

Ming-Chen Hsu, David Kamensky, Fei Xu, Josef Kiendl, Chenglong Wang, Michael CH Wu, Joshua Mineroff, Alessandro Reali, Yuri Bazilevs, and Michael S Sacks. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with t-splines and fung-type material models. *Computational mechanics*, 55(6):1211–1225, 2015.

Kostadin Ivanov and Maria Avramova. Challenges in coupled thermal–hydraulics and neutronics simulations for lwr safety analysis. *Annals of Nuclear Energy*, 34(6):501–513, 2007.

David E Keyes, Lois C McInnes, Carol Woodward, William Gropp, Eric Myra, Michael Pernice, John Bell, Jed Brown, Alain Clo, Jeffrey Connors, et al. Multiphysics simulations: Challenges and opportunities. *The International Journal of High Performance Computing Applications*, 27 (1):4–83, 2013.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.

Ulrich Küttler and Wolfgang A Wall. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational mechanics*, 43(1):61–72, 2008.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Ekachai Lelarasmee, Albert E Ruehli, and Alberto L Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE transactions on computer-aided design of integrated circuits and systems*, 1(3):131–145, 1982.

Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Chenkai Mao, Robert Lupoiu, Tianxiang Dai, Mingkun Chen, and Jonathan A Fan. Towards general neural surrogate solvers with specialized neural accelerators. *arXiv preprint arXiv:2405.02351*, 2024.

Joaquim RRA Martins and John T Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA journal*, 51(11):2582–2599, 2013.

Tanya Marwah, Ashwini Pokle, J. Zico Kolter, Zachary C. Lipton, Jianfeng Lu, and Andrej Risteski. Deep equilibrium based neural operators for steady-state PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15716–15737, 2023.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62, 2023.

James M. Ortega and Werner C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Computer science and applied mathematics. Academic Press, 1970. ISBN 978-0-12-528550-6.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Ashiqur Rahman, Robert J George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A Yeh, Jean Kossaifi, et al. Pretraining codomain attention neural operators for solving multiphysics pdes. *Advances in Neural Information Processing Systems*, 37: 104035–104064, 2024.

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

Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*, volume 66 of *Classics in Applied Mathematics*. SIAM, revised edition, 2011.

Khemraj Shukla, Patricio Clark Di Leoni, James Blackshire, Daniel Sparkman, and George Em Karniadakis. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *Journal of Nondestructive Evaluation*, 39(3):61, 2020.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. *Advances in neural information processing systems*, 36:71242–71262, 2023.

Amin Totounferoush, Axel Schumacher, and Miriam Schulte. Partitioned deep learning of fluid-structure interaction. *arXiv preprint arXiv:2105.06785*, 2021.

Lloyd N. Trefethen and Mark Embree. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005. doi: 10.1515/9780691213101.

Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in neural information processing systems*, 33:6111–6122, 2020.

Richard S. Varga. *Matrix Iterative Analysis*. Springer, 2 edition, 2000.

Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

Ezra Winston and J Zico Kolter. Monotone operator equilibrium networks. *Advances in neural information processing systems*, 33:10718–10728, 2020.

Xiongye Xiao, Defu Cao, Ruochen Yang, Gaurav Gupta, Gengshuo Liu, Chenzhong Yin, Radu Balan, and Paul Bogdan. Coupled multiwavelet neural operator learning for coupled partial differential equations. *arXiv preprint arXiv:2303.02304*, 2023.

Liyu Zerihun and Mark Shinyoung Lee. Time without timesteps: Simulating coupled dynamical systems via self-consistency. *arXiv preprint arXiv:2609.03358*, 2026.

Tao Zhang, Zhenhai Liu, Feipeng Qi, Yongjun Jiao, and Tailin Wu. M2pde: compositional generative multiphysics and multi-component pde simulation. *arXiv preprint arXiv:2412.04134*, 2024.

A NOTATION

Table 2 summarizes the mathematical notation employed throughout the manuscript for coupled physical equations, iterative numerical solves, error propagation analysis, and equilibrium sensitivity gradients. Throughout, state and parameter derivatives hold the problem data $\mathbf{x}$ fixed. Unsubscripted norms are Euclidean for vectors and induced spectral norms for matrices unless another induced norm is stated. Symbols used only within a single derivation are defined where they appear, and a hat marks a numerical approximation or a quantity evaluated at an approximate state.

Table 2: **Summary of mathematical notation.** Core notation and symbols used throughout the manuscript.

Symbol	Meaning
$M,\ n_i,\ N,\ I$	Number of fields, dimension of field i, total state dimension $N = \sum_i n_i$, and the identity matrix ($N \times N$ unless stated otherwise).
$\Omega_i,\ q_i,\ f_i$	Spatial domain, number of components, and right-hand side of the governing equation of field i.
$\mathbf{u} = (u_1, \ldots, u_M),\ \mathbf{x},\ \mathbf{c}$	Stacked physical fields, fixed problem data $\mathbf{x} = (\mathbf{u}^n, \Delta t, \mathbf{c})$, and given conditions.
$\theta,\ \theta_i,\ \Theta$	All module parameters, parameters of module i, and the subset selected for training.
$\mathcal{M}_{\theta_i,i},\ \mathcal{M}_\theta$	Learned model for field i and the coupled model stacking all model outputs.
$F_\theta,\ S_\theta$	Fluid and structure modules $\mathcal{M}_{\theta_1,1},\ \mathcal{M}_{\theta_2,2}$ of the two-field example in Appendix B.1.
$\mathcal{M}_i,\ \mathcal{M},\ \mathbf{u}^\star_{\mathrm{true}}$	Physical operator for field i, the stacked operator, and its coupled solution.
$\mathbf{u}^\star,\ \mathbf{u}^\star_\theta$	Exact equilibrium of the learned equations; θ is omitted when fixed.
$\mathbf{u}_{\mathrm{gt}}$	Ground truth from coupled simulations, the joint state used by the loss and by prediction error.
$\mathcal{T}$	Training set of paired problem data and ground truth in Algorithm 1.
$\tilde{\mathbf{u}},\ u_i^{\mathrm{ref}},\ \ell_{\mathrm{pretrain},i}$	Reference partner fields used as inputs in module-wise pre-training, the corresponding reference output, and the expected module-wise pre-training loss of module i.
$\mathbf{u}_{\mathrm{out}}$	Numerical output at the selected training or deployment setting.
$\mathbf{w},\ \mathbf{y},\ \hat{\mathbf{u}}$	Detached solver candidate, output of the recorded exchange $\mathbf{y} = G_\theta(\mathbf{w})$, and a generic approximate root; the implemented loss uses $\hat{\mathbf{u}} = \mathbf{y}$.
$C(\mathbf{u}),\ C,\ P$	Coupling Jacobian $C(\mathbf{u}) = \partial_{\mathbf{u}} \mathcal{M}_\theta(\mathbf{u}) \in \mathbb{R}^{N \times N}$; C without an argument is its value at the equilibrium in context, and $C(\hat{\mathbf{u}})$ its value at a numerical state; parameter Jacobian $P = \partial_\theta \mathcal{M}_\theta$.
$C_{ij}(\mathbf{u}),\ C_{ij},\ C_{\mathrm{low}},\ C_{\mathrm{diag}},\ C_{\mathrm{up}}$	Derivative blocks $C_{ij}(\mathbf{u}) = \partial_{u_j} \mathcal{M}_{\theta_i,i}(\mathbf{u})$ and $C_{ij} = C_{ij}(\mathbf{u}^\star_\theta)$; $C_{\mathrm{low}}, C_{\mathrm{diag}}, C_{\mathrm{up}}$ are the strictly block-lower, block-diagonal, and strictly block-upper parts of C.
$G_\theta,\ J_G,\ P_G$	One complete field exchange $\mathbf{u}^+ = G_\theta(\mathbf{u})$ with updated fields u_i^+, and derivatives $J_G = \partial_{\mathbf{u}} G_\theta$, $P_G = \partial_\theta G_\theta$. Synchronous exchange has $G_\theta = \mathcal{M}_\theta$; subscripts jac and gs identify exchange order.
$H_{\theta,d},\ J_H,\ P_H$	Damped exchange $H_{\theta,d} = (1-d)\mathbf{u} + d\, G_\theta$, with $J_H = (1-d)I + d\, J_G$ and $P_H = d\, P_G$.
$d,\ K,\ k$	Damping factor $d \in (0, 1]$, finite unrolling depth, and exchange index.
$J_{H,k},\ P_{H,k},\ Z_0,\ Z_k$	Derivatives along the finite trajectory, initial sensitivity $Z_0 = \mathrm{d}\mathbf{u}_0/\mathrm{d}\theta$, and trajectory sensitivity $Z_k = \mathrm{d}\mathbf{u}_k/\mathrm{d}\theta$.
R_{phys}	Physical equation residual $R_{\mathrm{phys}} = \mathbf{u} - \mathcal{M}(\mathbf{u}; \mathbf{x})$.
$R,\ R_G,\ r(\mathbf{u})$	Equation residual $R = \mathbf{u} - \mathcal{M}_\theta(\mathbf{u})$; exchange residual $R_G = \mathbf{u} - G_\theta(\mathbf{u})$, equal to R under synchronous exchange; relative exchange residual $r = \|R_G\|/\|\mathbf{u}\|$.
$\mathcal{D},\ L_C$	Convex set containing both equilibria and Lipschitz constant of the full model Jacobian on that set.
$\mathbf{u}^\dagger,\ \mathbf{p}$	In the error analysis, the exact physical solution $\mathbf{u}^\star_{\mathrm{true}}$ and learned equilibrium $\mathbf{u}^\star_\theta$, respectively.

Table 2, continued

Symbol	Meaning
$\mathbf{e},\ \boldsymbol{\delta},\ \Delta$	Absolute field error norms, module approximation bounds at $\mathbf{u}^\star_{\mathrm{true}}$, and $\Delta = \|\boldsymbol{\delta}\|_2$ in Corollary 5.
M_C	Growth constant $M_C = \sup_{k\geq 1} \|C^k\|_2/\rho(C)^k$ of the powers of C in Theorem 1; defined when $\rho(C) > 0$, and finite if and only if every eigenvalue of modulus $\rho(C)$ is semisimple (Lemma 4).
c_{nl}	Nonlinear remainder constant $2L_C\|(I-C)^{-1}\|_2^3$ in Corollary 5.
$\mathbf{h},\ \mathbf{b},\ \mathbf{q}$	Signed equilibrium error, module bias at the physical solution, and Taylor remainder, with $\mathbf{h} = A\mathbf{b} + A\mathbf{q}$.
$H,\ B,\ A,\ \eta$	Error norm $H = \|\mathbf{h}\|_2$, actual bias norm $B = \|\mathbf{b}\|_2 > 0$, resolvent $A = (I-C)^{-1}$, and local remainder factor $\eta = L_C\|A\|_2 H/2$ in the error analysis.
$\boldsymbol{\psi}_\rho,\ \boldsymbol{\psi}_\lambda,\ \alpha,\ \alpha_\lambda$	Left eigenvectors for eigenvalue $+\rho(C)$ or λ, chosen with unit norm; α is the largest $\|\boldsymbol{\psi}_\rho^*\mathbf{b}\|/B$ over such vectors (0 if $+\rho$ is not an eigenvalue), and $\alpha_\lambda = \|\boldsymbol{\psi}_\lambda^*\mathbf{b}\|/B$; $*$ denotes conjugate transpose.
$g_\lambda,\ (1-\rho)^{-1}$	Coefficient gain $g_\lambda = \|1-\lambda\|^{-1}$ of $A = (I-C)^{-1}$ on a left eigenmode, and its ceiling for $\rho = \rho(C) < 1$ in Corollary 2. Complex modes use the modulus of the complex projection coefficient.
$\rho(\cdot),\ s_\rho$	Matrix spectral radius; FSI diagnostic $s_\rho = \sqrt{\rho(J_{G,\mathrm{gs}})}$, equal to $\rho(C)$ at the evaluation state with its fluid field recomputed from the structural field.
$\ell,\ \mathcal{L},\ \mathcal{L}_K,\ \mathcal{L}_{\mathrm{sys}}$	Per-sample system loss, equilibrium objective, finite-depth objective, and general system-level training objective.
$\mathbf{g},\ \mathbf{g}_\star,\ \mathbf{g}_K$	Loss gradient with respect to the state at a specified state, an equilibrium, or the terminal iterate.
$\boldsymbol{\lambda}$	Adjoint of the module equations in (7).
$\boldsymbol{\lambda}_G,\ \hat{\boldsymbol{\lambda}},\ \mathbf{a}$	Adjoint of the exchange equations in (28), a candidate adjoint, and its linear-system residual; $\mathbf{a}_y$ is this residual at the recorded exchange output $\mathbf{y}$.
$\varepsilon_f,\ N_f$	Forward solver tolerance and iteration cap.
$\beta,\ \Gamma_G,\ \varepsilon_J,\ \varepsilon_P,\ \varepsilon_g$	Uniform derivative bounds and trajectory deviations in Proposition 9.
$c_R,\ c_f,\ c_b,\ L_{P_G}$	Local residual-to-state and residual-to-gradient constants in Lemma 8, and a Lipschitz constant for P_G.
$u,\ v,\ D_u,\ D_v$	GS species concentrations and their diffusion coefficients; chemical species in the reaction are denoted U, V.
$f_{\mathrm{feed}},\ k_{\mathrm{kill}}$	GS feed and kill rates.
μ	FSI viscosity.
$t,\ t_n,\ \Delta t,\ T,\ \Delta t_{\mathrm{sim}}$	Physical time, time level with index n, prediction step size, and simulation duration; Δt_{sim} denotes the internal time step of the FSI simulation.

B PROOFS AND DERIVATIONS

The physical module equations define the coupled equilibrium independently of the specific numerical exchange format used to solve them. This section establishes the equivalence between different exchange schemes, proves the module-to-system error bound, and characterizes the mathematical relationship between finite-depth unrolled gradients and the implicit equilibrium adjoint.

B.1 EXCHANGE SCHEMES FOR M MODULES

Each physical module predicts one field using estimates of the fields on which it depends. A *field exchange* is one round in which every module is evaluated once to update these predictions. Throughout this round, the model parameters θ and the problem data $\mathbf{x}$, including the prescribed conditions and any given past states, remain fixed. We write u_i for field i before the round and u_i^+ for its updated prediction. The superscript $+$ denotes a new coupling estimate at the same physical prediction target, not advancement to another physical time. This subsection explains two ways to choose the inputs within a round and why they give the same equilibrium sensitivity.

Synchronous and staggered exchanges. We use a fluid–structure example to introduce the two exchange schemes and explain which field estimates each module receives. The fluid prediction depends on an estimate of the structural displacement at the interface, while the structural prediction depends on the load obtained from the fluid prediction. Write u_1 for the fluid state and u_2 for the structural state. Let F_θ map a structural estimate to a fluid prediction, and let S_θ map a fluid estimate to a structural prediction, including the extraction of the interface load. These are the two learned modules $\mathcal{M}_{\theta_1,1}$ and $\mathcal{M}_{\theta_2,2}$, respectively. With the fixed data $\mathbf{x}$ suppressed, the coupled equations are

$$u_1^\star = F_\theta(u_2^\star), \qquad u_2^\star = S_\theta(u_1^\star).$$

The star denotes an equilibrium, where each module's input agrees with the other module's output. We write $\mathbf{u}^\star = \mathbf{u}_\theta^\star(\mathbf{x})$, as in the main text, with $u_i^\star$ denoting its field i. Starting from the current estimates (u_1, u_2), a *synchronous exchange*, also called a Jacobi update, uses the old estimates for both predictions,

$$\underbrace{u_1^+ = F_\theta(u_2)}_{\text{fluid uses old structural estimate}}, \qquad \underbrace{u_2^+ = S_\theta(u_1)}_{\text{structure uses old fluid estimate}}.$$

Both predictions become available for the next round. A *staggered exchange*, also called a Gauss–Seidel update, instead updates the fluid first and immediately supplies its new prediction to the structure module,

$$\underbrace{u_1^+ = F_\theta(u_2)}_{\text{update fluid first}}, \qquad \underbrace{u_2^+ = S_\theta(u_1^+)}_{\text{structure uses new fluid prediction}}.$$

The distinction is which input estimate the structure module receives. At equilibrium, old and updated predictions coincide, so both procedures satisfy the same two coupled equations.

For M fields, staggered exchange uses earlier modules' updated predictions within the same round. We now extend the two-field example to M fields and define the coupled model and iteration maps used in the derivative calculations below. Let $u_i \in \mathbb{R}^{n_i}$ contain the discretized values of field i, and let $\mathbf{u} = (u_1, \dots, u_M) \in \mathbb{R}^N$. The learned module $\mathcal{M}_{\theta_i,i}$ predicts field i, with parameters θ_i. The vector θ collects all module parameters. A module may use only a subset of the fields, although we write the full state as its argument. The coupled model

$$\mathcal{M}_\theta(\mathbf{u};\mathbf{x}) = \big(\mathcal{M}_{\theta_1,1}(\mathbf{u};\mathbf{x}), \dots, \mathcal{M}_{\theta_M,M}(\mathbf{u};\mathbf{x})\big)$$

collects the module outputs evaluated at the same input state. The coupled equations are $\mathbf{u} = \mathcal{M}_\theta(\mathbf{u};\mathbf{x})$. Separately, G_θ denotes the operation that performs one complete exchange, so $\mathbf{u}^+ = G_\theta(\mathbf{u};\mathbf{x})$.

For synchronous exchange, every module receives the incoming state $\mathbf{u}$,

$$u_i^+ = \mathcal{M}_{\theta_i,i}(u_1, \dots, u_M;\mathbf{x}), \qquad i = 1, \dots, M.$$

Thus $G_{\theta,\mathrm{jac}} = \mathcal{M}_\theta$, where the subscript jac identifies the Jacobi update. For staggered exchange, choose the order in which modules are evaluated and number them $1, \dots, M$ in that order. When

module i is evaluated, fields $1, \dots, i-1$ have already been updated. The other fields still have their incoming values. The update is therefore

$$u_i^+ = \mathcal{M}_{\theta_i,i}\big(\underbrace{u_1^+, \dots, u_{i-1}^+}_{\text{updated in this round}}, \underbrace{u_i, \dots, u_M}_{\text{incoming estimates}}; \mathbf{x}\big), \qquad i = 1, \dots, M.$$

The resulting complete exchange is denoted $G_{\theta,\mathrm{gs}}$, with gs identifying the Gauss–Seidel update. This definition also allows a module to depend on its own incoming estimate u_i. Directly repeating $\mathbf{u}_{k+1} = G_\theta(\mathbf{u}_k; \mathbf{x})$ gives a synchronous or staggered Picard-type iteration, with one evaluation per module per round. A Newton solver instead computes corrections to solve the coupled residual equation. The Jacobians below describe the module and exchange maps, rather than the Newton update.

Exchange order can change both the state and parameter Jacobians. We now compare how the two exchanges respond to small changes in state and parameters. Define

$$C = \partial_{\mathbf{u}} \mathcal{M}_\theta, \qquad P = \partial_\theta \mathcal{M}_\theta, \qquad J_G = \partial_{\mathbf{u}} G_\theta, \qquad P_G = \partial_\theta G_\theta.$$

Here C and J_G describe the response to changes in the input fields, while P and P_G describe the response to changes in model parameters at fixed input fields. The state Jacobians C and J_G are $N \times N$ matrices. The (i, j) block $C_{ij} = \partial_{u_j} \mathcal{M}_{\theta_i,i} \in \mathbb{R}^{n_i \times n_j}$ describes how module i responds to a change in field j.

Evaluate these derivatives at a common equilibrium $\mathbf{u}^\star$. Split $C = C_{\mathrm{low}} + C_{\mathrm{diag}} + C_{\mathrm{up}}$ according to the chosen update order. The matrix C_{low} contains the blocks C_{ij} with $j < i$, corresponding to inputs updated earlier in the round. The matrix C_{diag} contains the blocks C_{ii}, corresponding to each module's own input. The matrix C_{up} contains the blocks C_{ij} with $j > i$, corresponding to inputs updated later. All other blocks in each of these three matrices are zero. For the staggered exchange, the chain rule gives

$$\mathrm{d}\mathbf{u}^+ = C_{\mathrm{low}}\, \mathrm{d}\mathbf{u}^+ + (C_{\mathrm{diag}} + C_{\mathrm{up}})\, \mathrm{d}\mathbf{u} + P\, \mathrm{d}\theta,$$

where $\mathrm{d}\mathbf{u}$, $\mathrm{d}\mathbf{u}^+$ and $\mathrm{d}\theta$ denote infinitesimal changes in the incoming state, updated state and parameters. The first term uses $\mathrm{d}\mathbf{u}^+$ because later modules receive the predictions already updated in this round. At the common equilibrium, all module derivatives are evaluated at the same state. Moving this term to the left and solving for $\mathrm{d}\mathbf{u}^+$ yields

$$\begin{aligned} J_{G,\mathrm{jac}} &= C, & P_{G,\mathrm{jac}} &= P, \\ J_{G,\mathrm{gs}} &= (I - C_{\mathrm{low}})^{-1}(C_{\mathrm{diag}} + C_{\mathrm{up}}), & P_{G,\mathrm{gs}} &= (I - C_{\mathrm{low}})^{-1} P. \end{aligned} \tag{8}$$

Here I is the $N \times N$ identity matrix. The first row holds at every state because $G_{\theta,\mathrm{jac}} = \mathcal{M}_\theta$. The second row uses the common equilibrium. Since C_{low} has nonzero blocks only below its diagonal, $I - C_{\mathrm{low}}$ is block lower triangular with identity diagonal blocks and is always invertible.

Both exchanges give the same equilibrium sensitivity on a common regular solution branch. A *regular equilibrium* is a solution $\mathbf{u}^\star$ for which $I - C$ is invertible. When the module maps are jointly continuously differentiable in state and parameters, this condition allows the equilibrium to be followed as a locally unique differentiable function $\mathbf{u}^\star(\theta)$ as the parameters change. We call this function a local solution branch, and its derivative $\mathrm{d}\mathbf{u}^\star/\mathrm{d}\theta$ the equilibrium sensitivity. The following result concerns that derivative, rather than the derivative of a finite number of exchanges.

Proposition 3 (Scheme independence)**.** *For the exchange maps defined above, on states where the module evaluations are well-defined, the synchronous and staggered exchanges have the same fixed points. Suppose the module maps are jointly continuously differentiable in state and parameters on an open neighborhood of one such point. Then, at that point, $I - J_{G,\mathrm{gs}}$ is invertible if and only if $I - C$ is invertible. On the same regular solution branch, both exchanges give*

$$\frac{\mathrm{d}\mathbf{u}^\star}{\mathrm{d}\theta} = (I - C)^{-1} P.$$

Proof. At a staggered fixed point, $\mathbf{u}^+ = \mathbf{u}$. Substituting this equality into every module update gives $u_i = \mathcal{M}_{\theta_i,i}(\mathbf{u}; \mathbf{x})$ for all i, which are exactly the synchronous fixed-point equations. Conversely, if all these equations hold, the first module leaves its field unchanged, then the second does the same, and so on. Thus a coupled equilibrium is also a staggered fixed point.

Equation (8) gives

$$I - J_{G,\mathrm{gs}} = (I - C_{\mathrm{low}})^{-1}\big[(I - C_{\mathrm{low}}) - (C_{\mathrm{diag}} + C_{\mathrm{up}})\big] = (I - C_{\mathrm{low}})^{-1}(I - C).$$

Since $I - C_{\mathrm{low}}$ is invertible, the two matrices on the left and right are invertible together. When $I - C$ is invertible, the implicit function theorem makes the solution branch $\mathbf{u}^\star(\theta)$ continuously differentiable near this point, and differentiating the equilibrium equation $\mathbf{u}^\star = G_\theta(\mathbf{u}^\star; \mathbf{x})$ gives

$$(I - J_G)\frac{\mathrm{d}\mathbf{u}^\star}{\mathrm{d}\theta} = P_G.$$

For the staggered exchange, its solution is

$$(I - J_{G,\mathrm{gs}})^{-1}P_{G,\mathrm{gs}} = (I - C)^{-1}(I - C_{\mathrm{low}})(I - C_{\mathrm{low}})^{-1}P = (I - C)^{-1}P.$$

For the synchronous exchange, the same expression follows directly from $J_{G,\mathrm{jac}} = C$ and $P_{G,\mathrm{jac}} = P$. □

Consequently, the gradient of a differentiable loss evaluated on the same regular equilibrium branch is also independent of the exchange order. The two exchanges can nevertheless differ in convergence behavior, in their outputs after a finite number of rounds, and in approximate gradients. Regularity only excludes the eigenvalue 1 from C. It does not ensure convergence of either iteration.

For two fields without self-dependence, $\rho(J_{G,\mathrm{gs}}) = \rho(C)^2$ at a common equilibrium. Return to the two-field example above, where each module uses the other field but not its own estimate. At a common equilibrium, $C_{12} = \partial_{u_2} F_\theta$ and $C_{21} = \partial_{u_1} S_\theta$ are the two interface derivatives, so

$$C = \begin{pmatrix} 0 & C_{12} \\ C_{21} & 0 \end{pmatrix}, \qquad J_{G,\mathrm{gs}} = \begin{pmatrix} 0 & C_{12} \\ 0 & C_{21}C_{12} \end{pmatrix}.$$

The product $C_{21}C_{12}$ describes a perturbation of the structural estimate passing through the fluid module and then the structure module in one staggered exchange. For a square matrix X, its spectral radius $\rho(X)$ is the largest absolute value of its eigenvalues. Since $C^2 = \mathrm{diag}(C_{12}C_{21}, C_{21}C_{12})$ and the two products share their nonzero eigenvalues,

$$\rho(J_{G,\mathrm{gs}}) = \rho(C_{21}C_{12}) = \rho(C)^2. \tag{9}$$

This is the two-cyclic case of Young's relation (Varga, 2000, Ch. 4). It requires the stated two-field structure and a common evaluation point for the module derivatives. The blocks C_{12} and C_{21} may be rectangular because the two fields can have different numbers of degrees of freedom.

The NT and GS diagnostic probes use synchronous exchanges and estimate $\rho(C(\hat{\mathbf{u}}))$ at their numerical evaluation states $\hat{\mathbf{u}}$. FSI uses staggered exchange and reports $s_\rho = \sqrt{\rho(J_{G,\mathrm{gs}})}$. The FSI fluid module reads only the structural field and the structure module reads only the fluid field, so one staggered exchange from any state $\mathbf{u} = (u_1, u_2)$ evaluates $C_{12} = \partial_{u_2} F_\theta$ at u_2 and $C_{21} = \partial_{u_1} S_\theta$ at the updated fluid field $F_\theta(u_2)$. The Jacobian $J_{G,\mathrm{gs}}(\mathbf{u})$ keeps the block upper-triangular form above with lower-right block $C_{21}C_{12}$, and C_{12} and C_{21} are the off-diagonal blocks of $C(\bar{\mathbf{u}})$ at $\bar{\mathbf{u}} = (F_\theta(u_2), u_2)$, the state whose fluid field is recomputed from u_2. The argument leading to (9), with every block evaluated at $\bar{\mathbf{u}}$, then gives $\rho(J_{G,\mathrm{gs}}(\mathbf{u})) = \rho(C_{21}C_{12}) = \rho(C(\bar{\mathbf{u}}))^2$, that is, $s_\rho(\mathbf{u}) = \rho\big(C(\bar{\mathbf{u}})\big)$ at every state, which is why the main text reports the FSI diagnostic as $\rho(C)$. At an exact coupled equilibrium $\bar{\mathbf{u}} = \mathbf{u}^\star$, so s_ρ equals $\rho(C)$ there. The estimation protocols are given in Appendix C.4.

For the same two-field structure, the staggered equilibrium can also be found by solving only for the structural state,

$$u_2^\star = S_\theta(F_\theta(u_2^\star)), \qquad u_1^\star = F_\theta(u_2^\star).$$

Our FSI assembly iterates this staggered Dirichlet–Neumann exchange on the full state. Because the fluid module does not read the incoming fluid estimate, this full-state iteration has the same fixed points and exact equilibrium gradient as the reduced structural iteration. Structural displacement supplies the fluid-side interface boundary data, and the resulting fluid load supplies the structure-side boundary data (Degroote et al., 2009; Küttler & Wall, 2008).

B.2 Formal statements and proofs of the error bounds

This result relates errors made by individual modules to the error in their coupled equilibrium. Two effects contribute. A learned module can differ from its physical operator even when both receive the same inputs. In addition, errors in the partner fields change the inputs supplied to that module. In the fluid–structure example, the fluid prediction can therefore change both because the fluid module is imperfect and because the structural displacement supplied to it is inaccurate. The proof retains the full coupling Jacobian and separates the module bias, its linear propagation through the coupled equations, and the nonlinear remainder. The complete statements below provide the assumptions and constants omitted from the informal versions in the main text.

Formal statement of Theorem 1. Fix $\mathbf{x}$, and let $\mathbf{u}^\dagger = \mathbf{u}^\star_{\mathrm{true}}$ and $\mathbf{p} = \mathbf{u}^\star_\theta$ be the physical and learned coupled solutions, with $\mathbf{u}^\dagger, \mathbf{p} \in \mathcal{D}$ for a convex set $\mathcal{D} \subseteq \mathbb{R}^N$, $N = \sum_i n_i$. Suppose $\mathcal{M}_\theta$ is continuously differentiable on an open set containing $\mathcal{D}$ and $\|C(\mathbf{v}) - C(\mathbf{z})\|_2 \le L_C \|\mathbf{v} - \mathbf{z}\|_2$ for all $\mathbf{v}, \mathbf{z} \in \mathcal{D}$. Define the signed error $\mathbf{h} = \mathbf{p} - \mathbf{u}^\dagger$, the module bias $\mathbf{b} = \mathcal{M}_\theta(\mathbf{u}^\dagger) - \mathbf{u}^\dagger$, $H = \|\mathbf{h}\|_2$, and $B = \|\mathbf{b}\|_2 > 0$. Let $C = C(\mathbf{p})$ have spectral radius $0 < \rho < 1$ with semisimple eigenvalues of modulus ρ, so that $M_C = \sup_{k \ge 1} \|C^k\|_2 / \rho^k$ is finite (Lemma 4). Set $A = (I - C)^{-1}$ and assume the locality condition $\eta = \frac{1}{2} L_C \|A\|_2 H \le \frac{1}{2}$. Let $\alpha \in [0, 1]$ be the largest value of $|\boldsymbol{\psi}_\rho^* \mathbf{b}|/B$ over unit left eigenvectors $\boldsymbol{\psi}_\rho^* C = \rho \boldsymbol{\psi}_\rho^*$, which measures the component of the bias along the left eigenvectors of $+\rho$, and set $\alpha = 0$ if $+\rho$ is not an eigenvalue of C. Then $\mathbf{h} = A\mathbf{b} + A\mathbf{q}$ for a nonlinear remainder $\mathbf{q}$ with $\|A\mathbf{q}\|_2 \le \eta H$, and the error amplification satisfies

$$\frac{\alpha}{(1+\eta)(1-\rho)} \le \frac{H}{B} \le \frac{1 + (M_C - 1)\rho}{(1-\eta)(1-\rho)}. \tag{10}$$

The following lemma bounds the linear propagation through the powers of C, which can grow transiently when C is nonnormal (Trefethen & Embree, 2005).

Lemma 4 (Powers of the coupling Jacobian)**.** *Let $C \in \mathbb{R}^{N \times N}$ have spectral radius $\rho = \rho(C) > 0$, and let $M_C = \sup_{k \ge 1} \|C^k\|_2 / \rho^k$, possibly infinite.*

(i) If $\rho < 1$ and $M_C < \infty$, then $I - C$ is invertible and

$$(I - C)^{-1} C = \sum_{k \ge 1} C^k, \qquad \|(I - C)^{-1} C\|_2 \le \sum_{k \ge 1} \|C^k\|_2 \le \frac{M_C \rho}{1 - \rho}.$$

(ii) $M_C < \infty$ if and only if every eigenvalue of C with modulus ρ is semisimple, that is, every Jordan block associated with an eigenvalue of modulus ρ is 1×1.

(iii) $M_C \ge 1$. If $C = V \Lambda V^{-1}$ over $\mathbb{C}$ with Λ diagonal, then $M_C \le \|V\|_2 \|V^{-1}\|_2$. If C is normal, then $M_C = 1$.

Proof. (i) By definition, $\|C^k\|_2 \le M_C \rho^k$ for all $k \ge 1$, so the Neumann series $\sum_{k \ge 0} C^k$ converges absolutely. Since $(I - C) \sum_{k=0}^{K} C^k = I - C^{K+1}$ and $C^{K+1} \to 0$, the series satisfies $(I - C) \sum_{k \ge 0} C^k = I$, so $I - C$ is invertible with $(I - C)^{-1} = \sum_{k \ge 0} C^k$. Hence $(I - C)^{-1} C = (I - C)^{-1} - I = \sum_{k \ge 1} C^k$. The triangle inequality and $\sum_{k \ge 1} \rho^k = \rho/(1 - \rho)$ give the bound.

(ii) Let $C = Q \Xi Q^{-1}$ be a Jordan decomposition over $\mathbb{C}$ with blocks $\Xi_l = \nu_l I_{m_l} + E_l$, where E_l is the $m_l \times m_l$ matrix with ones on the first superdiagonal and zeros elsewhere, so that $E_l^{m_l} = 0$ and $\|E_l\|_2 \le 1$. Since $C^k = Q \Xi^k Q^{-1}$ and $\|\Xi^k\|_2 = \max_l \|\Xi_l^k\|_2$,

$$\frac{\max_l \|\Xi_l^k\|_2}{\|Q\|_2 \|Q^{-1}\|_2} \le \|C^k\|_2 \le \|Q\|_2 \|Q^{-1}\|_2 \max_l \|\Xi_l^k\|_2.$$

Hence $M_C < \infty$ if and only if $\sup_{k \ge 1} \|\Xi_l^k\|_2 / \rho^k < \infty$ for every block. Because $\nu_l I_{m_l}$ and E_l commute, $\Xi_l^k = \sum_{s=0}^{\min(k, m_l - 1)} \binom{k}{s} \nu_l^{k-s} E_l^s$. If $\nu_l = 0$, then $\Xi_l^k = 0$ for $k \ge m_l$, so only finitely many ratios are nonzero. If $0 < |\nu_l| < \rho$, then $\|\Xi_l^k\|_2 / \rho^k \le \sum_{s=0}^{m_l - 1} \binom{k}{s} |\nu_l|^{-s} (|\nu_l|/\rho)^k$, which tends to zero as $k \to \infty$. If $|\nu_l| = \rho$ and $m_l = 1$, the ratio equals 1. If $|\nu_l| = \rho$ and $m_l \ge 2$, the $(1, 2)$ entry of Ξ_l^k is $k \nu_l^{k-1}$, so the ratio is at least k/ρ and is unbounded. The supremum is therefore finite exactly when every block with $|\nu_l| = \rho$ is 1×1.

(iii) For every $k \geq 1$, $\|C^k\|_2 \geq \rho(C^k) = \rho^k$. If $C = V\Lambda V^{-1}$, then $\|C^k\|_2 = \|V\Lambda^k V^{-1}\|_2 \leq \|V\|_2\|V^{-1}\|_2\,\rho^k$. A normal C admits a unitary V, for which the bound is 1, so $M_C = 1$. □

Proof of Theorem 1. Fix $\mathbf{x}$ and suppress it in operator arguments. Let $\mathbf{u}^\dagger = \mathbf{u}^\star_{\text{true}}$ satisfy $\mathbf{u}^\dagger = \mathcal{M}(\mathbf{u}^\dagger)$ and $\mathbf{p} = \mathbf{u}^\star_\theta$ satisfy $\mathbf{p} = \mathcal{M}_\theta(\mathbf{p})$. Both states lie in the convex set $\mathcal{D}$, on which $\mathcal{M}_\theta$ is continuously differentiable with L_C-Lipschitz Jacobian. Write

$$\mathbf{h} = \mathbf{p} - \mathbf{u}^\dagger, \quad \mathbf{b} = \mathcal{M}_\theta(\mathbf{u}^\dagger) - \mathbf{u}^\dagger, \quad H = \|\mathbf{h}\|_2, \quad B = \|\mathbf{b}\|_2 > 0, \quad C = C(\mathbf{p}).$$

The theorem assumes $0 < \rho = \rho(C) < 1$, $M_C < \infty$, and $\eta = L_C\|A\|_2 H/2 \leq 1/2$, where $A = (I - C)^{-1}$ exists by Lemma 4(i). The lower bound is trivial when $\alpha = 0$; otherwise take a unit left eigenvector $\boldsymbol{\psi}_\rho^* C = \rho\boldsymbol{\psi}_\rho^*$ attaining the maximum in the definition of α, so that $|\boldsymbol{\psi}_\rho^*\mathbf{b}| = \alpha B$.

Using the learned fixed-point equation and integrating along the segment between the two solutions gives

$$\begin{aligned}\mathbf{h} &= \mathcal{M}_\theta(\mathbf{p}) - \mathcal{M}_\theta(\mathbf{u}^\dagger) + \mathbf{b} \\ &= \int_0^1 C(\mathbf{u}^\dagger + \tau\mathbf{h})\,\mathbf{h}\,\mathrm{d}\tau + \mathbf{b} = C\mathbf{h} + \mathbf{b} + \mathbf{q},\end{aligned}$$

where

$$\mathbf{q} = \int_0^1 \big[C(\mathbf{u}^\dagger + \tau\mathbf{h}) - C\big]\mathbf{h}\,\mathrm{d}\tau.$$

Convexity keeps the segment in $\mathcal{D}$. Since C is evaluated at $\mathbf{p} = \mathbf{u}^\dagger + \mathbf{h}$, the Lipschitz condition yields

$$\|\mathbf{q}\|_2 \leq \int_0^1 L_C(1-\tau)H^2\,\mathrm{d}\tau = \frac{L_C}{2}H^2. \tag{11}$$

Applying A to $(I - C)\mathbf{h} = \mathbf{b} + \mathbf{q}$ gives the exact identity

$$\mathbf{h} = A\mathbf{b} + A\mathbf{q}, \qquad \|A\mathbf{q}\|_2 \leq \eta H. \tag{12}$$

The triangle and reverse triangle inequalities imply $|H - \|A\mathbf{b}\|_2| \leq \eta H$. Since $\eta < 1$, this is equivalent to

$$\frac{\|A\mathbf{b}\|_2}{1+\eta} \leq H \leq \frac{\|A\mathbf{b}\|_2}{1-\eta}. \tag{13}$$

To bound the resolvent, use $A = I + AC$ and Lemma 4(i),

$$\|AC\|_2 = \Big\|\sum_{k\geq 1} C^k\Big\|_2 \leq \frac{M_C\rho}{1-\rho}, \qquad \|A\|_2 \leq 1 + \frac{M_C\rho}{1-\rho} = \frac{1 + (M_C - 1)\rho}{1-\rho}. \tag{14}$$

Inserting $\|A\mathbf{b}\|_2 \leq \|A\|_2 B$ into (13) proves the upper bound in (10).

For the lower bound, $\boldsymbol{\psi}_\rho^*(I - C) = (1-\rho)\boldsymbol{\psi}_\rho^*$ implies $\boldsymbol{\psi}_\rho^* A = (1-\rho)^{-1}\boldsymbol{\psi}_\rho^*$. As $\|\boldsymbol{\psi}_\rho\|_2 = 1$,

$$\|A\mathbf{b}\|_2 \geq |\boldsymbol{\psi}_\rho^* A\mathbf{b}| = \frac{|\boldsymbol{\psi}_\rho^*\mathbf{b}|}{1-\rho} = \frac{\alpha B}{1-\rho}.$$

Combining this inequality with the lower side of (13) proves the lower bound in (10). □

Formal statement of Corollary 2. Let $C \in \mathbb{R}^{N\times N}$ have spectral radius $\rho = \rho(C) < 1$ and set $A = (I - C)^{-1}$. For every eigenvalue λ of C with nonzero left eigenvector $\boldsymbol{\psi}_\lambda \in \mathbb{C}^N$, $\boldsymbol{\psi}_\lambda^* C = \lambda\boldsymbol{\psi}_\lambda^*$, and every $\mathbf{b} \in \mathbb{R}^N$, the coefficient identity $\boldsymbol{\psi}_\lambda^* A\mathbf{b} = \boldsymbol{\psi}_\lambda^*\mathbf{b}/(1-\lambda)$ holds. The modal gain $g_\lambda = |1-\lambda|^{-1}$ satisfies $g_\lambda \leq (1-\rho)^{-1}$, with equality exactly when $\lambda = +\rho$.

Proof of Corollary 2. Let $C \in \mathbb{R}^{N\times N}$ satisfy $\rho = \rho(C) < 1$. Every eigenvalue λ obeys $|\lambda| \leq \rho < 1$, so 1 is outside the spectrum and $A = (I - C)^{-1}$ exists. For a nonzero left eigenvector $\boldsymbol{\psi}_\lambda^* C = \lambda\boldsymbol{\psi}_\lambda^*$, multiplying $\boldsymbol{\psi}_\lambda^*(I - C) = (1-\lambda)\boldsymbol{\psi}_\lambda^*$ on the right by A yields

$$\boldsymbol{\psi}_\lambda^* A = (1-\lambda)^{-1}\boldsymbol{\psi}_\lambda^*.$$

Multiplication by any $\mathbf{b}$ proves the coefficient identity. The reverse triangle inequality gives

$$|1-\lambda| \geq 1 - |\lambda| \geq 1 - \rho > 0,$$

so the modulus g_λ of this scalar multiplier is at most $(1-\rho)^{-1}$. Both inequalities are equalities exactly when $|\lambda| = \rho$ and λ is a nonnegative real number, that is, when $\lambda = +\rho$. The argument applies to every left eigenvector, including those of defective or nonnormal matrices, and uses neither M_C nor a nonzero projection $\boldsymbol{\psi}_\lambda^*\mathbf{b}$. □

Modal coefficients and the full equilibrium error. Under the conditions of Theorem 1, applying a unit left eigenvector to the error identity (12) gives

$$\boldsymbol{\psi}_\lambda^* \mathbf{h} = \frac{\boldsymbol{\psi}_\lambda^* \mathbf{b}}{1-\lambda} + \boldsymbol{\psi}_\lambda^* A\mathbf{q}, \qquad |\boldsymbol{\psi}_\lambda^* A\mathbf{q}| \le \eta H. \tag{15}$$

Thus the coefficient multiplier in Corollary 2 acts on the linear contribution to the actual error, whose remaining term is controlled by the theorem's local nonlinear assumptions. For a complex eigenvalue, g_λ measures the gain in the modulus of the complex coefficient $\boldsymbol{\psi}_\lambda^* \mathbf{b}$. The Euclidean gain along an arbitrary real direction in the corresponding two-dimensional invariant subspace also depends on its coordinate geometry. More generally, mixtures of nonorthogonal modes and Jordan generalized modes can have gains exceeding this coefficient ceiling. Controlling the norm of their combined response requires the full resolvent or an additional bound such as M_C in Theorem 1.

Uniform conditions give the reciprocal spectral-gap order. Along a family satisfying the theorem's assumptions with $+\rho \in \operatorname{spec}(C)$, $\alpha \ge \alpha_0 > 0$, and $M_C \le M_0 < \infty$ uniformly, the bounds give

$$\frac{2\alpha_0}{3(1-\rho)} \le \frac{H}{B} \le \frac{2M_0}{1-\rho}. \tag{16}$$

Thus H/B has order $(1-\rho)^{-1}$ as $\rho \to 1^-$ within such a family. The lower bound uses the projection of the actual bias onto the left eigenvector of the eigenvalue $+\rho$. The modal gain ceiling $(1-\rho)^{-1}$ in Corollary 2 depends on ρ alone, so it can be compared directly across separately trained models, and a smaller ρ gives a lower ceiling.

Corollary 5 (Module approximation bounds)**.** *Under the assumptions for the upper bound in Theorem 1, allowing $B = 0$, suppose nonnegative bounds δ_i satisfy $\|\mathcal{M}_{\theta_i,i}(\mathbf{u}^\dagger) - \mathcal{M}_i(\mathbf{u}^\dagger)\|_2 \le \delta_i$. Set $\boldsymbol{\delta} = (\delta_1, \dots, \delta_M)^\top$, $\Delta = \|\boldsymbol{\delta}\|_2$, $e_i = \|h_i\|_2$, $\mathbf{e} = (e_1, \dots, e_M)^\top$, $A = (I - C)^{-1}$, and $c_{\mathrm{nl}} = 2L_C\|A\|_2^3$. Then*

$$H = \|\mathbf{e}\|_2 \le \Delta + \frac{M_C\rho}{1-\rho}\Delta + c_{\mathrm{nl}}\Delta^2, \tag{17}$$

and, for each field i,

$$e_i \le \delta_i + \frac{M_C\rho}{1-\rho}\Delta + c_{\mathrm{nl}}\Delta^2. \tag{18}$$

Proof. Since $\mathbf{u}^\dagger = \mathcal{M}(\mathbf{u}^\dagger)$, the module bounds imply $\|b_i\|_2 \le \delta_i$ and $B \le \Delta$. Equation (12) and $\eta \le 1/2$ give $H \le \|A\|_2\Delta + H/2$, hence

$$H \le 2\|A\|_2\Delta.$$

Substituting this estimate into the nonlinear remainder bound gives

$$\begin{aligned} \|A\mathbf{q}\|_2 &\le \frac{L_C}{2}\|A\|_2 H^2 \\ &\le 2L_C\|A\|_2^3\Delta^2 = c_{\mathrm{nl}}\Delta^2. \end{aligned} \tag{19}$$

Using $A = I + AC$ in (12) separates the direct bias, linear coupling contribution, and nonlinear remainder,

$$\mathbf{h} = \mathbf{b} + AC\mathbf{b} + A\mathbf{q}. \tag{20}$$

Taking the norm and applying (14) and (19) proves (17). For field i, the same identity gives

$$h_i = b_i + [AC\mathbf{b}]_i + [A\mathbf{q}]_i.$$

Each block norm is at most the norm of its full vector, so $\|b_i\|_2 \le \delta_i$ and the same bounds prove (18). □

The comparison uses the same module bounds and field norms. With the input fields prescribed at $\mathbf{u}^\star_{\mathrm{true}}$, module i has error at most δ_i. Corollary 5 adds nonnegative coupling and nonlinear terms to these same module bounds. The lower bound in Theorem 1 instead uses the actual bias norm B and its projection α; Δ is an upper bound on B and cannot replace it in that lower bound. The quantity H is the absolute Euclidean distance between the learned equilibrium and the exact fixed point of the discrete physical operators, and H/B is amplification relative to the bias norm. The reported system error averages field-wise relative errors against the ground truth $\mathbf{u}_{\mathrm{gt}}$ from coupled simulations and has a different normalization.

Proposition 6 (Error at a regular coupled equilibrium). *Let $\mathbf{u}^\dagger = \mathcal{M}(\mathbf{u}^\dagger)$ and $\mathbf{p} = \mathcal{M}_\theta(\mathbf{p})$ lie in a convex set $\mathcal{D}$, and suppose $\mathcal{M}_\theta$ is continuously differentiable on an open set containing $\mathcal{D}$ with L_C-Lipschitz Jacobian $C(\cdot)$ on $\mathcal{D}$. Define $\mathbf{h}$, $\mathbf{b}$, H, and $B > 0$ as in Theorem 1, and set $C = C(\mathbf{p})$. Assume $I - C$ is invertible and $\eta = L_C\|(I-C)^{-1}\|_2 H/2 \le 1/2$. For any eigenvalue λ of C and unit left eigenvector $\boldsymbol{\psi}_\lambda^* C = \lambda \boldsymbol{\psi}_\lambda^*$, set $\alpha_\lambda = |\boldsymbol{\psi}_\lambda^* \mathbf{b}|/B$. Then*

$$\frac{\alpha_\lambda}{(1+\eta)|1-\lambda|} \le \frac{H}{B} \le \frac{\|(I-C)^{-1}\|_2}{1-\eta}. \tag{21}$$

Proof. The Taylor identity (12) and comparison (13) require only the stated regularity and locality assumptions. Moreover, $\boldsymbol{\psi}_\lambda^*(I-C)^{-1} = (1-\lambda)^{-1}\boldsymbol{\psi}_\lambda^*$, so $\|(I-C)^{-1}\mathbf{b}\|_2 \ge \alpha_\lambda B/|1-\lambda|$. These inequalities prove both sides of (21). □

The modal lower bound depends on the distance $|1-\lambda|$ and bias excitation, while the upper bound uses the full resolvent norm. Regular roots can have $\rho(C) \ge 1$. That inequality alone determines neither the distance of an eigenvalue from 1 nor the resulting equilibrium error. For $0 < \rho(C) < 1$, a positive real eigenvalue $+\rho(C)$ supplies the distance $1 - \rho(C)$ used by Theorem 1. Its actual amplification still depends on the bias projection and, for the upper bound, on the growth constant M_C. If $B = 0$, the same regularity and locality conditions give $H \le \eta H$ in (12), hence $H = 0$, and the ratio H/B is then undefined.

Nonlinear modules require a local comparison. The Lipschitz condition controls the change in the full Jacobian along the segment between the two equilibria. The locality condition $\eta \le 1/2$ makes the quadratic remainder small enough to yield the two-sided comparison (13). The two equilibria are assumed to exist, and the allowed separation depends on L_C and $\|(I-C)^{-1}\|_2$. When $+\rho$ is an eigenvalue, $0 < \rho < 1$, and $L_C > 0$, the inequality $\|(I-C)^{-1}\|_2 \ge (1-\rho)^{-1}$ makes locality require $H \le (1-\rho)/L_C$. The order statement therefore concerns families that remain within this local regime. The constant $c_{\mathrm{nl}} = 2L_C\|(I-C)^{-1}\|_2^3$ in Corollary 5 also depends on the coupled model. For affine modules, the Jacobian is constant, so $L_C = 0$, $\eta = 0$, $\mathbf{q} = 0$, and $c_{\mathrm{nl}} = 0$, and the locality condition is then automatic. For nonlinear modules, both the quadratic term and the local assumptions remain part of the bound.

B.3 Solver independence

This subsection asks whether the gradient of an equilibrium loss depends on how the equilibrium was found. A root-finding solver, such as Picard or Newton iteration, is a numerical procedure for finding a state that satisfies the coupled equations. Different solvers can take different intermediate steps. If they reach the same regular equilibrium, the exact equilibrium gradient is determined by the coupled equations at that state. We first establish this statement, then explain the errors introduced by incomplete forward and backward solves.

The equilibrium equations determine the gradient on a regular solution branch. For fixed problem data $\mathbf{x}$, use the equation residual defined in Section 3.1,

$$R(\mathbf{u}, \theta; \mathbf{x}) = \mathbf{u} - \mathcal{M}_\theta(\mathbf{u}; \mathbf{x}).$$

An exact equilibrium satisfies $R(\mathbf{u}^\star, \theta; \mathbf{x}) = 0$. As in Appendix B.1, regularity means that $I - C$ is invertible, where $C = \partial_{\mathbf{u}} \mathcal{M}_\theta$ is evaluated at the equilibrium. A local solution branch is the differentiable function $\mathbf{u}^\star(\theta)$ that follows this same solution as parameters change. This qualification matters when the coupled equations have several solutions. Proposition 7 is the reverse accumulation of fixed-point derivatives (Christianson, 1994; Griewank & Walther, 2008) applied to the coupled equations.

Proposition 7 (Solver independence). *Let $\mathbf{u}^\star$ be a regular equilibrium on a local solution branch. Assume that $\mathcal{M}_\theta$ is jointly continuously differentiable in state and parameters near $\mathbf{u}^\star$ and that the state loss $\ell(\mathbf{u}, \mathbf{u}_{\mathrm{gt}})$ is continuously differentiable. Then the gradient of $\ell(\mathbf{u}^\star(\theta), \mathbf{u}_{\mathrm{gt}})$ is $P^\top(I - C^\top)^{-1}\mathbf{g}_\star$, where $\mathbf{g}_\star = \nabla_{\mathbf{u}}\ell(\mathbf{u}^\star, \mathbf{u}_{\mathrm{gt}})$ and every factor is evaluated at $(\mathbf{u}^\star, \theta)$. Any numerical procedure that returns $\mathbf{u}^\star$ therefore yields the same implicit gradient, independently of its iteration history.*

Proof. The implicit function theorem ensures that the local solution branch is differentiable. Differentiate $R(\mathbf{u}^\star(\theta), \theta; \mathbf{x}) = 0$ while keeping $\mathbf{x}$ fixed. Since $\partial_{\mathbf{u}} R = I - C$ and $\partial_\theta R = -P$, where $P = \partial_\theta \mathcal{M}_\theta$, the chain rule gives

$$(I - C)\frac{\mathrm{d}\mathbf{u}^\star}{\mathrm{d}\theta} - P = 0, \qquad \frac{\mathrm{d}\mathbf{u}^\star}{\mathrm{d}\theta} = (I - C)^{-1}P.$$

Let $\mathbf{g}_\star = \nabla_{\mathbf{u}} \ell(\mathbf{u}^\star, \mathbf{u}_{\mathrm{gt}})$ be the loss derivative with respect to the equilibrium state. This is $\mathbf{g}$ in (7). Applying the chain rule once more gives

$$\nabla_\theta \ell = P^\top (I - C^\top)^{-1} \mathbf{g}_\star.$$

Here $\nabla_\theta \ell$ denotes the total parameter gradient of the composite loss $\ell(\mathbf{u}^\star(\theta), \mathbf{u}_{\mathrm{gt}})$, as in (7). Equivalently, solve $(I - C^\top)\boldsymbol{\lambda} = \mathbf{g}_\star$ for the adjoint vector $\boldsymbol{\lambda}$ and compute $P^\top \boldsymbol{\lambda}$, as in (7). This linear solve avoids forming the full parameter sensitivity matrix $\mathrm{d}\mathbf{u}^\star/\mathrm{d}\theta$. Every quantity in the expression is evaluated at the same equilibrium, so the result contains no dependence on the solver's intermediate steps. □

Proposition 3 shows that synchronous and staggered exchanges give this same exact gradient. The adjoint vectors for the different equations need not coincide. It is the resulting parameter gradient that agrees. A solver must still reach the specified solution for the proposition to apply. For example, Picard iteration can be unstable at a regular root that another solver can reach. The proposition concerns the derivative of the equilibrium solution branch. A finite solver trajectory that reaches the root at one parameter value need not have the same derivative, because its output at nearby parameter values can depart from that branch. The relation between finite-trajectory and equilibrium gradients is established in Appendix B.4.

The local gradient error bound includes both forward and adjoint residuals. Proposition 7 identifies the exact equilibrium gradient. We now ask how accurately a numerical computation approximates that gradient when both the forward and backward solves are stopped with nonzero residuals. This situation can arise with Picard, Newton, or other solvers under finite stopping tolerances or iteration budgets. The bound below separates the contribution from an approximate equilibrium state from the contribution of an approximate adjoint, showing why gradient accuracy depends on both solves.

To measure these two contributions, first define the forward exchange residual

$$R_G(\mathbf{u}, \theta; \mathbf{x}) = \mathbf{u} - G_\theta(\mathbf{u}; \mathbf{x}),$$

which measures how much one more complete exchange would change the state. Synchronous exchange has $G_\theta = \mathcal{M}_\theta$, so $R_G = R$. Staggered exchange uses updated inputs within the round. Its residual can differ from R away from equilibrium even though the two residuals have the same zeros.

Let $\hat{\mathbf{u}}$ be an approximate state near the regular equilibrium $\mathbf{u}^\star$. Define $\hat{J}_G = \partial_{\mathbf{u}} G_\theta(\hat{\mathbf{u}}; \mathbf{x})$ and $\hat{P}_G = \partial_\theta G_\theta(\hat{\mathbf{u}}; \mathbf{x})$, both evaluated at that state. Let $\hat{\boldsymbol{\lambda}}$ be the adjoint returned by the backward solver. Its linear-system residual is

$$\mathbf{a} = (I - \hat{J}_G^\top)\hat{\boldsymbol{\lambda}} - \nabla_{\mathbf{u}} \ell(\hat{\mathbf{u}}, \mathbf{u}_{\mathrm{gt}}).$$

Thus $R_G(\hat{\mathbf{u}}, \theta; \mathbf{x})$ measures the remaining error in satisfying the forward fixed-point equation, while $\mathbf{a}$ measures the remaining error in solving the backward linear equation at $\hat{\mathbf{u}}$. For this single example, the resulting gradient estimate is $\widehat{\nabla_\theta \ell} = \hat{P}_G^\top \hat{\boldsymbol{\lambda}}$. Minibatch gradients average the per-example contributions.

Lemma 8 (Gradient error under inexact solves)**.** *Let $\mathbf{u}^\star$ be a regular equilibrium, and let G_θ be the synchronous or staggered exchange. Suppose the module maps are jointly continuously differentiable in state and parameters on an open neighborhood of $\mathbf{u}^\star$, and that J_G, P_G and $\nabla_{\mathbf{u}} \ell$ are Lipschitz in the state on a closed ball $\mathcal{U}$ centered at $\mathbf{u}^\star$. If the radius of $\mathcal{U}$ is small enough, there are constants c_R, c_f, c_b such that every $\hat{\mathbf{u}} \in \mathcal{U}$ and every candidate adjoint $\hat{\boldsymbol{\lambda}}$ satisfy $\|\hat{\mathbf{u}} - \mathbf{u}^\star\|_2 \le c_R \|R_G(\hat{\mathbf{u}}, \theta; \mathbf{x})\|_2$ and*

$$\|\widehat{\nabla_\theta \ell} - \nabla_\theta \ell(\mathbf{u}^\star(\theta), \mathbf{u}_{\mathrm{gt}})\|_2 \le c_f \|R_G(\hat{\mathbf{u}}, \theta; \mathbf{x})\|_2 + c_b \|\mathbf{a}\|_2,$$

where $\widehat{\nabla_\theta \ell} = \hat{P}_G^\top \hat{\boldsymbol{\lambda}}$ and $\mathbf{a}$ are defined above.

Proof. By Proposition 3, $I - J_G$ is invertible at $\mathbf{u}^\star$, and $P_G^\top (I - J_G^\top)^{-1}\mathbf{g}_\star$ evaluated there equals the equilibrium gradient $\nabla_\theta \ell(\mathbf{u}^\star(\theta), \mathbf{u}_{\mathrm{gt}})$ for either exchange. Because J_G is continuous, a small enough radius keeps $\|(I - J_G^\top)^{-1}\|_2$ bounded on $\mathcal{U}$. First solve the adjoint exactly at $\hat{\mathbf{u}}$ and call the resulting vector $\boldsymbol{\lambda}_{\mathrm{exact}}(\hat{\mathbf{u}})$. Subtracting its equation from the approximate adjoint equation gives

$$\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}_{\mathrm{exact}}(\hat{\mathbf{u}}) = (I - \hat{J}_G^\top)^{-1}\mathbf{a}.$$

Multiplying by $\hat{P}_G^\top$ bounds the backward-solve contribution by $c_b\|\mathbf{a}\|_2$. Separately, the inverse function theorem gives a local state-error bound $\|\hat{\mathbf{u}} - \mathbf{u}^\star\|_2 \le c_R\|R_G(\hat{\mathbf{u}}, \theta; \mathbf{x})\|_2$, where c_R is a local constant. The exactly solved adjoint gradient at $\mathbf{u} \in \mathcal{U}$ is $P_G(\mathbf{u})^\top (I - J_G(\mathbf{u})^\top)^{-1}\nabla_{\mathbf{u}}\ell(\mathbf{u}, \mathbf{u}_{\mathrm{gt}})$, a product of bounded Lipschitz factors on $\mathcal{U}$, where the inverse is Lipschitz because $X^{-1} - Y^{-1} = X^{-1}(Y - X)Y^{-1}$. This gradient therefore changes between $\hat{\mathbf{u}}$ and $\mathbf{u}^\star$ by at most a constant times $\|\hat{\mathbf{u}} - \mathbf{u}^\star\|_2 \le c_R\|R_G(\hat{\mathbf{u}}, \theta; \mathbf{x})\|_2$, which gives the forward contribution $c_f\|R_G(\hat{\mathbf{u}}, \theta; \mathbf{x})\|_2$. □

These constants depend on the derivative bounds and on the conditioning of the equations. A poorly conditioned system can amplify a small residual.

For fixed parameters and states in this neighborhood, the computed gradient therefore converges to the exact equilibrium gradient as both residuals tend to zero. Reducing the forward residual alone leaves the backward contribution uncontrolled. The bound provides a local way to assess gradient accuracy using the two equation residuals, with the conditioning-dependent constants determining how small those residuals must be. Applying this bound to the implementation in Appendix C.3 requires an additional term because its state and parameter Jacobians are evaluated at different states.

This implementation distinguishes the solver candidate $\mathbf{w}$ from the state $\mathbf{y} = G_\theta(\mathbf{w}; \mathbf{x})$ used in the loss. The candidate is detached, meaning that automatic differentiation does not trace the forward solver iterations that produced it. One exchange is then recorded for parameter differentiation. The implemented backward rule uses $J_G(\mathbf{y})$ in the adjoint equation and $P_G(\mathbf{w})$ in the parameter gradient. Let

$$\mathbf{a}_y = (I - J_G(\mathbf{y})^\top)\hat{\boldsymbol{\lambda}} - \nabla_{\mathbf{y}}\ell(\mathbf{y}, \mathbf{u}_{\mathrm{gt}}).$$

Applying Lemma 8 at $\hat{\mathbf{u}} = \mathbf{y}$ and accounting for the different parameter-derivative evaluation point gives

$$\begin{aligned}\|P_G(\mathbf{w})^\top \hat{\boldsymbol{\lambda}} - \nabla_\theta \ell(\mathbf{u}^\star(\theta), \mathbf{u}_{\mathrm{gt}})\|_2 \le c_f\|R_G(\mathbf{y}, \theta; \mathbf{x})\|_2 + c_b\|\mathbf{a}_y\|_2 \\ + L_{P_G}\|R_G(\mathbf{w}, \theta; \mathbf{x})\|_2\,\|\hat{\boldsymbol{\lambda}}\|_2.\end{aligned}$$

Here L_{P_G} is a local Lipschitz constant for P_G, and both $\mathbf{w}$ and $\mathbf{y}$ must lie in the neighborhood where the bounds hold. The extra term follows from $\mathbf{w} - \mathbf{y} = R_G(\mathbf{w}, \theta; \mathbf{x})$. It shows that the mismatch between the derivative evaluation states is itself controlled by a forward exchange residual, multiplied by the candidate adjoint norm. When the solver candidate $\mathbf{w}$ reaches $\mathbf{u}^\star$, the recorded exchange gives $\mathbf{y} = \mathbf{w}$ and this additional term vanishes. An exact adjoint solve then recovers the equilibrium gradient.

Preserving the full equilibrium gradient requires differentiating the reconstructed field. The two-field structure in Appendix B.1 allows us to solve for the structural field and reconstruct the fluid field. We now show how this reduced computation can preserve the gradient of a loss that supervises both fields. The required derivative must include both the change in the solved field and the direct effect of the parameters on the reconstructed field.

The full-state formula above treats all supervised fields as components of $\mathbf{u}$. For the two-field structure in Appendix B.1, $u_1 = F_\theta(u_2)$ and $u_2 = S_\theta(u_1)$, each module depends on the other field and has no self-dependence. To solve these equations using only the structural state u_2, define the reduced exchange $G_{\theta,\mathrm{red}} = S_\theta \circ F_\theta$. Then $u_2^\star = G_{\theta,\mathrm{red}}(u_2^\star)$, and the other field is reconstructed as $u_1^\star = F_\theta(u_2^\star)$. The loss as a function of the reduced state is

$$\tilde{\ell}(u_2, \theta) = \ell\big((F_\theta(u_2), u_2), \mathbf{u}_{\mathrm{gt}}\big).$$

Even when u_2 is held fixed, changing the parameters can change the reconstructed field. Let $\boldsymbol{\lambda}_2$ be the adjoint for this reduced equation. At a regular reduced equilibrium, the corresponding adjoint

and gradient are

$$\big(I - (\partial_{u_2} G_{\theta,\mathrm{red}})^\top\big)\boldsymbol{\lambda}_2 = \nabla_{u_2}\tilde{\ell},$$
$$\nabla_\theta \ell = (\partial_\theta G_{\theta,\mathrm{red}})^\top \boldsymbol{\lambda}_2 + \nabla_\theta \tilde{\ell}(u_2,\theta)\Big|_{u_2 \text{ fixed}}.$$

All terms are evaluated at $u_2 = u_2^\star$. The matrix I here is the $n_2 \times n_2$ identity. The final term is the partial parameter gradient of $\tilde{\ell}$, which differentiates the field reconstruction while holding u_2 fixed. Under the same differentiability assumptions, block elimination in $I - C$ shows that its invertibility is equivalent to that of $I - C_{21}C_{12} = I - \partial_{u_2} G_{\theta,\mathrm{red}}$. The reconstructed solution therefore follows the same local equilibrium branch. By the chain rule, reducing the number of solved state variables preserves the full-state equilibrium gradient when the loss is differentiated through the reconstruction, including its direct parameter dependence. Omitting that direct term can change the gradient even if the reduced equilibrium is solved exactly. At an exact equilibrium, the full-state staggered adjoint $\boldsymbol{\lambda}_G$ in (28) has first block g_1 and second block $\boldsymbol{\lambda}_2$, where $\mathbf{g} = (g_1, g_2)$, so its parameter gradient $P_{G,\mathrm{gs}}^\top \boldsymbol{\lambda}_G$ agrees with this formula and includes the direct term.

An exact forward solve does not guarantee an exact one-step phantom gradient. The residual analysis also explains the limitation of a cheaper backward rule that skips the adjoint solve. In particular, improving the forward solution alone need not make the one-step phantom gradient exact, because the backward approximation can retain a nonzero residual at the equilibrium.

The one-step phantom rule in (29) sets the candidate adjoint equal to the loss gradient instead of solving the linear system. At an exact root, it takes $\hat{\boldsymbol{\lambda}} = \mathbf{g}_\star$. Substituting this choice into the adjoint equation leaves residual $-J_G^\top \mathbf{g}_\star$, which can be nonzero even though the forward residual is zero. When $\rho(J_G) < 1$, this rule keeps only the zeroth-order term of the adjoint's Neumann series. Under synchronous exchange, the omitted part of the gradient is $P^\top[(I - C)^{-1}C]^\top \mathbf{g}_\star$, whose operator is the transpose of the coupling operator $(I - C)^{-1}C$ in (20). Under the spectral assumptions of Theorem 1, Lemma 4(i) bounds its norm by $\|P\|_2\, M_C \rho(C)/(1 - \rho(C))\, \|\mathbf{g}_\star\|_2$. Consequently, once the forward state is exact, further forward iterations cannot recover the adjoint terms omitted by this rule. The phantom gradient can still differ from the equilibrium gradient because of its backward approximation. This is the distinction between accurately locating the equilibrium and accurately differentiating its dependence on the parameters.

B.4 Unrolling and the equilibrium gradient

With $K = 1$, $d = 1$ and the iteration started at the root, the first part of Proposition 9 below reduces to the one-step phantom gradient of Geng et al. (2021). Truncated Neumann sums of this kind also arise in reverse accumulation through fixed-point iterations (Christianson, 1994; Griewank & Faure, 2002). For $\rho(C) < 1$, the implicit adjoint has the Neumann expansion

$$\boldsymbol{\lambda} = (I - C^\top)^{-1}\mathbf{g}_\star = \sum_{p=0}^{\infty} (C^\top)^p \mathbf{g}_\star, \tag{22}$$

where $\mathbf{g}_\star = \nabla_{\mathbf{u}} \ell(\mathbf{u}^\star, \mathbf{u}_{\mathrm{gt}})$. The actual finite trajectory instead follows $\mathbf{u}_{k+1} = H_{\theta,d}(\mathbf{u}_k; \mathbf{x})$, with $H_{\theta,d}(\mathbf{u}; \mathbf{x}) = (1 - d)\mathbf{u} + d\, G_\theta(\mathbf{u}; \mathbf{x})$ and fixed $d \in (0, 1]$. Write $J_{H,k} = \partial_{\mathbf{u}} H_{\theta,d}|_{\mathbf{u}_k}$, $P_{H,k} = \partial_\theta H_{\theta,d}|_{\mathbf{u}_k}$, $Z_0 = \mathrm{d}\mathbf{u}_0/\mathrm{d}\theta$, and $\mathbf{g}_K = \nabla_{\mathbf{u}} \ell(\mathbf{u}_K, \mathbf{u}_{\mathrm{gt}})$. For one example, the exact finite-depth gradient is

$$\nabla_\theta \mathcal{L}_K = Z_0^\top J_{H,0}^\top \cdots J_{H,K-1}^\top \mathbf{g}_K + \sum_{k=0}^{K-1} P_{H,k}^\top J_{H,k+1}^\top \cdots J_{H,K-1}^\top \mathbf{g}_K, \tag{23}$$

with an empty product equal to the identity. Our initial states are parameter-independent, so $Z_0 = 0$.

At a common root, write $J_H = (1 - d)I + d\, J_G$ and $P_H = d\, P_G$. The root derivative remains unchanged, yielding

$$(I - J_H)^{-1} P_H = (I - J_G)^{-1} P_G = (I - C)^{-1} P.$$

When the trajectory starts at the root, finite-depth backpropagation is exactly the truncated Neumann series of the iterated map (Proposition 9). Otherwise, the trajectory terms bounded there measure

the deviation. For the undamped synchronous exchange, $J_H = C$ and $P_H = P$. For other choices, the implicit root derivative agrees, while individual truncated terms can differ. Convergence of the corresponding Neumann series additionally requires $\rho(J_H) < 1$.

Proposition 9 (Truncation at the fixed point and trajectory error)**.** *Suppose the module maps are jointly continuously differentiable in state and parameters on an open set containing the trajectory and $\mathbf{u}^\star$, and that ℓ is continuously differentiable. If $\mathbf{u}_0 = \mathbf{u}^\star$ at the differentiated parameter value and this initial value is held constant, then the gradient in* (23) *equals $P_H^\top \sum_{p=0}^{K-1} (J_H^\top)^p \mathbf{g}_\star$. More generally, at a coupled root $\mathbf{u}^\star$ of the learned equations and with Euclidean operator norms, let $\boldsymbol{\lambda}$ solve* (7) *and suppose $\beta = \max\{\|J_H\|, \max_{1\le k\le K-1} \|J_{H,k}\|\} < 1$. Let $\Gamma_G = \max\{\|P_H\|, \max_{0\le k\le K-1} \|P_{H,k}\|\}$, $\varepsilon_J = \max_{1\le k\le K-1} \|J_{H,k} - J_H\|$, $\varepsilon_P = \max_{0\le k\le K-1} \|P_{H,k} - P_H\|$ and $\varepsilon_g = \|\mathbf{g}_K - \mathbf{g}_\star\|$. For $K = 1$, no trajectory state derivative enters, so $\beta = \|J_H\|$ and $\varepsilon_J = 0$. For $Z_0 = 0$,*

$$\begin{aligned}\|\nabla_\theta \mathcal{L}_K - P^\top \boldsymbol{\lambda}\| \le \frac{\Gamma_G \beta^K}{1-\beta}\|\mathbf{g}_\star\| + \frac{\varepsilon_P}{1-\beta}\|\mathbf{g}_\star\| \\ + \frac{\Gamma_G \varepsilon_J}{(1-\beta)^2}\|\mathbf{g}_\star\| + \frac{\Gamma_G}{1-\beta}\varepsilon_g.\end{aligned} \tag{24}$$

Here $P^\top \boldsymbol{\lambda}$ is the equilibrium gradient; it exists under this assumption even if $\rho(C) \ge 1$, because $\|J_H\| < 1$ makes $I - J_G$, and hence $I - C$ by Proposition 3, invertible. For a parameter-dependent initial state, extend the state-derivative maxima to $0 \le k \le K-1$ and add $\|Z_0\|\beta^K \|\mathbf{g}_K\|$ to the bound.

Proof. At the fixed point, every $J_{H,k} = J_H$, $P_{H,k} = P_H$ and $\mathbf{g}_K = \mathbf{g}_\star$, proving the first claim by substitution. Compare the finite trajectory sum with $P_H^\top \sum_{p=0}^{K-1} (J_H^\top)^p \mathbf{g}_\star$. Replacing terminal loss gradients and parameter derivatives contributes at most $\Gamma_G \varepsilon_g/(1-\beta)$ and $\varepsilon_P \|\mathbf{g}_\star\|/(1-\beta)$. Telescoping a product of p state derivatives bounds its difference from $(J_H^\top)^p$ by $p\beta^{p-1}\varepsilon_J$. Summing uses $\sum_{p\ge 1} p\beta^{p-1} = (1-\beta)^{-2}$. The missing tail contributes $\Gamma_G \|\mathbf{g}_\star\| \beta^K/(1-\beta)$. The infinite damped-exchange gradient equals the equilibrium gradient by the common-root identity above. The initial-state term follows directly from (23). □

Under uniform bounds Γ_G and $\beta < 1$, the truncation tail decays geometrically with increasing depth K. Because ε_J and ε_P are maxima over the whole trajectory, the bound does not by itself give convergence as $K \to \infty$. That limit is given by Corollary 10 below, a case of the convergence of derivatives of iterative processes (Gilbert, 1992; Griewank & Faure, 2002). However, because varying the training depth modifies the optimization objective and typically leads to different converged model parameters, fixed-parameter sensitivity bounds do not imply a monotonic performance improvement for models trained at greater depths.

Corollary 10 (Convergence of unrolled gradients)**.** *Fix θ and a parameter-independent initial state. Suppose the module maps are jointly continuously differentiable in state and parameters on an open set containing the trajectory and $\mathbf{u}^\star$, ℓ is continuously differentiable, $\mathbf{u}_k \to \mathbf{u}^\star$, and $\rho(J_H) < 1$. Then $\nabla_\theta \mathcal{L}_K \to P^\top \boldsymbol{\lambda}$ as $K \to \infty$, where $\boldsymbol{\lambda}$ solves* (7)*.*

Proof. Continuity of $H_{\theta,d}$ makes $\mathbf{u}^\star$ a fixed point of $H_{\theta,d}$, hence a common root of the exchanges. Since $\rho(J_H) < 1$, the matrix $I - J_H = d(I - J_G)$ is invertible, so $I - C$ is invertible by Proposition 3 and $\boldsymbol{\lambda}$ exists. Choose an induced norm with $\|J_H\| < 1$, which exists because $\rho(J_H) < 1$. The parameter derivatives $Z_k = \mathrm{d}\mathbf{u}_k/\mathrm{d}\theta$ satisfy $Z_{k+1} = J_{H,k} Z_k + P_{H,k}$. Set $Z^\star = (I - J_H)^{-1} P_H$. Subtracting its defining equation gives

$$Z_{k+1} - Z^\star = J_{H,k}(Z_k - Z^\star) + (J_{H,k} - J_H)Z^\star + (P_{H,k} - P_H).$$

Continuity gives $J_{H,k} \to J_H$ and $P_{H,k} \to P_H$. The first term therefore contracts uniformly for sufficiently large k, while the other terms vanish, so $Z_k \to Z^\star$. For a continuously differentiable loss, $\mathbf{g}_K \to \mathbf{g}_\star$, and the unrolled gradient $Z_K^\top \mathbf{g}_K$ converges to $(Z^\star)^\top \mathbf{g}_\star = P^\top \boldsymbol{\lambda}$ by the exchange identities above. □

C IMPLEMENTATION DETAILS

C.1 PHYSICAL BENCHMARK SYSTEMS

Fig. 5 illustrates the spatial domain configurations and cross-field interface topologies across the three multiphysics benchmarks. The top panels present the physical geometries, spatial partitioning, and boundary layouts, while the bottom panels map the directional exchange of predicted fields and interface variables between interacting modules. Specifically, Fig. 5a highlights the three-way cyclic coupling in neutron–thermal transport, Fig. 5b depicts the symmetric mutual feedback between reacting chemical species, and Fig. 5c details the partitioned Dirichlet–Neumann exchange across the fluid–structure interface. Below, we give the governing equations of GS and FSI, the physical description of NT, whose equations follow Gao et al. (2026), and the discrete configuration and dataset split of each system.

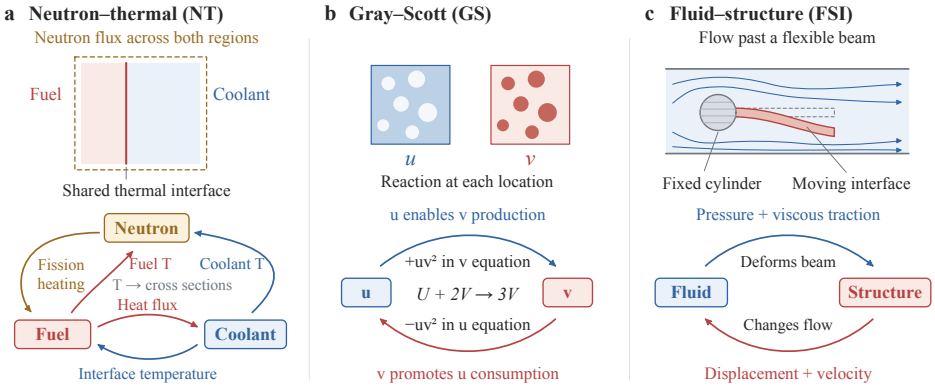

Figure 5: **Multiphysics benchmark configurations and field exchange topologies.** Physical domain schematics (top) and cross-field coupling interfaces (bottom) for (a) neutron–thermal coupling (NT), (b) Gray–Scott reaction–diffusion (GS), and (c) fluid–structure interaction (FSI).

Neutron–thermal (NT). This system couples neutron transport to fuel and coolant temperatures through temperature-dependent cross sections, where neutron fission heats the fuel, the fuel conducts heat to the coolant, and the coolant temperature modulates the macroscopic cross sections governing neutron flux. Each sample represents a sixteen-frame transient on a 64×20 spatial grid. The neutron flux spans the entire domain, the fuel temperature occupies the left 64×8 subregion, and four coolant channels (temperature, two velocity components, and pressure) occupy the right 64×12 subregion. Each module takes the time-dependent boundary flux and its coupled partner fields over the entire trajectory as inputs, predicting its own field across all sixteen frames in a single non-autoregressive forward pass. Exchanging whole sixteen-frame trajectories between modules in this way is a waveform relaxation (Lelarasmee et al., 1982).

We adopt the `NTcouple` benchmark introduced by Gao et al. (2026), comprising 4000 decoupled cases for Stage I, 400 coupled cases for Stage II, and 1000 held-out coupled cases on the same 64×20 grid. The Stage I data on NT are generated by solving each physical process separately, with partner fields taken from reference trajectories. With reference partner fields as inputs, the module-wise pre-training errors are 0.0040, 0.0027, and 0.0089 for neutron, fuel, and coolant, respectively, measuring accuracy on isolated single physics rather than in coupled inference. Consequently, on NT the pretrained modules also never encounter the coupled distribution before assembly.

Gray–Scott reaction–diffusion (GS). Two chemical species diffuse and react on a periodic square domain. The reaction kinetics are autocatalytic ($U + 2V \rightarrow 3V$), converting the substrate u into the activator v wherever v is present, such that the identical reaction term enters both equations with opposite signs and neither species can advance independently of the other. Differing diffusion rates sustain characteristic spatial patterns. Each temporal window takes three historical frames of both species and twelve future frames of the partner species as inputs, predicting twelve future frames of its own species. In Stage I module-wise pre-training, partner frames are supplied from reference

trajectories, whereas in Stage II and deployment, they are replaced by partner module predictions, so the exchange is again a waveform relaxation over the window.

The concentrations obey

$$\partial_t u = D_u \nabla^2 u - uv^2 + f_{\text{feed}}(1-u), \qquad \partial_t v = D_v \nabla^2 v + uv^2 - (f_{\text{feed}} + k_{\text{kill}})v. \tag{25}$$

The trajectories are generated using an implicit–explicit (IMEX) pseudospectral solver on $[0,1]^2$ with periodic boundary conditions on a 64×64 spatial grid, treating diffusion implicitly in Fourier space and reaction kinetics explicitly, with $D_u = 2 \times 10^{-5}$, $D_v = 10^{-5}$, feed rate $f_{\text{feed}} = 0.03$, and kill rate $k_{\text{kill}} = 0.06$ (time step 1, 200 warm-up steps, saving every 20 numerical steps to record 1000 frames per trajectory). Each trajectory starts from $u = 1$, $v = 0$ with three to eight randomly placed rectangular patches set to $u = 0.5$, $v = 0.25$, plus Gaussian noise of standard deviation 0.01, clipped to $[0, 1]$. Samples are windows of fifteen consecutive saved frames (three input and twelve target frames) taken with a stride of ten frames, giving 99 samples per trajectory. The dataset contains 25 trajectories, with 20 allocated for training and 5 held out. Stage I module-wise pre-training uses 1584 samples extracted from 16 coupled trajectories, while Stage II post-training uses 396 samples from 4 distinct trajectories. The held-out trajectories comprise 495 samples.

Fluid–structure interaction (FSI). A flexible beam is mounted behind a rigid cylinder in a fluid channel, where fluid stress deforms the solid beam, and the displaced interface in turn modifies the fluid domain boundary. The channel dimensions are 2.5×0.41, the cylinder has radius 0.05 centered at $(0.2, 0.2)$, and the attached beam has length 0.35 and thickness 0.02. Fluid viscosity μ governs the Reynolds number and modulates the strength of the two-way interface coupling. Each sample advances the coupled state over a time increment of $\Delta t = 0.05\,\text{s}$. The fluid module takes the current fluid state, interface displacement, and displacement increment as nodal features and predicts the next fluid state via a residual update, while the structure module takes, on all 104 solid nodes, the current structural displacement, the reference node coordinates, the interface pressure at the current and next time steps, and a mask marking the loaded interface nodes, and predicts the structural displacement at the next time step. The next-step interface pressure is read from the pressure that the fluid module predicts at the 53 interface nodes, an operation without trainable parameters, and neither module receives states earlier than the current time.

The fluid obeys the incompressible Navier–Stokes equations in arbitrary Lagrangian–Eulerian (ALE) form on the moving fluid domain $\Omega_f(t)$, and the beam is a St. Venant–Kirchhoff solid written on its reference domain Ω_s,

$$\begin{aligned} &\rho_f\big(\partial_t \mathbf{v} + ((\mathbf{v}-\mathbf{w})\cdot\nabla)\mathbf{v}\big) = \nabla\cdot\boldsymbol{\sigma}_f, \qquad \nabla\cdot\mathbf{v} = 0, \qquad \boldsymbol{\sigma}_f = -p\mathbf{I} + \mu(\nabla\mathbf{v} + \nabla\mathbf{v}^\top), \\ &\rho_s\,\partial_{tt}\mathbf{d} = \nabla\cdot(\mathbf{F}\mathbf{S}), \quad \mathbf{S} = \lambda_s\,\mathrm{tr}(\mathbf{E})\mathbf{I} + 2\mu_s\mathbf{E}, \quad \mathbf{E} = \tfrac{1}{2}(\mathbf{F}^\top\mathbf{F} - \mathbf{I}), \quad \mathbf{F} = \mathbf{I} + \nabla\mathbf{d}, \end{aligned} \tag{26}$$

where ∂_t in the fluid equation is taken at fixed mesh coordinates and $\mathbf{w}$ is the velocity of the fluid mesh, whose displacement extends the interface displacement into $\Omega_f(t)$. On the fluid–structure interface, the fluid velocity equals $\partial_t\mathbf{d}$ and the tractions balance, $\boldsymbol{\sigma}_f\mathbf{n} = (\det\mathbf{F})^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^\top\mathbf{n}$. The channel walls and the cylinder are no-slip, the outlet has zero pressure, and the inlet velocity is $v_x = 16\,r(t)\,U_{\max}\,y(H-y)(y-\xi_1 H/2)(y-\xi_2 H/2)/[H^4(1-\xi_1)(1-\xi_2)]$, $v_y = 0$, with channel height $H = 0.41$, $U_{\max} = 2.94175$, and ramp $r(t) = \frac{1}{2}\big(1-\cos(\pi t/2)\big)$ for $t < 2$ and $r(t) = 1$ afterwards. This profile equals $U_{\max}$ at mid-channel, and the inlet parameters $\xi_1 \le \xi_2$, both taken from $\{-4, -2, 0, 2, 4, 6\}$, set its asymmetry. In SI units, the densities are $\rho_f = \rho_s = 10^3$ and the Lamé parameters are $\mu_s = 2 \times 10^6$ and $\lambda_s = 4 \times 10^6$.

The trajectories follow the Turek–Hron benchmark configuration released by Rahman et al. (2024) on an unstructured mesh with 1317 nodes and 2474 elements. The dataset spans seven fluid viscosities $\mu \in \{0.5, 1, 2, 3, 4, 5, 10\}$, comprising $\mu \in \{0.5, 1, 5, 10\}$ from the public release and $\mu \in \{2, 3, 4\}$ generated using the same monolithic ALE solver via TurtleFSI (Bergersen et al., 2020) on FEniCS ($T = 10.005$, internal time step $\Delta t_{\text{sim}} = 0.005$, saving every two steps to yield 1001 frames per trajectory, across 21 inlet combinations (ξ_1, ξ_2) per viscosity). Both stages share an 81%/19% trajectory split (118 training and 29 held-out trajectories) stratified across viscosities, with module-wise pre-training pairs extracted from coupled simulations. The held-out trajectories comprise 725 samples (25 uniformly spaced samples per trajectory), with nine trajectories representing the strong-coupling regime ($\mu \in \{0.5, 1\}$) and twenty representing the weak-coupling regime ($\mu \ge 2$).

C.2 PHYSICAL MODULE ARCHITECTURES

Within each physical system, Frozen, the training controls, and our post-training methods share identical physical module architectures across pre-training and post-training, ensuring that these controlled comparisons isolate the training paradigm rather than model capacity (Table 3). The assembled models contain 47,508,518 parameters on NT, 4,034,578 on GS, and 2,202,451 on FSI.

For NT and GS, the physical modules employ the convolutional neural operator (CNO) architecture of Raonic et al. (2023), a lift–encode–bottleneck–decode–project structure with anti-aliased resampling layers that realizes operators between continuous function spaces. On NT, the convolution operates jointly over space and time, enabling a single forward pass to output the complete sixteen-frame trajectory. The coolant module uses twice the channel multiplier of the neutron and fuel modules because it predicts four physical channels rather than one, accounting for approximately two-thirds of the assembled parameters. On GS, the spatial domain is periodic, utilizing circular padding with identical network dimensions for both species modules.

For FSI, the two physical modules employ structurally heterogeneous architectures to accommodate differing physical representations. The fluid module is a MeshGraphNet (Pfaff et al., 2020) defined over the 1317-node unstructured mesh, comprising a feature encoder, six message-passing layers, and a decoder with a uniform hidden width of 128. The structure module is a multilayer perceptron (MLP) with three hidden layers, mapping the current displacement and interface pressure to the next-step displacement. It takes the seven input features of all 104 solid nodes as one 728-dimensional vector and outputs the 208 displacement components.

Table 3: **Architectures and hyperparameters of physical modules.** Model configurations and parameter counts across benchmarks.

	NT			GS		FSI	
Hyperparameter	neutron	fuel	coolant	u	v	fluid	structure
Backbone	CNO	CNO	CNO	CNO	CNO	MeshGraphNet	MLP
Activation	LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU	ReLU	ReLU
N_layers	4	4	4	3	3	—	—
N_res	1	1	1	1	1	—	—
N_res_neck	6	6	6	6	6	—	—
channel_multiplier	16	16	32	16	16	—	—
in_size	64	64	64	64	64	—	—
Hidden width	—	—	—	—	—	128	1024, 512, 256
Message-passing layers	—	—	—	—	—	6	—
node_in / edge_in	—	—	—	—	—	8 / 3	—
Input / output size	—	—	—	—	—	—	728 / 208
Parameters	7.94 M	7.94 M	31.63 M	2.02 M	2.02 M	2,202,451 together	

C.3 TRAINING CONFIGURATIONS AND EXPERIMENTAL PROTOCOLS

Table 4 summarizes and compares the evaluated composition frameworks, training controls, and our multiphysics post-training implementations across their initializations, partner-field inputs, supervised target points, and gradient backward paths. In contrast to a deep equilibrium layer trained end to end from scratch, the fixed-point map in post-training is assembled from separately pretrained physical modules along their physical interfaces, so its equilibrium is a coupled state of physical fields and post-training starts from modules that have already learned isolated single physics.

Optimization and training schedules. Stage I module-wise pre-training trains each physical module to predict its own field from reference partner fields. Post-training methods and MW-CT initialize from these shared weights, while EquiPT without pre-training trains directly from random initialization. Stage II executes 2000, 5000, and 10 000 parameter updates on NT, GS, and FSI, respectively. On NT, Unrolled-K and MW-CT use a peak learning rate of 10^{-4}, while EquiPT uses 2×10^{-5} to maintain numerical stability during the linear adjoint solve. Optimization hyperparameters for all physical systems are summarized in Table 5. The loss objective on NT sums field-wise MSEs, and GS evaluates the mean over both chemical species. For FSI, all trained methods normalize loss

Table 4: **Comparison of multiphysics composition frameworks.** Summary of model configurations, training objectives, and backward update rules across evaluated methods.

Method	Initialization	Partner fields in Stage II	Supervised output	Backward rule
Existing practice				
Frozen (Martins & Hwang, 2013; Dubreuil et al., 2016; Arcones et al., 2022; Li et al., 2022)	Pretrained modules	—	—	—
GenCP (Gao et al., 2026)	Pretrained flow matching models	—	—	—
M2PDE (Zhang et al., 2024)	Pretrained diffusion models	—	—	—
Training controls				
MW-CT	Pretrained modules	Reference fields	Each module's output	Module-wise backprop.
EquiPT w/o pre-training	Without pre-training	Predicted fields	Numerical equilibrium	Implicit adjoint
Multiphysics post-training (ours)				
Unrolled-1	Pretrained modules	Predicted fields	State after one exchange	Through one exchange
Unrolled-K	Pretrained modules	Predicted fields	State after K exchanges	Through K exchanges
EquiPT	Pretrained modules	Predicted fields	Numerical equilibrium	Implicit adjoint

terms within each sample before averaging, via

$$\ell(\mathbf{u}, \mathbf{u}_{\mathrm{gt}}) = \sum_i \frac{\mathrm{MSE}(u_i, u_{i,\mathrm{gt}})}{\max\{\mathrm{MSE}(0, u_{i,\mathrm{gt}}), 10^{-12}\}}, \tag{27}$$

preventing high-amplitude frames from dominating the gradient signal. On GS, unrolled models employ two-step gradient accumulation to match EquiPT in batch size and sample exposure. For all forward evaluations, initial states depend solely on problem inputs rather than model parameters, using a constant spatial field on NT, the replicated final input frame on GS, and the initial interface state on FSI. Optimization employs the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with cosine learning-rate decay, decaying to 1% of the peak rate for Stage II post-training and to zero for Stage I and MW-CT. All configurations use gradient clipping at norm 1.0 without weight decay. For each training run, we evaluate the checkpoint that performs best on the validation set during training.

Damping conventions. For Unrolled-K with $K > 1$, we fix the system-specific damping factors in Table 5 across training depths and retain them during evaluation. Unrolled-1 uses $d = 1$ to apply one complete exchange. Equilibrium training and Anderson evaluation solve the undamped exchange map. For additional Picard evaluations of Frozen and equilibrium-trained models, we use $d = 1$ on NT, $d = 0.5$ on GS, and $d = 0.1$ on FSI. Unrolled models retain their training damping in the deployment-depth grid and the NT depth-response curves. The NT memory probe uses undamped exchanges throughout. Subsequent experiments follow these conventions unless they explicitly vary the setting.

Forward solving and gradient calculation. The overall post-training workflow is summarized in Algorithm 1, which executes two distinct gradient pathways at finite forward residuals. Unrolled-K records all K field exchanges and evaluates the exact gradient of its finite-depth objective via reverse-mode automatic differentiation. In contrast, EquiPT solves for a numerical equilibrium using Anderson acceleration (Geng & Kolter, 2023) (with a 50-iteration cap and absolute-residual tolerance 10^{-5}), returning a detached candidate $\mathbf{w}$. The subsequent differentiable exchange gives $\hat{\mathbf{u}} = \mathbf{y} = G_\theta(\mathbf{w}; \mathbf{x})$, which enters the loss objective $\ell(\mathbf{y}, \mathbf{u}_{\mathrm{gt}})$. Parameters are updated via the

Table 5: **Optimization hyperparameters and training configurations.** Training configurations for module-wise pre-training and multiphysics post-training across benchmarks (on NT Stage II, EquiPT adopts a learning rate of 2×10^{-5} for adjoint stability, while Unrolled-K uses 10^{-4}).

System	Steps	Batch	Peak lr	Forward solve	d for unrolling ($K > 1$)
			Stage I (module-wise pre-training)		
NT	10,000	16	3×10^{-4}	—	—
GS	10,000	8	10^{-3}	—	—
FSI	80,000	8	fluid 10^{-3}, structure 10^{-4}	—	—
			Stage II (multiphysics post-training)		
NT	2,000	2	10^{-4} (2×10^{-5})	Anderson	0.1
GS	5,000	4	10^{-4}	Anderson	0.5
FSI	10,000	2	2×10^{-5}	Anderson	0.1

Algorithm 1 Multiphysics post-training

Input: Pretrained parameters θ; paired input conditions and ground truth $\mathcal{T} = \{(\mathbf{x}, \mathbf{u}_{\mathrm{gt}})\}$.
Given: Field-exchange map G_θ; trainable subset Θ; solver tolerance target ε_f and iteration cap N_f (EquiPT), or depth K and damping d (Unrolled-K).

for batch $\mathcal{B}$ in $\mathcal{T}$ **do**
  Initialize the parameter-independent state $\mathbf{u}_0$
  if EquiPT **then**
    $\mathbf{w} \leftarrow \mathrm{EquilibriumSolve}(G_\theta(\cdot;\mathbf{x}), \mathbf{u}_0; \varepsilon_f, N_f)$, detached
    $\mathbf{y} \leftarrow G_\theta(\mathbf{w};\mathbf{x})$
    $\mathcal{L}_\mathcal{B} \leftarrow \mathrm{mean}_\mathcal{B}\, \ell(\mathbf{y}, \mathbf{u}_{\mathrm{gt}})$
    $\boldsymbol{\gamma} \leftarrow \mathrm{ImplicitGrad}(\mathcal{L}_\mathcal{B}, G_\theta, \mathbf{w}, \mathbf{y})$ by (28)
  else
    for $k = 0, \ldots, K-1$ **do**
      $\mathbf{u}_{k+1} \leftarrow (1-d)\mathbf{u}_k + d\, G_\theta(\mathbf{u}_k;\mathbf{x})$
    $\mathcal{L}_\mathcal{B} \leftarrow \mathrm{mean}_\mathcal{B}\, \ell(\mathbf{u}_K, \mathbf{u}_{\mathrm{gt}})$
    $\boldsymbol{\gamma} \leftarrow \mathrm{Backprop}(\mathcal{L}_\mathcal{B}, \mathbf{u}_0 \to \mathbf{u}_K)$
  $\theta_\Theta \leftarrow \mathrm{OptimizerStep}(\theta_\Theta, [\boldsymbol{\gamma}]_\Theta)$

implicit adjoint without storing forward exchange trajectories:

$$\big(I - J_G(\mathbf{y})^\top\big)\boldsymbol{\lambda}_G = \mathbf{g}(\mathbf{y}), \qquad \widehat{\nabla_\theta \mathcal{L}} = P_G(\mathbf{w})^\top \boldsymbol{\lambda}_G, \tag{28}$$

where $J_G(\mathbf{y}) = \partial_\mathbf{u} G_\theta(\mathbf{y};\mathbf{x})$ and $P_G(\mathbf{w}) = \partial_\theta G_\theta(\mathbf{w};\mathbf{x})$. At an exact coupled equilibrium $\mathbf{w} = \mathbf{y}$, this recovers the equilibrium gradient in (7) (Proposition 3). Under staggered exchange, $\boldsymbol{\lambda}_G = (I - C_{\mathrm{low}}^\top)\boldsymbol{\lambda}$ while $P_G^\top \boldsymbol{\lambda}_G = P^\top \boldsymbol{\lambda}$. The effect of inexact forward and adjoint solves is discussed in Appendix B.3. At evaluation, root-solving methods operate in relative-residual mode, retaining the residual-minimizing candidate within the iteration budget.

C.4 Deployment settings and spectral estimation

Deployment settings. Each model is evaluated under its optimal deployment setting. Unrolled models deliver the iterate supervised during training (K damped exchanges), whereas equilibrium-trained models deliver the numerical equilibrium from root solving. EquiPT, with and without pre-training, uses Anderson-50 across all three benchmarks. Frozen and MW-CT use Picard-200 on NT and FSI, alongside Anderson-50 on GS. Fig. 4 uses Anderson-50 for both compared methods to hold the numerical solver fixed. Fixed-depth evaluation returns the prescribed final iterate, whereas root solvers return the candidate that minimizes the residual within their iteration budget. Detailed deployment points, residuals, and inference latencies for Frozen, the training controls, and the post-training methods are reported in Table 9 (Appendix D.6).

Metric aggregation. Deployment errors are measured on the test sets. For field i of test sample n with T predicted frames, the field error is $e_{i,n} = \frac{1}{T}\sum_{t=1}^{T} \|\hat{u}_{i,n,t} - u_{i,n,t}\|_2 / \|u_{i,n,t}\|_2$, where the norm runs over all spatial points and channels of the field and T is 16 on NT, 12 on GS, and 1 on FSI. Errors are computed in the units of each dataset, namely fixed-bound min–max scaling to $[-1, 1]$ on NT with a logarithmic transform of the neutron flux, raw concentrations on GS, and per-channel standardization with training statistics on FSI. The system error is the unweighted mean of these field errors. NT weights test samples equally. GS and FSI average within each trajectory first and then weight trajectories equally so that longer trajectories do not dominate. For methods with repeated training realizations, Table 1 reports means and relative sample standard deviations across the three runs (Appendix D.1). GenCP and M2PDE report single-run errors. Field, depth, solver, and other diagnostic analyses use representative trained model checkpoints.

Estimating the local coupling diagnostic. The local coupling diagnostic is evaluated from the exchange Jacobian $J_G = \partial_{\mathbf{u}} G_\theta$ at the numerical states returned by each diagnostic probe. For every method, including Unrolled-15, the probe solves that model's learned equations with Anderson acceleration (at most 50 iterations, tolerance 10^{-5}) and evaluates the diagnostic at the returned state. For synchronous exchanges on NT and GS, $J_G(\hat{\mathbf{u}}) = C(\hat{\mathbf{u}})$ holds identically at each evaluation state $\hat{\mathbf{u}}$. All three probes differentiate the learned exchange with reverse-mode automatic differentiation. On NT and GS, where the states are too large to form J_G explicitly, we apply it through vector–Jacobian products, which act with $J_G^\top$ and leave the spectrum unchanged. On NT, we estimate the spectral radius via Gelfand's formula $\rho(J_G) = \lim_{k\to\infty} \|J_G^k\|^{1/k}$, taking the geometric mean of the per-iteration norm growth over iterations 21–60 of a 60-step power iteration on 30 uniformly sampled test samples. On GS, where dominant eigenvalues frequently occur in complex conjugate pairs and power iteration does not settle, we employ the Arnoldi iteration (Saad, 2011) with a Krylov subspace dimension of 160 on 50 uniformly sampled test samples. On FSI, the staggered exchange Jacobian is block upper triangular, so its nonzero eigenvalues are those of the 208×208 structural block $C_{21}C_{12}$. We form this block from the two interface Jacobians and compute its spectrum exactly by dense eigendecomposition. Fig. 6 uses the test samples from the trajectories with $\mu \leq 1$. We report $s_\rho = \sqrt{\rho(J_{G,\mathrm{gs}})}$, which equals $\rho(C)$ at the evaluation state with its fluid field recomputed from the structural field (Appendix B.1). Corollary 2 applies to each such matrix whenever its radius is below one, giving the ceiling $(1-\rho)^{-1}$ on left-eigenmode coefficient gains. An estimated radius $\hat{\rho} < 1$ yields an estimated ceiling $(1-\hat{\rho})^{-1}$. Spectral diagnostics are compared within each benchmark. Within-system distributions and the construction of the FSI viscosity curves using these estimators are analyzed in Appendix D.7.

D Additional experimental results

D.1 Repeated training runs

To assess empirical stability across random initializations, Frozen, the training controls, and our post-training methods are evaluated across three training realizations on NT, GS, and FSI with fixed data splits, network architectures, and training hyperparameters. All three benchmarks use seeds 1, 42, and 123. Within each realization, all methods initialized from pretrained modules use the same Stage I weights. Frozen has no Stage II and directly evaluates those weights. For these methods, Table 1 reports the sample mean and relative sample standard deviation ($100s/\bar{e}$) of system and per-field errors across the three runs, aggregating individual run errors according to the evaluation protocol in Appendix C.4. GenCP and M2PDE instead report one realization per benchmark, and their FSI adaptations use seed 42. Across all three benchmark systems, the repeated runs consistently confirm the substantial system error reductions achieved by multiphysics post-training.

D.2 Approximating the equilibrium adjoint

Both the one-step phantom rule (Geng et al., 2021) and the full implicit adjoint evaluate the identical system-level objective at a numerical equilibrium. At an exact coupled root, the one-step phantom rule approximates the inverse $(I - J_G^\top)^{-1}$ with the identity matrix I, which, when $\rho(J_G) < 1$, corresponds to truncating its Neumann series to the zeroth-order term:

$$\widehat{\nabla_\theta \mathcal{L}} = P_G^\top \nabla_{\mathbf{u}} \ell(\mathbf{u}^\star, \mathbf{u}_{\mathrm{gt}}), \qquad \text{with candidate adjoint } \hat{\boldsymbol{\lambda}} = \nabla_{\mathbf{u}} \ell(\mathbf{u}^\star, \mathbf{u}_{\mathrm{gt}}), \tag{29}$$

where $P_G = \partial_\theta G_\theta(\mathbf{u}^\star; \mathbf{x})$. Under synchronous exchange, the resulting gradient vector $P_G^\top \mathbf{g}_\star$ coincides identically with $P^\top \mathbf{g}_\star$, whereas under staggered exchange it incorporates intermediate within-step dependencies. The phantom update represents an approximate equilibrium gradient within the post-training paradigm because it supervises the equilibrium state delivered by the forward solver. Under synchronous exchange, P is block diagonal, so both the phantom rule and Unrolled-1 give each module the gradient of its own output. They differ in where the modules are evaluated and which state is supervised. The phantom rule evaluates the modules at the numerical equilibrium and supervises that state, whereas Unrolled-1 evaluates them at the initial state and supervises the state after one exchange.

Table 6 compares the prediction accuracy of equilibrium post-training under the one-step phantom rule versus the full implicit adjoint. To ensure a controlled comparison, paired training runs are conducted using Anderson-50 with matched forward search budgets, dataset partitions, pretrained initializations, trainable parameter subsets, optimizer hyperparameters, and update budgets (2×10^{-5} learning rate on NT for both rules). Both backward rules deliver substantial accuracy improvements over the Frozen baseline across all three benchmarks. In paired runs, the full adjoint achieves lower system error on NT (0.0077 vs. 0.0084) and GS (0.0161 vs. 0.0196), while the one-step phantom rule yields slightly lower error on FSI (0.0424 vs. 0.0430). The full adjoint thus improves accuracy over the phantom rule on NT and GS, while the computationally lighter phantom approximation already retains the primary accuracy gains of EquiPT. The full adjoint incurs additional computational cost. For example, on NT with batch size two, forward/backward peak memory allocation is 3.20 GiB for phantom versus 6.40 GiB for the adjoint, and matched 2000-step training runs require 4720 and 6609 seconds, respectively. Neither backward rule stores intermediate forward exchange trajectories, preserving $\mathcal{O}(1)$ memory complexity with respect to the forward solver iteration count.

Table 6: **Comparison of equilibrium backward differentiation rules.** Coupled prediction errors across benchmarks for equilibrium post-training with the one-step phantom update versus the full implicit adjoint.

Backward rule	NT	GS	FSI
One-step phantom	0.0084	0.0196	0.0424
Full adjoint	0.0077	0.0161	0.0430

D.3 Comparison with generative composition frameworks

The NT columns of Table 1 in Section 5.2 benchmark EquiPT against published generative composition frameworks on the NT dataset using the public codebase and data released by Gao et al. (2026), which originally reported performance for GenCP, M2PDE (Zhang et al., 2024), and an uncoupled neural surrogate. We evaluate the authors' released checkpoints on the test set with their inference scripts, without modifying their model configurations. Under their original metric convention (whole-trajectory relative L^2 error evaluated in dimensional physical units), our re-evaluation closely matches their published values.

Because Gao et al. (2026) evaluates trajectory-level dimensional errors whereas our benchmark computes the per-frame relative L^2 error in normalized units with coolant velocity channels pooled, the published numbers cannot be compared directly against our benchmark values. The NT columns of Table 1 therefore evaluate all methods under our standardized per-frame normalized protocol. Furthermore, their uncoupled surrogate and our Frozen baseline exhibit closely matched error magnitudes (0.2280 vs. 0.2056), so our frozen modular assembly is comparable to the uncoupled baseline reported in the existing literature.

For GS, neither work provides a released configuration. We adapt the authors' released code to GS and otherwise follow the original implementations. We configure the CNO backbone in both baselines to match EquiPT. Each generator is trained from random initialization for 10,000 updates with batch size eight, matching the GS pre-training budget in Table 5, using a peak learning rate of 10^{-4} with cosine decay and seed 42. Both baselines use the same data split and evaluation protocol as the GS benchmark in this paper.

For FSI, we adapt the released GenCP and M2PDE implementations to the graph-structured fluid field and vector-valued structural state. Their physical backbones follow the fluid and structure architectures used by EquiPT, and cross-field inputs use the same interface quantities. GenCP retains conditional flow matching and sequential Euler composition, and M2PDE retains its diffusion objective and two-round composition. Each generator is trained from random initialization for 80,000 updates with batch size eight and seed 42. Table 1 reports coupled predictions with one generative sample per test sample on the same FSI test set.

D.4 Training depth and deployment depth

Table 7 evaluates four training depths ($K \in \{3, 5, 8, 15\}$) across all three benchmarks to characterize the response of unrolled models across physical systems and training horizons. Across all twelve evaluated training configurations, unrolled models achieve their minimum prediction error at their training depth. Increasing training depth improves on-depth accuracy on NT (decreasing from 0.0161 at $K = 3$ to 0.0096 at $K = 15$) and FSI ($\mu \leq 1$, improving from 0.1385 at $K = 3$ to 0.1062 at $K = 15$). On GS, training depth has little effect on on-depth accuracy, with errors between 0.0183 and 0.0235 across $K \in \{3, 5, 8, 15\}$. Unrolled-1 and Unrolled-15 likewise reach 0.0201 and 0.0196 in Table 1, and this spread is comparable to the variation of GS Unrolled-15 across training runs (relative sample standard deviation 15.71%). Across all systems, unrolled models require matching deployment depth to training depth.

The two rightmost columns of Table 7 illustrate the failure modes encountered when deploying unrolled models beyond their supervised training depth into iterative root solvers. On GS, extending Picard iteration to 200 steps (P200) causes large error growth for shallow unrolled models, with the Unrolled-3 error escalating from 0.0183 to 1.1954 (65 times its supervised minimum). Under Anderson-50, the Unrolled-3 error is 0.0444, still worse than the Frozen baseline and 2.4 times that minimum. On NT, deploying the Unrolled-3 model under deep solvers similarly causes error to escalate to approximately 4.05 under both Picard-200 and Anderson-50 (252 times its on-depth error). These results illustrate the difference between the finite-depth and equilibrium objectives. Finite unrolling supervises the K-th iterate, whose prediction accuracy can differ from that of the learned equilibrium even for a contractive iteration.

Table 7: **Unrolled model prediction errors across training and deployment depths.** System error of Unrolled-K across training and deployment depths. Columns Pn and A50 denote n damped Picard exchanges and Anderson-50 root solving, respectively. Each row is one trained model (Appendix C.4). FSI rows use the strongly coupled test trajectories ($\mu \leq 1$).

Training K	P1	P3	P5	P8	P15	P50	P200	A50
				NT				
3	1.1990	**0.0161**	0.6547	1.2943	2.1730	3.7772	4.0504	4.0505
5	1.5245	0.5935	**0.0138**	0.5481	1.2114	2.5417	3.4248	3.4302
8	1.7903	1.0539	0.5479	**0.0114**	0.6851	2.2522	7.2106	8.9917
15	1.9990	1.4343	1.0235	0.5851	**0.0096**	0.5907	0.6473	0.6477
				GS				
3	0.0413	**0.0183**	0.0650	0.1926	0.6243	1.1698	1.1954	0.0444
5	0.0469	0.0350	**0.0235**	0.0503	0.1505	0.8612	0.9328	0.0456
8	0.0475	0.0414	0.0323	**0.0225**	0.0537	0.2765	0.7531	0.0550
15	0.0478	0.0430	0.0382	0.0310	**0.0212**	0.0474	0.0549	0.0527
				FSI				
3	0.4877	**0.1385**	0.3873	0.8454	1.6787	2.5383	2.5593	2.5525
5	0.5783	0.2843	**0.1199**	0.3361	0.8703	1.6188	1.6720	1.6489
8	0.6326	0.4098	0.2463	**0.1120**	0.3828	0.9359	0.9967	0.9931
15	0.6723	0.5103	0.3851	0.2490	**0.1062**	0.3602	0.3704	0.3781

D.5 SOLVER INDEPENDENCE AND PREDICTION ACCURACY

To evaluate solver independence, Table 8 assesses predictions on NT across four numerical root-finding algorithms, damped Picard iteration (fixed relaxation), Aitken dynamic relaxation, Anderson acceleration (multi-secant extrapolation), and an inexact Newton–Krylov method solving $R(\mathbf{u}) = 0$. All solvers operate on identical physical module equations with fixed weights. Picard, Aitken, and Anderson receive an iteration budget of 50 steps at batch size 100, while Newton–Krylov uses 20 outer steps with inner GMRES iterations at batch size six.

Across all four numerical solvers, EquiPT attains a system prediction error of 0.0077. For the Frozen baseline, system errors remain around 0.1897 to 0.1990. In contrast, the Unrolled-15 column illustrates the consequences of deploying an unrolled model with iterative root solvers rather than truncating at its supervised 15-step horizon. Across all three fixed-point solvers, Unrolled-15 yields the same degraded prediction error of 0.6477 (agreeing across solvers within 0.002%), which is about 68 times the error it attains at its training depth (0.0096). The similarly high errors across these three solver settings show that their low-residual outputs remain substantially less accurate than the iterate supervised during training.

Near an isolated root where $I - C$ is invertible, the residual-to-state bound of Lemma 8, applied with $G_\theta = \mathcal{M}_\theta$ so that $R_G = R$, bounds the distance from an approximate iterate $\hat{\mathbf{u}}$ to the exact equilibrium $\mathbf{u}^\star$, for $\hat{\mathbf{u}}$ sufficiently close to $\mathbf{u}^\star$, by $\|\hat{\mathbf{u}} - \mathbf{u}^\star\| \le c_R \|R(\hat{\mathbf{u}})\|$, where $R(\hat{\mathbf{u}}) = \hat{\mathbf{u}} - \mathcal{M}_\theta(\hat{\mathbf{u}})$ is the equation residual and c_R is a local conditioning constant. By the triangle inequality, the physical prediction error against the ground truth $\mathbf{u}_{\mathrm{gt}}$ satisfies $\|\hat{\mathbf{u}} - \mathbf{u}_{\mathrm{gt}}\| \le \|\mathbf{u}^\star - \mathbf{u}_{\mathrm{gt}}\| + c_R \|R(\hat{\mathbf{u}})\|$. This relation distinguishes residual convergence (internal self-consistency) from physical prediction error (fidelity to the ground truth). Driving the equation residual to zero merely guarantees convergence to the learned fixed point $\mathbf{u}^\star$, but cannot reduce the intrinsic modeling error $\|\mathbf{u}^\star - \mathbf{u}_{\mathrm{gt}}\|$. For instance, Newton–Krylov drives the relative residual to 1.31×10^{-4} for EquiPT and 2.20×10^{-4} for Frozen, but cannot bridge the order-of-magnitude accuracy gap between them. Finite unrolling optimizes the intermediate iterate $\mathbf{u}_K$, which depends on the exchange scheme and the number of exchanges, whereas EquiPT optimizes the fixed point $\mathbf{u}^\star$ directly, so the state it is trained on is the state that a converged root solver returns.

Table 8: **Evaluation of solver independence across numerical schemes on NT.** System prediction errors paired with relative exchange residuals (error/residual) across root solvers at the fixed weights of one trained model per method, with the same parameter-independent initial state for all solvers. Proposition 7 applies when the solvers reach the same regular solution branch. Newton–Krylov was not evaluated for Unrolled-15.

Solver	Frozen	EquiPT	EquiPT, per field	Unrolled-15
Picard-50	0.1898 / 7.17e-4	0.0077 / 3.17e-4	0.0039 / 0.0086 / 0.0108	0.6477 / 1.60e-4
Aitken-50	0.1990 / 1.85e-3	0.0077 / 2.31e-4	0.0039 / 0.0085 / 0.0108	0.6477 / 1.78e-4
Anderson-50	0.1902 / 6.65e-4	0.0077 / 1.38e-4	0.0039 / 0.0086 / 0.0108	0.6477 / 1.14e-4
Newton–Krylov	0.1897 / 2.20e-4	0.0077 / 1.31e-4	0.0039 / 0.0086 / 0.0108	—

D.6 COMPUTATIONAL COST PROFILES

Training memory consumption is primarily determined by the differentiation framework, whereas inference execution time depends on the chosen deployment solver and its iteration budget. We characterize both dimensions for the methods evaluated in Section 5.3.

Training memory. Peak memory for finite unrolling scales linearly with unrolling depth K, measuring an empirical increment of approximately 3.0 GiB per exchange step on NT with batch size two (measuring 3.19, 15.56, 24.21, and 45.22 GiB at $K = 1, 5, 8, 15$, respectively, as shown in Fig. 3c). Linear extrapolation indicates that an 80 GiB accelerator is exhausted near $K \approx 26$, causing unrolled training at $K = 27$ and $K = 50$ to fail with out-of-memory errors. In contrast, EquiPT maintains an $\mathcal{O}(1)$ memory profile of 6.40 GiB under the full implicit adjoint and 3.20 GiB under the one-step phantom rule, completely bypassing the storage of intermediate forward exchange trajectories.

Inference time. Complete inference execution times are measured on an NVIDIA A100-SXM4-80GB GPU at standard test batch sizes (100 for NT, eight for GS, and four for FSI) across three repeated timing runs over a standardized sequence of batches following an initial warm-up, recording iterative solver execution via CUDA synchronization without data loading overhead. One sample covers a sixteen-frame transient on NT, twelve future frames on GS, and one 0.05 s step on FSI. The per-sample times in Table 9 are the per-batch means divided by these batch sizes.

Table 9 reports the deployment points, prediction errors, relative exchange residuals, and mean inference latencies for Frozen, the training controls, and the post-training methods. Finite unrolling delivers low inference latency under fixed iteration budgets (e.g., 1.311 s for Unrolled-1 and 19.80 s for Unrolled-15 on NT). For equilibrium-trained models, accelerated solvers reduce runtime relative to standard Picard iteration. On NT, Anderson-50 requires 65.98 s per batch, an approximately $4\times$ speedup over Picard-200 (263.7 s). On FSI, Anderson-50 satisfies the 10^{-5} relative residual threshold in only 0.0437 s per batch (median residual 4.5×10^{-7}), achieving a $29\times$ wall-clock speedup compared to Picard-200 (1.278 s) while delivering identical system prediction accuracy on the same checkpoint. EquiPT modules can therefore be deployed with accelerated numerical solvers to reduce runtime.

Table 9: **Inference performance across deployment settings.** Coupled prediction errors, relative exchange residuals, and wall-clock inference times (seconds per batch and per sample on an NVIDIA A100 GPU) across benchmarks. Errors are the means reported in Table 1. Relative exchange residuals (means over test samples) and inference times are measured on a representative checkpoint of each method, as in the depth and solver analyses (Table 7 and Table 8).

System	Method	Deployment point	System error	Relative exchange residual	Time (s/batch)	Time (s/sample)
NT	Frozen	Picard-200	0.2056	5.51e-4	263.8	2.638
	MW-CT	Picard-200	0.0123	6.39e-3	259.6	2.596
	EquiPT w/o pre-training	Anderson-50	0.0787	3.77e-2	65.75	0.6575
	Unrolled-1	1 exchange	0.0704	1.50	1.311	0.0131
	Unrolled-15	15 exchanges	0.0094	5.54e-1	19.80	0.1980
	EquiPT	Anderson-50	0.0081	1.38e-4	65.98	0.6598
GS	Frozen	Anderson-50	0.0313	4.30e-4	4.641	0.5801
	MW-CT	Anderson-50	0.0350	3.87e-4	4.634	0.5793
	EquiPT w/o pre-training	Anderson-50	0.0353	4.95e-5	4.630	0.5788
	Unrolled-1	1 exchange	0.0201	3.35e-2	0.0920	0.0115
	Unrolled-15	15 exchanges	0.0196	3.68e-3	1.378	0.1723
	EquiPT	Anderson-50	0.0161	2.60e-4	4.636	0.5795
FSI	Frozen	Picard-200	0.1728	1.98e-4	1.319	0.3297
	MW-CT	Picard-200	0.1650	5.41e-4	1.326	0.3316
	EquiPT w/o pre-training	Anderson-50	0.1051	7.08e-6	0.0638	0.0160
	Unrolled-1	1 exchange	0.0542	3.76e-2	0.0134	0.0033
	Unrolled-15	15 exchanges	0.0417	4.83e-2	0.1025	0.0256
	EquiPT	Anderson-50	0.0428	1.91e-6	0.0437	0.0109

D.7 Coupling sensitivity and spectral diagnostics

Fig. 6 presents the empirical distributions of estimated $\rho(C(\hat{\mathbf{u}}))$ on NT and GS and s_ρ on FSI in the strongly coupled regime across Frozen assembly, Unrolled-15, and EquiPT. Post-training shifts these distributions toward smaller values on all three systems. These diagnostics evaluate the coupling Jacobian at the corresponding numerical states. Using the expression in Corollary 2, we convert each estimated radius $\hat{\rho} < 1$ to an estimated modal gain ceiling $(1 - \hat{\rho})^{-1}$, and smaller estimates give lower ceilings. EquiPT's spectral shifts accompany lower system errors. Some probes give estimates at or above one. At a regular equilibrium, Proposition 6 relates amplification to how close the eigenvalues of C lie to one. Unrolled-15 also lowers these estimates, and its error is lowest at the 15-exchange output it is trained on. The error grows as exchanges continue toward the equilibrium, which its finite-depth objective does not supervise (Section 5.3 and Appendix D.4).

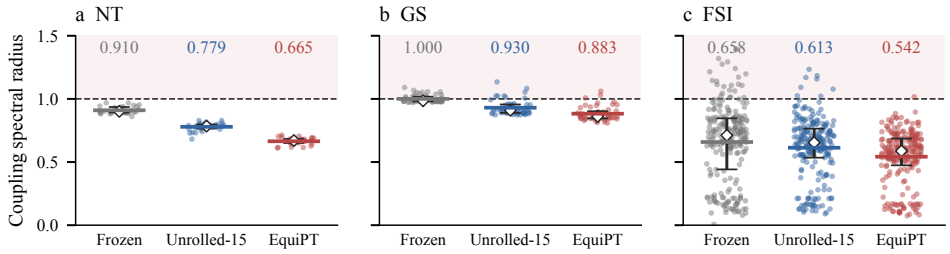

Figure 6: **Local coupling spectral diagnostics across benchmarks.** Distributions of estimated $\rho(C(\hat{\mathbf{u}}))$ on NT and GS and s_ρ on FSI in the strongly coupled regime ($\mu \leq 1$) for Frozen assembly, Unrolled-15, and EquiPT. Numbers above the distributions give their means. Values are compared within each system. The dashed line marks $\rho(C) = 1$.

Fig. 4 analyzes model performance and coupling diagnostics across the test samples from the held-out trajectories spanning fluid viscosities $\mu \in \{0.5, 1, 2, 3, 4, 5, 10\}$. Both Frozen and EquiPT are evaluated under Anderson-50. To prevent trajectories with varying sample counts from biasing the aggregate, evaluation metrics are first averaged across samples within each trajectory and subsequently averaged with equal trajectory weighting at each viscosity tier. Shaded regions highlight the strongly coupled regime ($\mu \leq 1$). Dashed trend lines in Fig. 4a and Fig. 4b depict ordinary least squares fits across the seven viscosity tiers in log–log coordinates, while panel c traces the local coupling diagnostic s_ρ of the frozen modular assembly, defined in Appendix C.4.

D.8 Trainable parameter subsets

EquiPT can effectively adapt targeted sub-blocks within each physical module while preserving their original neural architectures and physical exchange interfaces. Table 10 benchmarks five distinct trainable parameter subsets on NT under an identical data budget (400 coupled training samples) and iteration budget (2000 steps), evaluated under Anderson-50. As the proportion of trainable parameters increases, the system prediction error steadily improves from 0.0257 when adapting only the output projection layers (0.26% of parameters) down to 0.0081 for full-parameter adaptation. Notably, adapting only the decoder-side components (17.90% of parameters) achieves a system error of 0.0083, which is within 2.5% of full-parameter adaptation (0.0081). Across the four restricted configurations (0.26% to 17.90%), error reductions are most pronounced on the fuel ($3.74\times$ reduction) and coolant ($3.17\times$ reduction) temperature fields, whereas the neutron flux error varies by only $1.26\times$, indicating that post-training refines mainly the thermal fields (fuel and coolant temperature) that dominate system error. Full adaptation uses the EquiPT learning rate in Table 5 (2×10^{-5}), while restricted subsets employ 10^{-4}.

Table 10: **Ablation of trainable parameter subsets for EquiPT on NT.** Prediction errors across five parameter fine-tuning configurations. Each restricted subset is one trained model. The all-parameter row reports the EquiPT means over three training runs in Table 1.

		Relative L^2 error			
			of each field		
Trainable components	Parameters	System	neutron	fuel	coolant
Output projection	121,158 (0.26%)	0.0257	0.0049	0.0370	0.0353
Lift & encoder	5,369,792 (11.30%)	0.0118	0.0041	0.0120	0.0194
Inverse decoder	6,183,744 (13.02%)	0.0104	0.0041	0.0116	0.0154
Decoder & projection	8,504,358 (17.90%)	0.0083	**0.0039**	0.0099	**0.0111**
All parameters	47,508,518 (100%)	**0.0081**	0.0040	**0.0086**	0.0116

D.9 QUALITATIVE FIELD VISUALIZATIONS

The NT and GS examples in Fig. 2 are expanded in Fig. 7 and Fig. 8 to show all three NT fields and both GS species. The NT and GS samples were selected for their large Frozen errors, and the FSI sample in Fig. 2 is the strongly coupled test sample ($\mu = 1$) on which Frozen has its largest structural error, so these panels show difficult cases for Frozen. Fig. 2 shows the coolant temperature at the final frame of the NT sample, species u at the last of the twelve predicted frames of the GS sample, and the horizontal velocity after one $0.05\,\mathrm{s}$ step of the FSI sample. All field panels use 200 Picard exchanges for both methods, with the damping factors of Appendix C.3. Predicted field panels share the ground truth's color range, while absolute error maps use a unified scale spanning from zero to their joint maximum across methods. NT fields are displayed in normalized units, whereas GS concentrations (u, v) maintain independent ranges matching their distinct reaction–diffusion scales. The insets of Fig. 2 sort the absolute error over the spatial points after dividing it by the root mean square of the ground truth field, with nodal area weights on FSI. FSI errors are taken at corresponding mesh nodes, while each field is drawn on the geometry predicted by its method.

Fig. 7 shows the neutron flux, fuel temperature, and coolant temperature at the terminal time frame in NT. Compared with Frozen, EquiPT more closely matches the temperature distributions of the ground truth, yielding substantially lower prediction errors across both fuel and coolant temperature fields. Fig. 8 shows the concentration fields of chemical species u and v in GS. Frozen distorts the shapes and locations of the reaction patterns, whereas EquiPT captures the reaction pattern morphology with high fidelity across both chemical species.

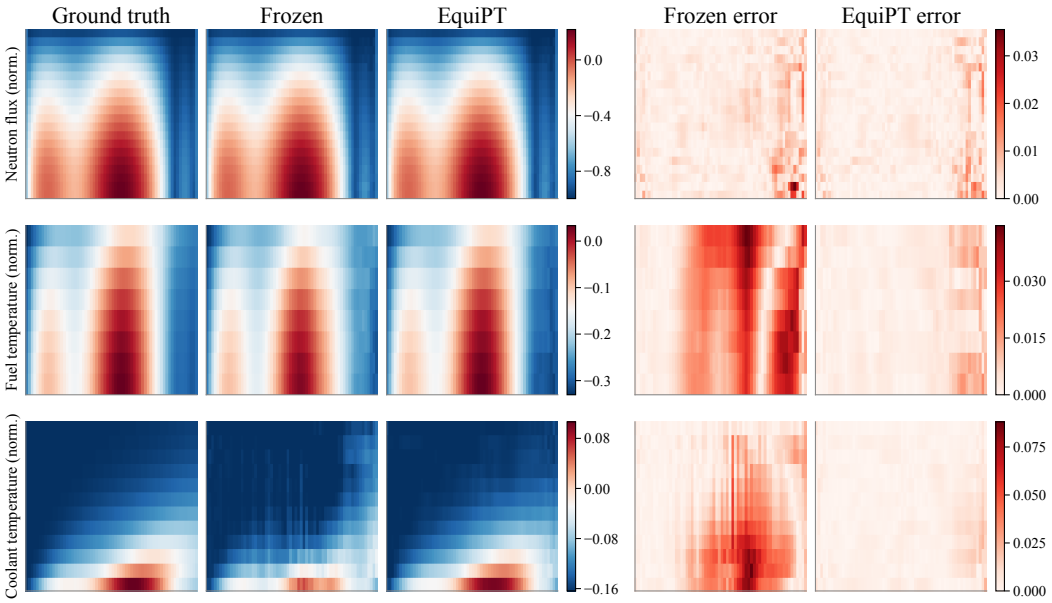

Figure 7: **Qualitative physical field predictions and error distributions on NT.** Predicted physical fields and error distributions for Frozen assembly and EquiPT at the final frame of the NT test sample in Fig. 2. Coolant velocity and pressure are included in the coolant error of Table 1 but are not shown.

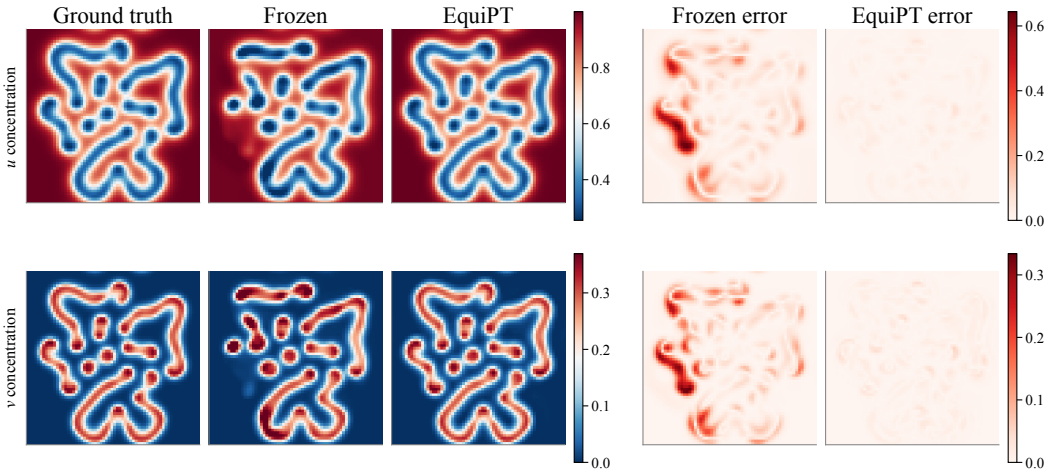

Figure 8: **Qualitative concentration field predictions and error distributions on GS.** Predicted concentration fields and error distributions for chemical species u and v at the last predicted frame of the GS test sample in Fig. 2.