LEARNING PDE DYNAMICS BETWEEN SUBMANIFOLDS USING GREEN'S OBSERVATION OPERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many physical systems are driven and observed only on lower-dimensional submanifolds of a larger spatial domain, while their dynamics are governed by the ambient medium occupying that domain. Examples include laser-heated parts imaged by an infrared camera, and ground-level emissions measured on a sensor plane. Full-domain solvers, however, compute the entire volume for every new source although only the observation submanifold is needed, and black-box surrogates do not exploit that the ambient medium remains fixed. We introduce the *Green's Observation Operator (GObO)*, which maps the ambient medium once to the Green's kernel of a linear PDE restricted to the source and observation submanifolds. New sources then cost one lower-dimensional integral and no network evaluation. Exponential rates in the kernel yield an exact finite streaming state with horizon-independent memory; we prove its stability and an approximation rate for the restricted heat kernel. On three-dimensional heat conduction and advection–diffusion with collocated and distinct source and observation geometries, GObO trained on static sources predicts responses to moving sources zero-shot with 4–8$\times$ lower error than black-box surrogates, at 1.4 ms per query after a single conditioning pass. The same kernel transfers across resolutions and admits corrections for mild nonlinearities, including radiative losses and temperature-dependent conductivity, without retraining, at the cost of lower in-distribution accuracy.

1 INTRODUCTION

Many systems in science and engineering are described by partial differential equations (PDEs). Examples include heat flow in manufactured parts, airflow around vehicles, blood flow in arteries, or acoustic wave propagation. In many applications, however, the full state inside the domain is not observed. Instead, measurements are available only on a lower-dimensional part of the domain, such as a cross-section, a boundary face, or more generally, a *submanifold*. Likewise, the input that drives the system, represented by a forcing, may act only on a lower-dimensional region, such as a heated surface, an area on the ground, or a boundary.

Laser-based metal additive manufacturing gives a concrete example of this setting, where laser locally heats and melts the material (Sajadi et al., 2025) on a surface, while an infrared camera observes only a lower-dimensional temperature field. Thus, both the input and the observed output are lower-dimensional, but their relation still depends on the full material through which heat travels. Figure 1 illustrates this setting schematically, with a laser acting on a source submanifold and the resulting temperature being observed on an observation submanifold coupled through a heterogeneous ambient medium.

Similarly, in atmospheric transport, pollutants or greenhouse gases may be released from an area on the ground and then carried through the three-dimensional atmosphere by wind and diffusion. Measurements, however, might only cover a two-dimensional concentration field on a vertical plane (Hashmonay et al., 2001). Related settings include surface-pressure prediction in aerodynamics (Li et al., 2023b), wall-shear-stress prediction in hemodynamics (Suk et al., 2024), or nearfield acoustic holography (Olivieri et al., 2021).

In these settings, the mapping between lower-dimensional source and observation submanifolds is still governed by the full ambient medium. The restricted state is therefore generally not closed

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.

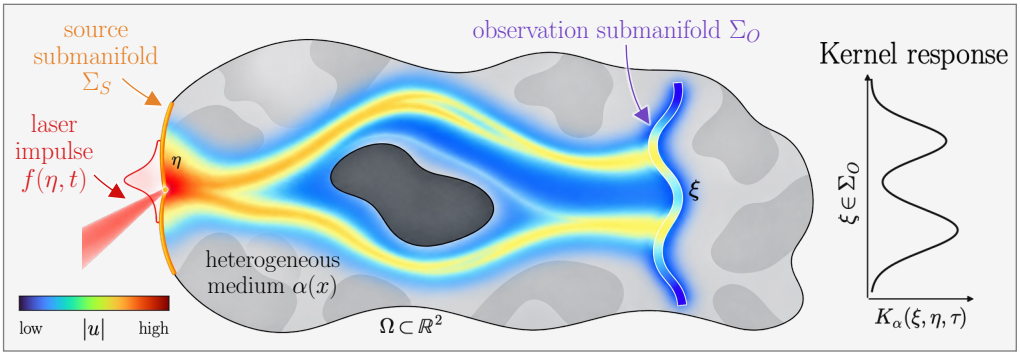


Figure 1: A laser acts on a source submanifold and drives an observed temperature response on the observation submanifold. The ambient medium affects the temperature flux. Classical PDE solvers model the transport through the full volume, whereas our GObO directly models the map between source and observation submanifolds.

and cannot be modeled independently of the surrounding domain. Classical PDE solvers retain this dependence by resolving the full ambient state, but consequently compute throughout the volume even when only the response on the observation submanifold is needed. This additional computation increases prediction time, which can limit its use in online prediction or planning settings.

We introduce the *Green's Observation Operator* (GObO), which learns a time-dependent Green's kernel directly between the source and observation submanifolds, avoiding full-domain computation. GObO enables rapid prediction under many different source terms and generalizes across source configurations beyond those observed during training. Our main contributions are:

- We introduce the **Green's Observation Operator (GObO)**, a time-dependent Green's kernel model acting directly between source and observation submanifolds conditioned on the ambient medium.
- We develop a **structured kernel parametrization** and prove its exponential approximation rate for the restricted heat kernel (Theorem 1), together with exact finite-state recurrent evaluation and its stability (Propositions 3–4).
- We **benchmark GObO** against PDE solvers, neural operators, and learned Green's function models. It achieves 4–8$\times$ lower error under source generalization, up to 3.5$\times$ lower error than the learned Green's function baseline, and over 100$\times$ faster evaluation of new sources than black-box neural operators.

2 Prediction between submanifolds needs a tailored approach

In this section, we formalize prediction between source and observation submanifolds and motivate a model that works directly on the submanifolds while retaining the ambient dynamics.

2.1 Submanifolds of closed systems are not closed

We study a map between *lower-dimensional* submanifolds whose output, unlike the full PDE state, does not form a closed dynamical system. We consider a time-dependent PDE on an ambient spatial domain $\Omega \subset \mathbb{R}^d$ over a time interval $[0, T]$. Its solution is $u_\alpha(x, t)$, while $\alpha : \Omega \to \mathbb{R}^p$ describes the known medium and f drives the system. With differential operator $\mathcal{P}_\alpha$, we write the PDE as

$$\mathcal{P}_\alpha u_\alpha = f(x, t), \qquad (x, t) \in \Omega \times (0, T], \tag{1}$$

together with suitable initial and boundary conditions. The source f acts on a source submanifold $\Sigma_S \subseteq \Omega$, i.e. $f|_{\Omega \setminus \Sigma_S} \equiv 0$. The solution is observed on a possibly different observation submanifold $\Sigma_O \subseteq \Omega$. Denoting the observed trace by $v_\alpha := u_\alpha|\Sigma_O$, we seek the map

$$\mathcal{S}_\alpha : \quad f \longmapsto v_\alpha. \tag{2}$$

Although both source and output are lower-dimensional, this map still depends on the full ambient medium. Therefore, the observed trace on Σ_O is generally not closed. While the full state $u_\alpha(\cdot, t)$ determines its future, its restriction $v_\alpha(\cdot, t)$ generally does not. Information can propagate through the unobserved domain and later influence Σ_O again. Consequently, the surrounding volume therefore acts as a hidden memory. We next consider two approaches to this problem. Classical PDE solvers compute the solution in the full ambient domain, whereas data-driven methods seek to learn the source-to-observation map directly.

2.2 PDE SOLVERS REQUIRE CLOSED SYSTEMS

Classical PDE solvers obtain the observation trace by first solving the closed dynamical system in the full ambient domain. They typically compute u_α by discretizing the governing equations throughout the full volume (LeVeque, 2007). The resulting state can then be restricted to Σ_O, but this requires computing the full solution even when only the lower-dimensional trace is needed. Moreover, because the trace is not a closed state, each new source generally requires solving the ambient PDE again. This overhead accumulates across repeated predictions and, on the heat benchmark of Section 5, makes the finite-volume solver about $2750\times$ slower than our model, as shown in Table 1.

	Ours	Finite-volume solver
Time for 1000 source predictions/solves	**1.7** s	1.3 h

Table 1: Classical solvers repeat a volumetric solve for every source although only the lower-dimensional trace is required, making repeated prediction expensive. See details in Section 5.5.

2.3 BLACK-BOX APPROACHES MISS KEY OPERATOR STRUCTURE OR DYNAMICS

General black-box surrogates such as DeepONet and the Fourier Neural Operator (FNO) can learn mappings between function spaces, including on submanifolds, but miss key structure of our setting (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). They can be conditioned on the ambient medium α, but process α and the source f jointly through a nonlinear mapping. Each new source therefore requires another network evaluation, even when the medium is unchanged, and generalization to source configurations beyond those observed during training is not built into the representation.

Neural Green's functions provide a more structured alternative by learning a source-independent kernel that is applied to different sources (Boullé et al., 2022b; Yoo et al., 2026). This makes them well-suited to generalization across sources. For transient problems, however, the Neural Green's Operator (NGO) and DGNet learn a one-step kernel and roll it out iteratively (Melchers et al., 2026; Tan et al., 2026). This requires a closed system state, which v_α does not provide, making these constructions not directly applicable to our setting.

2.4 PROPOSED GOBO DIRECTLY MODELS NON-CLOSED SUBMANIFOLD DYNAMICS

The proposed Green's Observation Operator addresses these limitations by representing the effect of the ambient dynamics directly between the source and observation submanifolds. Its time-dependent kernel captures how the full medium couples the two submanifolds without reconstructing the volumetric state or requiring the observation trace to form a closed dynamical system. Once this medium-dependent representation is learned, new sources can be evaluated without another network pass.

We introduce a structured parametrization of this kernel in the following sections. Besides supporting efficient approximation, it enables a recurrent computation that compresses the system memory into a running state. We first introduce the Green's-function formulation underlying this construction.

3 A PRIMER ON GREEN'S FUNCTIONS

Green's functions provide a useful representation of the solution of linear PDEs. We briefly review the ideas needed below and refer to Stakgold & Holst (1979) for a comprehensive treatment. In the notation of Section 2, a Green's function is the response to a unit impulse at location y and time s:

$$\mathcal{P}_{\alpha,(x,t)} G_\alpha(x,t;y,s) = \delta(x-y)\delta(t-s).$$

For a time-independent linear system, superposition gives the response to a source F as

$$u_\alpha(x,t) = \int_0^t \int_\Omega G_\alpha(x,y,t-s) F(y,s)\, dy\, ds\,. \tag{3}$$

This simplified display assumes zero initial data and homogeneous Neumann boundary conditions. Appendix B.1 develops the full setting with initial and boundary inputs. The kernel G_α depends on the medium α, but not on F, so it can be reused across source histories.

We now specialize the general problem in Section 2 to the parabolic PDE family studied throughout this work:

$$\partial_t u_\alpha + b(x)\cdot\nabla u_\alpha - \nabla\cdot\big(\kappa(x)\nabla u_\alpha\big) = F, \tag{4}$$

Here $\alpha = (b,\kappa)$ collects the transport field b and the spatially varying diffusivity κ. Setting $b = 0$ gives the heat equation, while nonzero b gives advection–diffusion.

Proposition 1 (Existence of the Green's kernel, *informal*)**.** *Under the standard assumptions in Appendix B.1, for each admissible $\alpha = (b,\kappa)$ the parabolic problem equation 4 with homogeneous Neumann boundary conditions admits a Green's kernel G_α that is continuous up to the boundary for positive time.*

The precise statement and proof are given in Appendix B.1. Under mild integrability assumptions on F, the integral in equation 3 is finite (Daners, 2000). Because the PDE is linear, a single Green's kernel can predict the response to any source in a fixed medium through equation 3. However, the kernel changes with the medium and is available analytically only in a few simple cases (Davies, 1989). The next section therefore develops a method for learning the map $\alpha \mapsto G_\alpha$ in our setting of prediction between submanifolds.

4 GREEN'S OBSERVATION OPERATOR

This section develops the Green's Observation Operator (GObO). Our construction first restricts the ambient Green kernel to the source and observation submanifolds and then introduces a tractable parametrization for learning and prediction.

4.1 RESTRICTING THE AMBIENT GREEN KERNEL TO SUBMANIFOLDS

Rather than learning the ambient kernel G_α between arbitrary points in Ω, GObO learns only its restriction from sources on Σ_S to observations on Σ_O, as introduced in Section 2. To express this restricted map in coordinates, we use bi-Lipschitz C^1 charts

$$X_S : U_S \subset \mathbb{R}^{m_S} \to \overline{\Omega} \qquad \text{and} \qquad X_O : U_O \subset \mathbb{R}^{m_O} \to \overline{\Omega} \tag{5}$$

that define a source manifold $\Sigma_S := X_S(U_S)$ and an observation manifold $\Sigma_O := X_O(U_O)$ respectively. In these coordinates, restricting the observation variable is straightforward and simply amounts to evaluating the solution on Σ_O, so the observed trace from equation 2 becomes $v_\alpha(\xi,t) := u_\alpha(X_O(\xi),t)$. Restricting the source variable requires a different construction, since Σ_S generally is a null set in the ambient space and its contribution would disappear under naïve integration in equation 3.

Instead, we will follow the standard singular-source construction used in immersed-interface and immersed-boundary methods (LeVeque & Li, 1994; Peskin, 2002; Kandilarov & Vulkov, 2007). First suppose that $\Sigma_S \subset \Omega$ and let $f_\Sigma(y,t)$ denote source strength per unit surface measure. For smooth test functions $\varphi \in C_c^\infty(\Omega)$, with $d\sigma$ denoting the surface measure on Σ_S, we represent this source by the distribution

$$F(\cdot,t) := f_\Sigma(\cdot,t)\delta_{\Sigma_S}, \qquad \langle F(\cdot,t),\varphi\rangle = \int_{\Sigma_S} f_\Sigma(y,t)\varphi(y)\, d\sigma(y), \tag{6}$$

which can also be obtained as the distributional limit of smooth volumetric sources supported in shrinking neighborhoods of Σ_S (Friedlander, 1998). Substituting equation 6 into equation 3 reduces the ambient to a surface integral. In chart coordinates, and using the area formula, this can be written as

$$v_\alpha(\xi,t) = \int_0^t \int_{U_S} K_\alpha(\xi,\eta,t-s)\, f(\eta,s)\, d\eta\, ds, \tag{7}$$

with $f(\eta,t) := f_\Sigma(X_S(\eta),t)$ and surface element $J_S(\eta) := \sqrt{\det(DX_S(\eta)^\top DX_S(\eta))}$ directly absorbed into the kernel $K_\alpha(\xi,\eta,\tau) := G_\alpha(X_O(\xi),X_S(\eta),\tau)\, J_S(\eta)$. Absorbing the surface element into K_α keeps inference in the form of a standard chart integral.

If we instead consider $\Sigma_S \subset \partial\Omega$ as an inward Neumann flux, we treat F as a usual boundary condition. Appendix B.1 shows that, in the absence of volumetric forcing, this yields the same integral representation as in equation 7. Therefore, in the following, we use equation 7 for both interior and boundary source manifolds. The regularity result below provides the corresponding continuous extension of the kernel to the boundary, so we also allow $\Sigma_O \subset \bar{\Omega}$. It remains to verify that restricting the kernel to the measure-zero source manifold preserves integrability.

Proposition 2 (Integrability of the restricted kernel, *informal*)**.** *For every finite $T > 0$, the source integral of the restricted kernel is bounded by $C_T \tau^{-1/2}$ for $0 < \tau \le T$, uniformly over the observation manifold. In particular, the kernel is integrable over source space and time.*

The precise statement and proof are given in Appendix B.1. In particular, this justifies that the integral in equation 7 exists in all settings considered below.

4.2 Spectral separation of time and space in the restricted kernel

Spectral decomposition of the kernel provides a tractable parametrization, which is needed because a naive pointwise parametrization is prohibitively expensive. The latter would represent the kernel with a coordinate-based network $\big((\xi,\eta,\tau),\alpha\big) \mapsto K_\alpha(\xi,\eta,\tau) \in \mathbb{R}$ which requires too many network evaluations to compute the space–time integral. For a 2D domain discretized with 64 points per spatial axis and 64 time steps, for instance, a single prediction would already require $64^2 \times 64^2 \times 64 > 10^9$ evaluations. Our parametrization is motivated by a classical spectral decomposition, which also separates time and space in the restricted kernel.

Indeed, for strongly elliptic second-order operators on bounded Lipschitz domains, spectral expansions in eigenfunctions and generalized eigenfunctions are established by Agranovich (2008). If the operator has no nontrivial generalized eigenfunctions, this reduces to the biorthogonal eigenfunction expansion

$$G_\alpha(x,y,\tau) = \sum_{k=1}^{\infty} e^{-\lambda_k \tau} r_k(x) \ell_k(y), \tag{8}$$

where r_k and ℓ_k are the corresponding right and left eigenfunctions and eigenvalues λ_k, which may be complex. We restrict this expansion to the source and observation manifolds and, to obtain a tractable parametrization for optimization, truncate the sum after M terms:

$$K_\alpha(\xi,\eta,\tau) = \sum_{k=1}^{\infty} e^{-\lambda_k \tau} \psi_k(\xi)\varphi_k(\eta) \approx \sum_{k=1}^{M} e^{-\lambda_k \tau} \psi_k(\xi)\varphi_k(\eta) =: K_{\alpha,M}(\xi,\eta,\tau), \tag{9}$$

with restricted spatial factors $\psi_k(\xi) := r_k(X_O(\xi))$ and $\varphi_k(\eta) := \ell_k(X_S(\eta))J_S(\eta)$. Thus, the source-side factor absorbs the surface element.

4.3 Exponential convergence of spectral truncation for the heat kernel

For the heat equation, this truncation is particularly well behaved, and its approximation error decays exponentially in $M^{2/d}$ with the number M of retained terms. This behavior follows from the spectral structure of the heat kernel. Its expansion holds pointwise, and the regular growth of the underlying operator's eigenvalues causes higher modes to be rapidly suppressed at positive times. Consequently, once we stay a positive time τ_0 away from the singularity at $\tau = 0$, the restricted kernel satisfies the explicit truncation bound stated next.

Theorem 1 (Approximation of the restricted heat kernel, *informal*)**.** *Assume the setting of Propositions 1 and 2 with $b = 0$. For each fixed medium α and every $\tau_0 > 0$, there are constants $C_{\tau_0}, c > 0$, with c independent of τ_0, such that*

$$\sup_{\tau \geq \tau_0} \|K_\alpha(\cdot,\cdot,\tau) - K_{\alpha,M}(\cdot,\cdot,\tau)\|_{L^2(U_O \times U_S)} \leq C_{\tau_0} e^{-c\tau_0 M^{2/d}}. \tag{10}$$

The precise statement and proof are given in Appendix B.2. This justifies the truncated representation in equation 9 and leaves the spatial factors as the remaining objects to represent.

4.4 Finite-dimensional parametrization of the restricted kernel

We approximate the remaining spatial factors of the truncated kernel in finite-dimensional bases $\mathbf{\Phi}_S(\eta) \in \mathbb{R}^{D_S}$ and $\mathbf{\Phi}_O(\xi) \in \mathbb{R}^{D_O}$ on the source and observation submanifolds, respectively. For each retained mode, we represent the two spatial factors by coefficient vectors $\mathbf{v}_k(\alpha) \in \mathbb{R}^{D_S}$ and $\mathbf{u}_k(\alpha) \in \mathbb{R}^{D_O}$. This gives

$$K_{\theta,\alpha}(\xi,\eta,\tau) = \sum_{k=1}^{M} e^{-\lambda_k \tau} \underbrace{\big(\mathbf{\Phi}_O(\xi)^\top \mathbf{u}_k(\alpha)\big)\big(\mathbf{v}_k(\alpha)^\top \mathbf{\Phi}_S(\eta)\big)}_{=:\, \mathbf{\Phi}_O(\xi)^\top A_k(\alpha) \mathbf{\Phi}_S(\eta)} \tag{11}$$

where $A_k(\alpha) := \mathbf{u}_k(\alpha)\mathbf{v}_k(\alpha)^\top$ collects the coefficients.

A conditioning network maps α to the factors $\mathbf{u}_k(\alpha)$ and $\mathbf{v}_k(\alpha)$ of the rank-one matrices $\{A_k\}$. Thus, the network learns only the medium-dependent coefficients, while source predictions are then obtained by numerical integration. Since the predicted response is linear in each A_k for fixed rates, a squared loss is convex in the coefficient matrices, but not in their individual factors. Results on factored low-rank optimization suggest, under assumptions that our problem does not fully satisfy, that rank-one factors can stall at stationary points that do not minimize the convex loss. Additional factor columns can supply descent directions away from such points (Burer & Monteiro, 2005; Haeffele & Vidal, 2015; Zhang, 2025). Motivated by these results, we use a higher-rank factorization as an optimization heuristic and parametrize

$$A_k(\alpha) = U_k(\alpha) V_k(\alpha)^\top,$$

with $V_k(\alpha) \in \mathbb{R}^{D_S \times L}$ and $U_k(\alpha) \in \mathbb{R}^{D_O \times L}$ to allow for higher rank. We elaborate on this in Appendix F.6, and the experiment in Section 5.4 demonstrates the effect empirically.

4.5 Markovian streaming via an augmented kernel state

The kernel parametrization in equation 11 also admits a Markovian augmented-state representation that can be used for recurrent streaming prediction. Since the dynamics on the observation submanifold are not closed, naively evaluating the Green's integral would require reintegration over the full source history at every time step with growing cost over longer prediction horizons. On a time grid where the source is piecewise constant, the augmented formulation is exact and replaces this growing history by a fixed-dimensional state whose size is independent of the prediction horizon. The observed state can then be reconstructed from the augmented state.

Proposition 3 (Exact streaming evaluation, *informal*)**.** *On a uniform grid $t_i = i\Delta$, a piecewise-constant projected source $\widehat{f^i}$ can be evaluated exactly with one recurrent state per temporal mode, updated by*

$$g_k^{i+1} = e^{-\lambda_k \Delta} g_k^i + \left(\int_0^\Delta e^{-\lambda_k s}\, ds \right) V_k(\alpha)^\top \widehat{f^i}. \tag{12}$$

The observation is reconstructed as $v_\alpha(\xi, t_i) = \mathbf{\Phi}_O(\xi)^\top \sum_{k=1}^{M} U_k(\alpha) g_k^i$.

Furthermore, Proposition 4 shows that the exponential update makes this recurrence stable whenever the rates have nonnegative real parts. For both propositions, precise statements and proofs are given in Appendix B.4.

Proposition 4 (Stability of the streaming state, *informal*)**.** *If $\mathrm{Re}(\lambda_k) \geq 0$, the recurrence in Proposition 3 does not amplify state perturbations. If the inequality is strict, bounded projected sources yield uniformly bounded states.*

4.6 Reusing the linear kernel for mild nonlinearities

The learned kernel describes a linear system, but classical techniques allow us to reuse it for certain mild nonlinearities. One such technique handles source-like nonlinearities by alternating the learned linear propagation with a nonlinear source update, in the spirit of Strang splitting (Strang, 1968). When the nonlinear term can be evaluated from the predicted trace, the streaming state makes these repeated updates inexpensive. Appendix B.5.1 develops this construction for radiative heat loss.

A second technique handles state-dependent material laws through a change of variables. For nonlinear heat conduction, the Kirchhoff transform either reduces the problem exactly to the learned linear response or leaves a transformed nonlinear system that can be advanced approximately using the streaming state. Appendix B.5.2 develops both cases.

5 Experiments

Our experiments show that GObO learns the dynamics between source and observation submanifolds while enabling efficient prediction and source generalization. We compare against black-box neural operators and learned Green's function approaches.

5.1 Experimental setting

We study three-dimensional heat conduction and advection–diffusion with different configurations of source and observation submanifolds. The datasets are obtained from finite-volume solves and specified in Appendix C. All results report the mean relative error and standard deviation over 3 seeds. Evaluation tests predictive accuracy across unseen inputs f and medium configurations α. Our GObO implementation uses cosine basis functions whose coefficients are learned with a medium-conditioned U-FNO-style cosine spectral backbone (Wen et al., 2022; Liu et al., 2023). The temporal rates are not learned. Further details on the architecture, baselines and experimental details are given in Appendices D and E, with ablations of these choices in Appendix F.

5.2 GObO learns a kernel that generalizes across sources

The kernel learned by GObO remains accurate across different source classes, although this comes at some cost on in-distribution accuracy. We train all models on static source pulses and evaluate them on held-out static pulses and on sources moving along a straight line. Figure 2 illustrates this transfer in the heat setting. While the medium-conditioned FNO-2D does not recover the location, shape, or magnitude of the moving response, GObO continues to track it closely.

system	method	static	moving
heat	GObO	0.011 ± 0.001	$\mathbf{0.012 \pm 0.000}$
	adapted NGO, $S = 1$	0.039 ± 0.001	0.043 ± 0.000
	medium-conditioned FNO-2D	$\mathbf{0.001 \pm 0.000}$	0.101 ± 0.004
advection–diffusion	GObO, fixed rates	0.114 ± 0.018	$\mathbf{0.101 \pm 0.020}$
	medium-conditioned FNO-2D	$\mathbf{0.104 \pm 0.009}$	0.430 ± 0.015

Table 2: GObO transfers to unseen source configurations. All models are trained on static sources and evaluated on static and moving sources without retraining. Shown is the mean relative error and standard deviation.

Table 2 quantifies this behavior for both heat and advection–diffusion. FNO-2D achieves the lowest error on static heat sources, but its error increases by about $100\times$ for moving sources, whereas the error of GObO changes only slightly. The adapted NGO, another learned Green's function approach, shows similarly stable transfer across source classes, but with errors around $4\times$ higher than GObO. This pattern is consistent across additional source and observation configurations, where the FNO-2D error increases by about 4–7$\times$ while GObO remains stable (Appendix F.2.3). Additional black-box baselines, including FNO-3D and a backbone-matched spectral predictor, show similar behavior (Appendix F.2.1).

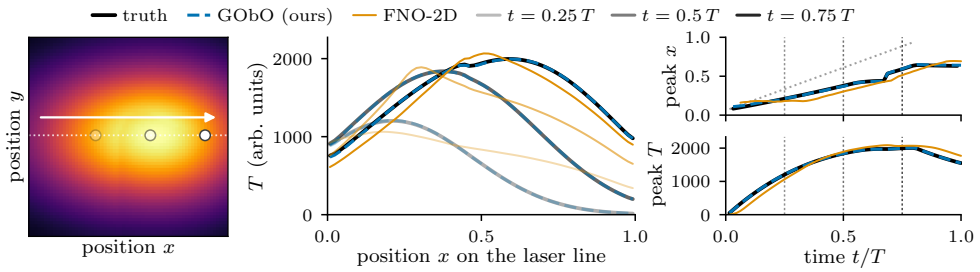
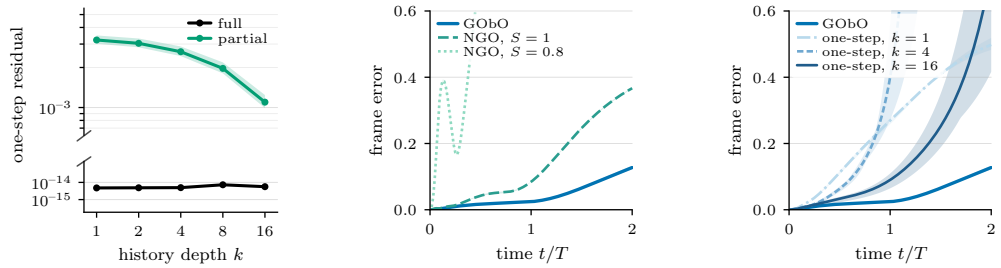


Figure 2: GObO tracks the moving response, whereas FNO does not track it accurately. Left: true final temperature and source trajectory. Middle: temperature profiles at the three indicated times. Right: position and magnitude of the temperature peak.

5.3 The augmented kernel state captures non-closed submanifold dynamics

Autoregressive prediction from the restricted state requires information about its history, whereas GObO carries the missing information in an augmented kernel state. We first verify this in a controlled regression setting by predicting the next state from different amounts of observation history. As shown in Figure 3(a), additional history reduces the error by about $3\times$ for the restricted trace, while a closed two-dimensional heat equation on the same manifold is predicted to machine precision from a single state. This isolates the effect of the non-closed dynamics. The same issue appears in recurrent Green's models. Transient NGO learns a purely spatial kernel in a one-step recurrence with a spectral cap S for stabilization, but yields about $3.5\times$ higher rollout error than GObO and degrades over longer horizons (Figure 3(b)). To separate this effect from architectural differences, we also place a purely spatial GObO kernel in the same one-step setting, matched to the full model except for the temporal exponential rates. Its predictions again improve with additional history, but remain less accurate than full GObO (Figure 3(c)). These results show that the non-closed dynamics are consequential and that GObO's augmented kernel state captures the information needed to propagate them. Experimental details and complete results are given in Appendices E.3 and F.3.2.



(a) full and partial observations (b) GObO vs. NGO (c) one-step GObO variant

Figure 3: The restricted observation requires additional state information for recurrent prediction. (a) Next-step regression error with increasing observation history, compared with a closed two-dimensional heat system. (b) Rollout error of GObO and transient NGO with spectral cap S. (c) History dependence of the purely spatial one-step GObO variant.

5.4 The kernel parametrization captures the spatiotemporal structure

The proposed spectral kernel parametrization is important for capturing the joint spatial and temporal structure of the response. Replacing it with a naive tensor decomposition over space and time increases the error by 3–4$\times$ (Appendix F.4.2). This difference is also visible in the impulse response of Figure 8, where GObO captures the exponential temporal structure while the tensor ablation does not (Appendix F.4.1). We further isolate the factor search rank L in a controlled example with curved manifolds, a homogeneous medium, and an analytically known kernel. Here, the task reduces to direct regression of the kernel coefficients. Increasing L from 1 to 8 reduces the fitting objective by about $14\times$, with little improvement afterward, which motivates our choice of $L = 16$ as a safety

margin. This agrees with the full-setting ablation in Appendix F.2.2; the controlled recovery and full rank sweep are reported in Appendices F.4.4 and F.4.5.

5.5 GObO enables rapid source evaluation and nonlinear corrections

Once learned, the GObO kernel supports inexpensive source evaluation and extensions of the underlying linear model. Constructing the kernel requires one neural-network pass per medium, comparable in cost to one FNO-2D prediction, while subsequent source evaluations are $134\times$ faster on CPU and $5.6\times$ faster on GPU. Table 3 summarizes these costs for batch size 1, with similar amortization for batched and streaming evaluation in Appendix F.5.1.

method	frequency	CPU	GPU, $b = 1$
FV solver, matched accuracy	per source	4.6 s	n/a
FNO-2D forward	per source	194.1 ms	2.50 ms
GObO conditioning	per medium	218.3 ms	10.1 ms
GObO query	per source	**1.45** ms	**0.45** ms

Table 3: GObO constructs one kernel per medium and then evaluates new sources faster than FNO-2D or the numerical solver.

Its basis-function representation also supports arbitrary evaluation locations on the observation submanifold, with numerically negligible additional error at $2\times$ the training resolution. The complete resolution-transfer study is given in Appendix F.5.2. Finally, the same kernel further supports the nonlinear corrections of Section 4.6 without retraining, improving accuracy by more than $11\times$ for radiative feedback and up to $29\times$ for temperature-dependent conductivity (Appendix F.5.3). The corrected traces closely follow the corresponding nonlinear reference solutions in Figures 11 and 12.

6 Related Work

We summarize the most directly related work here and provide an extended discussion of related work in Appendix A.

Neural operators and surrogates. Neural operators can learn maps between function spaces, including maps from sources on one submanifold to observations on another. However, they usually treat this relation as a black box and do not explicitly impose linearity in the source. GObO instead encodes this structure through a reusable restricted Green kernel.

PDEs on manifolds. Recent methods solve PDEs posed intrinsically on manifolds or learn their solution operators (Burbulla, 2023; Chen et al., 2024; Liao et al., 2025). In our setting, the PDE instead evolves in the ambient domain and the observed submanifold is not closed. GObO therefore models how the surrounding medium couples the source and observation submanifolds.

Learning Green's functions. Learned Green functions often identify a fixed operator (Boullé et al., 2022b). Recent methods condition steady-state representations on geometry (Yoo et al., 2026) or Green operators on PDE coefficients, while transient approaches rely on fixed time-stepping schemes. GObO instead learns the continuous time-dependent kernel between submanifolds.

7 Conclusion

We introduced the Green's Observation Operator, a method for learning a Green's kernel that maps directly between source and observation submanifolds of a larger ambient domain. GObO preserves source linearity and causality and admits a horizon-independent streaming form, enabling rapid predictions and generalization to new sources. Across heat conduction and advection–diffusion, this structure trades some in-distribution accuracy for reliable source transfer and cheap reuse. The same kernel also supports resolution transfer, longer rollouts, and extensions to mild nonlinearities without retraining the linear kernel. Future work could extend the approximation and stability guarantees to broader classes of PDEs and submanifold settings, and explore more general nonlinear dynamics.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets, to assist in the writing of proofs, to implement methods, to assist with translation, and to design or provide feedback on research methodology or experiments. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide critical ingredients for proving mathematical claims, to propose or refine hypotheses, to clean and reformat datasets, to support qualitative and thematic data analysis, or to interpret results. Additionally, we used generative AI tools to create or modify scientific figures or images, to create or edit software code, to draft parts of a research paper, to summarize or analyze existing literature, to search for and identify relevant literature, and to format references. We have reviewed all AI-assisted work. AI-assisted code was read and tested by the authors, AI-assisted figures were checked against the underlying numerical results, and AI-drafted text was verified against the experiments and the sources it describes. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All theoretical claims of the main text are stated with their assumptions and proved in Appendix B. Appendix C reports the data-generation procedures, parameters, and splits needed to regenerate the simulated datasets. Appendix D specifies the basis, model architecture, kernel parametrization, training settings, and common evaluation metric. Appendix E reports the baselines and all experiment-specific protocols, including the source-transfer, streaming, timing, resolution-transfer, and nonlinear evaluations. Complete numerical results and ablations are given in Appendix F. Upon acceptance we will release the full code base, including the data-generation, training, and evaluation scripts.

REFERENCES

Mikhail Semenovich Agranovich. Spectral boundary value problems in lipschitz domains for strongly elliptic systems in banach spaces h p σ and b p σ. *Functional Analysis and Its Applications*, 42(4):249–267, 2008.

Ziad Aldirany, Régis Cottereau, Marc Laforest, and Serge Prudhomme. Operator approximation of the wave equation based on deep learning of green's function. *Computers & Mathematics with Applications*, 159:21–30, 2024.

High-Performance Alloys. Special metals corporation, 2008.

Donald G Aronson. Non-negative solutions of linear parabolic equations. *Annali della Scuola Normale Superiore di Pisa-Scienze Fisiche e Matematiche*, 22(4):607–694, 1968.

Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.

Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. Learning dynamical systems from partial observations. *arXiv preprint arXiv:1902.11136*, 2019.

Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, and Patrick Gallinari. Modelling spatiotemporal dynamics from earth observation data with neural differential equations. *Machine Learning*, 111 (6):2349–2380, 2022.

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

Srinadh Bhojanapalli, Nicolas Boumal, Prateek Jain, and Praneeth Netrapalli. Smoothed analysis for low-rank solutions to semidefinite programs in quadratic penalty form. In *Conference on learning theory*, pp. 3243–3270. PMLR, 2018.

Nicolas Boullé, Christopher J Earls, and Alex Townsend. Data-driven discovery of green's functions with human-understandable deep learning. *Scientific reports*, 12(1):4824, 2022a.

Nicolas Boullé, Seick Kim, Tianyi Shi, and Alex Townsend. Learning green's functions associated with time-dependent partial differential equations. *Journal of Machine Learning Research*, 23 (218):1–34, 2022b.

Oussama Boussif, Yoshua Bengio, Loubna Benabbou, and Dan Assouline. Magnet: Mesh agnostic neural pde solver. *Advances in Neural Information Processing Systems*, 35:31972–31985, 2022.

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

Samuel Burbulla. Physics-informed neural networks for transformed geometries and manifolds. *arXiv preprint arXiv:2311.15940*, 2023.

Samuel Burer and Renato DC Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical programming*, 103(3):427–444, 2005.

Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.

Horatio Scott Carslaw. *Introduction to the Mathematical Theory of the Conduction of Heat in Solids*. Macmillan and Company, limited, 1906.

Gengxiang Chen, Xu Liu, Qinglu Meng, Lu Chen, Changqing Liu, and Yingguang Li. Learning neural operators on riemannian manifolds. *National Science Open*, 3(6), 2024.

Sung Woong Cho, Jae Yong Lee, and Hyung Ju Hwang. Learning time-dependent pde via graph neural networks and deep operator network for robust accuracy on irregular grids. *Journal of Computational Physics*, pp. 114430, 2025.

Jongkeun Choi and Seick Kim. Green's function for second order parabolic systems with neumann boundary condition. *Journal of Differential Equations*, 254(7):2834–2860, 2013.

Daniel Daners. Heat kernel estimates for operators with boundary conditions. *Mathematische Nachrichten*, 217(1):13–41, 2000.

Edward Brian Davies. *Heat kernels and spectral theory*. Number 92. Cambridge university press, 1989.

Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Aarti Singh, and Barnabas Poczos. Gradient descent can take exponential time to escape saddle points. *Advances in neural information processing systems*, 30, 2017.

Antoine Fabre and Jordan Hristov. On the integral-balance approach to the transient heat conduction with linearly temperature-dependent thermal diffusivity. *Heat and Mass Transfer*, 53(1):177–204, 2017.

Friedrich Gerard Friedlander. *Introduction to the Theory of Distributions*. Cambridge University Press, 1998.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International conference on machine learning*, pp. 1233–1242. PMLR, 2017.

Jianghang Gu, Ling Wen, Yuntian Chen, and Shiyi Chen. An explainable operator approximation framework under the guideline of green's function. *Journal of Computational Physics*, pp. 114244, 2025.

Wooseok Ha, Haoyang Liu, and Rina Foygel Barber. An equivalence between critical points for rank constraints versus low-rank factorizations. *SIAM Journal on Optimization*, 30(4):2927–2955, 2020.

Benjamin D Haeffele and René Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015.

Ram A Hashmonay, David F Natschke, Keith Wagoner, D Bruce Harris, Edgar L Thompson, and Michael G Yost. Field evaluation of a method for estimating gaseous fluxes from area sources using open-path fourier transform infrared. *Environmental science & technology*, 35(11):2309–2313, 2001.

Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning space-time continuous latent neural pdes from partially observed states. *Advances in Neural Information Processing Systems*, 36:26372–26395, 2023.

Juri D Kandilarov and Lubin G Vulkov. The immersed interface method for two-dimensional heat-diffusion equations with singular own sources. *Applied numerical mathematics*, 57(5-7):486–497, 2007.

Choong S Kim. Thermophysical properties of stainless steels. Technical report, Argonne National Lab., Ill.(USA), 1975.

Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D Shields. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1):5101, 2024.

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

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on learning theory*, pp. 1246–1257. PMLR, 2016.

D Lesnic, L Elliott, and DB Ingham. A note on the determination of the thermal properties of a material in a transient nonlinear heat conduction problem. *International communications in heat and mass transfer*, 22(4):475–482, 1995.

Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

Randall J LeVeque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.

Zijie Li, Saurabh Patil, Francis Ogoke, Dule Shu, Wilson Zhen, Michael Schneier, John R Buchanan Jr, and Amir Barati Farimani. Latent neural pde solver: A reduced-order modeling framework for partial differential equations. *Journal of Computational Physics*, 524:113705, 2025.

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

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

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023b.

Yunfeng Liao, Jiawen Guan, and Xiucheng Li. Curvature-aware graph attention for pdes on manifolds. In *Forty-second International Conference on Machine Learning*, 2025.

Guochang Lin, Fukai Chen, Pipi Hu, Xiang Chen, Junqing Chen, Jun Wang, and Zuoqiang Shi. Bi-greennet: learning green's functions by boundary integral network. *Communications in Mathematics and Statistics*, 11(1):103–129, 2023.

Ziyuan Liu, Yuhang Wu, Daniel Zhengyu Huang, Hong Zhang, Xu Qian, and Songhe Song. Spfno: Spectral operator learning for pdes with dirichlet and neumann boundary conditions. *arXiv preprint arXiv:2312.06980*, 2023.

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

Robert I McLachlan and G Reinout W Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.

HA Melchers, JHM Prins, and Michael RA Abdelmalik. Neural green's operators for parametric partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 455: 118893, 2026.

Changhong Mou, Binghang Lu, and Guang Lin. Neural-pod: A plug-and-play neural operator framework for infinite-dimensional functional nonlinear proper orthogonal decomposition. *arXiv preprint arXiv:2602.15632*, 2026.

Tuhin Mukherjee, JS Zuback, A De, and Tarasankar DebRoy. Printability of alloys for additive manufacturing. *Scientific reports*, 6(1):19717, 2016.

Ozisik M Necati et al. Heat conduction. *ed: John Wiley & Sons, Inc*, 1993.

Robin Nittka. Regularity of solutions of linear second order elliptic and parabolic boundary value problems on lipschitz domains. *Journal of Differential Equations*, 251(4-5):860–880, 2011.

Marco Olivieri, Mirco Pezzoli, Fabio Antonacci, and Augusto Sarti. A physics-informed neural network approach for nearfield acoustic holography. *Sensors*, 21(23):7834, 2021.

El-Maati Ouhabaz. *Analysis of heat equations on domains.(LMS-31)*. Princeton University Press, 2009.

Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

Harshwardhan Praveen, Jacob Brown, and Christopher Earls. chebgreen: Learning and interpolating continuous empirical green's functions from data. *Computer Physics Communications*, pp. 109867, 2025.

Pouyan Sajadi, Mostafa Rahmani Dehaghani, Yifan Tang, and G Gary Wang. Two-dimensional temperature field prediction with in-situ data in metal additive manufacturing using physics-informed neural networks. *Engineering Applications of Artificial Intelligence*, 150:110636, 2025.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

Jacob Seidman, Georgios Kissas, Paris Perdikaris, and George J Pappas. Nomad: Nonlinear manifold decoders for operator learning. *Advances in Neural Information Processing Systems*, 35: 5601–5613, 2022.

Ivar Stakgold and Michael J Holst. *Green's functions and boundary value problems*, volume 9. Wiley New York, 1979.

Robert Stephany and Christopher Earls. Weak-pde-learn: A weak form based approach to discovering pdes from noisy, limited data. *Journal of Computational Physics*, 506:112950, 2024.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M Wolterink. Mesh neural networks for se (3)-equivariant hemodynamics estimation on the artery wall. *Computers in Biology and Medicine*, 173:108328, 2024.

Yingjie Tan, Quanming Yao, and Yaqing Wang. Dgnet: Discrete green networks for data-efficient learning of spatiotemporal pdes. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 26748–26773, 2026.

Jingwei Tang, Vinicius C Azevedo, Guillaume Cordonnier, and Barbara Solenthaler. Neural green's function for laplacian systems. *Computers & Graphics*, 107:186–196, 2022.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

Tian Wang and Chuang Wang. Latent neural operator for solving forward and inverse pde problems. *Advances in Neural Information Processing Systems*, 37:33085–33107, 2024.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for pdes on arbitrary domains. *Advances in Neural Information Processing Systems*, 38:155423–155501, 2026.

Yang Yang, Xihong Zhang, Kunpang Kou, and Chichiu Lam. Green's function fourier operator: a spectral learning framework for linear differential equations with arbitrary boundary conditions. *Engineering Analysis with Boundary Elements*, 191:106948, 2026.

Seungwoo Yoo, Kyeongmin Yeo, Jisung Hwang, and Minhyuk Sung. Neural green's functions. *Advances in Neural Information Processing Systems*, 38:47961–47982, 2026.

Gavin Zhang, Salar Fattahi, and Richard Y Zhang. Preconditioned gradient descent for overparameterized nonconvex burer–monteiro factorization with global optimality certification. *Journal of Machine Learning Research*, 24(163):1–55, 2023.

Lulu Zhang, Tao Luo, Yaoyu Zhang, Weinan E, Zhi-Qin John Xu, and Zheng Ma. Mod-net: A machine learning approach via model-operator-data network for solving pdes. *Communications in Computational Physics*, 32(2):299–335, 2022. doi: 10.4208/cicp.OA-2021-0257.

Richard Zhang, Cédric Josz, Somayeh Sojoudi, and Javad Lavaei. How much restricted isometry is needed in nonconvex matrix recovery? *Advances in neural information processing systems*, 31, 2018.

Richard Y Zhang. Improved global guarantees for the nonconvex burer–monteiro factorization via rank overparameterization: Ry zhang. *Mathematical Programming*, 213(1):1009–1038, 2025.

Zhiwei Zhao, Changqing Liu, Yingguang Li, Zhibin Chen, and Xu Liu. Diffeomorphism neural operator for various domains and parameters of partial differential equations. *Communications Physics*, 8(1):15, 2025.

Weiheng Zhong and Hadi Meidani. Physics-informed geometry-aware neural operator. *Computer Methods in Applied Mechanics and Engineering*, 434:117540, 2025.

Overview of the Appendix

- Appendix A places GObO among operator-learning, manifold, and learned Green-function methods.
- Appendix B provides the proofs and derivations used by the method.

- Appendix C defines the equations, media, sources, observation manifolds, and numerical reference solutions.
- Appendix D describes the GObO architecture and the protocol shared across experiments.
- Appendix E gives the experiment-specific models and evaluation procedures.
- Appendix F provides the complete experimental results and ablations.

A Extended Related Work

A.1 Operator learning for PDEs

Neural operators learn mappings between function spaces and have become a common tool for data-driven models of PDEs. DeepONet and FNO use branch–trunk approximation and spectral convolution. Kovachki et al. (2023) give a general function-space framework. Later work extends these ideas to irregular meshes and graphs. Examples include message-passing PDE solvers (Brandstetter et al., 2022), mesh-agnostic neural PDE solvers (Boussif et al., 2022), and graph-enhanced DeepONets (Cho et al., 2025). We use the same function-space view but take the output to be a measured submanifold.

A.2 Partial observations and reduced representations

PDE dynamics can also be learned from limited measurements through forecasting models, latent states, weak forms, and reduced models. Partially observed PDE models use neural networks, recurrent models, or learned bases to forecast from limited measurements (Ayed et al., 2019; 2022), while latent neural PDE models infer continuous spatio-temporal dynamics from partial and noisy observations (Iakovlev et al., 2023). Weak-form methods discover PDE structure from sparse or noisy measurements (Stephany & Earls, 2024). Reduced and latent operator-learning methods, including Latent DeepONet (Kontolati et al., 2024), latent neural PDE solvers (Li et al., 2025), Latent Neural Operators (Wang & Wang, 2024), nonlinear manifold decoders (Seidman et al., 2022), and neural POD approaches (Mou et al., 2026), learn compact representations of high-dimensional states. GObO instead makes the observed submanifold the prediction space and uses the ambient medium to determine its response. The nonlinear construction in Section 4.6 also connects the learned kernel to projection-based model reduction (Benner et al., 2015; Carlberg et al., 2017).

A.3 Geometry and physical context

Many learned PDE models depend on geometry, coefficients, and boundary conditions. These quantities can enter through domain encodings, coordinate maps, graph representations, signed-distance functions, and residual-based losses. Examples include PI-GANO (Zhong & Meidani, 2025), geometry-aware operator transformers (Wen et al., 2026), Geo-FNO (Li et al., 2023a), and diffeomorphism-based neural operators (Zhao et al., 2025). GINO (Li et al., 2023b) is especially relevant because it predicts on surfaces in large three-dimensional systems. It uses full-field supervision and full-domain latent states. GObO uses only observations on the manifold and represents their response with an explicit kernel.

A.4 Nonlinear methods built on linear solvers

Operator splitting originates with Strang (1968) and is surveyed by McLachlan & Quispel (2002). Implicit–explicit methods treat stiff and non-stiff terms differently (Ascher et al., 1997). Exponential integrators propagate a linear part exactly and approximate the remaining increment (Hochbruck & Ostermann, 2010). The Kirchhoff transform is classical for temperature-dependent heat conduction (Carslaw, 1906; Necati et al., 1993). Each method assumes access to a linear solver. Section 4.6 uses the learned kernel as that solver. Its streaming form advances the linear state exactly, while source linearity allows it to evaluate the intermediate forcings created by the nonlinear method.

A.5 Green's-function and kernel-based operators

We separate methods by whether the problem is steady or transient and by how the learned kernel is represented. Pointwise methods evaluate a coordinate network at each source, observation, and possibly time coordinate. This is flexible, but resolving a dense space–time kernel requires many network evaluations and becomes impractical in the three-dimensional transient setting considered here. Structured methods instead produce a reusable finite representation that can be applied to many sources after 1 inference or identification step.

	pointwise kernel values	reusable representation
steady	GreenLearning, MOD-Net, BI-GreenNet, and GreensONet (Boullé et al., 2022a; Zhang et al., 2022; Lin et al., 2023; Gu et al., 2025)	discrete and spectral Green representations (Tang et al., 2022; Praveen et al., 2025; Yoo et al., 2026; Yang et al., 2026)
transient	GreenONet (Aldirany et al., 2024)	full-kernel identification (Boullé et al., 2022b), one-step NGO, DGNet, and GObO

Table 4: Representative learned Green-function methods. Pointwise methods query individual kernel values. Reusable methods produce a finite operator representation that is applied without another network query at each point.

Most prior work treats steady problems. Among transient approaches, GreenONet represents time through pointwise coordinate queries (Aldirany et al., 2024). The full space–time recovery of Boullé et al. (2022b) avoids these repeated queries, but identifies the kernel of 1 fixed system rather than predicting a kernel for a new medium. NGO and DGNet learn or construct a one-step update. Repeated application then requires the updated quantity to contain the full state needed for the next step. This is not true for the partial observations considered here.

GObO instead maps each new medium once to a structured kernel over temporal lag between the source and observation submanifolds. It does not evaluate a network at every kernel point, form a dense Green matrix, reconstruct the ambient state, or assume that the observation is a closed state. The same representation supports direct source convolution and the finite streaming state derived in Section 4.5. NGO is therefore the closest baseline because it also preserves source linearity and conditions its Green representation on PDE coefficients, while differing in its one-step treatment of time.

B Proofs and Derivations

This appendix collects the arguments that support the statements in Section 4. We follow the order of the method section, so that each argument can be read alongside the corresponding construction.

B.1 Green representations for initial, interior, and boundary data

For an autonomous linear parabolic problem, the initial condition and the interior forcing are propagated by the same solution semigroup. Boundary data enter through a boundary control operator and may therefore have a different kernel. Writing these contributions separately gives the general

representation

$$u_\alpha(x,t) = \underbrace{\int_\Omega G_\alpha(x,y,t)u_0(y)\,dy}_{\text{initial condition}} + \underbrace{\int_0^t \int_\Omega G_\alpha(x,y,t-s)F(y,s)\,dy\,ds}_{\text{interior forcing}} + \underbrace{\int_0^t \int_{\partial\Omega} G^\partial_\alpha(x,z,t-s)g(z,s)\,d\sigma(z)\,ds}_{\text{boundary input}}. \tag{13}$$

Thus the initial and forcing kernels coincide for the time-independent systems considered here. The boundary kernel G^∂_α depends on how the boundary data enter the PDE. For an inward Neumann flux, the weak load is

$$\ell_s(\phi) = \int_{\partial\Omega} g(z,s)\phi(z)\,d\sigma(z). \tag{14}$$

Relative to the homogeneous Neumann problem, its response kernel is the boundary trace of the same Green kernel, $G^\partial_\alpha(x,z,\tau) = G_\alpha(x,z,\tau)$, with the sign fixed by the convention that positive g injects flux into the domain. Other boundary conditions can produce a different boundary kernel. Keeping the superscript in equation 13 makes this distinction explicit.

We now restrict the two source terms. The area formula is

$$\int_{\Sigma_S} h(y)\,d\sigma(y) = \int_{U_S} h(X_S(\eta))\sqrt{\det(DX_S(\eta)^\top DX_S(\eta))}\,d\eta. \tag{15}$$

For an interior manifold source $F = f_\Sigma \delta_{\Sigma_S}$, the forcing term in equation 13 becomes

$$v^F_\alpha(\xi,t) = \int_0^t \int_{U_S} K^F_\alpha(\xi,\eta,t-s)f(\eta,s)\,d\eta\,ds, \tag{16}$$

$$K^F_\alpha(\xi,\eta,\tau) := G_\alpha(X_O(\xi),X_S(\eta),\tau)\,J_S(\eta). \tag{17}$$

For a Neumann flux supported on a boundary patch $\Sigma_S \subset \partial\Omega$, the boundary term instead becomes

$$v^\partial_\alpha(\xi,t) = \int_0^t \int_{U_S} K^\partial_\alpha(\xi,\eta,t-s)f(\eta,s)\,d\eta\,ds, \tag{18}$$

$$K^\partial_\alpha(\xi,\eta,\tau) := G^\partial_\alpha(X_O(\xi),X_S(\eta),\tau)\,J_S(\eta). \tag{19}$$

For a Neumann input, K^F_α and K^∂_α are restrictions of the same Green kernel to different source sets. A more general boundary operator may instead change the boundary kernel. In either case, the source-to-observation map has the same form. The spatial integration is over U_S, the kernel depends on the full ambient medium, and no observation Jacobian appears because the state is evaluated pointwise on Σ_O. This is the common form written in equation 7. If initial, interior, and boundary inputs are present together, their three responses simply add. The initial condition can also be viewed as an interior impulse at time zero.

The interior distribution and the weak Neumann load also share a local flux interpretation. For a codimension-one surface, consider a divergence-form diffusion equation written locally as

$$-\nabla\cdot q = f_\Sigma \delta_{\Sigma_S}, \qquad q = \kappa\nabla u. \tag{20}$$

If $\Sigma_S \subset \Omega$ separates two regions and n points from the minus side to the plus side, integration over a shrinking pillbox gives

$$[q\cdot n]_{\Sigma_S} := q^+\cdot n - q^-\cdot n = -f_\Sigma. \tag{21}$$

Thus the singular source prescribes a jump in normal flux across an interior interface, as in the classical immersed-interface construction (LeVeque & Li, 1994). If $\Sigma_S \subset \partial\Omega$, only the interior side remains and equation 21 becomes the Neumann condition

$$-\kappa\nabla u\cdot n = f_\Sigma, \tag{22}$$

when positive f_Σ denotes flux injected into the domain. The boundary and interior flux balances are therefore the one-sided and two-sided versions of the same local construction.

We now state the analytic assumptions on the ambient problem. The result below is the full version of Proposition 1 in the main text.

Proposition 1 (Existence of the Green's kernel)**.** *Let $\Omega \subset \mathbb{R}^d$ be bounded and Lipschitz. Suppose that $\kappa : \Omega \to \mathbb{R}^{d\times d}$ is bounded, measurable, symmetric, and uniformly elliptic, so that for some $0 < \kappa_- \le \kappa_+ < \infty$,*

$$\kappa_-|z|^2 \le z^\top \kappa(x) z \le \kappa_+|z|^2 \tag{23}$$

for almost every $x \in \Omega$ and every $z \in \mathbb{R}^d$. Let $b \in L^\infty(\Omega;\mathbb{R}^d)$. For the advection–diffusion systems considered here, assume additionally that b is divergence-free in Ω and has zero normal trace on $\partial\Omega$. Then each admissible α in equation 4 admits a Green's kernel G_α with a representative that is continuous on $\overline{\Omega} \times \overline{\Omega} \times (0,\infty)$.

Proof. For scalar divergence-form parabolic equations with bounded measurable coefficients, Choi & Kim (2013) construct the Neumann Green's function. The boundary regularity result of Nittka (2011) gives the continuous representative at positive times. The drift $b\cdot\nabla u$ is a bounded lower-order term. It is covered by the non-symmetric kernel theory of Daners (2000). □

The kernel not only exists but also satisfies an Aronson-type Gaussian bound (Aronson, 1968; Daners, 2000; Choi & Kim, 2013; Ouhabaz, 2009). For every finite $T > 0$, there are constants $C_G, c > 0$, which may depend on T, such that

$$|G_\alpha(x,y,\tau)| \le C_G \tau^{-d/2} \exp\left(-c\frac{|x-y|^2}{\tau}\right) \tag{24}$$

for all $x, y \in \overline{\Omega}$ and $0 < \tau \le T$. Integrating over y and enlarging the domain to $\mathbb{R}^d$ gives

$$\sup_{x\in\overline{\Omega}} \int_\Omega |G_\alpha(x,y,\tau)|\, dy \le C_G \left(\frac{\pi}{c}\right)^{d/2} =: C_T. \tag{25}$$

The factor $\tau^{-d/2}$ is therefore canceled by integration over the full d-dimensional domain. This makes equation 3 finite, for example, whenever $F \in L^1(0,T;L^\infty(\Omega))$. Restricting the kernel to a $(d-1)$-dimensional source manifold leaves the integrable singularity $\tau^{-1/2}$ instead. The following result gives the corresponding bound uniformly over all observation points. It is the full version of Proposition 2 in the main text.

Proposition 2 (Integrability of the restricted kernel)**.** *Assume the setting of Proposition 1, with $d \ge 2$. Let $U_S \subset \mathbb{R}^{d-1}$ be compact and let $X_S : U_S \to \Sigma_S \subset \overline{\Omega}$ be bi-Lipschitz with constant $L \ge 1$, that is,*

$$L^{-1}|\eta - \eta'| \le |X_S(\eta) - X_S(\eta')| \le L|\eta - \eta'| \qquad \text{for all } \eta, \eta' \in U_S, \tag{26}$$

and let K_α include the surface Jacobian J_S as in equation 7. Then for every finite $T \ge 0$ there is a constant C_T, independent of the observation point, such that for every $x = X_O(\xi) \in \overline{\Omega}$

$$\int_{U_S} |K_\alpha(\xi,\eta,\tau)|\, d\eta = \int_{\Sigma_S} |G_\alpha(x,y,\tau)|\, d\sigma(y) \le C_T \tau^{-1/2}, \qquad 0 < \tau \le T. \tag{27}$$

Consequently,

$$\sup_{\xi\in U_O} \int_0^T \int_{U_S} |K_\alpha(\xi,\eta,\tau)|\, d\eta\, d\tau \le 2C_T\sqrt{T} < \infty. \tag{28}$$

Proof. The equality in equation 27 is the area formula equation 15. Fix $x \in \overline{\Omega}$. Since Σ_S is compact, there is $\eta_0 \in U_S$ with $|x - X_S(\eta_0)| = \operatorname{dist}(x, \Sigma_S)$. For every $\eta \in U_S$ the triangle inequality and the choice of η_0 give

$$|X_S(\eta) - X_S(\eta_0)| \le |X_S(\eta) - x| + |x - X_S(\eta_0)| \le 2\,|X_S(\eta) - x|,$$

and the lower bound in equation 26 turns this into

$$|X_S(\eta) - x| \ge \frac{1}{2L}\,|\eta - \eta_0| \qquad \text{for all } \eta \in U_S. \tag{29}$$

If $x \in \Sigma_S$, then $x = X_S(\eta_0)$ and the factor 2 can be dropped.

The upper Lipschitz bound implies that DX_S exists almost everywhere and has operator norm at most L. Therefore,

$$J_S(\eta) = \sqrt{\det\big(DX_S(\eta)^\top DX_S(\eta)\big)} \le L^{d-1}. \tag{30}$$

The Gaussian bound equation 24 holds pointwise on $\overline{\Omega} \times \overline{\Omega}$, so it may be evaluated at $y = X_S(\eta)$. Combining it with equation 29 gives

$$|G_\alpha(x, X_S(\eta), \tau)| \le C_G \tau^{-d/2} \exp\Big(- \frac{c\,|\eta - \eta_0|^2}{4L^2\tau}\Big).$$

Multiplying by equation 30, integrating over U_S, and enlarging the domain of integration to $\mathbb{R}^{d-1}$ gives

$$\begin{aligned}
\int_{U_S} |K_\alpha(\xi, \eta, \tau)|\, d\eta &\le C_G L^{d-1} \tau^{-d/2} \int_{\mathbb{R}^{d-1}} \exp\Big(- \frac{c\,|\eta - \eta_0|^2}{4L^2\tau}\Big)\, d\eta \\
&= C_G L^{d-1} \tau^{-d/2} \Big(\frac{4\pi L^2 \tau}{c}\Big)^{(d-1)/2} \\
&= C_G L^{d-1} \Big(\frac{4\pi L^2}{c}\Big)^{(d-1)/2} \tau^{-1/2}.
\end{aligned} \tag{31}$$

Here we used the Gaussian integral $\int_{\mathbb{R}^n} e^{-|z|^2/(4s)}\, dz = (4\pi s)^{n/2}$ with $n = d-1$ and $s = L^2\tau/c$. Defining $C_T = C_G L^{d-1}(4\pi L^2/c)^{(d-1)/2}$ proves equation 27. This constant does not depend on x.

It remains to integrate the bound in time. Uniformly in ξ,

$$\int_0^T \int_{U_S} |K_\alpha(\xi, \eta, \tau)|\, d\eta\, d\tau \le C_T \int_0^T \tau^{-1/2}\, d\tau = 2C_T\sqrt{T},$$

which is equation 28. □

When the manifolds are separated, the exponential decay in the Gaussian bound removes even the remaining short-time singularity.

Remark 1 (Separated source and observation manifolds). If $\operatorname{dist}(\Sigma_S, \Sigma_O) = \delta > 0$, then for $x \in \Sigma_O$, $y \in \Sigma_S$, and $0 < \tau \le T$, equation 24 gives

$$|G_\alpha(x, y, \tau)| \le C_G \tau^{-d/2} e^{-c\delta^2/\tau} \le C_G \Big(\frac{d}{2ec\delta^2}\Big)^{d/2}, \tag{32}$$

where the second inequality is $s^{d/2} e^{-s} \le (d/2e)^{d/2}$ with $s = c\delta^2/\tau$. Hence

$$\sup_{0<\tau\le T} \sup_{\xi \in U_O} \int_{U_S} |K_\alpha(\xi, \eta, \tau)|\, d\eta \le C_G |\Sigma_S| \Big(\frac{d}{2ec\delta^2}\Big)^{d/2} < \infty.$$

The short-time singularity of Proposition 2 therefore appears only where the observation manifold touches the source manifold.

B.2 Approximation of the restricted kernel

For heat, the abstract descriptor is $\alpha = \kappa$, and the spatial operator A_κ is the variational Neumann realization of $-\nabla \cdot (\kappa \nabla)$. Its eigenpairs solve

$$-\nabla \cdot \big(\kappa(x) \nabla e_k(x)\big) = \lambda_k e_k(x), \qquad (\kappa \nabla e_k) \cdot n = 0 \quad \text{on } \partial\Omega. \tag{33}$$

For symmetric uniformly elliptic κ on a bounded Lipschitz domain, this operator is nonnegative and self-adjoint, and its resolvent is compact. The spectral theorem therefore gives an orthonormal basis $\{e_k\}_{k \ge 1}$ of $L^2(\Omega)$, with $0 = \lambda_1 \le \lambda_2 \le \cdots$ and $\lambda_k \to \infty$. For every $\tau > 0$, the corresponding heat kernel has the expansion

$$G_\kappa(x, y, \tau) = \sum_{k=1}^{\infty} e^{-\lambda_k \tau} e_k(x) e_k(y), \tag{34}$$

with convergence in $L^2(\Omega \times \Omega)$. These standard spectral facts for elliptic Neumann operators and their heat kernels are given, for example, by Davies (1989) and Ouhabaz (2009).

Before restricting this series to the manifolds, we strengthen its convergence. The Gaussian bound implies that the eigenfunctions grow at most polynomially in λ_k and that the eigenvalues grow at least like $k^{2/d}$. The exponential factor then gives uniform convergence for every time lag bounded away from zero. This makes the pointwise restriction well defined and reduces the truncation estimate to a tail sum. The following is the full version of Theorem 1 in the main text. It gives a uniform pointwise bound, from which the stated L^2 bound follows.

Theorem 1 (Approximation of the restricted heat kernel)**.** *Assume the setting of Proposition 1 with $b = 0$. Let $X_S : U_S \to \Sigma_S$ and $X_O : U_O \to \Sigma_O$ be bi-Lipschitz C^1 charts on compact parameter sets, with X_S satisfying the assumptions of Proposition 2, and let ψ_k, φ_k, and $K_{\alpha,M}$ be defined as in equation 9. Then the eigenfunctions e_k are continuous on $\overline{\Omega}$, and for every $\tau > 0$ the series in equation 9 converges uniformly on $U_O \times U_S$. Moreover, there is a constant $c > 0$ depending only on Ω, κ_-, κ_+, and the Gaussian constants in equation 24. For every $\tau_0 > 0$, there is a constant C_{τ_0} such that, for all $M \geq 1$,*

$$\sup_{\tau \geq \tau_0} \sup_{\xi \in U_O,\, \eta \in U_S} \frac{|K_\alpha(\xi, \eta, \tau) - K_{\alpha,M}(\xi, \eta, \tau)|}{J_S(\eta)} \leq C_{\tau_0} e^{-c\tau_0 M^{2/d}}. \tag{35}$$

In particular,

$$\sup_{\tau \geq \tau_0} \|K_\alpha(\cdot, \cdot, \tau) - K_{\alpha,M}(\cdot, \cdot, \tau)\|_{L^2(U_O \times U_S)} \leq |U_O|^{1/2} \|J_S\|_{L^2(U_S)} C_{\tau_0} e^{-c\tau_0 M^{2/d}}. \tag{10}$$

Proof. Let C_G and c_G denote the constants in equation 24 for $0 < t \leq 1$. We use the variational Neumann operator A_κ from equation 33, with orthonormal eigenpairs (λ_k, e_k) ordered as $0 = \lambda_1 < \lambda_2 \leq \cdots$.

We first use the Gaussian bound to control the heat operator from L^2 to L^∞. For $f \in L^2(\Omega)$, $x \in \overline{\Omega}$, and $0 < t \leq 1$, the Cauchy–Schwarz inequality gives

$$\begin{aligned} |(e^{-tA_\kappa} f)(x)| &= \left| \int_\Omega G_\alpha(x, y, t) f(y)\, dy \right| \\ &\leq C_G t^{-d/2} \left(\int_{\mathbb{R}^d} e^{-2c_G |x-y|^2/t}\, dy \right)^{1/2} \|f\|_{L^2(\Omega)} \\ &= C_G \left(\frac{\pi}{2c_G} \right)^{d/4} t^{-d/4} \|f\|_{L^2(\Omega)} =: C_1 t^{-d/4} \|f\|_{L^2(\Omega)}. \end{aligned} \tag{36}$$

The kernel is bounded and continuous on $\overline{\Omega} \times \overline{\Omega}$ for fixed $t > 0$. Dominated convergence therefore shows that $e^{-tA_\kappa} f$ is continuous on $\overline{\Omega}$.

Since $e^{-tA_\kappa} e_k = e^{-\lambda_k t} e_k$, equation equation 36 yields

$$\|e_k\|_{L^\infty(\Omega)} \leq C_1 t^{-d/4} e^{\lambda_k t}, \qquad 0 < t \leq 1.$$

It also shows that each e_k has a continuous representative on $\overline{\Omega}$. When $\lambda_k \geq 1$, we choose $t = 1/\lambda_k$. Otherwise, we choose $t = 1$. In both cases,

$$\|e_k\|_{L^\infty(\Omega)} \leq e C_1 (1 + \lambda_k)^{d/4}. \tag{37}$$

Thus, pointwise restriction of an eigenfunction introduces only polynomial growth in λ_k.

The Gaussian bound also gives the required growth of the eigenvalues. The squared Hilbert–Schmidt norm of $e^{-(t/2)A_\kappa}$ equals the squared $L^2(\Omega \times \Omega)$ norm of its kernel. Hence, for $0 < t \leq 1$,

$$\begin{aligned} \sum_{j=1}^{\infty} e^{-\lambda_j t} &= \int_\Omega \int_\Omega |G_\alpha(x, y, t/2)|^2\, dy\, dx \\ &\leq C_G^2 \left(\frac{t}{2} \right)^{-d} \int_\Omega \int_{\mathbb{R}^d} e^{-4c_G |x-y|^2/t}\, dy\, dx \\ &= C_G^2 |\Omega| 2^d \left(\frac{\pi}{4c_G} \right)^{d/2} t^{-d/2} =: C_3 t^{-d/2}. \end{aligned} \tag{38}$$

Because the eigenvalues are nondecreasing, the sum on the left is at least $ke^{-\lambda_k t}$. If $\lambda_k \geq 1$, choosing $t = 1/\lambda_k$ gives

$$ke^{-1} \leq C_3 \lambda_k^{d/2}, \qquad \lambda_k \geq (eC_3)^{-2/d} k^{2/d}.$$

Only finitely many indices $k \geq 2$ satisfy $\lambda_k < 1$. Decreasing the constant to include these indices gives some $c_0 > 0$ such that

$$\lambda_k \geq c_0 k^{2/d} \qquad \text{for every } k \geq 2. \tag{39}$$

We now combine these two estimates. Fix $\tau_0 > 0$ and let $\tau \geq \tau_0$. Equation equation 37 implies

$$\begin{aligned} e^{-\lambda_k \tau} |e_k(x) e_k(y)| &\leq e^2 C_1^2 (1 + \lambda_k)^{d/2} e^{-\lambda_k \tau_0} \\ &\leq C'_{\tau_0} e^{-\lambda_k \tau_0/2}, \end{aligned} \tag{40}$$

uniformly for $x, y \in \overline{\Omega}$, where

$$C'_{\tau_0} := e^2 C_1^2 \sup_{\lambda \geq 0} (1 + \lambda)^{d/2} e^{-\lambda \tau_0/2} < \infty.$$

Together with equation 39, this shows that the series

$$S(x, y, \tau) := \sum_{k=1}^{\infty} e^{-\lambda_k \tau} e_k(x) e_k(y)$$

converges absolutely and uniformly on $\overline{\Omega} \times \overline{\Omega} \times [\tau_0, \infty)$. Its sum is therefore continuous. By equation 34, the same series converges to $G_\alpha(\cdot, \cdot, \tau)$ in $L^2(\Omega \times \Omega)$. Since both functions are continuous and agree almost everywhere, they agree everywhere. As $\tau_0 > 0$ was arbitrary,

$$G_\alpha(x, y, \tau) = S(x, y, \tau) \qquad \text{for all } x, y \in \overline{\Omega} \text{ and } \tau > 0. \tag{41}$$

Evaluating equation 41 at $x = X_O(\xi)$ and $y = X_S(\eta)$, then multiplying by $J_S(\eta)$, gives

$$K_\alpha(\xi, \eta, \tau) = \sum_{k=1}^{\infty} e^{-\lambda_k \tau} \underbrace{e_k(X_O(\xi))}_{\psi_k(\xi)} \underbrace{e_k(X_S(\eta)) J_S(\eta)}_{\varphi_k(\eta)}, \tag{42}$$

with uniform convergence on $U_O \times U_S$.

It remains to bound the omitted terms. Every index in the tail satisfies $k \geq 2$ when $M \geq 1$. Using equation 40 and equation 39, with $a = c_0 \tau_0 / 2$, gives

$$\begin{aligned} \frac{|K_\alpha(\xi, \eta, \tau) - K_{\alpha, M}(\xi, \eta, \tau)|}{J_S(\eta)} &\leq \sum_{k > M} e^{-\lambda_k \tau} |e_k(X_O(\xi)) e_k(X_S(\eta))| \\ &\leq C'_{\tau_0} \sum_{k > M} e^{-a k^{2/d}}. \end{aligned} \tag{43}$$

For $k > M$, we have $k^{2/d} \geq M^{2/d}$ and therefore

$$e^{-a k^{2/d}} \leq e^{-a M^{2/d}/2} e^{-a k^{2/d}/2}.$$

Summing this inequality yields

$$\sum_{k > M} e^{-a k^{2/d}} \leq e^{-a M^{2/d}/2} \sum_{k=1}^{\infty} e^{-a k^{2/d}/2} =: S_a e^{-a M^{2/d}/2}, \qquad S_a < \infty. \tag{44}$$

Combining these estimates proves equation 35 with $C_{\tau_0} = C'_{\tau_0} S_a$ and $c = c_0/4$. In particular, c is independent of τ_0. Finally, multiplying the pointwise estimate by $J_S(\eta)$ and taking its $L^2(U_O \times U_S)$ norm gives equation 10. □

B.3 Optimization with additional search rank

We provide some intuition for the effect of search rank observed in Table 26 by examining the optimization landscape of the factorized objective. We treat the factors as free variables and hold the rates fixed. In the conditional model, however, the factors are outputs of a neural network, so the connection to our experiments is only qualitative.

For fixed rates, the observations depend linearly on the coefficient matrices $A = (A_1, \dots, A_M)$. The squared loss

$$\mathcal{L}(A) = \sum_i \|\mathcal{P}_i(A) - v_i\|_2^2, \qquad \mathcal{P}_i \text{ linear}, \tag{45}$$

is therefore a convex quadratic in A. After writing $A_k = U_k V_k^\top$, training instead minimizes

$$f(U, V) = \mathcal{L}\left(U_1 V_1^\top, \dots, U_M V_M^\top\right), \tag{46}$$

which is generally nonconvex in $U_k \in \mathbb{R}^{D_O \times L}$ and $V_k \in \mathbb{R}^{D_S \times L}$. In coefficient space, this resembles asymmetric low-rank matrix sensing with one measurement block per temporal mode. Our instance is poorly conditioned. In particular, the temporal design matrix in Appendix E.4.3 has condition number 5.8×10^{14}, and the objective has no balancing regularizer.

Let $G_k := \nabla_{A_k} \mathcal{L}$. The first-order stationary conditions for the factored objective are

$$G_k V_k = 0, \qquad G_k^\top U_k = 0 \qquad \text{for every } k. \tag{47}$$

These conditions are weaker than $G_k = 0$. For example, the conditions for a given slot hold when $U_k = V_k = 0$, regardless of its coefficient gradient. The factored objective can therefore be stationary even when the corresponding coefficient matrices do not minimize equation 127 (Ha et al., 2020, eq. (8)).

Rank deficiency gives a useful connection between the two objectives. For a twice differentiable function g and a factorization $X = AB^\top$ with L columns, every second-order stationary point of $g(AB^\top)$ satisfying $\operatorname{rank}(AB^\top) < L$ also satisfies $\nabla g(AB^\top) = 0$ (Ha et al., 2020, Theorem 1). Related arguments appear for Burer–Monteiro factorizations and regularized factorization models (Burer & Monteiro, 2005; Bhojanapalli et al., 2018; Haeffele & Vidal, 2015). Informally, unused factor directions allow the second-order behavior of the factored objective to test directions that are hidden from its first-order conditions.

Applying this results to one rate slot while holding the others fixed, a second-order stationary point with $\operatorname{rank}(A_k) < L$ must have $G_k = 0$. If every slot is rank deficient, then $\nabla_A \mathcal{L} = 0$, and convexity makes A a global minimizer of equation 127. At $L = 1$, this mechanism applies only to a zero rank slot. A nonzero rank-one factorization has no unused column. Additional columns make such unused directions possible, although there is no guarantee that our optimizer reaches a rank-deficient second-order stationary point.

The optimization landscape also depends on properties of the measurement operator. Under suitable restricted-isometry assumptions, factored matrix-sensing problems have no spurious local minima and their saddle points are strict (Ge et al., 2017). Counterexamples show that weaker isometry conditions can still permit spurious local minima (Zhang et al., 2018). Our temporal design is far from the well-conditioned setting assumed by the favorable results, so neither conclusion directly describes our loss. Optimization may also be slow without convergence to a spurious minimum. Gradient descent can be sensitive to ill-conditioning in low-rank estimation (Tong et al., 2021). In related factored linear models, singular directions may be learned sequentially with plateaus between them (Saxe et al., 2014), and even strict saddle points can take a long time to escape (Du et al., 2017). A finite training run cannot distinguish these plateaus from convergence to a poor local minimum.

Existing theory nevertheless gives some support for increasing the search rank. For positive-semidefinite Burer–Monteiro factorizations of smooth, strongly convex objectives, sufficient rank overparameterization makes every second-order stationary point globally optimal. Without overparameterization, the corresponding guarantee requires an objective with a condition number below three (Zhang, 2025). These results do not cover our rectangular, multi-rate factorization, and our loss is only weakly convex in unidentifiable directions. They therefore provide context rather than a guarantee. Similarly, gradient descent with random initialization avoids strict saddle points under suitable regularity and step-size assumptions (Lee et al., 2016), while rank deficiency can certify global optimality at a second-order stationary point (Zhang et al., 2023). On the other hand,

plain gradient descent may converge only sublinearly in the overparameterized setting (Zhang et al., 2023).

The results in Table 26 are consistent with the theoretical motivation for increasing search rank. At $L = 1$, the attained loss remains more than an order of magnitude above the convex coefficient-space optimum, whereas for $L \geq 8$, it approaches that optimum to within approximately one percent. Although increasing L affects both the set of representable coefficient matrices and the optimization landscape, the observed improvement provides empirical support for the search rank used in the remaining experiments. A theoretical characterization of the observed dependence on search rank is left to future work.

B.4 Exact streaming evaluation

The result below restates Proposition 3 from the main text. The proof separates the accumulated history from the final time interval.

Proposition 3 (Exact streaming evaluation). *Let $t_i = i\Delta$ be a uniform time grid and define*

$$\widehat{f}(t) := \int_{U_S} \mathbf{\Phi}_S(\eta) f(\eta, t)\, d\eta, \tag{48}$$

$$g_k(t) := \int_0^t e^{-\lambda_k (t-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds, \qquad k = 1, \ldots, M. \tag{49}$$

Suppose that the source is piecewise constant in time, with projected value $\widehat{f^i}$ on $[t_i, t_{i+1})$. Then, writing $g_k^i := g_k(t_i)$,

$$g_k^{i+1} = e^{-\lambda_k \Delta} g_k^i + \left(\int_0^\Delta e^{-\lambda_k s}\, ds \right) V_k(\alpha)^\top \widehat{f^i}. \tag{12}$$

The observation at t_i is

$$v_\alpha(\xi, t_i) = \mathbf{\Phi}_O(\xi)^\top \sum_{k=1}^{M} U_k(\alpha) g_k^i. \tag{50}$$

Thus the recurrence and readout are exactly equal to batch evaluation of the kernel for a piecewise-constant source.

Proof. We begin from the definition of the state at t_{i+1} and split its time integral at t_i:

$$\begin{aligned} g_k(t_{i+1}) &= \int_0^{t_{i+1}} e^{-\lambda_k (t_{i+1}-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \\ &= \int_0^{t_i} e^{-\lambda_k (t_{i+1}-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds + \int_{t_i}^{t_{i+1}} e^{-\lambda_k (t_{i+1}-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds. \end{aligned} \tag{51}$$

Since $t_{i+1} = t_i + \Delta$, the first integral becomes

$$\begin{aligned} &\int_0^{t_i} e^{-\lambda_k (t_{i+1}-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \\ &\quad = \int_0^{t_i} e^{-\lambda_k \Delta} e^{-\lambda_k (t_i - s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \\ &\quad = e^{-\lambda_k \Delta} \int_0^{t_i} e^{-\lambda_k (t_i - s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \\ &\quad = e^{-\lambda_k \Delta} g_k(t_i) = e^{-\lambda_k \Delta} g_k^i. \end{aligned} \tag{52}$$

On the interval $[t_i, t_{i+1})$, piecewise constancy gives $\widehat{f}(s) = \widehat{f}^i$. With the change of variables $r = t_{i+1} - s$, the second integral therefore becomes

$$\begin{aligned}
&\int_{t_i}^{t_{i+1}} e^{-\lambda_k(t_{i+1}-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \\
&\quad = V_k(\alpha)^\top \widehat{f}^i \int_{t_i}^{t_{i+1}} e^{-\lambda_k(t_{i+1}-s)}\, ds \\
&\quad = V_k(\alpha)^\top \widehat{f}^i \int_{\Delta}^{0} e^{-\lambda_k r}(-dr) \\
&\quad = \left(\int_0^{\Delta} e^{-\lambda_k s}\, ds\right) V_k(\alpha)^\top \widehat{f}^i. \qquad (53)
\end{aligned}$$

Combining the two parts gives

$$g_k^{i+1} = e^{-\lambda_k \Delta} g_k^i + \left(\int_0^{\Delta} e^{-\lambda_k s}\, ds\right) V_k(\alpha)^\top \widehat{f}^i, \qquad (54)$$

which is equation 12.

It remains to verify the readout. Substituting the kernel model equation 11 into the source-to-observation integral gives

$$\begin{aligned}
v_\alpha(\xi, t_i) &= \int_0^{t_i} \int_{U_S} K_{\theta,\alpha}(\xi, \eta, t_i - s) f(\eta, s)\, d\eta\, ds \\
&= \int_0^{t_i} \int_{U_S} \mathbf{\Phi}_O(\xi)^\top \sum_{k=1}^{M} e^{-\lambda_k(t_i-s)} U_k(\alpha) V_k(\alpha)^\top \mathbf{\Phi}_S(\eta) f(\eta, s)\, d\eta\, ds \\
&= \mathbf{\Phi}_O(\xi)^\top \sum_{k=1}^{M} U_k(\alpha) \int_0^{t_i} e^{-\lambda_k(t_i-s)} V_k(\alpha)^\top \left(\int_{U_S} \mathbf{\Phi}_S(\eta) f(\eta, s)\, d\eta\right) ds \\
&= \mathbf{\Phi}_O(\xi)^\top \sum_{k=1}^{M} U_k(\alpha) \int_0^{t_i} e^{-\lambda_k(t_i-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \\
&= \mathbf{\Phi}_O(\xi)^\top \sum_{k=1}^{M} U_k(\alpha) g_k^i. \qquad (55)
\end{aligned}$$

This is the stated observation formula. □

The next result restates Proposition 4 from the main text.

Proposition 4 (Stability of the streaming state)**.** *Let* $\rho_k = e^{-\lambda_k \Delta}$. *If* $\mathrm{Re}(\lambda_k) \geq 0$, *perturbations evolving under the same source satisfy*

$$\|\delta g_k^{i+1}\| \leq |\rho_k|\, \|\delta g_k^i\| \leq \|\delta g_k^i\|. \qquad (56)$$

If $\mathrm{Re}(\lambda_k) > 0$ *and* $\sup_i \|V_k(\alpha)^\top \widehat{f}^i\| < \infty$, *then the states* $\{g_k^i\}_{i \geq 0}$ *are uniformly bounded.*

Proof. Let g_k^i and $\widetilde{g}_k^i$ be two states driven by the same source, and set $\delta g_k^i := \widetilde{g}_k^i - g_k^i$. Their updates are

$$\begin{aligned}
g_k^{i+1} &= \rho_k g_k^i + \left(\int_0^{\Delta} e^{-\lambda_k s}\, ds\right) V_k(\alpha)^\top \widehat{f}^i, \\
\widetilde{g}_k^{i+1} &= \rho_k \widetilde{g}_k^i + \left(\int_0^{\Delta} e^{-\lambda_k s}\, ds\right) V_k(\alpha)^\top \widehat{f}^i. \qquad (57)
\end{aligned}$$

Subtracting the first equation from the second cancels the common input and gives

$$\begin{aligned}\delta g_k^{i+1} &= \widetilde{g}_k^{i+1} - g_k^{i+1} \\ &= \rho_k \widetilde{g}_k^{i} - \rho_k g_k^{i} \\ &= \rho_k \big(\widetilde{g}_k^{i} - g_k^{i}\big) = \rho_k \delta g_k^{i}. \end{aligned} \tag{58}$$

Moreover,

$$|\rho_k| = |e^{-\lambda_k \Delta}| = e^{-\operatorname{Re}(\lambda_k)\Delta} \le 1, \tag{59}$$

because $\operatorname{Re}(\lambda_k) \ge 0$. It follows that

$$\|\delta g_k^{i+1}\| = \|\rho_k \delta g_k^{i}\| = |\rho_k| \, \|\delta g_k^{i}\| \le \|\delta g_k^{i}\|. \tag{60}$$

For the second claim, let $a_k := \operatorname{Re}(\lambda_k) > 0$ and $B_k := \sup_i \|V_k(\alpha)^\top \widehat{f^i}\|$. Since the source is piecewise constant, the same bound holds at almost every time. The definition of the state then yields

$$\begin{aligned}\|g_k(t)\| &= \left\| \int_0^t e^{-\lambda_k (t-s)} V_k(\alpha)^\top \widehat{f}(s)\, ds \right\| \\ &\le \int_0^t |e^{-\lambda_k (t-s)}| \|V_k(\alpha)^\top \widehat{f}(s)\| \, ds \\ &\le B_k \int_0^t e^{-a_k (t-s)} \, ds \\ &= \frac{B_k}{a_k} \left(1 - e^{-a_k t}\right) \le \frac{B_k}{a_k}. \end{aligned} \tag{61}$$

Setting $t = t_i$ proves that the sequence $\{g_k^i\}_{i \ge 0}$ is uniformly bounded. □

B.5 Extensions to mild nonlinearities

This section develops the two nonlinear extensions summarized in Section 4.6. The first treats a nonlinear effect as a feedback source and alternates it with the learned linear propagation. The second uses a Kirchhoff transformation to rewrite a class of nonlinear heat equations in terms of a transformed field. Both constructions reuse the learned linear response. The reference medium is fixed throughout, so we omit the common argument α_0 from the factors and states below.

B.5.1 Radiative boundary feedback

Radiative heat loss provides a source-like nonlinearity whose strength grows rapidly with temperature and can therefore alter the peak response even when conduction is well described by a linear model. Given an applied flux q, we write the effective source as

$$f = q + \mathcal{N}(v_\alpha), \qquad \mathcal{N}(v_\alpha) := -\sigma\big((v_\alpha + u_\infty)^4 - u_\infty^4\big), \tag{62}$$

where u_∞ is the ambient temperature and $\mathcal{N}$ is the Stefan–Boltzmann radiative flux (Necati et al., 1993). If the source and observation manifolds coincide, this flux can be evaluated from the predicted temperature and fed back as a source.

In the spirit of operator splitting (Strang, 1968), we alternate linear propagation with this nonlinear source update. We use a semi-implicit step based on a linearization at the previous prediction (Ascher et al., 1997). Evaluating the full convolution at each step would revisit the entire source history; the streaming representation instead reduces the update to a fixed-cost state recurrence and a small solve in coefficient space.

For the coefficient-space construction, we assume that the coincident source and observation manifolds use the same orthonormal cosine basis. We retain the S/O subscripts to show the direction of each projection.

Let

$$w^i := \sum_{k=1}^{M} U_k g_k^i, \qquad v^i(\xi) := \mathbf{\Phi}_O(\xi)^\top w^i. \tag{63}$$

We write the projected applied flux and radiative flux as

$$\widehat{q}^{\,i} := \int_{U_S} \mathbf{\Phi}_S(\xi) q(\xi, t_i)\, d\xi, \tag{64}$$

$$\widehat{\mathcal{N}}(w) := \int_{U_S} \mathbf{\Phi}_S(\xi) \mathcal{N}\big(\mathbf{\Phi}_O(\xi)^\top w\big)\, d\xi. \tag{65}$$

At time t_i, set $\widehat{\mathcal{N}}^{\,i} := \widehat{\mathcal{N}}(w^i)$. Its derivative with respect to w is

$$J^i := \int_{U_S} \mathbf{\Phi}_S(\xi) \mathcal{N}'(v^i(\xi)) \mathbf{\Phi}_O(\xi)^\top\, d\xi. \tag{66}$$

To include the unknown radiative flux at t_{i+1}, we linearly interpolate the source over $[t_i, t_{i+1}]$. With $\nu_k := \lambda_k \Delta$, define

$$\gamma_0(\nu) := \frac{1 - (1+\nu)e^{-\nu}}{\nu^2}, \qquad \gamma_1(\nu) := \frac{\nu - 1 + e^{-\nu}}{\nu^2}, \tag{67}$$

$$B_\Delta := \sum_{k=1}^{M} \gamma_1(\nu_k) U_k V_k^\top, \tag{68}$$

where $\gamma_0(0) = \gamma_1(0) = 1/2$ by continuity. All terms that are known at the beginning of the step are collected in

$$\bar{g}_k^{i+1} := e^{-\nu_k} g_k^i + \Delta \gamma_0(\nu_k) V_k^\top \left(\widehat{q}^{\,i} + \widehat{\mathcal{N}}^{\,i}\right) + \Delta \gamma_1(\nu_k) V_k^\top \widehat{q}^{\,i+1}, \tag{69}$$

$$\bar{w}^{i+1} := \sum_{k=1}^{M} U_k \bar{g}_k^{i+1}. \tag{70}$$

Proposition 5 (Semi-implicit radiative update)**.** *A first-order expansion of the radiative flux at v^i gives the coefficient update*

$$\left(I - \Delta B_\Delta J^i\right) w^{i+1} = \bar{w}^{i+1} + \Delta B_\Delta \left(\widehat{\mathcal{N}}^{\,i} - J^i w^i\right). \tag{71}$$

Thus each time step requires one linear solve in the observation-basis coefficients.

Proof. For $0 \le r \le \Delta$, approximate the projected source by

$$\widehat{f}(t_i + r) \approx \left(1 - \frac{r}{\Delta}\right) \widehat{f}^{\,i} + \frac{r}{\Delta} \widehat{f}^{\,i+1}. \tag{72}$$

Splitting the state integral at t_i as in Proposition 3 then gives

$$\begin{aligned} g_k^{i+1} &\approx e^{-\nu_k} g_k^i + V_k^\top \widehat{f}^{\,i} \int_0^\Delta e^{-\lambda_k(\Delta - r)} \left(1 - \frac{r}{\Delta}\right) dr \\ &\quad + V_k^\top \widehat{f}^{\,i+1} \int_0^\Delta e^{-\lambda_k(\Delta - r)} \frac{r}{\Delta}\, dr \\ &= e^{-\nu_k} g_k^i + \Delta \gamma_0(\nu_k) V_k^\top \widehat{f}^{\,i} + \Delta \gamma_1(\nu_k) V_k^\top \widehat{f}^{\,i+1}. \end{aligned} \tag{73}$$

For radiative feedback,

$$\widehat{f}^{\,i} = \widehat{q}^{\,i} + \widehat{\mathcal{N}}^{\,i}, \qquad \widehat{f}^{\,i+1} = \widehat{q}^{\,i+1} + \widehat{\mathcal{N}}(w^{i+1}). \tag{74}$$

Using the definition of $\bar{g}_k^{i+1}$ therefore yields

$$g_k^{i+1} = \bar{g}_k^{i+1} + \Delta \gamma_1(\nu_k) V_k^\top \widehat{\mathcal{N}}(w^{i+1}). \tag{75}$$

After multiplication by U_k and summation over k, this becomes

$$w^{i+1} = \bar{w}^{i+1} + \Delta B_\Delta \widehat{\mathcal{N}}(w^{i+1}). \tag{76}$$

The first-order expansion at w^i is

$$\begin{aligned} \widehat{\mathcal{N}}(w^{i+1}) &\approx \widehat{\mathcal{N}}(w^i) + J^i(w^{i+1} - w^i) \\ &= \widehat{\mathcal{N}}^{\,i} + J^i(w^{i+1} - w^i). \end{aligned} \tag{77}$$

Substituting this expression into equation 76 and collecting the terms containing w^{i+1} gives equation 71. □

This feedback construction uses the linearity of the kernel in the source and could also be evaluated from the full convolution. The streaming form avoids repeating that convolution at every nonlinear step.

B.5.2 KIRCHHOFF TRANSFORMATION

The second construction concerns state-dependent medium properties, as arise when a material's conductivity changes with temperature. Given a reference medium α_0, write

$$\alpha_u(x,t) = \mathcal{A}\big(\alpha_0(x), u(x,t)\big), \qquad \alpha_0 : \Omega \to \mathbb{R}^p, \quad \mathcal{A} : \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}^p. \tag{78}$$

If α_u were known throughout the volume, we could evaluate a newly conditioned kernel K_{α_u}. In our setting, however, the constitutive law $\mathcal{A}$ is known while u is predicted only on the observation manifold, so the updated volumetric medium cannot generally be recovered.

For heat conduction, useful special cases arise when the spatial and temperature dependence separate. Let $\alpha_0(x)$ be the spatial reference conductivity and let $\widetilde{k}(u)$ and $c(u)$ be positive scalar conductivity and heat-capacity factors. We consider

$$\begin{aligned} \partial_t u &= \frac{1}{c(u)} \nabla \cdot \Big(\alpha_0(x) \widetilde{k}(u) \nabla u \Big), \qquad \alpha_0 : \Omega \to \mathbb{R}_{>0}, \quad \widetilde{k}, c : \mathbb{R} \to \mathbb{R}_{>0}, \\ r(u) &:= \frac{\widetilde{k}(u)}{c(u)}, \qquad \text{(A)} \quad r(u) \equiv 1, \qquad \text{(B)} \quad r(u) = \widetilde{k}(u). \end{aligned} \tag{79}$$

Setting A describes conductivity and heat capacity that vary by the same factor, so heat spreads at a temperature-independent rate (Lesnic et al., 1995). Setting B fixes $c \equiv 1$ and varies only the conductivity, so the propagation speed depends on the local temperature (Fabre & Hristov, 2017).

Both cases can be treated through the classical Kirchhoff construction (Carslaw, 1906; Necati et al., 1993). Define

$$\Theta_{\widetilde{k}}(r) := \int_0^r \widetilde{k}(s)\, ds, \qquad \theta := \Theta_{\widetilde{k}}(u). \tag{80}$$

It is strictly increasing and can therefore be inverted pointwise. This change of variables itself does not depend on the streaming formulation.

Proposition 6 (Kirchhoff transformation)**.** *The transformed field satisfies*

$$\partial_t \theta = r(u) \nabla \cdot (\alpha_0(x) \nabla \theta), \qquad r(u) := \frac{\widetilde{k}(u)}{c(u)}. \tag{81}$$

The imposed normal flux is unchanged. If $c(u) = \widetilde{k}(u)$, the equation for θ is exactly the linear heat equation. If $c(u) \equiv 1$, it becomes

$$\partial_t \theta = m(\theta) \nabla \cdot (\alpha_0(x) \nabla \theta), \qquad m(\theta) := r\Big(\Theta_{\widetilde{k}}^{-1}(\theta)\Big) = \widetilde{k}\Big(\Theta_{\widetilde{k}}^{-1}(\theta)\Big). \tag{82}$$

Proof. Differentiating the transform gives

$$\Theta'_{\widetilde{k}}(u) = \widetilde{k}(u). \tag{83}$$

The chain rule therefore gives

$$\nabla \theta = \widetilde{k}(u) \nabla u, \tag{84}$$

$$\partial_t \theta = \widetilde{k}(u) \partial_t u. \tag{85}$$

It follows that

$$\nabla \cdot \Big(\alpha_0(x) \widetilde{k}(u) \nabla u \Big) = \nabla \cdot (\alpha_0(x) \nabla \theta), \tag{86}$$

$$\partial_t u = \frac{1}{\widetilde{k}(u)} \partial_t \theta. \tag{87}$$

Substitution into equation 79 gives

$$\frac{1}{\widetilde{k}(u)}\partial_t\theta = \frac{1}{c(u)}\nabla\cdot(\alpha_0(x)\nabla\theta)\,. \tag{88}$$

Multiplying by $\widetilde{k}(u)$ proves equation 81. The same gradient identity gives

$$-\alpha_0(x)\widetilde{k}(u)\nabla u\cdot n = -\alpha_0(x)\nabla\theta\cdot n, \tag{89}$$

so the prescribed normal flux is unchanged. Setting $c(u)=\widetilde{k}(u)$ or $c(u)=1$ gives the two stated special cases. □

The two settings now lead to different uses of the learned response. Writing v_α^{lin} for the raw prediction of the linear kernel, both can be summarized as

$$\left(v_\alpha^{\mathrm{lin}},\widetilde{k},c\right)\xrightarrow{\mathcal{T}} v_\alpha. \tag{90}$$

In setting A, the transformation reduces the problem exactly to the linear heat equation. The raw prediction is therefore the transformed trace, $v_\alpha^{\mathrm{lin}}=\theta|_{\Sigma_O}$, and pointwise inversion gives

$$v_\alpha = \Theta_{\widetilde{k}}^{-1}(v_\alpha^{\mathrm{lin}}). \tag{91}$$

Thus $\mathcal{T}$ is exact and does not require the streaming state. In setting B, the multiplier $m(\theta)$ remains, so the transformed system does not fully linearize. Because this multiplier is defined throughout the volume while the model predicts only a trace, we introduce a trace-based approximation and use the streaming state to advance it in time.

For the conductivity used in the experiments, we set

$$\widetilde{k}(u)=1+\beta u, \tag{92}$$

and assume that $1+\beta u>0$ on the relevant solution range. The transform, its inverse, and the remaining multiplier when $c(u)=1$ are

$$\Theta_{\widetilde{k}}(u)=u+\frac{\beta}{2}u^2, \qquad \Theta_{\widetilde{k}}^{-1}(s)=\frac{-1+\sqrt{1+2\beta s}}{\beta}, \tag{93}$$

$$m(s)=\sqrt{1+2\beta s}. \tag{94}$$

The inverse formula is written for $\beta\neq 0$. For $\beta=0$, the transform is the identity and $m\equiv 1$.

We evaluate m on the predicted trace and use it to modify the corresponding time derivative. To express this update, we first collect the streaming variables in a single state vector.

Let

$$z:=\begin{bmatrix}g_1^\top & \cdots & g_M^\top\end{bmatrix}^\top, \qquad \dot z=-\Lambda z+B_f\widehat{f}, \tag{95}$$

where

$$\Lambda:=\mathrm{diag}(\lambda_1 I_L,\dots,\lambda_M I_L), \qquad C:=\begin{bmatrix}U_1 & \cdots & U_M\end{bmatrix}, \qquad B_f:=\begin{bmatrix}V_1^\top & \cdots & V_M^\top\end{bmatrix}^\top. \tag{96}$$

The transformed trace reconstructed from this state is

$$v^{\mathrm{lin}}(\xi):=\Theta_{\widetilde{k}}(u(X_O(\xi)))=\mathbf{\Phi}_O(\xi)^\top Cz. \tag{97}$$

Define the projected multiplication matrix and a regularized map from observation coefficients to state variables by

$$M(z):=\int_{U_O}\mathbf{\Phi}_O(\xi)m(v^{\mathrm{lin}}(\xi))\mathbf{\Phi}_O(\xi)^\top\,d\xi, \tag{98}$$

$$C_\varepsilon^\dagger:=(C^\top C+\varepsilon I)^{-1}C^\top. \tag{99}$$

Proposition 7 (Trace-based reduced approximation)**.** *Replacing the volumetric multiplier in equation 82 by its value on the predicted trace gives*

$$\dot z=C_\varepsilon^\dagger M(z)C\left(-\Lambda z+B_f\widehat{f}\right). \tag{100}$$

If $\beta=0$, $\varepsilon=0$, and C has full column rank, this reduces to the original linear streaming dynamics.

Proof. The linear streaming model predicts the observation coefficients and their time derivative as

$$w = Cz, \qquad \dot{w} = C\dot{z} = C(-\Lambda z + B_f \widehat{f}). \tag{101}$$

Multiplying the reconstructed derivative by $m(v^{\mathrm{lin}}(\xi))$ and projecting it back onto the observation basis gives the coefficient vector

$$M(z)C(-\Lambda z + B_f \widehat{f}). \tag{102}$$

We choose the state derivative by regularized least squares,

$$\dot{z} = \operatorname*{argmin}_y \left\| Cy - M(z)C(-\Lambda z + B_f \widehat{f}) \right\|^2 + \varepsilon \|y\|^2. \tag{103}$$

Its solution is exactly equation 100. If $\beta = 0$, then $m \equiv 1$ and orthonormality of the observation basis gives $M(z) = I$. For $\varepsilon = 0$ and full column rank, $C^\dagger C = I$, so

$$\dot{z} = -\Lambda z + B_f \widehat{f}. \tag{104}$$

□

Evolving equation 100 and then applying the inverse Kirchhoff transform gives the approximate map $\mathcal{T}$ for case B.

C Datasets

This section defines the equations, media, sources, observation manifolds, and numerical reference solutions used in the experiments. Model definitions and experiment-specific protocols follow in later sections.

C.1 Heat-Conduction Data

The temperature satisfies

$$\partial_t u = \nabla \cdot \big(\alpha(x) \nabla u\big) + f, \qquad u(\cdot, 0) = 0, \tag{105}$$

on the cube $\Omega = [0, 0.1]^3$ with homogeneous Neumann conditions. A surface flux on the top face is represented as a volumetric source in the top cell layer. The default observation is the temperature in that layer.

quantity	value
spatial grid	64^3 cells
time interval	$[0, 1]$
stored frames	$N_t = 64$
time integrator	Crank–Nicolson
substeps per frame	4 for linear, 32 for radiative, adaptive for configuration B
linear solver	preconditioned conjugate gradients
relative tolerance	10^{-8}
maximum iterations	200
solver precision	float64

Table 5: Reference discretization for the heat data.

The conservative finite-volume discretization uses harmonic averages of α at cell faces. Crank–Nicolson time stepping uses 4 substeps per stored frame for linear data and 32 for radiative data. Each linear system is solved by conjugate gradients with a constant-coefficient DCT preconditioner. The solver uses float64 and the learned models receive float32 fields. The 64 reported frames follow the zero initial condition, which is not stored as an additional frame.

The two-phase medium is obtained by smoothing a standard Gaussian field g_{med} with a Gaussian filter of standard deviation 0.02, thresholding it at its median, and assigning $\alpha_{\mathrm{solid}} = 5 \times 10^{-4}$ and $\alpha_{\mathrm{powder}} = 5 \times 10^{-6}$. The top 2 cell layers are set to the powder value. The interior-plane

medium	construction
two-phase	Threshold g_{med} at its median. Set $\alpha_{\mathrm{solid}} = 5 \times 10^{-4}$ and $\alpha_{\mathrm{powder}} = 5 \times 10^{-6}$. The top 2 cell layers are set to α_{powder}.
graded	Set $\varphi = \Phi(g_{\mathrm{med}}/\,\mathrm{std}(g_{\mathrm{med}}))$ and $\alpha = \alpha_{\mathrm{powder}} + (\alpha_{\mathrm{solid}} - \alpha_{\mathrm{powder}})\varphi$. For the interior-plane data, the endpoints are 10^{-4} and 10^{-2} and no powder cap is used.

Table 6: Media used in the heat experiments. The collocated experiments use the two-phase medium. The interior-plane experiments use the graded medium.

experiments instead use a graded medium between 10^{-4} and 10^{-2} without the powder cap. Every learned model receives the normalized coefficient

$$\alpha_{\mathrm{norm}} = \frac{\alpha - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}}. \tag{106}$$

For the two-phase data, $(\alpha_{\min}, \alpha_{\max}) = (\alpha_{\mathrm{powder}}, \alpha_{\mathrm{solid}})$. The graded data use their corresponding conductivity endpoints.

Every heat source is a Gaussian flux

$$g(x,t) = A(t) \exp\!\left(-\frac{\|x - c(t)\|^2}{2\sigma^2}\right), \qquad \sigma = 0.025, \tag{107}$$

with peak amplitude 10. Training sources are static pulses with centers uniformly distributed inside the surface margins and durations between 6 and 64 frames. Their centers are uniform in $[0.15, 0.85]^2 L_\Omega$, and the amplitude uses smooth $\sin^2$ ramps. Straight scans move at constant speed along a random line segment of length $0.3L_\Omega$ to $0.6L_\Omega$ and switch off at the end of the path. Their speed is sampled between $0.5L_\Omega$ and $1.5L_\Omega$ per unit time. Serpentine scans traverse 3 to 6 alternating horizontal lines in boxes whose widths range from $0.25L_\Omega$ to $0.5L_\Omega$ and whose heights range from $0.15L_\Omega$ to $0.4L_\Omega$. Half of these paths include a small sinusoidal perturbation. Only the static pulses are used for training.

Two further static-pulse test sets support appendix diagnostics. The rollout set contains 512 frames on $[0, 8]$ at the training frame spacing, while the frame-spacing set contains 128 frames on $[0, 1]$.

data set	samples
heat training	1024
static test	128
straight scans	128
serpentine scans	128
radiative test	128
Kirchhoff data	1024/128/128 per setting
interior-plane data	1024/128/128/128 per plane
resolution pairs	64

Table 7: Heat data used for training and evaluation. Multiple numbers denote training, static test, and the moving or shifted test sets.

The reported error is the mean of the per-sample relative errors

$$\mathcal{E}_{\mathrm{rel}} = \frac{1}{B} \sum_{b=1}^{B} \frac{\|\widehat{v}_b - v_b\|_2}{\|v_b\|_2 + 10^{-8}}. \tag{108}$$

Tables report the mean and population standard deviation over training seeds 0, 1, and 2 unless stated otherwise.

C.2 Distinct Observation Surfaces

The source and governing equation remain unchanged while the observation is moved into the volume.

observation	grid location	relation to the source
horizontal plane	$i_z = 48$	separated by about $L_\Omega/4$
vertical plane	$i_x = 32$	mid-plane whose top edge lies in the source face

Table 8: Interior observation surfaces. Both cases are covered by Proposition 2. The horizontal case also satisfies the uniform bound in Remark 1.

The horizontal and vertical data use identical media and sources, which isolates the effect of observation geometry. The selected graded-medium contrast produces a visible response throughout both planes while retaining meaningful dependence on the medium. Each plane has 1024 training samples and 128 samples for each static, straight, and serpentine test class.

C.3 Advection–Diffusion Data

A ground-level release is observed on an orthogonal sensor curtain. The concentration satisfies

$$\partial_t c + b\cdot\nabla c - \nabla\cdot(\kappa\nabla c) = S \tag{109}$$

on $[0, 9\,\mathrm{km}]^2 \times [0, 2\,\mathrm{km}]$ with homogeneous Neumann conditions. The grid has 64^3 cells and 64 stored frames over 2700 s. The diffusivity is

$$\kappa(x) = \bar{\kappa}\exp(0.4g(x)), \qquad \bar{\kappa} \sim U(300, 700)\ \mathrm{m^2 s^{-1}}, \tag{110}$$

where g is a band-limited cosine field with zero mean and unit variance, clipped to $|g| \le 2.5$. The velocity is the sum of a divergence-free circulation with amplitude $W \sim U(8, 12)$ m/s and a divergence-free perturbation whose RMS speed is $0.4W$. Both components are tangent to the boundary.

The source manifold is the ground plane and the observation manifold is the wall at $x = 9$ km. The emission enters as a volumetric source with a fixed Gaussian vertical profile centered at 50 m with width 300 m. The stored observation is the cell layer adjacent to the wall. Static training fumes have random location, radius between 400 and 1200 m, amplitude between 0.5 and 1, and exponential decay time between $T/2$ and $2T$. Moving tests use either 1 crosswind track or 2 to 4 alternating crosswind legs. Moving emitters keep a constant amplitude and radius and travel crosswind at 1.2 to 2.4 m/s. Static fumes are required to produce a curtain RMS of at least 0.005. Moving sources are not screened by their curtain response.

The finite-volume reference uses 32 substeps per frame. Vertical diffusion is implicit, while horizontal diffusion and first-order upwind advection are explicit. Negative concentrations are clipped after each step.

data set	samples
training	1024
static test	128
straight moving source	128
complex moving source	128

Table 9: Advection–diffusion data used for training and evaluation.

C.4 Nonlinear and Resolution-Transfer Data

The radiative set contains 128 static-pulse samples and stores both the nonlinear reference and the corresponding linear solution. The boundary loss is $-10^{-12}u^4$. The temperature-dependent conductivity experiments use $\widetilde{k}(u) = 1 + \beta u$ in the 2 configurations developed in Appendix B.5.2. The scale is set by the median peak temperature $u_{\max} = 1989.8$ of 64 linear training samples. The resulting values are 1.0051×10^{-4}, 2.5129×10^{-4}, and 5.0257×10^{-4} for the mild, strong, and severe settings. Each setting contains 1024 training samples, 128 static test samples, and 128 shifted test samples. Shifted sources are scans, rasters, or sums of 2 static pulses. Configuration B uses $\lceil 4(1 + \beta u^{(s)}_{\max})\rceil$ substeps per frame, where $u^{(s)}_{\max}$ is the peak of a linear solve for the same sample.

The resolution study uses 64 paired samples. A medium drawn on the 64^3 grid is prolonged as the same piecewise-constant field on the 128^3 grid. The same physical source is evaluated on each grid, and the numerical solver is run independently at both resolutions. The fine target averages the top 2 layers so that it represents the same physical slab as the coarse target.

D GOBO ARCHITECTURE AND COMMON PROTOCOL

D.1 COSINE BASIS AND KERNEL REPRESENTATION

All source and observation charts are rectangular. The outer boundaries of the physical domains have homogeneous Neumann conditions, so cosine functions are a natural basis because their normal derivatives vanish there. Their two-dimensional tensor products are indexed by 2 spatial frequencies and therefore form a regular coefficient grid that can be processed efficiently. We use the cell-centered DCT-II basis

$$\phi_a(i) = c_a \cos\left(\frac{\pi a(2i+1)}{2N}\right), \qquad c_0 = N^{-1/2}, \qquad c_a = (2/N)^{1/2} \text{ for } a > 0. \tag{111}$$

The first 32 functions in each direction give a 32×32 coefficient grid and $D_S = D_O = 1024$. Source projection and observation synthesis are

$$\widehat{F}^m = Df^m D^\top, \qquad v^m = D^\top \widehat{V}^m D. \tag{112}$$

The constant midpoint quadrature weight is absorbed into the learned source factors. Source and observation charts use the same function set. For collocated data, 1 transform matrix therefore serves both projection and synthesis. The basis is defined in continuous chart coordinates, so the same coefficients can be evaluated on another grid. This property is used in the resolution-transfer experiment.

The finite kernel used for heat is

$$K_{\theta,\alpha}(\xi,\eta,\tau) = \sum_{r=1}^{R} e^{-\mu_r \tau} \mathbf{\Phi}_O(\xi)^\top U_r(\alpha) V_r(\alpha)^\top \mathbf{\Phi}_S(\eta). \tag{113}$$

Here μ_r and R correspond to λ_k and M in the main text. The main model uses $R = 16$ rates and factor search rank $L = 16$. It shares source and observation factors for collocated heat data and uses separate factors for distinct manifolds. Each factor is normalized to unit Frobenius norm and a learned scalar amplitude sets its magnitude. Thus the coefficient matrix at rate r is

$$A_r(\alpha) = \sum_{\ell=1}^{L} a_{r\ell}(\alpha) \widehat{u}_{r\ell}(\alpha) \widehat{v}_{r\ell}(\alpha)^\top. \tag{114}$$

This normalization is used in every reported heat run.

The fixed rates are the lowest analytic decay rates of the homogeneous Neumann heat Green kernel on the two-dimensional chart. They provide a physical temporal dictionary for the dominant decay scales, while the learned factors in Eq. (114) represent the heterogeneous medium. The rates follow

$$\mu_{a,b} = \alpha_{\text{solid}} \frac{\pi^2}{L_\Omega^2}(a^2 + b^2). \tag{115}$$

The 16 retained values of $a^2 + b^2$ are $0, 1, 1, 2, 4, 4, 5, 5, 8, 9, 9, 10, 10, 13, 13, 16$. The zero rate represents the conserved mean mode. Repeated analytic rates retain independent learned spatial factors. In the learned-rate ablation, a pooled head predicts $d_r(\alpha)$ and replaces each nonzero ladder value by

$$\mu_r(\alpha) = \mu_r^{(0)} \exp(6 \tanh d_r(\alpha)). \tag{116}$$

The head is initialized at $d_r = 0$ and is regularized toward the analytic ladder. The conserved rate remains zero.

D.2 MEDIUM-CONDITIONING NETWORK

The conditioning network receives only the normalized three-dimensional medium. For each surface location, the corresponding depth column is augmented with a fixed 4-channel depth encoding. 3 one-dimensional convolutions reduce its 64 depth values to 8 positions. Averaging over them gives 16 features. 2 planar convolutions then produce a 32-channel field on the manifold grid. We append 16 fixed spatial coordinate channels and apply a channel map $48 \to 256 \to 64$.

The resulting field passes through 4 blocks of width 64. Each block combines a cosine spectral path, a pointwise linear path, and a small U-Net path. Its update is

$$h^{p+1} = \mathrm{GELU}(\mathcal{N}_p(\mathcal{S}_p h^p) + W_p h^p + \mathcal{U}_p h^p)\,. \tag{117}$$

Here $\mathcal{S}_p$ is a cosine spectral operator that retains the lowest 32×32 modes and mixes channels independently at each mode, $\mathcal{N}_p$ is normalization, W_p acts separately at each grid point, and $\mathcal{U}_p$ captures local features at several scales. The spectral path uses the same cosine transform as the kernel basis. After the 4 blocks, a separate truncated spectral layer applies the forward cosine transform and mixes channels separately at each mode. It stops directly on the 32×32 coefficient grid without applying the inverse transform.

quantity	setting
retained modes	32×32 in each chart
source and observation dimensions	$D_S = D_O = 1024$
temporal rates	$R = 16$
factor rank per rate	$L = 16$
hidden width	64
conditioning blocks	4 cosine spectral blocks with U-Nets and 1 truncated layer
collocated coupling	shared source and observation factors
distinct-manifold coupling	separate source and observation factors
parameter count	22.16 million

Table 10: Settings for the heat GObO used in the main experiments.

A coefficient head produces the source and observation factors in Eq. (114). A pooled head produces their amplitudes. For collocated heat data the same factor is used on both sides, while distinct manifolds use separate factors. The source is not an input to this network. It is projected and applied only after the medium-specific kernel has been formed, so later sources in the same medium require no further neural-network evaluation.

Figure 4 places this architecture beside the adapted NGO system network. NGO predicts a dense coefficient matrix for its 2 linear temporal basis functions. GObO instead keeps the manifold grid through its conditioning blocks and produces low-rank factors on the coefficient grid. It therefore never forms the dense $D_O \times D_S$ matrix.

D.3 TEMPORAL PROPAGATION AND STREAMING STATE

The reported heat and advection–diffusion models treat the source as piecewise linear between frames. For $z = \mu\Delta$, define

$$\gamma_0(z) = \frac{1 - (1+z)e^{-z}}{z^2}, \qquad \gamma_1(z) = \frac{z - 1 + e^{-z}}{z^2}, \tag{118}$$

with their continuous limits at $z = 0$. The contribution of source frames f^{m-1} and f^m is then integrated exactly against each exponential. The equivalent dense form uses the lower-triangular propagator

$$P_r[m,s] = \Delta\Big(\gamma_1(\mu_r\Delta)e^{-\mu_r\Delta(m-s)}\mathbb{1}[s \le m] + \gamma_0(\mu_r\Delta)e^{-\mu_r\Delta(m-s-1)}\mathbb{1}[s \le m-1]\Big) \tag{119}$$

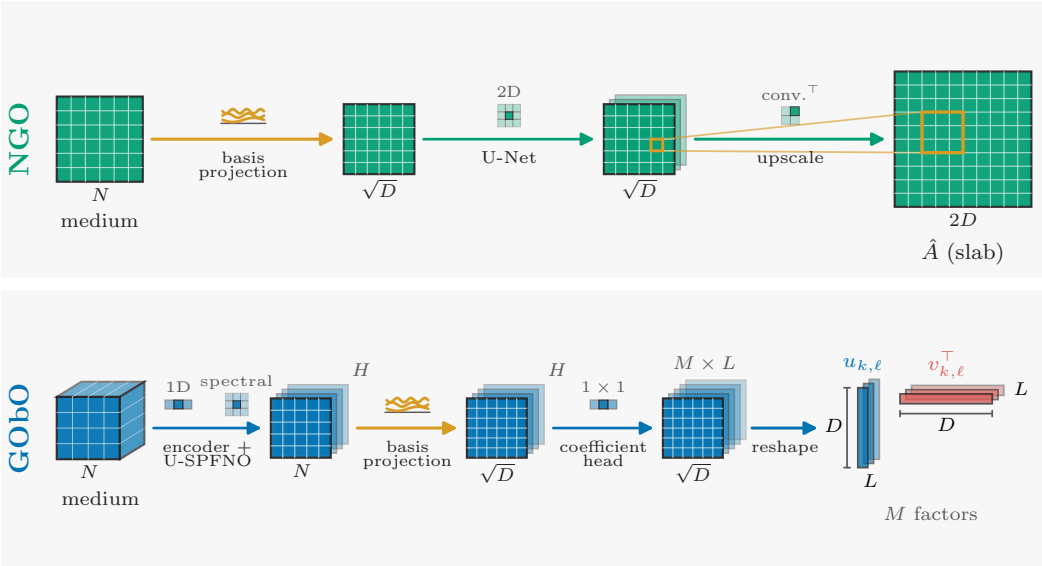


Figure 4: **Conditioning architectures.** The adapted NGO system network maps medium projections to a dense coefficient matrix. GObO maps the medium volume to low-rank factors for each analytic decay rate. The GObO source is applied only after these factors have been produced.

acting on the projected source coefficients $V_r(\alpha)^\top \widehat{f}^s$. Training evaluates the resulting convolution densely over all frames. At inference, the same computation is written as

$$\begin{aligned} g_r^m &= e^{-\mu_r \Delta} g_r^{m-1} + \Delta\gamma_0(\mu_r\Delta) V_r(\alpha)^\top \widehat{f}^{m-1} \\ &\quad + \Delta\gamma_1(\mu_r\Delta) V_r(\alpha)^\top \widehat{f}^m, \\ v^m &= \mathbf{\Phi}_O^\top \sum_r U_r(\alpha) g_r^m. \end{aligned} \tag{120}$$

with $g_r^1 = \Delta\gamma_1(\mu_r\Delta)V_r(\alpha)^\top \widehat{f}^1$ because the zero initial frame is not stored. The source coefficients are contracted with the source factors before propagation, and only the observation factors enter the synthesis. The state has $RL = 256$ scalar coefficients for the heat model. Changing the frame spacing only changes the analytical propagation weights. Dense and streaming evaluations agree to floating-point precision.

D.4 Advection–Diffusion Variant

The advection–diffusion model receives the diffusivity and 3 velocity components. The channels are scaled to order 1 using $\kappa/\kappa_{\mathrm{ref}}$, where $\kappa_{\mathrm{ref}} = 500\,\mathrm{m^2 s^{-1}}$, and by multiplying every velocity component by $T/L_x = 0.3$ s/m. One input path pools depth to the ground chart and a second pools the along-wind direction to the sensor curtain. Each path has its own channel map, 4 of the same conditioning blocks, and a final truncated spectral block. Both retain a 32×32 cosine coefficient grid and produce separate source and observation factors. Both factor grids are complex valued. Including the conjugate rates gives the real curtain field

$$v^m = \mathrm{Re}\left(2\sum_{r=1}^{8}\sum_{\ell=1}^{8} U_{r\ell} G_{r\ell}^m\right), \qquad G_{r\ell}^m = \sum_s P_r[m,s]\langle V_{r\ell}, \widehat{f}^s\rangle. \tag{121}$$

The factor 2 accounts for the conjugate rate.

The kernel uses rank $L = 8$ and the 8 lowest square modes (a, b). It parametrizes 8 complex rates together with their conjugates. In normalized time, the fixed rates satisfy

$$\mathrm{Re}\,\widetilde{\mu}_{a,b} = \kappa_{\mathrm{ref}} T \frac{\pi^2(a^2+b^2)}{L_x^2}, \qquad \mathrm{Im}\,\widetilde{\mu}_{a,b} = \widehat{u}(\alpha)\pi a, \tag{122}$$

where $\kappa_{\mathrm{ref}} = 500\,\mathrm{m^2 s^{-1}}$ and $\widehat{u}$ is the root-mean-square horizontal velocity scaled by T/L_x. The propagator runs in normalized time with $\Delta = 1/N_t$, so the rates are dimensionless. The real part represents diffusive decay and the imaginary part represents advective phase transport. Adding the conjugate terms gives a real-valued output. A learned-rate ablation perturbs both parts of each rate. The real part follows Eq. (116), while the imaginary part is multiplied by $1 + \tanh e_r$. This variant does not use the heat model's amplitude head or factor normalization. A scalar calibration sets its initial output scale.

D.5 Training and Evaluation

quantity	setting
loss	relative L_2 in equation 108
optimizer	AdamW with default moment parameters
learning rate	10^{-3}
weight decay	10^{-4}
schedule	cosine decay to zero
training steps	30,000
batch size	64
precision	float32
validation frequency	every 500 steps
validation samples	32 held-out training samples
reported checkpoint	final checkpoint
training seeds	$0, 1, 2$
gradient clipping	none

Table 11: Shared training settings. FNO-3D uses batch size 8. The interior-plane models use the changes described below.

Collocated heat data use all rotations and reflections of the square for data augmentation. Horizontal-plane data use the same transformations. The vertical-plane and advection–diffusion data use only the reflection that preserves their geometry. Inputs are normalized as stated in the dataset definitions. A source that enters a neural network is divided by its training-set standard deviation. GObO instead receives the source in physical units because it is applied only after conditioning. Targets remain in physical units. FNO-3D uses batch size 8. Interior-plane models use learning rate 3×10^{-4} and gradient clipping at 1. Adapted NGO uses Adam with learning rate 10^{-3} and no weight decay. All evaluations use Eq. (108) on the same splits.

E Experimental Details

This section gives only the choices that differ across experiments. Dataset, architecture, training, and metric definitions are given above. Numerical outcomes are collected in Appendix F.

E.1 Experimental setting

E.1.1 Baselines

FNO-2D receives a two-dimensional medium encoding together with all 64 source frames and predicts all 64 observation frames directly. It has width 64, 32 Fourier modes per direction, and 4 spectral layers. A 5-layer version tests additional depth. The backbone-matched spectral model uses GObO's medium encoder and 4 or 5 of the same cosine spectral blocks but replaces the kernel by direct regression of the trajectory. FNO-3D receives the conductivity volume and the surface source $f(x, y, t)$ arranged as a 64^3 array with time as the third axis, uses width 64 with 16 modes in each direction, and predicts the observed trajectory. The advection–diffusion FNO stacks the diffusivity and velocity fields as depth channels and maps the ground-source array directly to the sensor curtain. The nonlinear FNO baselines are retrained independently for each material law and contrast.

All baselines use the same data splits, loss, and evaluation metric as GObO. Model sizes are reported in Table 16.

E.2 GObO learns a kernel that generalizes across sources

E.2.1 Heat and Distinct-Surface Protocols

Every heat model is trained on static pulses and evaluated without retraining on held-out static pulses, straight scans, and serpentine scans. The headline experiment observes the source face. The distinct-surface experiments use separate source and observation factors. The horizontal plane retains all square rotations and reflections. The vertical plane uses reflection in the shared horizontal coordinate because rotations do not preserve its geometry.

E.2.2 Advection–Diffusion Protocol

The fixed-rate GObO is the primary model. Learned rates provide an ablation, and FNO-2D is the direct baseline. Training uses stationary fumes. Evaluation uses held-out fumes, straight crosswind motion, and alternating crosswind motion. Reflection in the crosswind coordinate is applied to the medium volume, source, and observation while reversing the corresponding velocity component.

E.3 The augmented kernel state captures non-closed submanifold dynamics

E.3.1 Adapted Transient NGO

The baseline follows the published time-dependent NGO, with only the minimal adaptations that our setting requires. Its conditioning network is shown next to GObO's in Figure 4.

NGO leaves the choice of spatial basis open. Its authors use splines, whereas we use the cosine basis of this paper so that the baseline is comparable with the other experiments. This is the only design choice we make differently. The remaining changes are not differences but adaptations that our setting necessitates. The stepped state is the observed trace because the full field that NGO advances is not available. The volumetric medium is projected onto the spatial basis at each depth, and the resulting maps are concatenated as channels for the system network.

NGO stabilizes its rollout by capping the norm of the step matrix at a threshold S. We report their published value $S = 0.8$ together with $S = 1$, because the norm of the true step map in our setting is close to 1.

E.3.2 One-Step GObO Variant

This diagnostic keeps GObO's medium encoder and spatial factors but replaces the lag-dependent kernel by a purely spatial recurrence. With k previous frames it predicts

$$\widehat{v}^m = \sum_{j=1}^{k} S_j(\alpha)\widehat{v}^{m-j} + B_0(\alpha)q_L^m + B_1(\alpha)q_R^m. \tag{123}$$

We use $k = 1$, 4, and 16. Each spatial map is represented by a rank-256 factorization with separate input and output factors. The concatenated state block $[S_1 \cdots S_k]$ is normalized by the same norm cap as NGO with $S = 1$. During training it receives the true previous frames and predicts the next frame. At evaluation it starts from the zero trace and is then advanced using its own predictions.

E.3.3 Best Linear One-Step Map

This fit removes neural-network approximation from the test of whether the observation contains enough state for a one-step update. For each of 3 fixed test media, each of the 64 modes in an 8×8 cosine block is excited once. Reported values are means over these 3 media. The reference responses define the restricted discrete Green operator. For history depth k, the best linear map is

$$W_k = \arg\min_W \sum_{p,m} \left\| W^\top \begin{bmatrix} \widehat{h}_p^{m-1} & \cdots & \widehat{h}_p^{m-k} & \widehat{f}_p^{m-1} & \widehat{f}_p^{m} \end{bmatrix}^\top - \widehat{h}_p^m \right\|_2^2. \tag{124}$$

The fit is evaluated on 24 additional band-limited pulses by 1-step residual and rollout error. The same procedure is repeated for a two-dimensional heat equation on the top layer of the same medium, observed over its whole domain. This control has access to the complete state and separates missing state in the partial observation from basis truncation and regression error.

E.4 THE KERNEL PARAMETRIZATION CAPTURES THE SPATIOTEMPORAL STRUCTURE

E.4.1 NAIVE TENSOR DECOMPOSITION

The naive alternative replaces the lag law by independent cosine bases for source time and observation time

$$K(\xi,\eta,t,s)=\sum_{c,c'=0}^{C-1}\tau_c(t)\tau_{c'}(s)\mathbf{\Phi}_O(\xi)^\top A_{cc'}(\alpha)\mathbf{\Phi}_S(\eta). \tag{125}$$

It uses the same medium encoder and rank-16 spatial factors as GObO. The main comparison uses $C=16$. A size-matched control uses $C=4$. This representation does not enforce zero response before a source is applied. It also has no finite streaming state and is defined only on the fixed time interval used to build its temporal bases.

E.4.2 ARCHITECTURE AND PARAMETRIZATION ABLATIONS

The retained ablations compare fixed and learned rates, shared and separate factors, and search ranks. A further control replaces GObO's learned volume-to-manifold encoder by the lowest 16^3 cosine projections of the medium followed by a parameter-matched U-Net. The kernel, rate ladder, factors, and recurrence remain unchanged. Reduced 16×16 and 8×8 basis controls test whether differences can be explained by basis size.

E.4.3 CONTROLLED CURVED-SURFACE RECOVERY

The controlled problem uses the homogeneous Neumann heat equation on the unit cube with diffusivity 1. For chart coordinates (p,q), set $a=p-1/2$ and $b=q-1/2$. The source and observation surfaces are

$$\begin{aligned}X_S(p,q)&=\left(\tfrac{1}{2}+0.7a+0.08\sin(2\pi q),\,\tfrac{1}{2}+0.8b,\,0.26-0.15a+0.15b+0.3ab\right),\\ X_O(p,q)&=\left(\tfrac{1}{2}+0.7a+0.05\cos(2\pi q),\,\tfrac{1}{2}+0.45b+0.1\sin(2\pi p),\,0.78-0.25a+0.2b-0.2a^2\right).\end{aligned} \tag{126}$$

The restricted reference kernel is computed from 343 ambient cosine modes and includes the source-surface area factor.

quantity	value
diffusivity	$\kappa=1$
time step and frames	$\Delta t=0.0125$ and $N_t=64$
first supervised frame	4, corresponding to $\tau_{\min}=0.05$
chart grid	64×64 cell-centered points
retained chart modes	16×16 cosine modes
ambient spectral cutoff	$n_j\le6$, giving 343 modes
displayed lags	0.05, 0.075, and 0.125
held-out sources	64 Gaussian chart bumps

Table 12: Settings for the controlled curved-manifold example.

The recovery figure uses 256 coefficient-space impulse probes, 16 distinct rates, and a truncated singular-value decomposition on frames with lag at least 0.05. Each fitted coefficient matrix is truncated to rank 8. The rank sweep is a separate fit with 63 rate slots, including repeated ambient eigenvalues. It minimizes the relative response error plus a factor norm penalty and is compared with a convex low-rank reference based on the nuclear norm. The table reports only the data-fit term for both methods. Recovered rank uses threshold $10^{-4}\sigma_1$.

E.5 GOBO ENABLES RAPID SOURCE EVALUATION AND NONLINEAR CORRECTIONS

E.5.1 TIMING PROTOCOL

GObO conditioning contains the only neural-network evaluation for a medium. Later all-frames and streaming queries use only the explicit kernel. FNO-2D requires a network evaluation for every

quantity	setting
CPU	AMD EPYC 7742 with 32 allocated cores
GPU	NVIDIA A100-SXM4 with 80 GB
software	Python 3.12.3 and PyTorch 2.13
numerical libraries	NumPy 2.1.0 and SciPy 1.17.1 with OpenBLAS
solver repetitions	1 warmup and 3 measurements
learned-model repetitions	3 warmups and 8 measurements
reported statistic	median

Table 13: Environment used for the timing measurements.

source. Batch size 1 measures latency. Batch size 64 measures throughput and shares 1 conditioned GObO across the batch. GPU measurements are synchronized. The matched-accuracy solver is the cheapest tested setting whose error against the production solver is below GObO's in-distribution error of 0.011. The latency measurements use 1 test medium and 1 source. The finite-volume rows use the production configuration with 1 thread. Each learned CPU row uses the fastest result obtained with 1, 8, or 32 threads.

E.5.2 Resolution-Transfer Protocol

mode	computation
M0	Compare the 2 numerical reference solutions.
M1	Run the complete learned model on the fine grid.
M2	Run the learned model on the coarse grid and evaluate its final continuous spectral representation on the fine grid.
M3	Run only the GObO medium encoder on the coarse grid, then project the fine-grid source and evaluate the operator on the fine grid.

Table 14: Resolution-transfer modes used in Section F.5.2.

M1 computes the positional encodings on the 64-cell training grid and resamples them to the fine grid. The positional-encoding ablation instead uses raw fine-grid indices. M2 resynthesizes GObO coefficients with the fine-grid cosine basis and uses zero-padded Fourier synthesis for FNO. M3 rebuilds the source and observation basis matrices at the fine resolution while reusing factors conditioned on the coarse medium.

E.5.3 Radiative Feedback Protocol

The radiative law uses coefficient 10^{-12} and ambient temperature 0. We evaluate an explicit update using the previous-frame radiative flux and a linearized implicit update in coefficient space. Both use the unchanged GObO kernel and the exact propagation weights in the state update. The reference uses 32 finite-volume substeps per stored frame. In the implicit correction matrix, $\gamma_1(\nu_k)$ is replaced by its small-ν limit $1/2$. The state update itself retains the exact weights.

E.5.4 Temperature-Dependent Conduction Protocol

setting	$\beta u_{\max}$	conductivity ratio
mild	0.2	1.2
strong	0.5	1.5
severe	1.0	2.0

Table 15: Conductivity contrasts used in the nonlinear experiments. The range covers reported process windows for 316L, IN718, and Ti-6Al-4V (Kim, 1975; Alloys, 2008; Mukherjee et al., 2016).

Configuration A applies the exact Kirchhoff transform $\Theta = u + \beta u^2/2$, evaluates the unchanged linear checkpoint, and applies the inverse transform pointwise. Configuration B advances the re-

duced kernel state with the trace-based state equation from Appendix B.5.2. The regularized left inverse uses $\varepsilon = 10^{-8}\|S^\top S\|_2$. The learned state update uses 4 substeps per frame. The local clock is frozen within a substep and the resulting affine system is advanced by its matrix exponential.

The fixed-point comparison uses damping 0.5, residual scale 1, tolerance 10^{-3}, and at most 20 iterations. A sample is marked as diverged after 2 successive residual increases. The reported iterate is the one with the smallest residual and does not use the truth for selection. The time-derivative rows obtain $\partial_t\theta$ by finite differences over frames, while the generator row uses the exact derivative of the modal representation. Statistics pool the 128 samples from all 3 seeds.

F Additional Experimental Results

The results follow the same order as the main experiments. Protocols are not repeated here.

F.1 Experimental setting

F.1.1 Model-Size Comparison

model	parameters	scope
GObO	22.16M	heat, fixed rates
FNO-2D	16.91M	heat
matched spectral model	17.89M	heat
FNO-3D	67.18M	heat
advection GObO, fixed rates	44.01M	advection–diffusion
advection GObO, learned rates	44.14M	advection–diffusion
advection FNO-2D	16.96M	advection–diffusion
FNO-2D, 5 layers	21.11M	heat
matched spectral model, 5 blocks	22.33M	heat

Table 16: **Model sizes.** Fixed-rate and learned-rate advection models are listed separately.

The table follows the framework convention in which 1 complex tensor entry is counted as 1 parameter. FNO spectral weights therefore contain 2 real numbers per counted complex parameter.

F.2 GObO learns a kernel that generalizes across sources

F.2.1 Heat Source Transfer

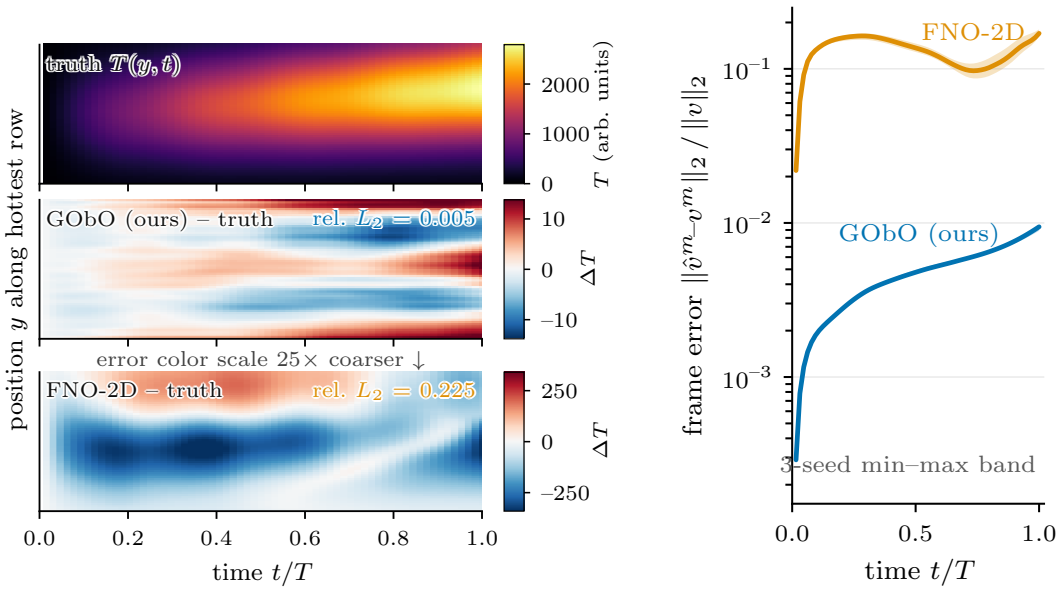
The full comparison confirms the trade-off shown in the main text. Direct models are strongest on static sources but degrade sharply under motion. GObO changes little across the 3 source classes.

	relative L_2			absolute L_2		
method	static	straight	serpentine	static	straight	serpentine
GObO	0.011 ± 0.001	0.012 ± 0.000	0.005 ± 0.000	2838 ± 160	2253 ± 85	2447 ± 12
FNO-2D	0.001 ± 0.000	0.101 ± 0.004	0.137 ± 0.007	282 ± 17	23330 ± 810	63590 ± 3400
FNO-3D	0.004 ± 0.000	0.185 ± 0.003	0.426 ± 0.008	970 ± 6	46670 ± 360	199500 ± 3900
matched spectral model	0.003 ± 0.001	0.204 ± 0.002	0.217 ± 0.004	370 ± 50	48130 ± 700	100800 ± 1900

Table 17: **Complete source-transfer comparison.** Values are means and population standard deviations over 3 seeds. The moving sets are zero-shot.

The aggregate errors show that the difference is not limited to 1 path type or 1 normalization. GObO remains stable in relative and absolute error on both moving tests, while every direct model loses its static-source advantage. The following frame-resolved view shows how that failure develops during a scan.

The displacement visible in the signed error explains why the relative error grows even though the direct model remains accurate on static sources.



(a) response and signed error

(b) per-frame relative error

Figure 5: **Per-frame behavior on serpentine scans.** GObO tracks the moving peak throughout the trajectory, whereas FNO-2D develops a persistent displacement error. Curves show the 3-seed min–max band.

F.2.2 Baseline Capacity and GObO Design

Increasing the depth of either direct baseline does not repair moving-source transfer. Within GObO, fixed rates and shared factors give the most reliable result. Search rank 1 is insufficient.

baseline architecture	static	straight	serpentine
FNO-2D, 4 layers	0.001 ± 0.000	0.101 ± 0.004	0.137 ± 0.007
FNO-2D, 5 layers	0.001 ± 0.000	0.103 ± 0.002	0.149 ± 0.005
matched spectral model, 4 layers	0.003 ± 0.001	0.204 ± 0.002	0.217 ± 0.004
matched spectral model, 5 layers	0.004 ± 0.001	0.226 ± 0.002	0.234 ± 0.013

Table 18: **Baseline-depth ablation** (relative L_2). Increasing depth does not resolve the moving-source failure of either direct-output architecture.

The unchanged or worse transfer error after adding depth rules out insufficient baseline capacity as the primary explanation. We next vary the structure inside GObO to determine which parts of its factorization matter.

GObO factorization	static	straight	serpentine
fixed rates, shared factors, $L = 16$	$\mathbf{0.011 \pm 0.001}$	$\mathbf{0.012 \pm 0.000}$	$\mathbf{0.005 \pm 0.000}$
learned rates, shared factors, $L = 16$	0.011 ± 0.002	0.012 ± 0.002	0.006 ± 0.001
fixed rates, separate factors, $L = 16$	0.019 ± 0.001	0.016 ± 0.001	0.008 ± 0.000
learned rates, separate factors, $L = 16$	0.013 ± 0.002	0.013 ± 0.002	0.006 ± 0.001
fixed rates, shared factors, $L = 1$	0.130 ± 0.007	0.123 ± 0.005	0.074 ± 0.008

Table 19: **Rate and factor-structure ablation.** Here L is the factor search rank used for each fixed or learned rate, matching the terminology of Tables 26 and 26.

These controls support the use of the analytic fixed rates, shared factors, and $L = 16$ in the headline heat model.

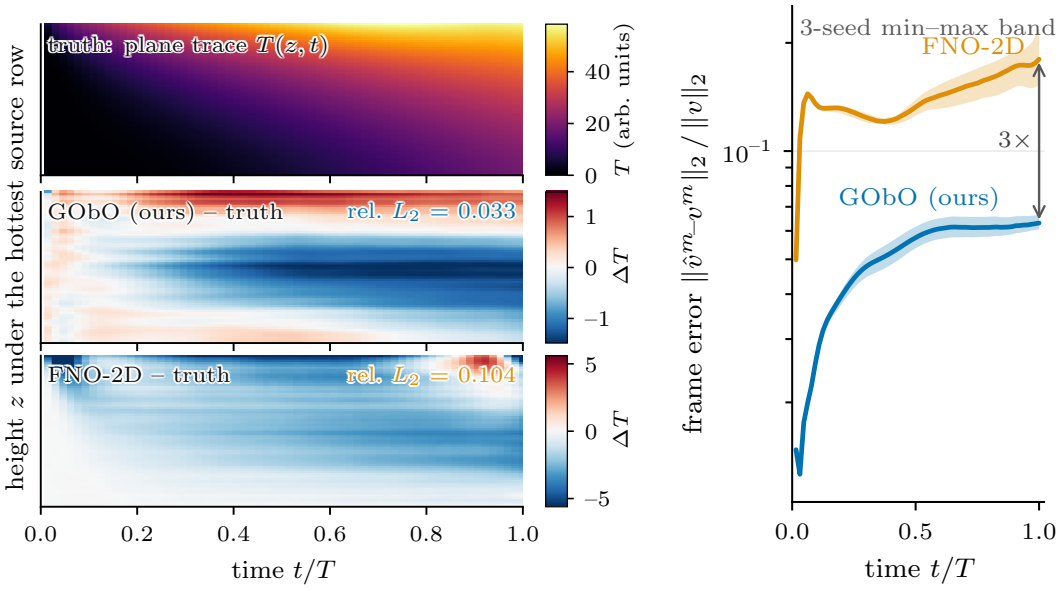
F.2.3 Distinct Observation Surfaces

The same source-transfer pattern holds when source and observation surfaces are distinct.

observation	method	static	straight	serpentine
horizontal plane	GObO	0.042	**0.044**	**0.027**
horizontal plane	FNO-2D	**0.027**	0.194	0.108
vertical mid-plane	GObO	0.064	**0.062**	**0.054**
vertical mid-plane	FNO-2D	0.066	0.240	0.148

Table 20: **Source transfer with non-collocated observations.** The horizontal plane is separated from the source by $L_\Omega/4$, the vertical plane is orthogonal to the source face.

The table shows that separating the manifolds does not remove GObO's transfer advantage. The vertical-plane example below resolves this result in time and shows that the aggregate gap is sustained throughout the moving scan.



(a) vertical trace and signed error

(b) per-frame relative error

Figure 6: **A moving source observed on the vertical mid-plane.** For the displayed sample, GObO has relative error 0.033 and FNO-2D has 0.104, the aggregate per-frame curves show that the gap persists over the test set.

The result shows that the learned kernel transfers across sources even when the response is observed away from and orthogonal to the source surface.

F.2.4 Advection–Diffusion Source Transfer

Fixed physical rates keep GObO close to its static error under both source shifts. Learned rates degrade all 3 source classes. FNO-2D is competitive on static fumes but does not transfer to motion.

The smaller gap on stationary fumes shows that both models can learn the in-distribution map. The moving release below illustrates the different behavior after the source leaves the stationary training distribution.

The curtain response confirms the quantitative result without suggesting the catastrophic qualitative failure observed for heat.

	relative L_2			absolute L_2		
method	static	straight	complex	static	straight	complex
GObO, fixed rates	0.114 ± 0.018	0.101 ± 0.020	0.139 ± 0.035	1.88 ± 0.19	2.04 ± 0.31	2.00 ± 0.24
GObO, learned rates	0.300 ± 0.021	0.313 ± 0.016	0.352 ± 0.028	5.19 ± 0.19	6.68 ± 0.23	6.13 ± 0.26
FNO-2D	0.104 ± 0.009	0.430 ± 0.015	0.364 ± 0.007	1.77 ± 0.15	9.33 ± 0.36	6.09 ± 0.17

Table 21: **Complete source-generalization results on the sensor curtain.** Values are means and population standard deviations over 3 seeds.

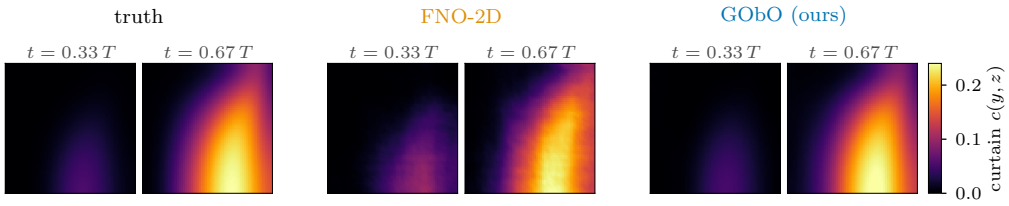


Figure 7: **A moving ground-level release observed on a vertical sensor curtain.** The qualitative comparison complements the aggregate advection–diffusion errors in Table 21.

F.3 The augmented kernel state captures non-closed submanifold dynamics

F.3.1 Adapted Transient NGO

The $S = 1$ model reaches 1-step error 0.012 and rollout errors 0.039, 0.043, and 0.027 on static, straight, and serpentine sources. The published value $S = 0.8$ clips every sample and gives static rollout error 0.721.

The $S = 1$ result is therefore the informative NGO comparison used in the main table, while the published threshold is retained to document the adaptation.

F.3.2 One-Step Prediction from Partial Observations

On the fully observed two-dimensional control, the best 1-frame map has residual 4×10^{-15} and additional history does not improve it. On the three-dimensional trace, the residual is 3.2×10^{-3} at $k = 1$ and 1.1×10^{-3} at $k = 16$. The corresponding best linear rollout errors are 0.043 and 0.016. This contrast shows that the observed trace is not a closed state.

The learned one-step GObO variants follow the same trend inside the training window. Longer history lowers static rollout error from 0.103 to 0.032. Beyond that window, the longer recurrences become unstable, while GObO remains bounded through the reported diagnostic.

	params	one-step (static)	rollout, static	straight	serpentine	rescaled fraction	ms/frame
GObO	22.16 M	–	0.011 ± 0.001	0.012 ± 0.000	0.005 ± 0.000	–	0.13
A. NGO as published							
NGO, $S = 0.8$	0.02 M	0.163 ± 0.000	0.721 ± 0.000	0.760 ± 0.000	0.552 ± 0.001	1.00	0.09
NGO, $S = 1$	0.02 M	0.012 ± 0.000	0.039 ± 0.001	0.043 ± 0.000	0.027 ± 0.001	1.00	0.09
B. One-step GObO variants							
one-step, $k = 1$	22.28 M	0.021 ± 0.000	0.103 ± 0.003	0.103 ± 0.009	0.058 ± 0.003	1.00	0.08
one-step, $k = 4$	22.43 M	0.014 ± 0.007	0.091 ± 0.012	0.108 ± 0.016	0.052 ± 0.007	1.00	0.09
one-step, $k = 16$	23.05 M	0.015 ± 0.004	0.032 ± 0.005	0.033 ± 0.006	0.017 ± 0.003	1.00	0.09

Table 22: One-step schemes on the trace. Relative error, mean $\pm$ population standard deviation over 3 seeds, 128 samples per set. "One-step" is the error of a single step from the true previous frames, the quantity these rows are trained on, the 3 rollout columns feed the model its own previous frames from rest. "Rescaled fraction" indicates whether the norm cap is active on the first static test sample, averaged over the 3 seeds. Frame-spacing transfer requires retraining for every one-step row. Parameter counts are of the built models.

Inside the training interval, longer histories partly compensate for the missing state and the $k = 16$ variant is the strongest one-step model. This does not establish a stable evolution law. The next table

rolls every model beyond the interval used for training and exposes the resulting accumulation of error.

cumulative error on	$[0, 1T]$	$[0, 2T]$	$[0, 3T]$	$[0, 4T]$
GObO	0.010 ± 0.000	0.047 ± 0.002	0.105 ± 0.004	0.158 ± 0.007
NGO, $S = 0.8$	0.702 ± 0.000	0.812 ± 0.000	0.832 ± 0.000	0.841 ± 0.000
NGO, $S = 1$	0.036 ± 0.001	0.142 ± 0.001	0.237 ± 0.001	0.297 ± 0.001
one-step, $k = 1$	0.099 ± 0.001	0.227 ± 0.005	0.326 ± 0.008	0.396 ± 0.012
one-step, $k = 4$	0.090 ± 0.012	diverged	diverged	diverged
one-step, $k = 16$	0.030 ± 0.006	0.136 ± 0.022	0.58 ± 0.20	diverged

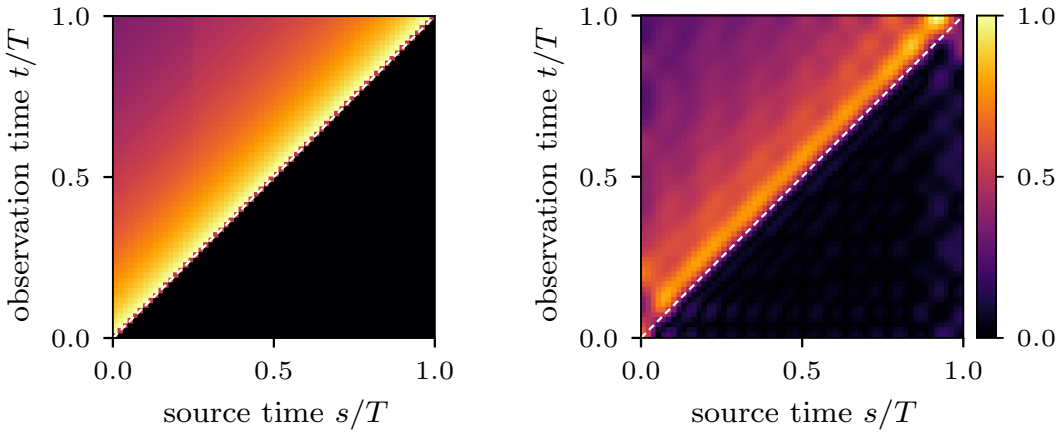
Table 23: One-step schemes, horizon. Cumulative relative error on the first $h \cdot 64$ frames of the decaying-source test set (128 samples, source off before T), mean $\pm$ population standard deviation over 3 seeds. GObO by the streaming recurrence, the one-step rows by rollout. "Diverged" is a rollout that has left the scale of the data by that horizon (a cumulative error above 1 in at least 1 seed). The $S = 0.8$ row has decayed to a fraction of the truth instead.

The horizon comparison shows that lower error inside the training window does not imply stable repeated prediction from partial observations.

F.4 The kernel parametrization captures the spatiotemporal structure

F.4.1 Impulse-Response Structure

Figure 8 compares both learned responses with the reference impulse response for the same medium and source location. GObO captures its lag-dependent decay and onset in this example. The generic space–time basis does not reproduce this structure and responds before the impulse is applied.



(a) GObO kernel parametrization

(b) separable space–time parametrization

Figure 8: **Impulse-response comparison.** The panels show the response produced by the proposed kernel parametrization and by the naive separable space–time alternative.

This response-level comparison links the lower GObO error to the temporal structure imposed by its lag-dependent kernel.

F.4.2 Naive Tensor-Decomposition Accuracy

Removing the lag law increases static error from 0.011 to 0.035 and moving error from 0.012 to 0.045. Its pre-source response reaches 0.52 of its peak. The size-matched $C = 4$ control is less accurate still, with static error 0.126 and pre-source response 0.91 of its peak.

The extra parameters of the separable model do not compensate for replacing the lag-dependent kernel and do not provide a reusable streaming state.

	params	static	straight	serpentine	$\Delta/2$	Δ (control)	2Δ
GObO ($e^{-\lambda_k(t-s)}$, 16 rates)	22.16 M	0.011 ± 0.001	0.012 ± 0.000	0.005 ± 0.000	0.010 ± 0.000	0.011 ± 0.000	0.011 ± 0.000
separable space–time kernel, $C = 16$	22.67 M	0.035 ± 0.001	0.045 ± 0.002	0.051 ± 0.008	0.031 ± 0.001	0.031 ± 0.001	0.033 ± 0.001

Table 24: Separable space–time basis. Relative error equation 108, mean $\pm$ population standard deviation over 3 seeds, 128 samples per set. The last 3 columns use the same independently generated frame-spacing test set.

F.4.3 Architecture and Medium-Pathway Ablations

Replacing the learned volume encoder by fixed medium projections increases static error to 0.103 at a similar parameter count. An MLP on the same projections reaches 0.020 but uses 137.23 million parameters. Reducing GObO's basis to 16×16 gives static error 0.009, while an 8×8 basis gives 0.019. The comparison therefore cannot be explained by basis size alone.

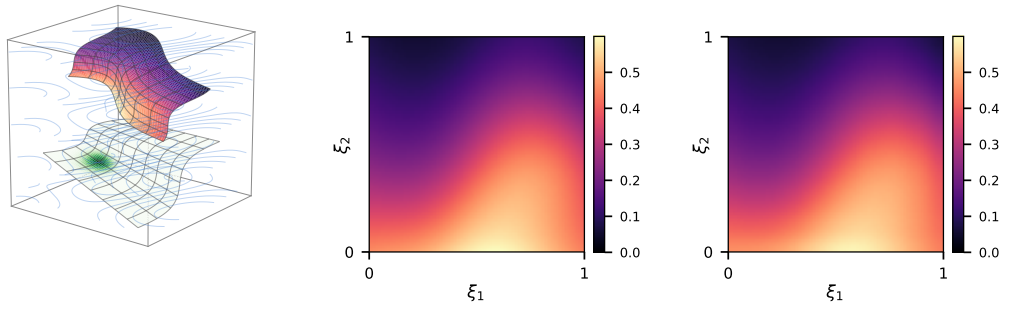
	params	static	straight	serpentine	$\Delta/2$	Δ (control)	2Δ	ms/frame
GObO (cosine spectral conditioning)	22.16 M	0.011 ± 0.001	0.012 ± 0.000	0.005 ± 0.000	0.010 ± 0.000	0.011 ± 0.000	0.011 ± 0.000	0.13
medium projections + U-Net	22.73 M	0.103 ± 0.012	0.102 ± 0.011	0.068 ± 0.011	0.110 ± 0.013	0.110 ± 0.013	0.111 ± 0.013	0.14

Table 25: Medium pathway. Relative error, mean $\pm$ population standard deviation over 3 seeds, 128 samples per set. Columns as in Table 24, ms/frame is measured per streaming step.

The learned volume-to-manifold pathway is therefore important independently of the retained basis size.

F.4.4 Controlled Curved-Surface Recovery

The recovered kernel has slice errors 9.4×10^{-3}, 4.3×10^{-3}, and 1.6×10^{-3} at lags 0.05, 0.075, and 0.125. Its held-out error on 64 Gaussian chart sources is 0.027.



(a) source and observation manifolds

(b) reference kernel slice

(c) recovered kernel slice

Figure 9: **Kernel recovery on distinct curved manifolds.** The source lies on a saddle and the response is observed on a separate curved surface. The final panels compare the reference and recovered restricted kernel at $\tau = 0.05$.

The agreement across separated curved surfaces provides a controlled check that the factorization can recover the intended restricted kernel.

F.4.5 Factor-Rank Sweep

The data-fit objective reaches its elbow at $L = 8$. The recovered rank remains 7 from $L = 8$ through $L = 64$. At $L = 256$, additional singular values lie at the reporting threshold and increase the reported rank without improving the objective. We use $L = 16$ in the main experiments to remain beyond the elbow. The held-out error for this separate optimization experiment is 0.106 on 16 sources.

The stable rank over the range around the elbow motivates the conservative $L = 16$ choice used in the learned heat kernels.

F.5 GObO enables rapid source evaluation and nonlinear corrections

F.5.1 Repeated-Source Timing

All interquartile ranges are within about 3% of the median. Dense and streaming evaluation agree to 5.2×10^{-7}. The matched-accuracy finite-volume setting has relative error 0.0096 against the production solve.

The one-time conditioning cost is amortized over every later source. The implemented costs below separate this one-time work from dense and streaming queries.

Together, the measurements and costs show that conditioning is paid once and that later sources use the explicit kernel without another network pass.

F.5.2 Resolution Transfer

Coarse computation followed by fine synthesis and coarse conditioning followed by a fine-grid query both reach the 0.026 reference discretization gap. The second path is specific to GObO because the direct FNO consumes the source before its learned layers.

The numerical values separate end-to-end fine-grid evaluation from reuse of the coarse-grid operator. Figure 10 makes this distinction explicit by tracing where the fine-grid source enters each model.

The end-to-end fine-grid path is sensitive to how coordinates are represented. The operator-reuse paths do not evaluate the encoder on the fine grid and are unaffected by this choice.

The grid-independent basis evaluation, rather than end-to-end fine-grid inference, is what makes the strongest transfer modes reach the numerical discretization floor.

F.5.3 Radiative Feedback

Both the explicit and linearized implicit corrections have relative error 0.027 ± 0.001 over 128 samples and 3 seeds. Their absolute errors are 6.7×10^3 and 6.5×10^3. The uncorrected linear trace ranges from 0.002 to 0.31 relative error across samples. The curves show that the correction removes the peak overshoot and follows the final profile.

The corrected trajectories show that feedback can be added around the frozen linear kernel without retraining it on radiative data.

search rank L	objective	recovered rank	coefficient error
coefficient optimum	1.804×10^{-4}	7	n/a
1	2.593×10^{-3}	1	0.986
2	8.714×10^{-4}	2	0.730
4	2.563×10^{-4}	4	0.380
8	1.826×10^{-4}	7	0.215
16	1.788×10^{-4}	7	0.160
32	1.801×10^{-4}	7	0.120
64	1.810×10^{-4}	7	0.095
256	1.808×10^{-4}	18^*	0.039

Table 26: **Complete search-rank sweep.** Recovered rank uses the threshold $10^{-4}\sigma_1$, coefficient error is $\|A - A_{\mathrm{cvx}}\|_F / \|A_{\mathrm{cvx}}\|_F$. The objective reaches its elbow at $L = 8$, and $L = 16$ is used in the heat experiments. At $L = 256$, the additional singular values lie at the reporting threshold, making the rank count threshold-sensitive.

operation	frequency	CPU	GPU $b = 1$	GPU $b = 64$/sample
FV assembly and preconditioner	per medium	0.6 s	n/a	n/a
FV production trajectory	per source	63.5 s	n/a	n/a
FV matched-accuracy trajectory	per source	4.6 s	n/a	n/a
FNO-2D forward pass	per source	194.1 ms	2.50 ms	0.38 ms
GObO conditioning	per medium	218.3 ms	10.1 ms	4.29 ms
GObO batch response	per source	1.45 ms	0.45 ms	0.008 ms
GObO streaming response	per source	57.0 ms	25.9 ms	0.41 ms

Table 27: **Operation-level timing.** A batch of 64 sources shares one conditioned GObO, the finite-volume solver was evaluated on CPU only.

method	time	memory
finite-volume solver	$O(N_t n_{\mathrm{sub}} I_{\mathrm{CG}} N^3 \log N)$	$O(N^3)$
FNO-2D	$O(PN^2(C^2 + C \log N))$	$O(N^2 C)$
GObO conditioning	$O(N^3)$	operator factors
GObO batch	$O(N_t[N^2 \log N + RL(D_S + D_O)] + N_t^2 RL)$	$O(N_t N^2 + N_t^2 R)$
GObO streaming	$O(N_t[N^2 \log N + RL(D_S + D_O)])$	$O(N^2 + RL)$

Table 28: **Implemented asymptotic costs.** Streaming removes the quadratic dependence on history length.

F.5.4 Temperature-Dependent Conduction

The exact transform in configuration A keeps GObO near its linear accuracy across all contrasts and shifted sources. The trace-based update in configuration B also improves every tested setting without retraining.

The table establishes the trend over all 3 conductivity contrasts and both source classes. The severe examples below show how the corrections change the spatial temperature profile rather than only reducing an aggregate norm.

Both constructions preserve the source-transfer benefit of the linear model while correcting the nonlinear material response.

F.5.5 Fixed-Point Comparison

The fixed-point study separates convergence of the outer iteration from the surrogate's accuracy on the effective sources generated by that iteration. GObO improves substantially under both residual constructions, while the direct FNO remains inaccurate on shifted sources.

The iteration results attribute the remaining shifted-source error to the surrogate rather than to failure of the nonlinear outer solve.

F.6 Optimization with additional search rank

The following argument provides intuition for the effect of search rank observed in Table 26. We first treat the factors as free variables and hold the rates fixed. In the conditional model, however, the factors are outputs of a neural network, so the connection to our experiments is only qualitative.

For fixed rates, the observations depend linearly on the coefficient matrices $A = (A_1, \dots, A_M)$. The squared loss

$$\mathcal{L}(A) = \sum_i \|\mathcal{P}_i(A) - v_i\|_2^2, \qquad \mathcal{P}_i \text{ linear}, \tag{127}$$

is therefore a convex quadratic in A. Introducing the factorizations $A_k = U_k V_k^\top$ gives

$$f(U, V) = \mathcal{L}\left(U_1 V_1^\top, \dots, U_M V_M^\top\right), \tag{128}$$

which is generally nonconvex in $U_k \in \mathbb{R}^{D_O \times L}$ and $V_k \in \mathbb{R}^{D_S \times L}$. In coefficient space, this resembles asymmetric low-rank matrix sensing with one measurement block per temporal mode. Our

mode	GObO	FNO-2D
M0. compare truths at 64 and 128	0.026	
M1. end-to-end at 128	0.078 ± 0.022	0.044 ± 0.002
M2. coarse compute, fine readout	$\mathbf{0.025 \pm 0.000}$	0.059 ± 0.000
M3. condition at 64, operator at 128	$\mathbf{0.025 \pm 0.000}$	n/a

Table 29: **Resolution transfer** $64 \rightarrow 128$ (relative L_2 against truth at 128, 3 seeds, 64 paired samples). M2 and M3 reach the discretization floor, M3 is structurally unavailable to FNO.

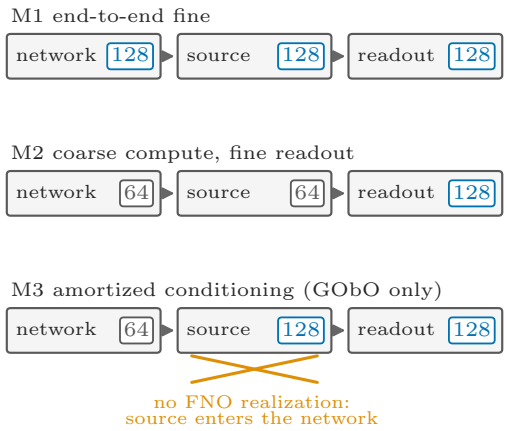


Figure 10: **Resolution-transfer paths.** The source enters FNO before its learned layers, so there is no FNO analogue of the GObO-only M3 path.

instance is highly ill-conditioned: the temporal design matrix in Appendix E.4.3 has condition number 5.8×10^{14}, and the objective has no balancing regularizer.

Let $G_k := \nabla_{A_k} \mathcal{L}$. The first-order stationary conditions for the factored objective are

$$G_k V_k = 0, \qquad G_k^\top U_k = 0 \qquad \text{for every } k. \tag{129}$$

These conditions are weaker than $G_k = 0$. For example, they hold for a given slot when $U_k = V_k = 0$, regardless of its coefficient gradient. The factored objective can therefore be stationary even when the corresponding coefficient matrices do not minimize equation 127 (Ha et al., 2020, eq. (8)).

Rank deficiency provides a useful connection between the two objectives. For a twice differentiable function g and a factorization $X = AB^\top$ with L columns, every second-order stationary point of $g(AB^\top)$ satisfying $\operatorname{rank}(AB^\top) < L$ also satisfies $\nabla g(AB^\top) = 0$ (Ha et al., 2020, Theorem 1). Related arguments appear for Burer–Monteiro factorizations and regularized factorization models (Burer & Monteiro, 2005; Bhojanapalli et al., 2018; Haeffele & Vidal, 2015). Intuitively, unused factor directions allow the second-order behavior to detect coefficient gradients that are hidden from the first-order conditions.

Applying this results to one rate slot while holding the others fixed, a second-order stationary point with $\operatorname{rank}(A_k) < L$ must satisfy $G_k = 0$. If every slot is rank deficient, then $\nabla_A \mathcal{L} = 0$, and convexity makes A a global minimizer of equation 127. At $L = 1$, this mechanism applies only to zero-rank slots, whereas additional columns allow unused directions even for nonzero coefficient matrices.

Related theoretical motivation for increasing search rank comes from positive-semidefinite Burer–Monteiro factorizations of smooth, strongly convex objectives. Sufficient rank overparameterization makes every second-order stationary point globally optimal, whereas the corresponding guarantee without overparameterization requires an objective with condition number below three (Zhang, 2025). These results suggest potential optimization benefits from increasing the search rank. However, they do not cover our rectangular, multi-rate factorization, and our coefficient-space loss need

M1 positional encoding	GObO	FNO-2D
raw cell-index features	0.292 ± 0.056	0.047 ± 0.001
continuous reference features	0.078 ± 0.022	0.044 ± 0.002

Table 30: **Positional-encoding ablation for end-to-end fine evaluation.** M2 and M3 do not run the encoder on the fine grid and are unaffected by this choice.



(a) peak temperature over time

(b) hottest-row profile at $t = T$

Figure 11: **Radiative feedback with the original linear checkpoint.** (a) Peak surface temperature over time, the shaded band between the unchanged linear kernel and the truth is the heat that radiation removes. (b) Temperature at $t = T$ along the hottest row. The unchanged kernel overshoots the peak, whereas the streamed semi-implicit correction captures both saturation and the final profile without retraining.

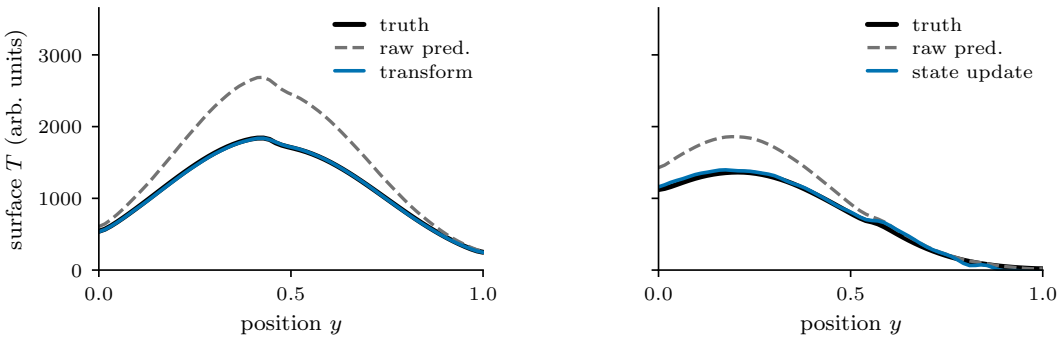
not satisfy the required strong-convexity assumptions. Nor is there a guarantee that our optimizer reaches a rank-deficient second-order stationary point.

The remaining stationary points depend on the measurement operator. Under suitable restricted-isometry assumptions, factored matrix-sensing problems have no spurious local minima and their saddle points are strict (Ge et al., 2017). Counterexamples show that weaker isometry conditions can still permit spurious local minima (Zhang et al., 2018). We have not established these assumptions for our highly ill-conditioned temporal design, so neither result directly characterizes our loss. Optimization may also be slow without convergence to a spurious minimum. Gradient descent can be sensitive to ill-conditioning in low-rank estimation (Tong et al., 2021). In related factored linear models, singular directions may be learned sequentially with plateaus between them (Saxe et al., 2014), and even strict saddle points can take a long time to escape (Du et al., 2017). The final loss of a finite training run therefore cannot distinguish these plateaus from convergence to a poor local minimum.

The results in Table 26 are consistent with the theoretical motivation for increasing search rank. At $L = 1$, the attained loss remains more than an order of magnitude above the convex coefficient-space optimum, whereas for $L \geq 8$, it approaches that optimum to within approximately one percent. Although increasing L affects both the set of representable coefficient matrices and the optimization landscape, the observed improvement provides empirical support for the search rank used in the remaining experiments. A theoretical characterization of the observed dependence on search rank is left to future work.

		mild 0.2		strong 0.5		severe 1.0	
method	retraining	static	shifted	static	shifted	static	shifted
Configuration A. exact Kirchhoff transform							
FNO trained for each setting	6	**0.001**	0.104	**0.001**	0.116	**0.001**	0.103
linear FNO + transform	0	**0.001**	0.096	0.002	0.107	0.002	0.099
GObO, no correction	0	0.053	0.069	0.122	0.151	0.232	0.263
GObO + exact transform	0	0.011	**0.009**	0.010	**0.009**	0.010	**0.009**
Configuration B. trace-based state update							
FNO trained for each setting	6	**0.001**	0.101	**0.001**	0.104	**0.001**	0.108
GObO, no correction	0	0.025	0.031	0.054	0.064	0.096	0.115
GObO + trace-based update	0	0.014	**0.012**	0.021	**0.019**	0.030	**0.030**

Table 31: **Temperature-dependent conduction without retraining.** Relative errors are means over 3 seeds. Every standard deviation is at most 0.002 for the GObO rows and at most 0.007 for the retrained FNO. Shifted sources include scans, rasters, and sums of 2 pulses.



(a) configuration A. exact Kirchhoff transform

(b) configuration B. trace-based state update

Figure 12: **2 nonlinear material laws corrected with 1 linear checkpoint.** Surface temperature along the hottest row at $t = T$ for the severe-contrast tier ($k_{\max}/k_{\min} = 2$), 1 static test sample per configuration. Each panel overlays the truth, the uncorrected GObO prediction, and the corrected prediction. Panel (a) uses the exact Kirchhoff transform and panel (b) uses the trace-based state update. Both panels share axes, line styles, and temperature scale. Table 31 quantifies the contrast trends.

model	residual/state	steps	mild static	mild shifted	strong static	strong shifted
FNO trained for setting	n/a	0	0.001	0.101	0.001	0.104
linear FNO + iteration	time derivative	≤ 20	0.011 (2)	0.097 (29)	0.026 (6)	0.093 (26)
linear GObO + iteration	time derivative	≤ 20	0.011 (3)	**0.009** (2)	0.015 (3)	**0.014** (3)
linear GObO + iteration	generator	≤ 20	0.011 (3)	**0.009** (2)	0.016 (4)	**0.014** (3)
GObO + trace-based update	generator state	1	0.014	0.012	0.021	0.019
FV solver + iteration	time derivative	≤ 20	0.002 (1)	n/a	0.007 (3)	n/a

Table 32: **Fixed-point attribution for configuration B.** Entries are median relative errors over non-diverged samples, parentheses give half the interquartile range in units of the last digit.