Knowledge-Informed Kernel State Reconstruction from Heterogeneous Partial Observations

Luca Muscarnera [*1] **Silas Ruhrberg Estévez** [*1] **Samuel Holt** [1] **Evgeny Saveliev** [1] **Mihaela van der Schaar** [1]

Abstract

Real-world scientific systems are rarely observed through complete, regularly sampled state trajectories. Instead, measurements are often partial, noisy, and heterogeneous, providing fragmented views of latent dynamical states. We introduce `MAAT` (Model Aware Approximation of Trajectories), a framework for knowledge-informed Kernel State Reconstruction in partially observed dynamical systems. `MAAT` formulates reconstruction in a reproducing kernel Hilbert space and incorporates heterogeneous observation operators together with semantic and structural priors, including non-negativity, conservation constraints, and domain-specific measurement models. This yields smooth, physically consistent state estimates with analytic time derivatives, providing a principled interface between fragmented measurements and downstream mechanistic discovery methods such as symbolic regression. Across nine scientific benchmarks, multiple noise regimes, and a real-world COVID-19 dataset, `MAAT` substantially reduces trajectory and derivative reconstruction error relative to strong baselines.

1. Introduction

In many scientific and clinical domains, the variables are not directly observable (Rubanova et al., 2019). Instead, systems are measured through heterogeneous data sources that provide only partial and indirect information about the underlying dynamics. For example, in oncology, tumour burden or clonal composition cannot be directly or continuously observed (Bartolomucci et al., 2025). Clinicians instead rely on sparse, high-specificity measurements such as imaging or genomic assays, alongside dense but indirect biomarkers derived from blood panels or physiological signals (Sivapalan et al., 2023). These observations differ in temporal resolution, noise characteristics, and their relationship to the latent state, making it non-trivial to reconstruct a coherent trajectory of disease progression.

This setting induces a fundamental data fusion problem: recovering physically meaningful latent trajectories from fragmented observations with heterogeneous structure. Mechanistic models expressed as differential equations provide a principled framework for reasoning about such systems (Strogatz, 2018; Huang et al., 2013; Yu & Wang, 2024). However, their application in practice is limited by the lack of reliable state trajectories and derivatives.

Existing approaches treat this reconstruction problem only partially. Classical smoothing methods produce continuous trajectories but ignore domain constraints and heterogeneous observation operators (Rasmussen & Williams, 2005). State-space models enable sensor fusion but require specifying transition dynamics a priori (Kalman, 1960). More flexible latent models accommodate partial observations but rely on black-box representations that obscure mechanistic structure (Chen et al., 2018). As a result, state reconstruction is typically treated as a preprocessing step, rather than as a *representational bottleneck* that determines which downstream analyses are possible. We address this limitation by formulating state reconstruction as a knowledge-informed inference problem in function space, rather than as a purely numerical preprocessing step.

Contributions

Conceptual. We identify knowledge-informed state reconstruction from heterogeneous partial observations as a central bottleneck for mechanistic modeling.

Technical. We introduce `MAAT`, a kernel-based framework that embeds latent trajectories in a reproducing kernel Hilbert space and incorporates heterogeneous observation operators together with semantic and structural priors directly into the reconstruction objective.

Empirical. Across nine benchmark dynamical systems and a real-world COVID-19 dataset, `MAAT` improves trajectory and derivative reconstruction under multiple noise regimes relative to strong baselines.

[*]Equal contribution [1]DAMTP, University of Cambridge, Cambridge, UK. Correspondence to: Luca Muscarnera <lm2152@cam.ac.uk>.

Accepted at ICML 2026 SD4H Workshop

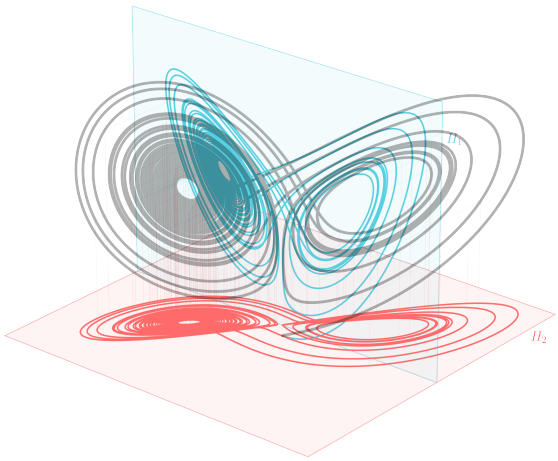

Figure 1. Observation operators represent different projections of the true underlying dynamics, that are observable through dense sampling

2. Problem Formulation

We consider a time-dependent dynamical system characterized by the state variable $x(t) \in \mathbb{R}^d$, governed by a system of first-order ordinary differential equations (ODEs):

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0, \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ is an unknown vector field. In real-world settings, the time derivative $\dot{x}(t)$ is not directly observable. Measurements instead provide partial and noisy observations of the latent state $x(t)$, yielding a dataset $\mathcal{D} = \{(t_i, y_i, \mathcal{H}_i)\}_{i=1}^N$ of N observations collected at irregular timestamps t_i. Each observation y_i is related to the latent state $x(t_i)$ through a linear observation operator $\mathcal{H}_i$:

$$y_i = \mathcal{H}_i(x(t_i)) + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \Sigma), \tag{2}$$

where ϵ_i denotes measurement noise. The generalized observation model captures a core trade-off in experimental design: constraints on sensing, cost, and invasiveness necessitate balancing temporal resolution against state specificity. Consequently, datasets often combine sparse direct measurements (e.g., gene expression) with dense aggregated signals (e.g., blood biomarkers). Moreover, available domain knowledge can be leveraged not only via statistical regularization, but by enforcing interpretable, mechanistic constraints to guide state reconstruction.

Optimization Framework. Standard smoothing techniques such as splines fail to account for the structural constraints $\mathcal{C}$ or the heterogeneous operators $\mathcal{H}_i$. Instead, we formulate the reconstruction as a risk minimization problem in a Reproducing Kernel Hilbert Space (RKHS), denoted by $\mathcal{H}_K$. We seek the trajectory function $\hat{x} \in \mathcal{H}_K$ that minimizes the regularized empirical risk including domain knowledge of the form:

$$\begin{aligned}\hat{x} = \operatorname*{argmin}_{x \in \mathcal{H}_K} & \underbrace{\sum_{i=1}^{N} \|y_i - \mathcal{H}_i(x(t_i))\|^2}_{\text{Data Fidelity}} + \\ & + \underbrace{\lambda_1 \|x\|^2_{\mathcal{H}_K}}_{\text{Smoothness}} + \underbrace{\lambda_2 \mathcal{R}_{\text{phys}}(x, \mathcal{C})}_{\text{Physical Priors}},\end{aligned} \tag{3}$$

where $\|\cdot\|_{\mathcal{H}_K}$ denotes the RKHS norm, and $\mathcal{R}_{\text{phys}}$ is a penalty term enforcing domain knowledge (defined in Section 3). This formulation is equivalent to Maximum A Posteriori (MAP) estimation under Gaussian process priors, providing a principled framework to reconstruct derivatives.

3. Knowledge-informed Kernel Regression

We introduce `MAAT` (Model Aware Approximation of Trajectories), a framework for knowledge informed kernel regression for state estimation in dynamical systems. Unlike prior methods, `MAAT` goes beyond simple interpolation of the measurements but does the regression under the constraints provided by the physical knowledge about the variables in the systems and the interactions of the subcomponents. A schematic overview of the framework is provided in Figure 2:

Kernel State Reconstruction (KSR). We model the trajectory of each state variable $j \in \{1, \ldots, d\}$ as a function in a Reproducing Kernel Hilbert Space (RKHS), $\widehat{x}_j(t) = \sum_{\ell=1}^{N} u_{\ell j} \kappa(t, t_\ell)$, where κ is a smooth kernel (e.g., Gaussian) and $U \in \mathbb{R}^{N \times d}$ is a matrix of coefficients to be learned. This is a form of kernel ridge regression, and follows from the Representer Theorem. The coefficients U are found by minimizing a composite loss function that balances fidelity to both snapshots and linear signals, along with regularization terms:

$$\begin{aligned}\min_U \frac{w_s}{N_{\text{obs}}} \|\mathbf{K}^{\text{obs}}\mathbf{U} - \mathbf{X}^{\text{obs}}\|_F^2 + \sum_i \frac{w_i}{N} \|\mathbf{K}\mathbf{U}\mathbf{H}_i^\top - \mathbf{Y}\|_F^2 \\ + \gamma \|\dot{\mathbf{K}}\mathbf{U} - F(\mathbf{K}\mathbf{U})\|_F^2 + \lambda \|\mathbf{U}\|_F^2,\end{aligned} \tag{4}$$

where $K^{\text{obs}}_{k\ell} = \kappa(t^{\text{obs}}_k, t_\ell)$ and $K_{i\ell} = \kappa(t_i, t_\ell)$ are, respectively, the matrix of the kernel-based inner products between the timesteps where full state is observed and where the signal is observed and the matrix of the kernel-based inner products between the timesteps in which the fine grained signals are observed. The additional penalty from deviation from a prior ($\gamma \|\dot{\mathbf{K}}\mathbf{U} - F(\mathbf{K}\mathbf{U})\|_F^2$) models the deviation from prior hypotheses on the dynamics of the system.

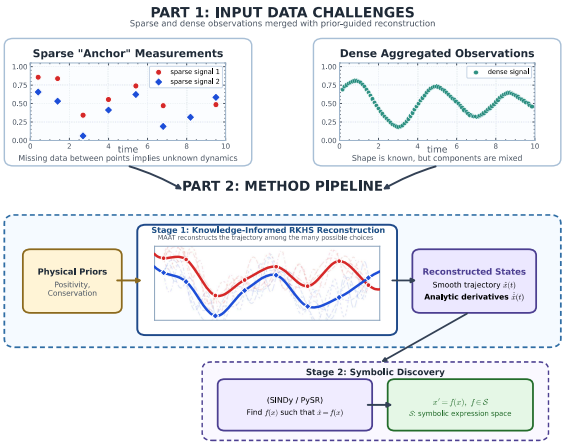

Figure 2. **Overview of MAAT.** Sparse anchor measurements and dense aggregate observations are combined with physical priors to produce reconstructed states through knowledge-informed kernel regression yielding smooth trajectories and analytical derivatives that can be used for symbolic regression.

The object $\dot{\mathbf{K}}$ denotes the time derivative of the kernel Gram matrix $\mathbf{K}$. To see how this arises, note that the time derivative of the reconstructed trajectory at a new time point t is given by

$$\partial_t \hat{\boldsymbol{x}}(t) = \partial_t \sum_j \boldsymbol{e}_j \kappa(t, t_j) = \mathbf{U}\, \partial_t \kappa(t, \boldsymbol{t}),$$

where the equality follows from the linearity of the model. Thus, the derivative operator acts only on the kernel. Since commonly used kernels (e.g., Gaussian kernels) are $\mathcal{C}^\infty$, their derivatives admit closed-form expressions and can be computed efficiently.

Theoretical Justification. This KSR approach is motivated by two key theoretical properties. First, the composite loss function is a valid surrogate for the true L^2 reconstruction error.

Lemma 1 (Composite loss is a calibrated surrogate). Let $H : \mathbb{R}^d \to \mathbb{R}^p$ be a bounded linear observation operator. For any candidate trajectory $\hat{x} \in L^2([0,T];\mathbb{R}^d)$ and true trajectory x, define the risk $\mathcal{R}(\hat{x}) = \|x - \hat{x}\|_{L^2}^2 + \|H(x - \hat{x})\|_{L^2}^2$. Then

$$\|x - \hat{x}\|_{L^2}^2 \;\le\; \mathcal{R}(\hat{x}) \;\le\; (1 + \|H\|^2)\, \|x - \hat{x}\|_{L^2}^2.$$

Hence minimizing $\mathcal{R}$ is equivalent to minimizing the L^2 reconstruction error up to a constant factor.

Sketch. The upper bound follows from the triangle inequality and the definition of the operator norm, $\|H(x-\hat{x})\|_{L^2}^2 \le \|H\|^2 \|x - \hat{x}\|_{L^2}^2$. The lower bound is immediate. A full proof is in the Appendix B. □

Second, KSR provides derivative estimates that are fundamentally more robust to noise than standard numerical differentiation.

Proposition 1 (FD noise floor vs KSR). Assume additive i.i.d. noise with variance σ^2 on $x(t)$ sampled with step Δt. Central differences yield derivative error $\mathbb{E}\left[\|\widehat{x}_{\mathrm{FD}} - \dot{x}\|_2^2\right] = \mathcal{O}(\Delta t^4) + \Omega(\sigma^2/\Delta t^2)$, which has an irreducible error floor. For KSR with regularization λ, the analytic derivative estimator satisfies $\mathbb{E}\left[\|\widehat{x}_{\mathrm{KSR}} - \dot{x}\|_2^2\right] = \mathcal{O}(\lambda) + \mathcal{O}(\sigma^2/n)$. Thus, unlike finite differences, KSR avoids high-frequency noise amplification and admits a standard bias–variance trade-off. A proof sketch is provided in the Appendix B. This robustness is critical, as accurate derivatives are the most important input for successful symbolic regression (Brunton et al., 2016).

4. RELATED WORK

We position MAAT within the landscape of derivative estimation and knowledge-informed machine learning using the structural criteria in Table 4. A broader comparison with prior work is deferred to Appendix C.

Numerical and Smoothing Baselines. Classical derivative estimation forms the foundation of most symbolic regression pipelines. While *Finite differences* are computationally efficient, they amplify noise in low-SNR regimes. Windowed methods like *Savitzky–Golay* (Steinier et al., 1972) and variational approaches like *TVRegDiff* (Chartrand, 2011) improve robustness but provide discrete numerical outputs rather than analytic forms. While *Cubic Splines* (de Boor, 1978) and *RBF Smoothing* (Buhmann, 2003) provide differentiable surrogates, they typically only operate on single-channel, regularly sampled data.

Probabilistic and Filtering Estimators. *Gaussian Processes (GPs)* (Rasmussen & Williams, 2005) and *Kalman Filters* (Kalman, 1960) offer a principled treatment of uncertainty and irregular sampling. GPs provide analytic derivatives through kernel differentiation; however, standard kernels struggle to scale to high-dimensional dynamical systems and do not natively support structural physical priors like mass conservation across state transitions.

Deep Latent Dynamics. Modern deep learning approaches, such as *Neural ODEs* (Chen et al., 2018), utilize neural vector fields to represent latent dynamics. *Universal Differential Equations* (UDEs) (Rackauckas et al., 2020) embed neural networks within mechanistic scaffolds, improving inductive bias and data efficiency, but they do not address heterogeneous observation operators.

Comparison with Physics-Informed Kernel Learning A distinct but complementary line of research is the recently proposed *Physics-Informed Kernel Learning* (PIKL) framework (Doumèche et al., 2025). Similar to MAAT, PIKL reformulates the learning problem in a Reproducing Kernel Hilbert Space (RKHS) to overcome the training

Table 1. State reconstruction MSE (↓) semi-synthetic benchmark datasets. Values are mean ± confidence interval. Best result for each dataset-backend pair is bolded. Noise type: Gaussian.

Method	Backend	Dynamical systems			Epidemiology / dynamics			Oncology / viral		
		CRC	Cons.	Neut.	SEIR	SEIRH	TMDD	Tumor	TDI	Viral
RBF	PySR	$1.5 \times 10^{-1} \pm 1.8 \times 10^{-2}$	$1.2 \times 10^{2} \pm 1.7 \times 10^{1}$	$2.9 \times 10^{-2} \pm 2.4 \times 10^{-2}$	$1.1 \times 10^{-2} \pm 9.2 \times 10^{-4}$	$9.0 \times 10^{-3} \pm 7.9 \times 10^{-4}$	$1.4 \times 10^{0} \pm 1.3 \times 10^{-1}$	$1.2 \times 10^{1} \pm 1.2 \times 10^{0}$	$5.6 \times 10^{1} \pm 1.4 \times 10^{1}$	$9.2 \times 10^{-3} \pm 1.4 \times 10^{-3}$
	SINDy	$1.5 \times 10^{-1} \pm 1.8 \times 10^{-2}$	$1.2 \times 10^{2} \pm 1.7 \times 10^{1}$	$2.9 \times 10^{-2} \pm 2.3 \times 10^{-2}$	$1.1 \times 10^{-2} \pm 9.2 \times 10^{-4}$	$9.0 \times 10^{-3} \pm 7.9 \times 10^{-4}$	$1.4 \times 10^{0} \pm 1.3 \times 10^{-1}$	$1.2 \times 10^{1} \pm 1.3 \times 10^{0}$	$5.7 \times 10^{1} \pm 1.5 \times 10^{1}$	$9.3 \times 10^{-3} \pm 1.4 \times 10^{-3}$
Cubic	PySR	$2.3 \times 10^{-1} \pm 4.1 \times 10^{-2}$	$2.1 \times 10^{2} \pm 4.7 \times 10^{1}$	$4.7 \times 10^{-2} \pm 3.2 \times 10^{-2}$	$1.8 \times 10^{-2} \pm 1.9 \times 10^{-3}$	$1.5 \times 10^{-2} \pm 2.9 \times 10^{-3}$	$2.2 \times 10^{0} \pm 2.3 \times 10^{-1}$	$2.0 \times 10^{1} \pm 3.2 \times 10^{0}$	$1.6 \times 10^{2} \pm 4.1 \times 10^{1}$	$1.5 \times 10^{-2} \pm 2.6 \times 10^{-3}$
	SINDy	$2.3 \times 10^{-1} \pm 4.1 \times 10^{-2}$	$2.1 \times 10^{2} \pm 4.5 \times 10^{1}$	$4.7 \times 10^{-2} \pm 3.2 \times 10^{-2}$	$1.8 \times 10^{-2} \pm 2.0 \times 10^{-3}$	$1.5 \times 10^{-2} \pm 2.9 \times 10^{-3}$	$2.2 \times 10^{0} \pm 2.3 \times 10^{-1}$	$1.9 \times 10^{1} \pm 2.7 \times 10^{0}$	$1.6 \times 10^{2} \pm 4.0 \times 10^{1}$	$1.5 \times 10^{-2} \pm 2.5 \times 10^{-3}$
GP	PySR	$3.7 \times 10^{-1} \pm 3.3 \times 10^{-1}$	$1.8 \times 10^{2} \pm 1.9 \times 10^{2}$	$7.5 \times 10^{-2} \pm 6.8 \times 10^{-2}$	$5.9 \times 10^{-3} \pm 6.9 \times 10^{-3}$	$2.9 \times 10^{-3} \pm 2.1 \times 10^{-3}$	$\mathbf{8.4 \times 10^{-2} \pm 3.5 \times 10^{-2}}$	$1.2 \times 10^{1} \pm 1.4 \times 10^{1}$	$2.9 \times 10^{2} \pm 2.8 \times 10^{2}$	$3.1 \times 10^{-2} \pm 4.8 \times 10^{-2}$
	SINDy	$3.1 \times 10^{-1} \pm 2.7 \times 10^{-1}$	$1.6 \times 10^{2} \pm 1.8 \times 10^{2}$	$1.7 \times 10^{-1} \pm 4.1 \times 10^{-1}$	$2.6 \times 10^{-3} \pm 2.2 \times 10^{-3}$	$8.6 \times 10^{-4} \pm 4.2 \times 10^{-4}$	$8.7 \times 10^{-2} \pm 2.9 \times 10^{-2}$	$1.3 \times 10^{1} \pm 1.7 \times 10^{1}$	$3.7 \times 10^{2} \pm 3.4 \times 10^{2}$	$1.3 \times 10^{-2} \pm 1.2 \times 10^{-2}$
Kalman	PySR	$1.1 \times 10^{-2} \pm 1.4 \times 10^{-3}$	$\mathbf{9.0 \times 10^{0} \pm 1.5 \times 10^{0}}$	$2.5 \times 10^{-3} \pm 2.5 \times 10^{-3}$	$7.8 \times 10^{-4} \pm 8.6 \times 10^{-5}$	$7.7 \times 10^{-4} \pm 9.5 \times 10^{-5}$	$1.0 \times 10^{-1} \pm 1.6 \times 10^{-2}$	$8.7 \times 10^{-1} \pm 1.3 \times 10^{-1}$	$5.1 \times 10^{1} \pm 1.4 \times 10^{1}$	$7.1 \times 10^{-4} \pm 1.4 \times 10^{-4}$
	SINDy	$1.1 \times 10^{-2} \pm 1.4 \times 10^{-3}$	$1.3 \times 10^{1} \pm 1.9 \times 10^{0}$	$2.5 \times 10^{-3} \pm 2.3 \times 10^{-3}$	$8.4 \times 10^{-4} \pm 8.3 \times 10^{-5}$	$7.8 \times 10^{-4} \pm 9.6 \times 10^{-5}$	$1.0 \times 10^{-1} \pm 1.5 \times 10^{-2}$	$8.8 \times 10^{-1} \pm 1.3 \times 10^{-1}$	$4.7 \times 10^{1} \pm 1.2 \times 10^{1}$	$8.1 \times 10^{-4} \pm 1.3 \times 10^{-4}$
Linear	PySR	$7.6 \times 10^{-2} \pm 9.0 \times 10^{-3}$	$6.3 \times 10^{1} \pm 9.2 \times 10^{0}$	$1.5 \times 10^{-2} \pm 1.3 \times 10^{-2}$	$5.7 \times 10^{-3} \pm 5.4 \times 10^{-4}$	$4.7 \times 10^{-3} \pm 4.5 \times 10^{-4}$	$7.0 \times 10^{-1} \pm 6.1 \times 10^{-2}$	$6.0 \times 10^{0} \pm 7.7 \times 10^{-1}$	$5.1 \times 10^{1} \pm 1.2 \times 10^{1}$	$4.7 \times 10^{-3} \pm 8.0 \times 10^{-4}$
	SINDy	$7.7 \times 10^{-2} \pm 8.9 \times 10^{-3}$	$6.4 \times 10^{1} \pm 9.4 \times 10^{0}$	$1.5 \times 10^{-2} \pm 1.3 \times 10^{-2}$	$5.7 \times 10^{-3} \pm 5.5 \times 10^{-4}$	$4.8 \times 10^{-3} \pm 4.5 \times 10^{-4}$	$7.0 \times 10^{-1} \pm 6.2 \times 10^{-2}$	$6.0 \times 10^{0} \pm 7.4 \times 10^{-1}$	$5.2 \times 10^{1} \pm 1.3 \times 10^{1}$	$4.8 \times 10^{-3} \pm 7.9 \times 10^{-4}$
NeuralODE	PySR	$5.3 \times 10^{1} \pm 9.9 \times 10^{1}$	$6.3 \times 10^{11} \pm 1.4 \times 10^{12}$	$5.9 \times 10^{1} \pm 7.8 \times 10^{1}$	$9.3 \times 10^{-1} \pm 6.1 \times 10^{-1}$	$4.2 \times 10^{-1} \pm 1.4 \times 10^{-1}$	$1.8 \times 10^{0} \pm 1.7 \times 10^{0}$	$1.3 \times 10^{10} \pm 2.9 \times 10^{10}$	$5.1 \times 10^{3} \pm 7.1 \times 10^{3}$	$1.5 \times 10^{0} \pm 1.2 \times 10^{0}$
	SINDy	$3.0 \times 10^{0} \pm 3.4 \times 10^{0}$	$1.6 \times 10^{2} \pm 2.3 \times 10^{2}$	$2.0 \times 10^{0} \pm 3.8 \times 10^{0}$	$5.8 \times 10^{-1} \pm 3.1 \times 10^{-1}$	$3.3 \times 10^{-1} \pm 1.3 \times 10^{-1}$	$2.2 \times 10^{0} \pm 2.2 \times 10^{0}$	$1.7 \times 10^{2} \pm 1.7 \times 10^{2}$	$6.3 \times 10^{1} \pm 5.8 \times 10^{1}$	$9.1 \times 10^{-1} \pm 7.8 \times 10^{-1}$
MAAT	PySR	$\mathbf{4.0 \times 10^{-3} \pm 1.7 \times 10^{-3}}$	$2.0 \times 10^{1} \pm 1.6 \times 10^{1}$	$\mathbf{3.4 \times 10^{-4} \pm 4.6 \times 10^{-4}}$	$\mathbf{2.6 \times 10^{-5} \pm 5.0 \times 10^{-6}}$	$\mathbf{1.7 \times 10^{-5} \pm 1.5 \times 10^{-6}}$	$1.3 \times 10^{-1} \pm 2.0 \times 10^{-1}$	$\mathbf{3.0 \times 10^{-1} \pm 1.9 \times 10^{-1}}$	$\mathbf{4.9 \times 10^{0} \pm 6.5 \times 10^{0}}$	$\mathbf{4.7 \times 10^{-5} \pm 2.9 \times 10^{-5}}$
	SINDy	$\mathbf{1.5 \times 10^{-3} \pm 1.6 \times 10^{-4}}$	$\mathbf{5.8 \times 10^{0} \pm 3.1 \times 10^{0}}$	$\mathbf{4.3 \times 10^{-4} \pm 5.2 \times 10^{-4}}$	$\mathbf{7.9 \times 10^{-5} \pm 1.1 \times 10^{-5}}$	$\mathbf{4.1 \times 10^{-5} \pm 5.0 \times 10^{-6}}$	$\mathbf{4.8 \times 10^{-3} \pm 4.2 \times 10^{-4}}$	$\mathbf{1.2 \times 10^{-1} \pm 4.2 \times 10^{-2}}$	$\mathbf{1.8 \times 10^{0} \pm 4.7 \times 10^{-1}}$	$\mathbf{1.3 \times 10^{-4} \pm 2.6 \times 10^{-5}}$

instabilities and lack of theoretical guarantees associated with Physics-Informed Neural Networks (PINNs). However, the two methods target fundamentally different objectives. PIKL addresses the *forward* or *hybrid modeling* problem: it assumes the differential operator $\mathcal{D}$ is **known** a priori (e.g., the wave or heat equation) and utilizes the kernel formulation to minimize a physics-informed risk $\mathcal{J}(f) = \|Y - f(X)\|^2 + \lambda \|\mathcal{D}f\|^2$, efficiently approximating the kernel via Fourier features to solve the PDE.

5. EXPERIMENTS

We evaluate MAAT on a series of diverse benchmarks to assess its performance in state reconstruction and downstream equation discovery. To evaluate the effectiveness of our state reconstruction, we assess the derived trajectories for downstream symbolic regression (SR) using two of the most popular SR algorithms: SINDy (Brunton et al., 2016) and PySR (Cranmer, 2023).

Experimental Setup. We evaluate our method on a total of ten datasets spanning diverse domains. Detailed descriptions of the underlying dynamical systems and data-generation procedures are provided in Appendix D. For baseline comparisons in the state-reconstruction stage, we consider a broad set of standard smoothing and inference methods. Each method is used to reconstruct state trajectories and derivatives, which are then provided as input to the same symbolic regression (SR) engines for fair comparison.

Penalizing Derivative Magnitude. To test the robustness of our approach against noise, we analyze the impact of penalizing the magnitude of the derivative during reconstruction. We observed that in high-noise regimes, standard smoothing techniques often preserve high-frequency artifacts, leading to erroneous derivative estimates. By adding a penalty term to the derivative magnitude, MAAT recovers smoother, more physically plausible trajectories than baselines (see Table 1). Additional results on correlated Gaussian and Student T noise are provided in Appendix C.

Incorporating Dynamical System Structure Knowledge. We evaluated the ability of MAAT to incorporate *structural priors* derived from known system semantics. We consider the SEIR and SEIRH epidemiological models, where domain knowledge permits enforcing conservation of mass, non-negativity of all compartments, and monotonicity constraints implied by irreversible transitions (specifically, $R'(t) \geq 0$ and $S'(t) \leq 0$). By injecting this physical prior into the system, we can substantially reduce the MSE of the recovered trajectories of MAAT (see Table 2).

Table 2. Effect of structural priors on MAAT. Values are state reconstruction MSE (↓), reported as mean ± confidence interval. Best result per setting is bolded.

Noise	Dataset	PySR		SINDy	
		Plain	+priors	Plain	+priors
Corr. Gauss.	SEIR	$2.39 \times 10^{-5} \pm 3.96 \times 10^{-6}$	$\mathbf{2.00 \times 10^{-5} \pm 1.80 \times 10^{-6}}$	$7.87 \times 10^{-5} \pm 9.34 \times 10^{-6}$	$\mathbf{7.57 \times 10^{-5} \pm 9.87 \times 10^{-6}}$
	SEIRH	$1.73 \times 10^{-5} \pm 2.61 \times 10^{-6}$	$\mathbf{1.52 \times 10^{-5} \pm 1.24 \times 10^{-6}}$	$4.22 \times 10^{-5} \pm 6.04 \times 10^{-6}$	$\mathbf{3.85 \times 10^{-5} \pm 6.80 \times 10^{-6}}$
Gaussian	SEIR	$2.58 \times 10^{-5} \pm 5.01 \times 10^{-6}$	$\mathbf{2.19 \times 10^{-5} \pm 1.57 \times 10^{-6}}$	$7.91 \times 10^{-5} \pm 1.09 \times 10^{-5}$	$\mathbf{7.57 \times 10^{-5} \pm 1.01 \times 10^{-5}}$
	SEIRH	$1.71 \times 10^{-5} \pm 1.51 \times 10^{-6}$	$\mathbf{1.48 \times 10^{-5} \pm 1.64 \times 10^{-6}}$	$4.08 \times 10^{-5} \pm 4.99 \times 10^{-6}$	$\mathbf{3.72 \times 10^{-5} \pm 5.19 \times 10^{-6}}$
Student-t	SEIR	$2.37 \times 10^{-5} \pm 3.07 \times 10^{-6}$	$\mathbf{2.07 \times 10^{-5} \pm 1.29 \times 10^{-6}}$	$7.69 \times 10^{-5} \pm 8.19 \times 10^{-6}$	$\mathbf{7.38 \times 10^{-5} \pm 7.69 \times 10^{-6}}$
	SEIRH	$1.65 \times 10^{-5} \pm 1.14 \times 10^{-6}$	$\mathbf{1.36 \times 10^{-5} \pm 1.37 \times 10^{-6}}$	$4.12 \times 10^{-5} \pm 4.28 \times 10^{-6}$	$\mathbf{3.68 \times 10^{-5} \pm 4.14 \times 10^{-6}}$

Real-world data modeling. We further evaluate our method on a real-world epidemiological dataset from the COVID-19 pandemic. As shown in Table 3, MAAT achieves substantially lower reconstruction error compared to all baselines when coupled with SINDy, demonstrating its effectiveness in recovering realistic disease dynamics from noisy observational data.

6. DISCUSSION

We present MAAT, a framework for knowledge-informed state reconstruction from heterogeneous partial observations. By combining kernel-based continuous-time reconstruction with domain-specific observation operators and priors, MAAT provides physically meaningful trajectories and analytic derivatives for downstream mechanistic analysis, including symbolic regression.

Table 3. **COVID-19 benchmark (SINDy).** Test MSE across 5 seeds (mean ± 95% CI).

Method	Test MSE	95% CI
MAAT	$\mathbf{6.33 \times 10^{-5}}$	$\pm 1.07 \times 10^{-5}$
RBF	9.64×10^{-4}	$\pm 6.51 \times 10^{-4}$
Savitzky–Golay	9.73×10^{-4}	$\pm 6.47 \times 10^{-4}$
TVRegDiff	9.73×10^{-4}	$\pm 6.47 \times 10^{-4}$
Linear	9.80×10^{-4}	$\pm 6.53 \times 10^{-4}$
Kalman filter	9.89×10^{-4}	$\pm 6.68 \times 10^{-4}$
Cubic	9.99×10^{-4}	$\pm 6.72 \times 10^{-4}$
Gaussian Process	6.92×10^{-2}	$\pm 4.55 \times 10^{-2}$

Limitations. The effectiveness of MAAT depends on the

informativeness of the available observations and the validity of the injected priors. If measurements are too sparse, observation operators are uninformative, or priors are misspecified, reconstruction may be biased or underdetermined relative to purely data-driven smoothers. Our current implementation also assumes that relevant structural information, such as subsystem organization or admissible constraints, is known *a priori*. While this is natural in domains such as quantitative systems pharmacology (QSP), where modular structure is often defined through absorption, distribution, receptor binding, and response subsystems (Kaddi et al., 2018), this assumption may not hold in less structured settings.

Nonlinear/Uncertain observation operators & Future Work MAAT works under the assumption that observation operators are known and linear. While this assumption is practically reasonable in many cases, the introduction of nonlinear operators or operators subject to uncertainty may enrich the spectrum of the possible phenomena that can be captured. We note, however, that nonlinearity can be introduced in MAAT sacrificing the convenience of the regression problem in Matrix Form. In fact if we define $\mathcal{H} : \mathbb{R}^d \to \mathbb{R}^m$ as a nonlinear operator acting on $\boldsymbol{x}$, the parametrization of $\widehat{\boldsymbol{x}}$ through control variables $\{u_{ij}\}_{ij}$ is still transparent to optimization, although regularity conditions on $\mathcal{H}$ and a priori knowledge on its nature are needed.

Clinical and Translational Impact. A central motivation for MAAT is its alignment with how mechanistic modeling is practiced in medicine and pharmacology. In QSP, models are typically assembled from interpretable modules that describe biological and pharmacological processes, which are then coupled to explain patient-level outcomes (Helmlinger et al., 2019). MAAT supports this workflow by fusing sparse, subsystem-specific measurements with denser indirect signals while enforcing clinically meaningful constraints. This makes it well suited to model-informed drug development, where transparent and physiologically plausible models are needed for dose selection, safety evaluation, and regulatory decision-making (U.S. FDA, 2023; European Medicines Agency, 2018; Peterson & Riggs, 2015).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Badiale, M. and Cravero, I. A nonlinear ode model for a consumeristic society. *Mathematics*, 12(8):1253, April 2024. ISSN 2227-7390.

Bartolomucci, A., Nobrega, M., Ferrier, T., Dickinson, K., Kaorey, N., Nadeau, A., Castillo, A., and Burnier, J. V. Circulating tumor dna to monitor treatment response in solid tumors and advance precision oncology. *npj Precision Oncology*, 9(1), March 2025. ISSN 2397-768X.

Bjørnstad, O. N., Shea, K., Krzywinski, M., and Altman, N. The seirs model for infectious disease dynamics. *Nature Methods*, 17(6):557–558, June 2020. ISSN 1548-7105.

Brucker, J., Bessler, W. G., and Gasper, R. A grey-box model with neural ordinary differential equations for the slow voltage dynamics of lithium-ion batteries: Model development and training. *Journal of The Electrochemical Society*, 170(12):120537, December 2023. ISSN 1945-7111.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, March 2016. ISSN 1091-6490.

Buhmann, M. D. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, July 2003. ISBN 9780511543241.

Carnerero, A. D., Ramirez, D. R., Limon, D., and Alamo, T. Kernel-based state-space kriging for predictive control. *IEEE/CAA Journal of Automatica Sinica*, 10(5): 1263–1275, May 2023. ISSN 2329-9274.

Carè, A., Carli, R., Libera, A. D., Romeres, D., and Pillonetto, G. Kernel methods and gaussian processes for system identification and control: A road map on regularized kernel-based learning for control. *IEEE Control Systems*, 43(5):69–110, October 2023. ISSN 1941-000X.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, October 2019. ISSN 1091-6490.

Chartrand, R. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011:1–11, May 2011. ISSN 2090-5572.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Cranmer, M. Interpretable machine learning for science with pysr and symbolicregression.jl, 2023.

d'Ascoli, S., Becker, S., Schwaller, P., Mathis, A., and Kilbertus, N. ODEFormer: Symbolic regression of dynamical systems with transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TzoHLiGVMo.

de Boor, C. *A Practical Guide to Splines*, volume Volume 27. 01 1978. doi: 10.2307/2006241.

dePillis, L. Mathematical model of colorectal cancer with monoclonal antibody treatments. *British Journal of Medicine and Medical Research*, 4(16):3101–3131, January 2014. ISSN 2231-0614.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. Ode parameter inference using adaptive gradient matching with gaussian processes. In Carvalho, C. M. and Ravikumar, P. (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 216–228, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR.

Doumèche, N., Bach, F., Biau, G., and Boyer, C. Physics-informed kernel learning. *Journal of Machine Learning Research*, 26(124):1–39, 2025.

Dugan, O., Dangovski, R., Costa, A., Kim, S., Goyal, P., Jacobson, J., and Soljačić, M. Occamnet: A fast neural model for symbolic regression at scale, 2020.

European Centre for Disease Prevention and Control. Data on daily new cases of covid-19 in eu/eea countries. https://www.ecdc.europa.eu/en/publications-data/data-daily-new-cases-covid-19-eueea-country, 2020. Accessed: 2026-04-29.

European Medicines Agency. Guideline on the qualification and reporting of physiologically based pharmacokinetic (pbpk) modelling and simulation. https://www.ema.europa.eu/en/documents/scientific-guideline/draft-guideline-qualification-and-reporting-physiologically-based-pharmacokinetic-pbpk-modelling-and-simulation_en.pdf, 2018. Accessed 2025-10-02.

Friberg, L. E., Henningsson, A., Maas, H., Nguyen, L., and Karlsson, M. O. Model of chemotherapy-induced myelosuppression with parameter consistency across drugs. *Journal of Clinical Oncology*, 20(24):4713–4721, December 2002. ISSN 1527-7755.

Froese, V. and Hertrich, C. Training neural networks is NP-hard in fixed dimension. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Helmlinger, G., Sokolov, V., Peskov, K., Hallow, K. M., Kosinsky, Y., Voronova, V., Chu, L., Yakovleva, T., Azarov, I., Kaschek, D., Dolgun, A., Schmidt, H., Boulton, D. W., and Penland, R. C. Quantitative systems pharmacology: An exemplar model-building workflow with applications in cardiovascular, metabolic, and oncology drug development. *CPT: Pharmacometrics & Systems Pharmacology*, 8(6):380–395, June 2019. ISSN 2163-8306.

Holt, S., Qian, Z., and van der Schaar, M. Neural laplace: Learning diverse classes of differential equations in the laplace domain. In *ICML*, pp. 8811–8832, 2022.

Holt, S., Qian, Z., and van der Schaar, M. Deep generative symbolic regression. In *The Eleventh International Conference on Learning Representations*, 2023.

Holt, S., Qian, Z., Liu, T., Weatherall, J., and van der Schaar, M. Data-driven discovery of dynamical systems in pharmacology using large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Hsin, J., Agarwal, S., Thorpe, A., Sentis, L., and Fridovich-Keil, D. Symbolic regression on sparse and noisy data with gaussian processes. In *2025 American Control Conference (ACC)*, pp. 3170–3175. IEEE, July 2025.

Huang, S.-M., Abernethy, D. R., Wang, Y., Zhao, P., and Zineh, I. The utility of modeling and simulation in drug development and regulatory review. *Journal of Pharmaceutical Sciences*, 102(9):2912–2923, September 2013. ISSN 0022-3549.

Kacprzyk, K. and van der Schaar, M. No equations needed: Learning system dynamics without relying on closed-form ODEs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kaddi, C. D., Niesner, B., Baek, R., Jasper, P., Pappas, J., Tolsma, J., Li, J., van Rijn, Z., Tao, M., Ortemann-Renon, C., Easton, R., Tan, S., Puga, A. C., Schuchman, E. H., Barrett, J. S., and Azer, K. Quantitative systems pharmacology modeling of acid sphingomyelinase deficiency and the enzyme replacement therapy olipudase alfa is an innovative tool for linking pathophysiology and pharmacology. *CPT: Pharmacometrics & Systems Pharmacology*, 7(7):442–452, June 2018. ISSN 2163-8306.

Kaheman, K., Kutz, J. N., and Brunton, S. L. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2242), October 2020. ISSN 1471-2946.

Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1): 35–45, March 1960. ISSN 0021-9223.

Kermack, W. O. and McKendrick, A. G. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772): 700–721, August 1927. ISSN 2053-9150.

Lu, P. Y., Ariño Bernad, J., and Soljačić, M. Discovering sparse interpretable dynamics from partial observations. *Communications Physics*, 5(1), August 2022. ISSN 2399-3650.

Makke, N. and Chawla, S. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1), January 2024. ISSN 1573-7462.

Mehta, V., Char, I., Neiswanger, W., Chung, Y., Nelson, A., Boyer, M., Kolemen, E., and Schneider, J. Neural dynamical systems: Balancing structure and flexibility in physical prediction. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3735–3742, 2021. doi: 10.1109/CDC45484.2021.9682807.

Nowak, M. A. and May, R. M. *Virus dynamics: Mathematical principles of immunology and virology*. Oxford University PressOxford, November 2000. ISBN 9781383020816.

Peterson, M. and Riggs, M. Fda advisory meeting clinical pharmacology review utilizes a quantitative systems pharmacology (qsp) model: A watershed moment? *CPT: Pharmacometrics & Systems Pharmacology*, 4(3): 189–192, March 2015. ISSN 2163-8306.

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, March 2014. ISSN 0005-1098.

Qian, Z., Zame, W. R., Fleuren, L. M., Elbers, P., and van der Schaar, M. Integrating expert ODEs into neural ODEs: Pharmacology and disease progression. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Qian, Z., Kacprzyk, K., and van der Schaar, M. D-CODE: Discovering closed-form ODEs from observed trajectories. In *International Conference on Learning Representations*, 2022.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. Universal differential equations for scientific machine learning. August 2020.

Raissi, M., Perdikaris, P., and Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 0021-9991.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, November 2005. ISBN 9780262256834.

Rubanova, Y., Chen, R. T., and Duvenaud, D. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. Data-driven discovery of partial differential equations. *Science Advances*, 3(4), April 2017. ISSN 2375-2548.

Shakib, M., Tóth, R., Pogromsky, A., Pavlov, A., and van de Wouw, N. Kernel-based learning of stable nonlinear state-space models. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 2897–2902. IEEE, December 2023.

Sharma, A. and Jusko, W. J. Characteristics of indirect pharmacodynamic models and applications to clinical drug responses. *British Journal of Clinical Pharmacology*, 45 (3):229–239, March 1998. ISSN 1365-2125.

Shojaee, P., Meidani, K., Gupta, S., Farimani, A. B., and Reddy, C. K. Llm-sr: Scientific equation discovery via programming with large language models, 2024.

Simeoni, M., Magni, P., Cammia, C., De Nicolao, G., Croci, V., Pesenti, E., Germani, M., Poggesi, I., and Rocchetti, M. Predictive pharmacokinetic-pharmacodynamic modeling of tumor growth kinetics in xenograft models after administration of anticancer agents. *Cancer Research*, 64 (3):1094–1101, February 2004. ISSN 1538-7445.

Sivapalan, L., Murray, J. C., Canzoniero, J. V., Landon, B., Jackson, J., Scott, S., Lam, V., Levy, B. P., Sausen, M., and Anagnostou, V. Liquid biopsy approaches to capture tumor evolution and clinical outcomes during cancer immunotherapy. *Journal for ImmunoTherapy of Cancer*, 11(1):e005924, January 2023. ISSN 2051-1426.

Steinier, J., Termonia, Y., and Deltour, J. Smoothing and differentiation of data by simplified least square procedure. *Analytical Chemistry*, 44(11):1906–1909, September 1972. ISSN 1520-6882.

Stephens, T. gplearn: Genetic programming in python. https://gplearn.readthedocs.io/en/stable/index.html, 2019. First released 2015.

Strogatz, S. H. *Nonlinear Dynamics and Chaos*. CRC Press, 2018.

U.S. FDA. Model-informed drug development (midd) paired meeting program. https://www.fda.gov/drugs/development-resources/model-informed-drug-development-paired-meeting-program, 2023. Accessed 2025-10-02.

Virgolin, M. and Pissis, S. P. Symbolic regression is NP-hard. *Transactions on Machine Learning Research*, 2022. Preprint on arXiv:2207.01018.

Wang, H. and Zhou, X. Explicit estimation of derivatives from data and differential equations by gaussian process regression. *International Journal for Uncertainty Quantification*, 11(4):41–57, 2021. ISSN 2152-5080.

Yu, R. and Wang, R. Learning dynamical systems from data: An introduction to physics-guided deep learning. *Proceedings of the National Academy of Sciences*, 121 (27), June 2024. ISSN 1091-6490.

Zhai, Z.-M., Stern, B. D., and Lai, Y.-C. Bridging known and unknown dynamics by transformer-based machine-learning inference from sparse observations. *Nature Communications*, 16(1), August 2025. ISSN 2041-1723.

Zhou, Y., Li, J., Zhou, X., and Wang, H. Model-embedded gaussian process regression for parameter estimation in dynamical system, 2024.

Supplementary Materials for MAAT

A. Extended Related Works

Sample and Computational Complexity of Noisy Symbolic Regression Symbolic regression aims to identify an expression f from a library of primitives (variables, constants, operators, functions) that explains a dataset $\{(x_i, y_i)\}_{i=1}^N$. The challenge lies in the size of the hypothesis space: the number of candidate expressions grows combinatorially with both the input dimension d and the allowed expression depth. This renders even state-of-the-art search procedures computationally demanding. Recent theoretical work provides formal support for this intuition: (Virgolin & Pissis, 2022) show that symbolic regression is NP-hard under mild assumptions on the primitive set, implying that no polynomial-time algorithm exists unless $P = NP$. These hardness results highlight why practical SR algorithms must rely on heuristic search, structural priors, or regularization, particularly in the presence of noise.

Concretely, one can reduce a canonical NP-hard problem (e.g., SUBSET SUM or additive partitioning) to the decision problem of whether there exists a symbolic expression of bounded cost that fits the data within a specified error tolerance. Consequently, any algorithm that guarantees recovery of the globally optimal symbolic representation must, in the worst case, incur non-polynomial runtime in d. In practice, even heuristic or stochastic search procedures face the combinatorial explosion of the grammar: each additional feature or operator multiplies the number of candidate sub-expressions. Recent approaches (e.g., neural-symbolic methods (Dugan et al., 2020), transformer-based architectures, or reinforcement-learning–guided search (Makke & Chawla, 2024)) mitigate this via pruning, learned proposals, or modular decomposition. However, these strategies do not fundamentally escape the exponential scaling in d (or in the maximum depth) unless further structural assumptions, such as sparsity, separability, or modular factorization, are imposed.

Classical equation discovery. Recovering governing equations from data has a long history in system identification and sparse regression. Sparse Identification of Nonlinear Dynamics (SINDy) (Brunton et al., 2016) recovers parsimonious differential equations from feature libraries, with extensions for noise (Rudy et al., 2017) and implicit formulations (Kaheman et al., 2020). Genetic programming frameworks such as gplearn (Stephens, 2019) and evolutionary engines like PySR (Cranmer, 2023) broaden the search space beyond fixed polynomial libraries, recovering compact analytical expressions directly from data. More recently, complementary approaches replace genetic heuristics with modern deep learning: ODEFormer (d'Ascoli et al., 2024) leverages transformer architectures for symbolic regression of dynamical systems, while Deep Generative Symbolic Regression (Holt et al., 2023) employs generative models to efficiently explore the space of candidate equations. Together, these methods expand the toolkit for symbolic discovery, but they, too, typically assume full observability.

Black-box dynamical models. Neural differential equation frameworks such as Neural ODEs (Chen et al., 2018), DyNODE, and Latent ODEs (Rubanova et al., 2019) model vector fields with neural networks, enabling scalable learning under partial observability. However, this surrogate–distillation pipeline can degrade or collapse under substantial observation noise or partial observability, where errors in the learned latent representation propagate into the recovered symbolic form. These methods achieve strong predictive performance but lack closed-form interpretability. Nevertheless, they can serve as surrogates from which symbolic models are distilled, suggesting complementarity rather than exclusivity between black-box and symbolic approaches. A growing body of work therefore studies *grey-box* or *hybrid* neural ODEs, which incorporate prior knowledge such as known state variables, conservation laws, stability constraints, or partially specified equations into neural dynamics (Mehta et al., 2021; Brucker et al., 2023; Qian et al., 2021). These approaches improve identifiability and inductive bias relative to unconstrained Neural ODEs, but typically retain neural components in the governing equations, limiting closed-form interpretability and robustness under severe noise or sparse observations.

Hybrid and recent approaches. Universal Differential Equations (UDEs) (Rackauckas et al., 2020) combine mechanistic ODEs with neural components, embedding domain priors into flexible models but without mechanisms for handling fragmented heterogeneous data or structured interventions. More recent directions move beyond time-domain formulations:

Neural Laplace (Holt et al., 2022) learns in the Laplace domain to capture diverse dynamics at scale, and D-CODE (Qian et al., 2022) targets exact closed-form recovery. Large language models further extend this space: recent work (Holt et al., 2024; Shojaee et al., 2024) shows that LLMs can propose function libraries, encode domain knowledge, and refine candidate models, pointing toward a synthesis of symbolic, neural, and language-based methods.

Equation discovery under partial observations. In many scientific domains, full-state measurements are rarely available: biological, clinical, and physical systems are often only partially observed, corrupted by noise, or measured at mismatched time resolutions. This has motivated methods that seek to augment incomplete data with synthetic simulations or structural priors. For example, Zhai et al. (2025) show that supplementing scarce real measurements with synthetic trajectories can improve the identifiability of governing equations in clinical settings. Other approaches leverage probabilistic latent-variable models to infer hidden dynamics before applying regression or discovery methods (Champion et al., 2019; Lu et al., 2022). A complementary strategy is to encode known physical structure into the learning process: physics-informed neural networks (PINNs) enforce differential equation constraints within the training loss, enabling recovery of latent variables and dynamics even when only a subset of states are measured (Raissi et al., 2019). Despite these advances, most existing work assumes homogeneous data sources and does not explicitly address the integration of heterogeneous, subsystem-specific observations, an issue particularly acute in fields like pharmacology or systems biology, where data are fragmented across scales.

Equation-free yet interpretable modeling. Beyond explicit equation discovery, several recent approaches aim to balance predictive power with interpretability without producing closed-form equations. Operator-learning methods represent dynamical evolution directly through learned integral operators, enabling transparent structural analysis even in the absence of symbolic expressions (Kacprzyk & van der Schaar, 2025). Similarly, mechanistic machine learning frameworks embed known conservation laws or monotonicity constraints into neural architectures, yielding models whose behavior can be interrogated and trusted despite lacking closed-form governing equations. These approaches illustrate that interpretability need not always hinge on explicit symbolic laws, but can also arise through structural constraints and transparent operator representations.

GP smoothing for trajectories . Classical Gaussian Processes (GPs) provide nonparametric, smooth interpolants with analytic derivative posteriors via kernel differentiation (Rasmussen & Williams, 2005). When all (or most) states are observed on a single grid, a GP prior over trajectories can effectively denoise and supply $\dot{x}(t)$ estimates. However, this paradigm does not natively address *partial observability*, *multi-rate heterogeneous sensors*, or *hierarchical priors* on cross-subsystem couplings. Works that explicitly target derivative recovery with GPs leverage the fact that derivatives of a GP are again GPs, enabling closed-form posterior means/variances for $\dot{x}$ (Wang & Zhou, 2021). These methods excel as denoisers/preprocessors but typically assume either fully (or densely) observed states and do not resolve the inverse problem of reconstructing latent states from images of linear observation operators $H_i x$ collected at different rates.

A direct line of work fits GPs to each observed channel (or a multi-output GP), extracts analytic derivatives, and then applies symbolic regression (SR) such as SINDy/PySR (Hsin et al., 2025). This reduces noise sensitivity in SR but still treats observation fusion as a *per-channel smoothing* step. Handling heterogeneous observation models, enforcing block-structured priors, or constraining library couplings generally requires bespoke kernel designs and does not provide an explicit mechanism for hierarchy-aware discovery. Adaptive Gradient Matching (AGM) uses GPs to interpolate states and matches ODE right-hand sides to GP derivatives to infer parameters without direct numerical integration (Dondelinger et al., 2013). AGM targets *parameter estimation for a specified ODE family*, not discovery of new symbolic structure.

Kernel-based state reconstruction. A related line of work uses kernel methods to reconstruct latent dynamical states from indirect, noisy, or irregular observations. Early system identification approaches framed state reconstruction as a regularized inverse problem, using reproducing kernel Hilbert space (RKHS) priors to interpolate trajectories under smoothness or stability assumptions (Pillonetto et al., 2014). Subsequent work extended these ideas to multi-output and operator-valued kernels, enabling joint reconstruction of multiple state components and improved handling of correlated signals (Carnerero et al., 2023; Shakib et al., 2023). More recent methods consider kernel-based formulations for learning continuous-time dynamics from sparse or asynchronous measurements, often emphasizing well-posedness and statistical consistency of the reconstruction step (Zhou et al., 2024; Carè et al., 2023).

While these approaches provide powerful tools for denoising and interpolating partially observed trajectories, they are typically developed for *state estimation* or *parameter identification* rather than symbolic structure discovery. In particular, most kernel-based reconstruction methods assume either a single observation operator or homogeneous sensing modalities,

Table 4. **Capability matrix for state and derivative estimators.** Comparison of classical numerical methods, probabilistic estimators, and deep learning approaches. MAAT satisfies the requirements for equation discovery under partial observability and structured priors.

Method	Analytic $\dot{x}$	Handles Irregular Data	Noise Robust	Fuses Heterog. Obs.	Supports Structure	Supports Priors
Numerical & Smoothing Baselines						
Finite Differences	✗	✗	✗	✗	✗	✗
Savitzky–Golay	~	✗	~	✗	✗	✗
TVRegDiff	✗	~	✓	✗	✗	~
Cubic Spline	✓	~	~	✗	✗	~
RBF Kernels	✓	✓	~	✗	✗	✗
Probabilistic & Filtering						
Gaussian Processes	✓	✓	✓	~	~	✓
Kalman Filters	~	✓	✓	✓	✗	✓
Deep Learning & Physics-Informed						
Neural ODEs	✓	✓	✓	~	✗	✗
MAAT (Ours)	✓	✓	✓	✓	✓	✓

and they do not explicitly address the integration of heterogeneous, subsystem-specific observations or the imposition of hierarchical sparsity over downstream model components. Moreover, kernel reconstructions are usually treated as preprocessing steps, without a principled mechanism for propagating reconstruction uncertainty or structural priors into subsequent equation discovery. MAAT builds on this kernel perspective but departs from prior work by explicitly coupling kernel state reconstruction with symbolic regression, enabling joint handling of heterogeneous partial observability and structured discovery of governing dynamics.

MAAT is designed for *heterogeneous partial observability* and *structural discovery*. Its kernel state reconstruction (KSR) jointly fits multiple observation operators H_i and sampling grids to recover a coherent latent trajectory $x(t)$ and analytic $\dot{x}(t)$ *before* SR, and then injects physics-aware priors via library masks to constrain cross-subsystem couplings. An overview of the different preprocessing pipelines and their properties is given in Table 4.

B. Theoretical analysis

In this appendix we present the theoretical foundations underlying our state reconstruction algorithm. We outline the motivation for its design, provide the analytical derivation, and discuss the results that justify its adoption within our framework.

B.1. Proof of Lemma 1

Definition 1 (Equivalence Between Distances). Let (X, d) and (X, d') be two metric spaces defined on the same underlying set X through two different metrics $d : X \times X \to \mathbb{R}_+$ and $d' : X \times X \to \mathbb{R}_+$. We say that d and d' are *equivalent*, written $d \sim d'$, if there exist constants $c, C > 0$ such that

$$\exists\, c, C \in \mathbb{R}_+ : cd(x_1, x_2) \leq d'(x_1, x_2) \leq Cd(x_1, x_2) \quad \forall x_1, x_2 \in X \tag{5}$$

Corollary 1. Let d, d' be two metrics such that $d \sim d'$. Then for any point $x^* \in X$ and any sequence $\{x_n\}_n^\infty \subset X$ such that

$$\lim_{n\to\infty} d(x_n, x^*) = 0 \tag{6}$$

it also holds that

$$\lim_{n\to\infty} d'(x_n, x^*) = 0 \tag{7}$$

and vice versa.

Proof. The forward direction follows immediately by the squeeze theorem. For the reverse implication, note that by assumption

$$d(x_1, x_2) \;\leq\; \tfrac{1}{c}\, d'(x_1, x_2) \;\leq\; \tfrac{C}{c}\, d(x_1, x_2), \tag{8}$$

and similarly

$$\tfrac{c}{C}\, d(x_1, x_2) \;\le\; \tfrac{1}{C}\, d'(x_1, x_2) \;\le\; d(x_1, x_2). \tag{9}$$

Hence

$$\tfrac{1}{C}\, d'(x_1, x_2) \;\le\; d(x_1, x_2) \;\le\; \tfrac{1}{c}\, d'(x_1, x_2), \tag{10}$$

showing the equivalence is symmetric, and the claim follows. □

Lemma 2 (Composite loss is a calibrated surrogate). Let $H : \mathbb{R}^d \to \mathbb{R}^p$ be a bounded linear observation operator with operator norm $\|H\| < \infty$. For any candidate trajectory $\hat{x} \in L^2([0, T]; \mathbb{R}^d)$ and true trajectory x, define the risk functional

$$\mathcal{R}(\hat{x}) = \|x - \hat{x}\|_{L^2}^2 + \|H(x - \hat{x})\|_{L^2}^2.$$

Then

$$\|x - \hat{x}\|_{L^2}^2 \;\le\; \mathcal{R}(\hat{x}) \;\le\; (1 + \|H\|^2)\, \|x - \hat{x}\|_{L^2}^2.$$

Proof. The lower bound is immediate, as $\|H(x - \hat{x})\|_{L^2}^2 \ge 0$. For the upper bound, we use the definition of the induced operator norm for H:

$$\|H\boldsymbol{v}\|_2 \le \|H\|\|\boldsymbol{v}\|_2 \quad \forall \boldsymbol{v} \in \mathbb{R}^d.$$

Applying this to the L^2 norm of the function $H(x(t) - \hat{x}(t))$:

$$\begin{aligned}
\|H(x - \hat{x})\|_{L^2}^2 &= \int_0^T \|H(x(t) - \hat{x}(t))\|_2^2 dt \\
&\le \int_0^T \|H\|^2 \|x(t) - \hat{x}(t)\|_2^2 dt \\
&= \|H\|^2 \int_0^T \|x(t) - \hat{x}(t)\|_2^2 dt \\
&= \|H\|^2 \|x - \hat{x}\|_{L^2}^2.
\end{aligned}$$

Substituting this result into the definition of $\mathcal{R}(\hat{x})$ gives:

$$\begin{aligned}
\mathcal{R}(\hat{x}) &= \|x - \hat{x}\|_{L^2}^2 + \|H(x - \hat{x})\|_{L^2}^2 \\
&\le \|x - \hat{x}\|_{L^2}^2 + \|H\|^2 \|x - \hat{x}\|_{L^2}^2 \\
&= (1 + \|H\|^2) \|x - \hat{x}\|_{L^2}^2.
\end{aligned}$$

Thus $\mathcal{R}(\hat{x})$ is bounded above and below by constant multiples of the true error $\|x - \hat{x}\|_{L^2}^2$. By Definition 1 and Corollary B.1, minimizing $\mathcal{R}(\hat{x})$ is equivalent for interpolating solutions to minimizing the true L^2 error, establishing the surrogate calibration. □

B.2. Proof Sketch for Proposition 1

Proposition 2 (FD noise floor vs KSR). Assume additive i.i.d. zero-mean noise $\epsilon(t)$ with variance σ^2 on measurements of $x(t)$ sampled with step size Δt. Using central finite differences (FD) to approximate $\dot{x}(t)$ yields a mean-squared derivative error of: $\mathbb{E}\left[\|\widehat{x}_{\mathrm{FD}} - \dot{x}\|_2^2\right] = \mathcal{O}(\Delta t^4) + \Omega(\sigma^2/\Delta t^2)$. For KSR (kernel ridge with a twice-differentiable kernel) with regularization λ and n samples, the analytic derivative estimator satisfies $\mathbb{E}\left[\|\widehat{x}_{\mathrm{KSR}} - \dot{x}\|_2^2\right] = \mathcal{O}(\lambda) + \mathcal{O}(\sigma^2/n)$.

Sketch. **Finite Differences (FD)**: The central difference approximation for $\dot{x}(t_i)$ is $\widehat{x}_{\mathrm{FD}}(t_i) = \frac{x(t_{i+1}) + \epsilon_{i+1} - (x(t_{i-1}) + \epsilon_{i-1})}{2\Delta t}$. The error has two components: a bias term from the Taylor expansion, which is $\mathcal{O}(\Delta t^2)$, and a variance term from the noise. The squared error, in the case $\mathbb{E}[\epsilon_i] = 0$, is:

$$\mathbb{E}\left[\left(\frac{x(t_{i+1}) - x(t_{i-1})}{2\Delta t} - \dot{x}(t_i)\right)^2 + \left(\frac{\epsilon_{i+1} - \epsilon_{i-1}}{2\Delta t}\right)^2\right].$$

The squared bias is $\mathcal{O}(\Delta t^4)$. The variance term is $\mathbb{E}\left[\frac{\epsilon_{i+1}^2 - 2\epsilon_{i+1}\epsilon_{i-1} + \epsilon_{i-1}^2}{4\Delta t^2}\right] = \frac{2\sigma^2}{4\Delta t^2} = \frac{\sigma^2}{2\Delta t^2}$. Thus, the total MSE is $\mathcal{O}(\Delta t^4) + \Omega(\sigma^2/\Delta t^2)$. As $\Delta t \to 0$, the variance term explodes.

Kernel State Reconstruction (KSR): Taking expectation over the noise realizations and applying the standard bias–variance decomposition, we obtain

$$\mathbb{E}[||\hat{\dot{x}} - \dot{x}||_2^2] = \sum_i^n \mathsf{Var}[\hat{\dot{x}}_i] + ||\mathbb{E}[\hat{\dot{x}}] - \dot{x}||_2^2 \tag{11}$$

We bound the two terms separately.

Because the Gaussian kernel is smooth, differentiation is a bounded operator and we obtain, for some constant C,

$$||\mathbb{E}[\hat{\dot{x}}] - \dot{x}||_2^2 \leq C||\mathbb{E}[\hat{\dot{x}}] - \dot{x}||_{\mathcal{H}}^2 = \mathcal{O}(\lambda)$$

By Lemma 2,

$$\hat{x} - \mathbb{E}\hat{x} = k(t_i)^\top(\mathbf{K} + \lambda\mathbf{I})^{-1}\boldsymbol{\epsilon}$$

for a matrix $\mathbf{K}$ induced by the observation operators, with $\boldsymbol{\epsilon}$ zero-mean noise.

The variance can then be expressed as

$$\mathsf{Var}[\hat{\dot{x}}_i] = \mathbb{E}\left\{\boldsymbol{\epsilon}^\top(\mathbf{K} + \lambda\mathbf{I})^{-1}\dot{k}(t_i)\dot{k}(t_i)^\top(\mathbf{K} + \lambda\mathbf{I})^{-1}\boldsymbol{\epsilon}\right\} \tag{12}$$

$$= \sigma_{noise}^2 \dot{k}(t_i)^\top(\mathbf{K} + \lambda\mathbf{I})^{-2}\dot{k}(t_i). \tag{13}$$

Differentiating the Gaussian kernel with respect to time gives

$$\dot{k}(t)^\top\dot{k}(t) = \sum_{t_i}\left(\frac{t - t_i}{\sigma^2}\right)^2 e^{-\frac{||t-t_i||^2}{\sigma^2}} = \mathcal{O}(n). \tag{14}$$

We also note that

$$(\mathbf{K} + \lambda\mathbf{I})^{-2} = \frac{1}{n^2}\left(\frac{1}{n}\mathbf{K} + \frac{1}{n}\lambda\mathbf{I}\right)^{-2}$$

and since, for any matrix $\mathbf{A}$ we have

$$\boldsymbol{v}^\top\mathbf{A}\boldsymbol{v} \sim \mathsf{Tr}[\mathbf{A}]||\boldsymbol{v}||^2$$

the n^{-1} factor acts as a normalization constant.

Hence,

$$\mathsf{Var}[\hat{\dot{x}}_i] \sim \sigma_{noise}^2 \underbrace{n}_{\mathcal{O}||\dot{k}||}\frac{1}{n^2}\mathsf{Tr}\left(\frac{1}{n}\mathbf{K} + \frac{1}{n}\lambda\mathbf{I}\right)^{-2} = \mathcal{O}\left(\frac{\sigma_{noise}^2}{n}\right).$$

proving that

$$\mathbb{E}[||\hat{\dot{x}} - \dot{x}||_2^2] = \mathcal{O}(\lambda) + \mathcal{O}\left(\frac{\sigma_{noise}^2}{n}\right)$$

□

B.3. Method derivation

Let $(t^{obs}, X^{obs}) \in \mathbb{R}^{N_{obs}} \times \mathbb{R}^{N_{obs} \times D}$ denote the set of full (possibly noisy) observations of the state trajectory $x(t) \in \mathbb{R}^D$. Each row of X^{obs} corresponds to one observed state, recorded at the corresponding entry in t^{obs}.

In contrast, let $Y \in \mathbb{R}^{N \times S}$ represent a collection of S observed signals, each measured on a common vector of time points $t = (t_1, \ldots, t_N) \in \mathbb{R}^N$. Each column of Y corresponds to one signal. We assume these signals arise through a linear observation operator $H \in \mathbb{R}^{S \times D}$ applied to the (unknown) full state matrix $X \in \mathbb{R}^{N \times D}$:

$$Y = XH^\top. \tag{15}$$

Our goal in this setting is to construct a differentiable, vector-valued function

$$\widehat{x} : \mathbb{R} \;\to\; \mathbb{R}^D$$

that approximates the true state trajectory $\boldsymbol{x}(t)$. We measure accuracy via the expected squared error

$$\mathbb{E}_{t\sim\mathcal{T}} \big\|\widehat{x}(t) - x(t)\big\|^2 = \int_{\mathrm{supp}(\mathcal{T})} p_{\mathcal{T}}(t)\, \big\|\widehat{x}(t) - x(t)\big\|^2\, dt, \tag{16}$$

where $\mathcal{T}$ denotes the distribution of observation times. Assuming $\mathcal{T}$ is uniform, this reduces to a rescaled L^2 distance between the reconstructed and true trajectories.

From the perspective of statistical learning theory, this task is equivalent to learning a function that maps points from a real interval (time) into a higher-dimensional space $\mathbb{R}^D$, while accounting for partial and noisy observations.

In light of such consideration and our assumption of a sparse-observations regime ($N_{obs} \ll N$), a further regularization step is required to estimate the state in a meaningful way. This result can be obtained both by incorporating information on the fidelity of the signal (i.e. how the signal, which is fully observed, is reconstructed by the application of the operator H on the predicted state) and by imposing further regularization constraints that may come from domain knowledge related to the specific nature of the dynamical system. Formally, we obtain the regularized risk functional

$$\mathcal{R}_{reg}(\hat{x}) := \mathbb{E}_{t\sim\mathcal{T}} \left\|\widehat{x}(t) - x(t)\right\|^2 + \mathbb{E}_{t\sim\mathcal{T}} \left\|H\widehat{x}(t) - Hx(t)\right\|^2 + \lambda\, \mathfrak{R}(\widehat{x} \mid \mathscr{C}), \tag{17}$$

where $\mathfrak{R}$ denotes a regularization operator and $\mathscr{C}$ represents the contextual knowledge used in constructing the regularization. In the absence of regularization ($\lambda = 0$), this reduces to the case described by Lemma 2, from which we infer the induced equivalence between the two risk measures.

To formulate the regression problem associated with the constructed risk measure, we express $\widehat{x}$ as a linear combination of features obtained via a map $\phi : \mathbb{R} \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$. Under this hypothesis, we can reparameterize the risk measure as

$$\mathcal{R}_{reg}(\{w_j\}_j) = \mathbb{E}_{t\sim\mathcal{T}} \sum_{j=1}^{D} \big(\phi(t)^\top w_j - x_j(t)\big)^2 + \mathbb{E}_{t\sim\mathcal{T}} \sum_{s=1}^{S} \left(\sum_{j=1}^{D} H_{sj}\, \phi(t)^\top w_j - y_s(t)\right)^2 + \lambda\, \mathfrak{R}(\{w_j\}_j \mid \mathscr{C}). \tag{18}$$

Replacing expectations by empirical averages yields

$$\begin{aligned} \widehat{\mathcal{R}}_{reg}(\{w_j\}_j) = {} & \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} \sum_{j=1}^{D} \Big(\phi(t_i^{obs})^\top w_j - X_{ij}^{obs}\Big)^2 \\ & + \frac{1}{N} \sum_{i=1}^{N} \sum_{s=1}^{S} \Big(\sum_{j=1}^{D} H_{sj}\, \phi(t_i)^\top w_j - Y_{is}\Big)^2 + \lambda\, \mathfrak{R}(\{w_j\}_j \mid \mathscr{C}). \end{aligned} \tag{19}$$

Since ϕ may be infinite-dimensional, we restrict $w_1, \ldots, w_D$ to the span of $\{\phi(t_i)\}_{i=1}^N$, writing

$$w_j = \sum_{\ell=1}^{N} u_{\ell j}\, \phi(t_\ell), \qquad j = 1, \ldots, D, \tag{20}$$

for coefficients $\{u_{\ell j}\} \subset \mathbb{R}$ arranged in a matrix $U \in \mathbb{R}^{N \times D}$. Using the kernel trick $\phi(x)^\top \phi(y) = \kappa(x, y)$, the empirical loss becomes

$$\begin{aligned}\widehat{\mathcal{R}}_{reg}(U) = \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} \sum_{j=1}^{D} \left(\sum_{\ell=1}^{N} u_{\ell j}\, \kappa(t_i^{obs}, t_\ell) - X_{ij}^{obs} \right)^2 \\ + \frac{1}{N} \sum_{i=1}^{N} \sum_{s=1}^{S} \left(\sum_{j=1}^{D} \sum_{\ell=1}^{N} H_{sj}\, u_{\ell j}\, \kappa(t_i, t_\ell) - Y_{is} \right)^2 + \lambda\, \mathfrak{R}(U \mid \mathscr{C}).\end{aligned} \tag{21}$$

Defining kernel matrices $[K^{obs}]_{i\ell} := \kappa(t_i^{obs}, t_\ell)$ and $[K]_{i\ell} := \kappa(t_i, t_\ell)$, we obtain the compact form

$$\widehat{\mathcal{R}}_{reg}(U) = \frac{1}{N_{obs}} \left\| K^{obs} U - X^{obs} \right\|_F^2 + \frac{1}{N} \left\| K U H^\top - Y \right\|_F^2 + \lambda\, \mathfrak{R}(U \mid \mathscr{C}). \tag{22}$$

with the reconstructed state being, again in matrix form

$$\widehat{X} = KU. \tag{23}$$

Interestingly, the choice of the Gaussian feature map allows explicit computation of the derivatives which can be employed in the construction of context aware constraints in $\mathcal{R}$. In fact for any differential operator T we obtain

$$\begin{aligned} T\widehat{x}_j(t) &= T\left\{ \phi(t)^\top \sum_i^N \phi(t_i) u_{ij} \right\} \\ &= T\left\{ \sum_i^N \kappa(t, t_i) u_{ij} \right\} \\ &= \sum_i^N T\kappa(t, t_i) u_{ij} \end{aligned}$$

with $T\kappa$, the image of the kernel κ through the linear operator T, being analytically computable for the case of the Gaussian feature map.

We note that our method intrinsically supports the inclusion of additional loss terms, which can encode prior knowledge or structural constraints. Such terms may be specified directly by domain experts or automatically suggested by large language models. For example, one may penalize large deviations in the dynamics by introducing a regularizer of the form

$$\mathfrak{R}(U) \;=\; \|\dot{K} U\|_F^2, \tag{24}$$

where $\dot{K}$ denotes the Gram matrix after applying the time-derivative operator. More generally, this mechanism allows the integration of expert priors (e.g., known critical points, concavity properties, or monotonicity constraints) into the reconstruction process. In this way, our framework mimics the role of a human statistician in guiding model specification, while retaining the flexibility to incorporate data-driven or automatically generated hypotheses.

C. Extended Experimental Results

C.1. Additional noise regimes

Tables 5 and 6 further demonstrate that `MAAT` consistently outperforms competing baselines under both correlated Gaussian noise and heavy-tailed Student-t noise.

C.2. Computational Complexity

A potential concern with kernel-based methods is the $\mathcal{O}(N^3)$ cost of direct Gram-matrix inversion. However, `MAAT` avoids this bottleneck entirely: our implementation relies on first-order optimization (Adam) rather than closed-form matrix

Table 5. State reconstruction MSE (↓) semi-synthetic benchmark datasets. Values are mean ± confidence interval. Best result for each dataset-backend pair is bolded. Noise type: Correlated Gaussian.

Method	Backend	Dynamical systems			Epidemiology / dynamics			Oncology / viral		
		CRC	Cons.	Neut.	SEIR	SEIRH	TMDD	Tumor	TDI	Viral
RBF	PySR	$4.3 \times 10^{-2} \pm 5.6 \times 10^{-3}$	$3.3 \times 10^{1} \pm 5.7 \times 10^{0}$	$8.8 \times 10^{-3} \pm 7.8 \times 10^{-3}$	$2.9 \times 10^{-3} \pm 2.8 \times 10^{-4}$	$2.7 \times 10^{-3} \pm 2.6 \times 10^{-4}$	$3.4 \times 10^{-1} \pm 4.7 \times 10^{-2}$	$3.1 \times 10^{0} \pm 3.9 \times 10^{-1}$	$1.8 \times 10^{1} \pm 4.1 \times 10^{0}$	$2.6 \times 10^{-3} \pm 2.4 \times 10^{-4}$
	SINDy	$4.3 \times 10^{-2} \pm 5.7 \times 10^{-3}$	$3.6 \times 10^{1} \pm 5.7 \times 10^{0}$	$8.7 \times 10^{-3} \pm 7.6 \times 10^{-3}$	$3.0 \times 10^{-3} \pm 2.7 \times 10^{-4}$	$2.7 \times 10^{-3} \pm 2.6 \times 10^{-4}$	$3.4 \times 10^{-1} \pm 4.6 \times 10^{-2}$	$3.1 \times 10^{0} \pm 3.6 \times 10^{-1}$	$1.9 \times 10^{1} \pm 4.1 \times 10^{0}$	$2.7 \times 10^{-3} \pm 2.2 \times 10^{-4}$
Cubic	PySR	$5.6 \times 10^{-2} \pm 8.4 \times 10^{-3}$	$4.2 \times 10^{1} \pm 6.3 \times 10^{0}$	$1.2 \times 10^{-2} \pm 1.1 \times 10^{-2}$	$3.8 \times 10^{-3} \pm 4.4 \times 10^{-4}$	$3.6 \times 10^{-3} \pm 4.2 \times 10^{-4}$	$4.5 \times 10^{-1} \pm 6.5 \times 10^{-2}$	$4.0 \times 10^{0} \pm 5.1 \times 10^{-1}$	$3.6 \times 10^{1} \pm 7.3 \times 10^{0}$	$3.5 \times 10^{-3} \pm 3.3 \times 10^{-4}$
	SINDy	$5.6 \times 10^{-2} \pm 8.5 \times 10^{-3}$	$4.5 \times 10^{1} \pm 6.6 \times 10^{0}$	$1.2 \times 10^{-2} \pm 1.1 \times 10^{-2}$	$3.9 \times 10^{-3} \pm 4.4 \times 10^{-4}$	$3.7 \times 10^{-3} \pm 4.2 \times 10^{-4}$	$4.5 \times 10^{-1} \pm 6.5 \times 10^{-2}$	$4.0 \times 10^{0} \pm 5.2 \times 10^{-1}$	$3.7 \times 10^{1} \pm 7.5 \times 10^{0}$	$3.6 \times 10^{-3} \pm 3.2 \times 10^{-4}$
GP	PySR	$2.3 \times 10^{-1} \pm 2.5 \times 10^{-1}$	$2.3 \times 10^{2} \pm 2.1 \times 10^{2}$	$4.8 \times 10^{-2} \pm 4.7 \times 10^{-2}$	$5.0 \times 10^{-2} \pm 8.7 \times 10^{-2}$	$7.7 \times 10^{-3} \pm 7.7 \times 10^{-3}$	$9.2 \times 10^{-2} \pm 4.1 \times 10^{-2}$	$4.0 \times 10^{1} \pm 4.1 \times 10^{1}$	$1.9 \times 10^{3} \pm 3.2 \times 10^{3}$	$2.6 \times 10^{-2} \pm 5.8 \times 10^{-2}$
	SINDy	$4.1 \times 10^{-1} \pm 2.5 \times 10^{-1}$	$2.4 \times 10^{2} \pm 3.3 \times 10^{2}$	$2.0 \times 10^{-1} \pm 4.7 \times 10^{-1}$	$5.9 \times 10^{-3} \pm 6.0 \times 10^{-3}$	$2.2 \times 10^{-3} \pm 2.2 \times 10^{-3}$	$8.0 \times 10^{-2} \pm 3.4 \times 10^{-2}$	$2.9 \times 10^{1} \pm 2.5 \times 10^{1}$	$2.1 \times 10^{1} \pm 1.9 \times 10^{1}$	$1.9 \times 10^{-2} \pm 1.4 \times 10^{-2}$
Kalman	PySR	$1.2 \times 10^{-2} \pm 1.6 \times 10^{-3}$	$\mathbf{1.0 \times 10^{1} \pm 2.9 \times 10^{0}}$	$2.6 \times 10^{-3} \pm 2.4 \times 10^{-3}$	$7.7 \times 10^{-4} \pm 1.3 \times 10^{-4}$	$7.3 \times 10^{-4} \pm 1.3 \times 10^{-4}$	$8.8 \times 10^{-2} \pm 1.1 \times 10^{-2}$	$9.1 \times 10^{-1} \pm 1.4 \times 10^{-1}$	$1.8 \times 10^{1} \pm 3.8 \times 10^{0}$	$7.0 \times 10^{-4} \pm 7.4 \times 10^{-5}$
	SINDy	$1.2 \times 10^{-2} \pm 1.7 \times 10^{-3}$	$1.3 \times 10^{1} \pm 2.8 \times 10^{0}$	$2.6 \times 10^{-3} \pm 2.3 \times 10^{-3}$	$8.3 \times 10^{-4} \pm 1.3 \times 10^{-4}$	$7.5 \times 10^{-4} \pm 1.3 \times 10^{-4}$	$8.8 \times 10^{-2} \pm 1.1 \times 10^{-2}$	$9.2 \times 10^{-1} \pm 1.4 \times 10^{-1}$	$1.8 \times 10^{1} \pm 4.0 \times 10^{0}$	$7.9 \times 10^{-4} \pm 6.7 \times 10^{-5}$
Linear	PySR	$2.8 \times 10^{-2} \pm 3.8 \times 10^{-3}$	$2.4 \times 10^{1} \pm 4.2 \times 10^{0}$	$6.0 \times 10^{-3} \pm 5.5 \times 10^{-3}$	$1.9 \times 10^{-3} \pm 2.1 \times 10^{-4}$	$1.8 \times 10^{-3} \pm 2.0 \times 10^{-4}$	$2.2 \times 10^{-1} \pm 3.1 \times 10^{-2}$	$2.1 \times 10^{0} \pm 2.6 \times 10^{-1}$	$1.7 \times 10^{1} \pm 3.7 \times 10^{0}$	$1.7 \times 10^{-3} \pm 1.6 \times 10^{-4}$
	SINDy	$2.8 \times 10^{-2} \pm 3.9 \times 10^{-3}$	$2.5 \times 10^{1} \pm 4.3 \times 10^{0}$	$6.0 \times 10^{-3} \pm 5.3 \times 10^{-3}$	$2.0 \times 10^{-3} \pm 2.1 \times 10^{-4}$	$1.8 \times 10^{-3} \pm 2.0 \times 10^{-4}$	$2.2 \times 10^{-1} \pm 3.1 \times 10^{-2}$	$2.1 \times 10^{0} \pm 2.7 \times 10^{-1}$	$1.8 \times 10^{1} \pm 3.9 \times 10^{0}$	$1.8 \times 10^{-3} \pm 1.5 \times 10^{-4}$
NeuralODE	PySR	$6.2 \times 10^{1} \pm 1.3 \times 10^{2}$	$4.8 \times 10^{12} \pm 1.1 \times 10^{13}$	$9.4 \times 10^{0} \pm 2.0 \times 10^{1}$	$1.1 \times 10^{0} \pm 6.1 \times 10^{-1}$	$4.8 \times 10^{-1} \pm 1.4 \times 10^{-1}$	$1.9 \times 10^{0} \pm 2.6 \times 10^{0}$	$8.6 \times 10^{4} \pm 1.4 \times 10^{5}$	$8.6 \times 10^{8} \pm 1.9 \times 10^{9}$	$6.0 \times 10^{-1} \pm 3.0 \times 10^{-1}$
	SINDy	$1.8 \times 10^{0} \pm 8.5 \times 10^{-1}$	$1.1 \times 10^{3} \pm 1.0 \times 10^{3}$	$7.4 \times 10^{-1} \pm 7.5 \times 10^{-1}$	$8.0 \times 10^{-1} \pm 6.7 \times 10^{-1}$	$3.2 \times 10^{-1} \pm 1.2 \times 10^{-1}$	$2.2 \times 10^{0} \pm 3.1 \times 10^{0}$	$2.4 \times 10^{2} \pm 2.5 \times 10^{2}$	$4.1 \times 10^{3} \pm 9.2 \times 10^{3}$	$5.2 \times 10^{-1} \pm 3.1 \times 10^{-1}$
MAAT	PySR	$\mathbf{3.6 \times 10^{-3} \pm 1.9 \times 10^{-3}}$	$1.1 \times 10^{1} \pm 1.3 \times 10^{1}$	$\mathbf{3.4 \times 10^{-4} \pm 3.5 \times 10^{-4}}$	$\mathbf{2.4 \times 10^{-5} \pm 4.0 \times 10^{-6}}$	$\mathbf{1.7 \times 10^{-5} \pm 2.6 \times 10^{-6}}$	$\mathbf{3.4 \times 10^{-2} \pm 3.7 \times 10^{-2}}$	$\mathbf{3.9 \times 10^{-1} \pm 2.8 \times 10^{-1}}$	$\mathbf{5.4 \times 10^{0} \pm 5.9 \times 10^{0}}$	$\mathbf{4.1 \times 10^{-5} \pm 1.9 \times 10^{-5}}$
	SINDy	$\mathbf{1.4 \times 10^{-3} \pm 1.3 \times 10^{-4}}$	$\mathbf{5.8 \times 10^{0} \pm 4.9 \times 10^{0}}$	$\mathbf{4.1 \times 10^{-4} \pm 4.6 \times 10^{-4}}$	$\mathbf{7.9 \times 10^{-5} \pm 9.3 \times 10^{-6}}$	$\mathbf{4.2 \times 10^{-5} \pm 6.0 \times 10^{-6}}$	$\mathbf{4.8 \times 10^{-2} \pm 6.3 \times 10^{-4}}$	$\mathbf{1.7 \times 10^{-1} \pm 6.1 \times 10^{-2}}$	$\mathbf{1.7 \times 10^{0} \pm 4.8 \times 10^{-1}}$	$\mathbf{1.3 \times 10^{-4} \pm 2.5 \times 10^{-5}}$

Table 6. State reconstruction MSE (↓) semi-synthetic benchmark datasets. Values are mean ± confidence interval. Best result for each dataset-backend pair is bolded. Noise type: Student T.

Method	Backend	Dynamical systems			Epidemiology / dynamics			Oncology / viral		
		CRC	Cons.	Neut.	SEIR	SEIRH	TMDD	Tumor	TDI	Viral
RBF	PySR	$1.5 \times 10^{-1} \pm 3.1 \times 10^{-2}$	$1.2 \times 10^{2} \pm 1.8 \times 10^{1}$	$3.1 \times 10^{-2} \pm 2.6 \times 10^{-2}$	$1.1 \times 10^{-2} \pm 2.3 \times 10^{-3}$	$9.0 \times 10^{-3} \pm 1.1 \times 10^{-3}$	$1.3 \times 10^{0} \pm 1.7 \times 10^{-1}$	$1.1 \times 10^{1} \pm 1.3 \times 10^{0}$	$5.7 \times 10^{1} \pm 1.3 \times 10^{1}$	$8.4 \times 10^{-3} \pm 1.4 \times 10^{-3}$
	SINDy	$1.5 \times 10^{-1} \pm 3.1 \times 10^{-2}$	$1.2 \times 10^{2} \pm 1.8 \times 10^{1}$	$3.1 \times 10^{-2} \pm 2.6 \times 10^{-2}$	$1.1 \times 10^{-2} \pm 2.3 \times 10^{-3}$	$9.1 \times 10^{-3} \pm 1.1 \times 10^{-3}$	$1.3 \times 10^{0} \pm 1.7 \times 10^{-1}$	$1.1 \times 10^{1} \pm 1.3 \times 10^{0}$	$5.8 \times 10^{1} \pm 1.3 \times 10^{1}$	$8.5 \times 10^{-3} \pm 1.4 \times 10^{-3}$
Cubic	PySR	$2.3 \times 10^{-1} \pm 4.2 \times 10^{-2}$	$1.9 \times 10^{2} \pm 3.6 \times 10^{1}$	$4.9 \times 10^{-2} \pm 3.7 \times 10^{-2}$	$1.6 \times 10^{-2} \pm 3.4 \times 10^{-3}$	$1.4 \times 10^{-2} \pm 1.9 \times 10^{-3}$	$2.1 \times 10^{0} \pm 5.0 \times 10^{-1}$	$1.9 \times 10^{1} \pm 2.0 \times 10^{0}$	$1.6 \times 10^{2} \pm 5.0 \times 10^{1}$	$1.4 \times 10^{-2} \pm 3.5 \times 10^{-3}$
	SINDy	$2.3 \times 10^{-1} \pm 4.2 \times 10^{-2}$	$1.9 \times 10^{2} \pm 3.6 \times 10^{1}$	$4.8 \times 10^{-2} \pm 3.7 \times 10^{-2}$	$1.6 \times 10^{-2} \pm 3.4 \times 10^{-3}$	$1.4 \times 10^{-2} \pm 1.9 \times 10^{-3}$	$2.1 \times 10^{0} \pm 5.0 \times 10^{-1}$	$1.9 \times 10^{1} \pm 2.0 \times 10^{0}$	$1.6 \times 10^{2} \pm 5.0 \times 10^{1}$	$1.5 \times 10^{-2} \pm 3.5 \times 10^{-3}$
GP	PySR	$6.8 \times 10^{-1} \pm 6.3 \times 10^{-1}$	$1.9 \times 10^{2} \pm 1.7 \times 10^{2}$	$6.4 \times 10^{-2} \pm 4.5 \times 10^{-2}$	$1.3 \times 10^{-2} \pm 1.3 \times 10^{-2}$	$7.4 \times 10^{-3} \pm 6.7 \times 10^{-3}$	$\mathbf{4.3 \times 10^{-2} \pm 1.2 \times 10^{-2}}$	$2.7 \times 10^{1} \pm 3.4 \times 10^{1}$	$4.6 \times 10^{2} \pm 4.8 \times 10^{2}$	$3.6 \times 10^{-2} \pm 4.9 \times 10^{-2}$
	SINDy	$3.9 \times 10^{-1} \pm 3.2 \times 10^{-1}$	$3.2 \times 10^{2} \pm 3.5 \times 10^{2}$	$9.0 \times 10^{-2} \pm 6.2 \times 10^{-2}$	$5.2 \times 10^{-3} \pm 4.6 \times 10^{-3}$	$3.9 \times 10^{-3} \pm 3.6 \times 10^{-3}$	$6.4 \times 10^{-2} \pm 4.1 \times 10^{-2}$	$3.2 \times 10^{1} \pm 2.5 \times 10^{1}$	$7.4 \times 10^{2} \pm 1.2 \times 10^{3}$	$1.4 \times 10^{-2} \pm 1.2 \times 10^{-2}$
Kalman	PySR	$1.1 \times 10^{-2} \pm 2.5 \times 10^{-3}$	$9.8 \times 10^{0} \pm 2.3 \times 10^{0}$	$2.6 \times 10^{-3} \pm 2.5 \times 10^{-3}$	$8.4 \times 10^{-4} \pm 2.6 \times 10^{-4}$	$6.6 \times 10^{-4} \pm 1.1 \times 10^{-4}$	$1.1 \times 10^{-1} \pm 1.9 \times 10^{-2}$	$8.0 \times 10^{-1} \pm 1.6 \times 10^{-1}$	$4.9 \times 10^{1} \pm 1.1 \times 10^{1}$	$6.5 \times 10^{-4} \pm 1.3 \times 10^{-4}$
	SINDy	$1.1 \times 10^{-2} \pm 2.6 \times 10^{-3}$	$1.3 \times 10^{1} \pm 2.1 \times 10^{0}$	$2.5 \times 10^{-3} \pm 2.3 \times 10^{-3}$	$8.9 \times 10^{-4} \pm 2.6 \times 10^{-4}$	$6.7 \times 10^{-4} \pm 1.1 \times 10^{-4}$	$1.0 \times 10^{-1} \pm 1.9 \times 10^{-2}$	$8.1 \times 10^{-1} \pm 1.6 \times 10^{-1}$	$4.9 \times 10^{1} \pm 1.2 \times 10^{1}$	$7.5 \times 10^{-4} \pm 1.2 \times 10^{-4}$
Linear	PySR	$7.5 \times 10^{-2} \pm 1.6 \times 10^{-2}$	$6.4 \times 10^{1} \pm 1.1 \times 10^{1}$	$1.6 \times 10^{-2} \pm 1.3 \times 10^{-2}$	$5.7 \times 10^{-3} \pm 1.3 \times 10^{-3}$	$4.6 \times 10^{-3} \pm 5.3 \times 10^{-4}$	$6.7 \times 10^{-1} \pm 9.8 \times 10^{-2}$	$5.9 \times 10^{0} \pm 6.6 \times 10^{-1}$	$5.3 \times 10^{1} \pm 1.2 \times 10^{1}$	$4.3 \times 10^{-3} \pm 7.9 \times 10^{-4}$
	SINDy	$7.4 \times 10^{-2} \pm 1.6 \times 10^{-2}$	$6.5 \times 10^{1} \pm 1.1 \times 10^{1}$	$1.6 \times 10^{-2} \pm 1.3 \times 10^{-2}$	$5.7 \times 10^{-3} \pm 1.3 \times 10^{-3}$	$4.6 \times 10^{-3} \pm 5.3 \times 10^{-4}$	$6.7 \times 10^{-1} \pm 9.7 \times 10^{-2}$	$5.8 \times 10^{0} \pm 6.5 \times 10^{-1}$	$5.3 \times 10^{1} \pm 1.2 \times 10^{1}$	$4.4 \times 10^{-3} \pm 7.8 \times 10^{-4}$
NeuralODE	PySR	$1.1 \times 10^{1} \pm 1.7 \times 10^{1}$	$2.1 \times 10^{10} \pm 3.4 \times 10^{10}$	$2.5 \times 10^{1} \pm 6.5 \times 10^{1}$	$9.6 \times 10^{-1} \pm 5.9 \times 10^{-1}$	$5.3 \times 10^{-1} \pm 3.2 \times 10^{-1}$	$1.0 \times 10^{0} \pm 7.5 \times 10^{-1}$	$7.5 \times 10^{7} \pm 1.2 \times 10^{8}$	$3.4 \times 10^{10} \pm 7.5 \times 10^{10}$	$1.7 \times 10^{0} \pm 2.4 \times 10^{0}$
	SINDy	$2.2 \times 10^{0} \pm 2.9 \times 10^{0}$	$4.1 \times 10^{3} \pm 5.1 \times 10^{3}$	$1.1 \times 10^{0} \pm 9.1 \times 10^{-1}$	$5.1 \times 10^{-1} \pm 2.6 \times 10^{-1}$	$4.5 \times 10^{-1} \pm 2.2 \times 10^{-1}$	$1.3 \times 10^{0} \pm 1.1 \times 10^{0}$	$2.6 \times 10^{2} \pm 2.8 \times 10^{2}$	$7.4 \times 10^{1} \pm 8.8 \times 10^{1}$	$7.9 \times 10^{-1} \pm 6.4 \times 10^{-1}$
MAAT	PySR	$\mathbf{5.6 \times 10^{-3} \pm 4.1 \times 10^{-3}}$	$\mathbf{5.1 \times 10^{0} \pm 4.8 \times 10^{0}}$	$\mathbf{3.6 \times 10^{-4} \pm 4.8 \times 10^{-4}}$	$\mathbf{2.4 \times 10^{-5} \pm 3.1 \times 10^{-6}}$	$\mathbf{1.7 \times 10^{-5} \pm 1.1 \times 10^{-6}}$	$4.3 \times 10^{-2} \pm 2.6 \times 10^{-2}$	$\mathbf{2.4 \times 10^{-1} \pm 1.4 \times 10^{-1}}$	$\mathbf{9.1 \times 10^{0} \pm 1.4 \times 10^{1}}$	$\mathbf{4.3 \times 10^{-5} \pm 2.2 \times 10^{-5}}$
	SINDy	$\mathbf{1.4 \times 10^{-3} \pm 1.4 \times 10^{-4}}$	$\mathbf{4.5 \times 10^{0} \pm 2.0 \times 10^{0}}$	$\mathbf{4.2 \times 10^{-4} \pm 5.0 \times 10^{-4}}$	$\mathbf{7.7 \times 10^{-5} \pm 8.2 \times 10^{-6}}$	$\mathbf{4.1 \times 10^{-5} \pm 4.3 \times 10^{-6}}$	$\mathbf{4.8 \times 10^{-2} \pm 6.1 \times 10^{-4}}$	$\mathbf{1.3 \times 10^{-1} \pm 3.0 \times 10^{-2}}$	$\mathbf{1.7 \times 10^{0} \pm 4.2 \times 10^{-1}}$	$\mathbf{1.2 \times 10^{-4} \pm 2.3 \times 10^{-5}}$

inversion. Each iteration requires only matrix-vector products with the kernel matrix, reducing per-step complexity to $\mathcal{O}(N^2)$. This makes the method scalable to long or high-resolution trajectories. The situation is analogous to neural network training: although exact loss minimization is NP-hard in the worst case (Froese & Hertrich, 2023), practical gradient-based optimizers achieve good solutions efficiently.

Table 7 reports wall-clock time and peak memory usage across all baselines, averaged over the benchmark suite on identical hardware. MAAT incurs moderate overhead relative to classical smoothers (e.g., splines, Savitzky–Golay), but remains substantially faster than Neural ODEs. While slower than lightweight SINDy-based pipelines, it achieves consistently superior reconstruction accuracy. Its memory footprint is higher than simple interpolation methods but remains comparable to other kernel and neural approaches.

Table 7. **Computational cost comparison.** Mean wall-clock time and peak memory usage across all benchmarks, separated by downstream backend.

	PySR		**SINDy**	
Method	**Time (s)**	**Mem (MB)**	**Time (s)**	**Mem (MB)**
Linear	569.44	1331.03	3.99	220.72
RBF	594.47	1330.07	10.65	222.54
Cubic	575.01	1334.95	3.96	221.07
Savitzky–Golay	583.39	1336.63	4.35	222.14
Kalman filter	586.86	1348.64	9.56	230.78
TVRegDiff	591.17	1340.55	4.15	222.15
Gaussian Process	668.85	1332.21	149.01	232.16
Neural ODE	1165.29	1482.51	489.41	457.45
MAAT (ours)	693.39	1445.34	183.87	322.86

D. Experimental Details

Our evaluation suite consists of several ODE models commonly used in computational biology and pharmacology. We evaluate MAAT on both standard dynamical systems benchmarks and clinically motivated pharmacological models, reflecting the clinical relevance emphasized in our motivation. At the same time, we demonstrate the applicability of our method beyond these settings by including representative models from additional scientific domains.

D.1. Dynamical systems

Colorectal cancer model We adopt the seven-state CRC–mAb–IL2–chemo system modelling tumour burden (T), natural killer cells (N), lymphocytes (L), circulating chemotherapy (C), monoclonal antibody (M), cytokines (I), and a secondary antibody pool (A) (dePillis, 2014).

Neutrophil life-cycle model The Friberg-type transit system tracks proliferating precursors (Prol), transit compartments (T_1, T_2, T_3), marrow reservoir (Reserv), and circulating neutrophils (Circ) (Friberg et al., 2002).

Consumeristic socio-ecological model Population (x), renewable resources (y), non-renewable resources (z), and wealth (w) (Badiale & Cravero, 2024).

D.2. Epidemiology

SEIR compartmental epidemic model Susceptible (S), exposed (E), infectious (I), and removed (R) populations evolve under homogeneous mixing (Kermack & McKendrick, 1927):

SEIRH epidemic model with hospitalisation Extending SEIR with a hospitalised class H and transition rate δ (Bjørnstad et al., 2020).

Tumour PK/PD (TMDD + RO–driven TGI) We couple a 3-compartment monoclonal-antibody PK (central C_c, peripheral C_p, tumour C_t) with target-mediated drug disposition (TMDD) at tumour/peripheral sites, and drive tumour growth inhibition (TGI) by receptor occupancy (RO). This construction is consistent with TMDD foundations and minimal/PBPK mAb models that include a tumour (interstitial) site and use RO as the pharmacodynamic driver of TGI (Simeoni et al., 2004). All tumour-rate constants are converted from per-day to per-hour in implementation.

D.3. Oncology/Viral dynamical systems

Tumour model Custom logistic model of tumor growth, studying the temporal evolution of its volume.

$$\begin{cases} \frac{dC_c}{dt} &= -\frac{CL}{V_c}C_c - \frac{Q_p}{V_c}(C_c - C_p) - \frac{Q_t}{V_c}(C_c - C_t) \\ \frac{dC_p}{dt} &= \frac{Q_p}{V_p}(C_c - C_p) \\ \frac{dC_t}{dt} &= \frac{Q_t}{V_t}(C_c - C_t) \\ \frac{dR_A}{dt} &= -k_{\text{on},A}\, C_t\, R_A + k_{\text{off},A}\,(R_{A,\text{tot}} - R_A) \\ \frac{dR_B}{dt} &= -k_{\text{on},B}\, C_p\, R_B + k_{\text{off},B}\,(R_{B,\text{tot}} - R_B) \\ \frac{dT}{dt} &= K_g\, T\left(1 - \frac{T}{T_{\max}}\right) - K_k\, \frac{R_{A,\text{bound}}}{IC_{50} + R_{A,\text{bound}}}\, T \end{cases} \tag{25}$$

with

$$C_c : \text{ Drug concentration in the central (plasma) compartment} \tag{26}$$
$$C_p : \text{ Drug concentration in the peripheral compartment} \tag{27}$$
$$C_t : \text{ Drug concentration in the tumour compartment} \tag{28}$$
$$R_A : \text{ Unbound receptor A concentration} \tag{29}$$
$$R_B : \text{ Unbound receptor B concentration} \tag{30}$$
$$R_{A,\text{tot}} : \text{ Total receptor A concentration} \tag{31}$$
$$R_{B,\text{tot}} : \text{ Total receptor B concentration} \tag{32}$$
$$R_{A,\text{bound}} : \text{ Bound receptor A concentration} = R_{A,\text{tot}} - R_A \tag{33}$$
$$T : \text{ Tumour volume} \tag{34}$$
$$V_c : \text{ Central compartment volume} \tag{35}$$
$$V_p : \text{ Peripheral compartment volume} \tag{36}$$
$$V_t : \text{ Tumour compartment volume} \tag{37}$$
$$CL : \text{ Clearance from central compartment} \tag{38}$$
$$Q_p : \text{ Inter-compartmental flow between central and peripheral} \tag{39}$$
$$Q_t : \text{ Inter-compartmental flow between central and tumour} \tag{40}$$
$$k_{\text{on},A},\ k_{\text{off},A} : \text{ Binding and unbinding rates for receptor A} \tag{41}$$
$$k_{\text{on},B},\ k_{\text{off},B} : \text{ Binding and unbinding rates for receptor B} \tag{42}$$
$$K_g : \text{ Tumour growth rate constant (converted to h}^{-1}) \tag{43}$$
$$K_k : \text{ Maximum tumour kill rate (converted to h}^{-1}) \tag{44}$$
$$T_{\max} : \text{ Carrying capacity (maximum tumour volume)} \tag{45}$$
$$IC_{50} : \text{ Half-maximal inhibitory concentration for tumour kill} \tag{46}$$

Viral dynamics model Target cells (T), eclipse-phase cells (E), productively infected cells (I), and free virions (V) (Nowak & May, 2000).

Tumour–drug–immune interaction We consider tumour cells (T), chemotherapeutic payload (M), immune effectors (N), an inflammatory cytokine (I), and a tissue-damage biomarker (K). This low-order chemo-immuno system follows classical tumour–immune ODEs with chemotherapy cytotoxicity and immunosuppression terms; cytokines and damage markers use standard turnover/indirect–response kinetics (Sharma & Jusko, 1998).

D.4. COVID-19 data

We use publicly available data from the European Centre for Disease Prevention and Control (ECDC), comprising daily reported COVID-19 cases and deaths for European countries (European Centre for Disease Prevention and Control, 2020). These observations are combined with population statistics to obtain normalized trajectories. To stabilize variance and facilitate learning, we apply a $\log(x + 1)$ transformation to the raw time series. We split countries into training and holdout sets to evaluate generalization across populations. The training set consists of Austria, Belgium, and France ($n = 3$), while the holdout set includes Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, and Finland ($n = 7$).

D.5. Dataset Generation

ODE Integration and Trajectories. For each dynamical system, we generate ground-truth trajectories by integrating the governing Ordinary Differential Equations (ODEs) using a deterministic fourth-order Runge–Kutta (RK4) solver. We use a fixed integration step size Δt specific to each system's timescale (see Table 8).

To evaluate robustness to model mismatch, we sample a unique parameter set for each dataset by applying multiplicative jitter (typically 5%–10%) to the default literature parameters. For each data split (Train, Validation, Test), we sample a fresh initial condition $\mathbf{x}(t_0)$ by applying a 10% per-dimension jitter to a nominal starting state, clipping to non-negative values where physically required.

Observation Model and Noise. We generate observed data $\mathbf{Y} \in \mathbb{R}^{T\times M}$ from the latent states $\mathbf{x}(t) \in \mathbb{R}^D$ via a linear observation operator $\mathbf{Y}(t) = \mathbf{H}\mathbf{x}(t) + \boldsymbol{\epsilon}$. The matrix $\mathbf{H}$ simulates heterogeneous sensor channels, including direct observations of state subsets, total sums (e.g., total population), or linear combinations.

We evaluate three noise regimes to test solver robustness:

- **Gaussian Noise:** $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- **Student-t Noise:** $\epsilon \sim t_\nu$ with degrees of freedom $\nu = 5$ (heavy-tailed).
- **Correlated Noise:** AR(1) processes with correlation coefficient $\alpha = 0.8$.

The noise scale σ is set to 5% of the mean absolute amplitude of the state variable ($\sigma = 0.05 \times \text{mean}|\mathbf{x}|$). Full-state snapshots are provided at evenly spaced indices, with the count scaling as $\approx 1.5\sqrt{T}$ to mimic sparse measurement settings.

Dataset Specifications. Table 8 details the grid dimensions and state variables for all systems evaluated.

Table 8. Dataset specifications. D: state dimension; Δt: integration step; N: number of time points per split. All trajectories start at $t_0 = 0$.

Dataset	D	Δt	N_{train}	$N_{\text{val/test}}$	$t_{\text{train}}^{\max}$	**State Variables**
Population & Ecological Dynamics						
SEIR	4	0.2	500	200	99.8	Susceptible, Exposed, Infected, Recovered
SEIRH	5	0.2	500	200	99.8	SEIR + Hospitalized
Viral	4	0.2	500	200	99.8	Target cells, Exposed, Infected, Virus
Consumeristic	4	0.2	500	200	99.8	x, y, z, w (Social dynamics)
Systems Biology & Pharmacology (PK/PD)						
Neutrophil	6	0.2	500	200	99.8	Proliferating, Transit$_{1\text{–}3}$, Reserve, Circulating
Colorectal	7	0.2	500	200	99.8	Tumor, Necrotic, Lymph, Cells, Macrophage, I, A
Tumor	6	0.2	500	200	99.8	PK/PD with binding & tumor volume
TMDD Lite	5	0.2	500	200	99.8	Drug, Receptor, Complex, Production, Internalization
Tumor-Drug-Imm	5	0.5	500	200	249.5	Tumor, Macrophage, NK cells, IL-2, Kill rate

D.6. MAAT Implementation Details

Kernel State Recovery (KSR). We model each state dimension $d \in \{1, \dots, D\}$ using a Gaussian Radial Basis Function (RBF) kernel:

$$\kappa_d(t, t') = \exp\Big(-\tfrac{(t-t')^2}{2\sigma_d^2}\Big).$$

The recovered trajectory $\hat{x}_d(t)$ and its time-derivative $\widehat{\dot{x}}_d(t)$ are expanded over the grid points $t_j \in \mathcal{T}$ as:

$$\hat{x}_d(t) = \sum_{j=1}^{T} U_{j,d}\, \kappa_d(t, t_j), \qquad \widehat{\dot{x}}_d(t) = \sum_{j=1}^{T} U_{j,d}\, \partial_t \kappa_d(t, t_j).$$

Loss Function and Optimization. We optimize the coefficient matrix $U \in \mathbb{R}^{T\times D}$ by minimizing a composite loss function $\mathcal{L}(U)$:

$$\begin{aligned}\mathcal{L}(U) = {} & w_s\, \mathbb{E}_{(t_k,\mathbf{s}_k)\in\mathcal{M}}\big[\|\hat{\mathbf{x}}(t_k) - \mathbf{s}_k\|_2^2\big] + w_i\, \mathbb{E}_{t\in\mathcal{T}}\big[\|\mathbf{H}\hat{\mathbf{x}}(t) - \mathbf{y}(t)\|_2^2\big] \\ & + \gamma\, \mathbb{E}_{t\in\mathcal{T}}\big[\|\widehat{\dot{\mathbf{x}}}(t) - \mathbf{f}_0(\hat{\mathbf{x}}(t))\|_2^2\big] + \lambda\, \|U\|_F^2 + w_+\, \|\min(\hat{\mathbf{x}}, 0)\|_2^2. \end{aligned} \tag{47}$$

We use fixed weights:

- **Weights:** $w_s = w_i = 1.0$, $\gamma = 10^{-3}$, $\lambda = 10^{-6}$.

Training Protocol. Training proceeds in two phases using the Adam optimizer ($\beta_1 = 0.99, \beta_2 = 0.999, \epsilon = 10^{-8}$):

1. **Length-scale Selection:** Kernel length-scales are set from the snapshot time variance, i.e., ($\sigma_d = \sqrt{\mathrm{Var}(\mathcal{T})}$), and kept fixed during training (no length-scale sweep/tuning in these runs).
2. **Optimization:** We optimize U with learning rate 1.0 for up to $20,000$ iterations, employing early stopping with a patience of $2,000$ steps based on validation loss.

Implementation is in JAX, utilizing JIT compilation for efficient batched evaluation.

D.7. Baseline Configurations

All baselines use fixed hyperparameters across datasets to ensure fair comparison.

Smoothing & Interpolation Baselines.

- **Cubic Spline:** Natural cubic splines (SciPy) with analytic derivatives.
- **RBF Interpolant:** Multiquadric RBFs with shape parameter ε selected via grid search over $\{0.25, \ldots, 4.0\} \times (1/\sqrt{\mathrm{Var}(\mathcal{T})})$. Derivatives via central differences ($\Delta = 10^{-3}\Delta t$).
- **Savitzky–Golay:** Window length 25, polynomial order 3, with boundary-aware padding. Analytic filter derivatives are used.
- **TV-Reg Diff:** Total Variation Regularized Differentiation with regularization $\alpha = 0.01$, implemented via a Savitzky–Golay proxy (window 21, order 3) for computational efficiency on large grids.

Probabilistic & Neural Baselines.

- **Gaussian Process (GP):** Independent RBF kernel GPs per dimension. Length-scales optimized via log-marginal likelihood maximization (3 random restarts) searching over factors $\{0.25, \ldots, 4.0\} \times \mathrm{std}(\mathcal{T})$.
- **Kalman Smoother:** Constant-velocity kinematic model ($q = 1.0, r = 0.1$). Derivatives extracted from the smoothed velocity states.
- **Neural ODE:** MLP vector field (width 64, depth 3, tanh activation). Integrated with Dopri5 ($dt_0 = 0.1$). Trained for 2,000 steps (Adam, lr=10^{-3}).

D.8. Symbolic Regression Back-Ends

Downstream equation discovery is performed on the trajectories recovered by `MAAT` or baselines.

- **PySINDy:** Sparse identification with a polynomial library (degree ≤ 2). Optimizer: STLS with threshold 0.1 and decay 0.9.
- **PySR:** Evolutionary search with 20 iterations, population 1,000. Allowed operators: $\{+, \times\}$. Selection criterion: Validation derivative MSE.

D.9. Computing Infrastructure

All experiments were conducted on a workstation with dual AMD EPYC 7713 CPUs (2 × 64 cores; 128 physical cores total) and an NVIDIA RTX 6000 Ada Generation GPU (49,140 MiB VRAM, 49 GB).