Inverse hidden state neural operators with observable state-space models

Andrew Gracyk

Abstract

We study inverse problems in partial differential equation (PDE) data-driven scenarios through the use of Mamba state-space model architectures. We constrain the respective ordinary differential equation (ODE) system using the concept known as observability in ODE control theory. By effectuating observability, the learning task of mapping PDE trajectories to the original hidden state is a well-defined mapping. Under a more classical objective in learning the Mamba inverse operator, it is possible the solution is naturally observable. We take this notion a step further: we specifically constrain observability, and it is guaranteed. The addition of this constraint comes at minimal expense when training the objective. Our strategies employ techniques found in resolution independent deep operator networks and consistency functions, belonging to the framework of consistency models, thus allowing sparsely sampled trajectories of the dynamics to be mapped back to the original hidden state. We develop a theoretical stability analysis for generalized state-space models. Our methods are applicable when developing a backwards numerical method is impossible. We deploy our methodologies on PDE-type data, providing empirical foundations for solving inverse problems with Mamba architectures.

1 Introduction

The neural operator, in its many diverse forms, is the prevalent archetype for learning dynamics, namely partial differential equations (PDEs), with data-driven exploratory techniques via machine learning. While a more primitive lens for learning PDEs is done with finite-dimensional vector spaces, the neural operator amends this PDE learning perspective via the operator learning task: instead of learned fixed, discretized vectors of PDE solutions, we learn the operator mapping from a function space to a function space. In this sense, neural operators offer additional benefits over more traditional frameworks, primarily mesh-invariance, as well as high in-distribution learning precision. In this investigation, we develop techniques in inverse problems using the Mamba neural operator paradigm by leveraging the concept of observability, as is studied from the classical ordinary differential equation (ODE) lens.

With our approaches, we train the Mamba neural operator in the classical sense to map initial data to the PDE solution over the course of its time continuum in a forward problem setting. Simultaneously, we train the backward problem to solve the inverse problem via mapping trajectories of the PDE, possibly irregularly sampled, to the original hidden state. The Mamba neural operator may naturally converge so that this backward mapping is plausible, but the well-definedness of this backward mapping is not guaranteed with the traditional training objective. By enforcing observability upon the learned state-space system, the existence of this backward mapping is assured. The additional constraint to enforce this condition is of low computational expense, thus our strategies are simplistic, numerically efficient, and comprehensive, as we solve both the forward and backward problems in one objective.

NOTE: it is possible the learned state-space model naturally converges to an observable system. With our method, the observability is guaranteed, and the condition to enforce this is at relatively modest cost.

2 Preliminaries

In practice, for training, we will assume we have trajectories of ambient data y sampled over mesh Ω and irregularly over an interval $[0, T]$ of the form

$$\mathcal{Y} = \bigcup_{i \in \mathcal{D}_{\text{training}}} \bigcup_{j} \left\{ \overline{y}_{t_j^{(i)}} \;\middle|\; y|_{t_j^{(i)}} \in \mathcal{C}(\mathbb{R}^N; \mathbb{R}^n) \cap W^{1,p}(\mathcal{X}, \mathbb{R}^n), t_j^{(i)} \in [0, T] \right\}, \tag{1}$$

where $\overline{y}_{t_j^{(i)}} = (y(x_1, t_j^{(i)}), \ldots, y(x_D, t_j^{(i)}))$ is the discretization $x_i \in \Omega \subseteq \mathcal{X}$, and we denote $\mathcal{D}_{\text{training}}$ the training set, $\mathcal{X} \subseteq \mathbb{R}^N$. Index i runs over training data and j runs over the number of samples in the trajectory per single instance of training data, being the observed ambiet PDE data.

2.1 The inverse problem of interest

Given an ambient observation in $\mathcal{Y}$, we seek an inverse mapping

$$\mathcal{F}^{-1} : \mathcal{Y} \to \mathcal{A}, \qquad \mathcal{A} = \left\{ h \in \mathcal{C}([0,T]; \mathbb{R}^{n_x = n_h}) \middle| h \text{ observable} \right\} \tag{2}$$

where $n_x = n_h$ is the hidden state dimension, and $\mathcal{A}$ is the function space of all possible hidden states that are observable.

2.2 Control theory

We provide background on control theory to establish our framework. First, we define the *reachable set* at t and the *overall reachable set* respectively as

$$\mathcal{C}(t) = \text{set of initial } x_0 \text{ in which there exists a control so } x(t) = 0 \tag{3}$$

$$\mathcal{C} = \bigcup_{t \geq 0} \mathcal{C}(t). \tag{4}$$

for some function $\alpha : [0, \infty) \to A$ we call a *control*. We say an ODE system is *controllable* if $\mathcal{C} = \mathbb{R}^n$. We say an ODE system

$$\begin{cases} \dot{x}(t) = Mx(t) \\ y(t) = Nx(t) \end{cases} \tag{5}$$

is observable if the system

$$\dot{z}(t) = M^T z(t) + N^T \alpha(t) \tag{6}$$

is controllable, where α is the control. The system

$$y(t) = Nx(t). \tag{7}$$

is *observable* if knowledge of y on any $[0, t]$ allows us to compute $x(0)$ under the system. Since the backward problem of the ambient dynamics may not be well-posed, we cannot necessarily solve it traditionally; however, observability guarantees the hidden state is unique, and thus it can be solved backwards.

Remark. Generally, we consider a local version of observability such that $\mathcal{C} \in \mathcal{X} \subseteq \mathbb{R}^n$ for some bounded set $\mathcal{X}$. We desire the image of the hidden state to be in this set. This gives us a more computationally practicable version of observability, as it not reasonable to examine all of $\mathbb{R}^d$.

Remark. Generally, it is sufficient to observe $y(t)$ sparsely in discrete locations.

2.3 Consistency functions

Consistency functions, as proposed in , are classically for diffusion models; however, our use in similar. We can learn the mapping

$$\Gamma : C(\mathbb{R}^{N_y} \times \mathbb{R}_+; \mathbb{R}^{n_y}) \cap W^{1,p}(\mathcal{X} \times \mathbb{R}_+; \mathbb{R}^{n_y}) \times [0,t] \to C(\mathbb{R}^{N_x} \times \mathbb{R}_+; \mathbb{R}^{n_x}) \cap W^{1,p}(\mathcal{X} \times \mathbb{R}_+; \mathbb{R}^{n_x}), \tag{8}$$

or more simply $\Gamma : y_t \times [0,t] \to x_0$, through a neural network F such that

$$F : \mathbb{R}^n \times [0,T] \times \theta_F \to \mathbb{R}^n, \qquad F_{\Theta_F}(y_t, t) \approx x_0. \tag{9}$$

Generally, solving such an inverse problem in the ambient space for y is not guaranteed to exist; however, by the principle of observability, such a mapping is well-defined in the hidden state-space. Thus, we learn x_0, and using our system, we can produce y_0 from this.

Consistency models map a single point along the trajectory to its origin. For observability, a single instance along the trajectory is insufficient to learn initial data x_0. In our case, we are interested in a (possibly sparse) collection of samples along the trajectory. In this sense, we build upon resolution

independent deep operator networks and apply them for our consistency functions. In this case, we are interested in the mapping

$$\Gamma^{\dagger} : (\prod_{j=1:t_j \in [0,T]}^{n_i} f_j \in C(\mathbb{R}^{N_y} \times \mathbb{R}_+; \mathbb{R}^{n_y}) \cap W^{1,p}(\mathcal{X}^{N_y} \times \mathbb{R}_+; \mathbb{R}^n) \times t_j) \to C(\mathbb{R}^{N_x} \times \mathbb{R}_+; \mathbb{R}^{n_x}) \cap W^{1,p}(\mathcal{X}^{N_x} \times \mathbb{R}_+; \mathbb{R}^{n_x}), \tag{10}$$

where the functions f_j are irregularly sampled with respect to the time domain n_i times over $t_j \in [0, T]$. More simply put,

$$\Gamma^{\dagger} : \underbrace{(f_1 \times t_1) \times \ldots \times (f_{n_i} \times t_{n_i})}_{\text{an } n_i \text{ irregular number of times}} \to x_0. \tag{11}$$

2.4 State space models

We learn a continuous-time mapping sequence-to-sequence

$$\begin{cases} \dot{h}(t) = Ah(t) + Bx(t) \\ y(t) = Ch(t) \end{cases} \tag{12}$$

for hidden state $h(t)$.

Note there is an addition of a $Bx(t)$ term here. This does not influence the observability condition of 2.2.

Using a discretization, our ODE system becomes

$$\begin{cases} h_t = \overline{A} h_{t-1} + \overline{B} x_t \\ y_t = Ch_t \end{cases} \tag{13}$$

using the equations

$$\overline{A} = \exp\{\Delta \cdot A\}, \qquad \overline{B} = (\Delta \cdot A)^{-1}(\exp\{\Delta \cdot A\} - I) \cdot \Delta B. \tag{14}$$

Using a convolution, we have

$$\overline{K} = (C\overline{B}, C\overline{AB}, \ldots, C\overline{A}^k B, \ldots), \tag{15}$$

which gives

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \\ \vdots \end{pmatrix} = y = u * \overline{K} = \underbrace{\begin{pmatrix} C\overline{B} & 0 & \ldots & 0 \\ C\overline{AB} & C\overline{B} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\overline{A}^k\overline{B} & C\overline{A}^{k-1}\overline{B} & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}}_{\tilde{K}} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_k \\ \vdots \end{pmatrix} \tag{16}$$

The convolution theorem is employed here, notably

$$\overline{y} = \mathcal{F}^{-1}[\mathcal{F}[u] \cdot \mathcal{F}[\overline{K}]] = \tilde{K}\overline{x}, \tag{17}$$

where the use of the discrete fast Fourier transform (FFT) $\mathcal{F}$ reduces the computational complexity required. Thus, it is of interest to examine $\mathcal{F}(\overline{K})$. Here, the Fourier transform of $\overline{K}$ is defined such that

$$\mathcal{F}[\overline{K}] = (\mathcal{F}[C\overline{B}], \mathcal{F}[C\overline{AB}], \ldots) \tag{18}$$

$$\mathcal{F}[C\overline{A}^k\overline{B}] = \sum_j (C\overline{A}^j\overline{B}) e^{-2\pi i k j / n_x}. \tag{19}$$

It is of consideration to derive a backward numerical method to solve the task instead of using machine learning techniques. The most apparent scheme is of the form

$$h_i = C^{-1} y_i \tag{20}$$

$$x_i = B^{-1}(h_i' - Ah_i) \tag{21}$$

$$h_{i-1} = h_i - (\Delta t)(Ah_i + Bx_i); \tag{22}$$

by substituting the last two, we get

$$h_{i-1} = h_i - (\Delta t)(Ah_i + BB^{-1}(h_i' - Ah_i)). \tag{23}$$

Solving the backward problem is solvable under conditions of invertibility on B, C. Yet, we see it is necessary B, C have left inverses, which cannot be true if B, C have more columns than rows, which corresponds to a high-dimensional hidden states. For the remainder of our investigation, we will assume C does not have a left inverse, i.e. C has columns than rows.

2.5 Consistency functions with numerous samples over the trajectory: an adaptation of resolution independent deep operator networks

We build upon the idea of consistency functions but for numerous samples of data mapped to the same origin. We build this method based off the work in for resolution independent deep operator networks. Construct a dictionary Φ. Let $h(t) \in \mathbb{R}^d$. We solve the objective

$$\mathbb{E}_{j \sim [N_{\text{training}}]} \, \mathbb{E}_{h^{(j)} \sim H_j} \, \mathbb{E}_{t_i \sim \Omega^{(j)}} \left[||\Phi^T(\overline{t}_i; \theta_\Phi)\delta^{(j)} - \overline{h}_i^{(j)}|| \right]. \tag{24}$$

Here, we have used notation

$$[N_{\text{training}}] = \{j \in \{1, \dots, N_{\text{training}}\} : N_{\text{training}} = \text{total number of hidden state trajectories}\} \tag{25}$$

$$H_j = \{h : h \in \mathcal{C}([0,T]; \mathbb{R}), \text{ i.e. } h \text{ is a possible hidden state}\} \tag{26}$$

$$\Omega^{(j)} = \bigcup_{i=1}^{n_j} \{t_i : t_i \in U((0,T]), n_j = \text{number of time samples at iterate } j\} \tag{27}$$

We denote the bar as notation of the collection respect to that sample. Typically in the inverse problem, $t_i = 0$, but this is not necessarily true and be taken arbitrarily if desired. It can be shown the dictionary coefficient $\delta^{(j)}$ is given by

$$\delta^{(j)} = (\Phi^{(j)}{\Phi^{(j)}}^T + \lambda I)^{-1}\Phi^{(j)}U^{(j)}, \qquad \Phi^{(j)} \in \mathbb{M}^{K \times n_i}, U^{(j)} \in \mathbb{M}^{n_i \times D} \tag{28}$$

where $\Phi^{(j)} = \Phi(\overline{t}^{(j)}; \theta_\Phi)$ is the basis function value at $\overline{t}_i = \{t_{j_1}, \dots, t_{j_i}\}$, and $U^{(j)}$ is the collection of ambient data realization such that n_i is the number of samples of the ambient data (typically low) at training data sample j. D is the PDE discretization dimension (typically taken high). Note the above solution is a form of ridge regression, which has a sparsity regularization term whose strength is controlled by λ. In particular, Φ is constructed using

$$\Phi(t; \theta_\Phi) = \begin{pmatrix} \phi_1(t_1) & \dots & \phi_1(t_{n_i}) \\ \vdots & \ddots & \vdots \\ \phi_{K=|\Phi|}(t_1) & \dots & \phi_{K=|\Phi|}(t_{n_i}) \end{pmatrix}. \tag{29}$$

The final dimensions of the terms to be solved in the objective function are

$$\Phi^T(t^{(i)}; \theta_\Phi) \in \mathbb{M}^{n_i \times K}, \delta \in \mathbb{M}^{K \times D}, \overline{h}(t) \in \mathbb{R}^D. \tag{30}$$

Finally, instead of passing δ into a deep operator network architecture, we pass it into a consistency style framework in the mapping

$$\Psi : (\delta \in \mathbb{R}^{K \times D}) \times (t \in [0,T]) \times \Theta_\Psi \to h(t) \in \mathbb{R}^D. \tag{31}$$

We get

$$h^{(j)}(0) \approx \Psi(\alpha^{(j)}; \theta_\Psi). \tag{32}$$

3 Methods

3.1 Training

We constrain the hidden states to be observable, thus we have a guarantee of well-definedness when solving the inverse problem among the hidden states, even without invertibility of our matrices. First,

we remark how our method can be approached via the lens of neural ordinary differential equations; however, this method is computationally costly. Instead, we propose alternative methodology via the lens of the observability matrix.

We consider a variety of approaches to enforce observability in the state-space system. First, we consider a training approach using an ODE solver. For a system to be observable, we desire $z(t) = 0$. Now, note that

$$z(t) - z(0) = \int_{[0,T]} \dot{z}(\tau) d\tau = \int_{[0,t]} M^T z(\tau) + N^T \alpha(\tau) d\tau. \tag{33}$$

Since we want $z(t) = 0$, we can rearrange and minimize the loss in a discretized setting. The integral can be calculated using an ODE solver. We take $M = B, N = C$. The observability constraint term can be solved using an ODE integration solver as presented in the corresponding repository of [1] such that

$$\mathbb{E}_{t_i} \mathbb{E}_{z(0) \sim \mathcal{X}} \left[\sum_i ||z(0) + \int_{[0,t_i]} B_{\theta_B}^T z(\tau) + C_{\theta_C}^T \alpha(\tau) d\tau|| \right] \tag{34}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \text{odeint}(B_{\theta_B}^T z(\tau) + C_{\theta_C}^T \alpha(\tau), z_{\text{sampled}}, \Omega_i), \tag{35}$$

where $\text{odeint} : \mathbb{R}^{N_h} \times \mathbb{R}^{N_h} \times \Omega_{t_i} \to \mathbb{R}^{N_h}$ is the ODE integration solver, and we have denoted $\Omega_i = \{i\Delta t : \Delta t \in \mathbb{R}^+\}$ a temporal domain. Since z is learned with a neural network the above computation is valid. If z were not learned, it is known $z(t) = 0$ for some t, but we do not know exactly what t based on $z(0)$, or we do not know exactly what $z(0)$ if t is chosen.

However, the above approach incurs high offline cost and is overall nontrivial. We may simplify the above computation by use of the observability matrix. This approach yields lesser expense. Define the observability matrix as

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \in \mathbb{M}^{(n_h \cdot N_c) \times n_h}. \tag{36}$$

The system is observable if this matrix is full column rank; however, this is problematic to evaluate in an objective. Computing the rank of the matrix is typically a non-differentiable process, thus we cannot include it in a loss function with a backpropagation-type algorithm. Yet, we remark the determinant is indeed a differentiable process, but we cannot take the determinant, since $\mathcal{O}$ is not square; however, if we enforce

$$\det(\mathcal{O}^T \mathcal{O}) > 0, \tag{37}$$

then we must have $\mathcal{O}$ is full column rank. Thus, the ODE system is observable. In practice, we can choose a constant $\text{const} > 0$ and enforce the condition

$$\text{relu}(\text{const} - \det(\mathcal{O}^T \mathcal{O})) \tag{38}$$

in the loss function.

Our total objective to train in this setting is

$$\min_{\theta = (\theta_A, \theta_B, \theta_C, \theta_\Psi, \theta_\Phi)} \underbrace{\mathbb{E}_t \mathbb{E}_x \left[||y - \tilde{K}_{\theta_K = (\theta_A, \theta_B, \theta_C)} x|| \right]}_{\text{state-space loss}} + \underbrace{\gamma_{\text{obs}} \text{relu}(\text{const} - \det(\mathcal{O}^T \mathcal{O}))}_{\text{observability constraint}} \tag{39}$$

$$+ \underbrace{\mathbb{E}_i \mathbb{E}_{h_0 \sim p_{h_0^{(i)}}} \left[||h_0^{(i)} - \Psi_{\theta_\Psi}(\delta^{(i)}; \theta_\Psi)|| \right]}_{\text{consistency loss}} + \underbrace{\mathbb{E}_i \mathbb{E}_{h_0^{(i)} \sim p_{h_0}} \left[||\Phi_{\theta_\Phi}^T(0; \theta_\Phi) \delta^{(i)} - h_0^{(i)}|| \right]}_{\text{dictionary learning}}. \tag{40}$$

3.2 A special case: a simplified C

A sufficient condition such that the system is observable is that C is full column rank; however, for us, this is not possible. The observability matrix of equation 36 is automatically full column rank if C is full column rank, but since we generally take C to have more columns than rows, this is of interest.

Instead, we may constrain our first handful of matrices in the observability block matrix to be full column rank in the span of this matrix concatenation. In particular, we seek

$$\text{column rank}\bigg(\begin{pmatrix} C \\ CA \\ \vdots \\ CA^p \end{pmatrix} \bigg) = \dim(\{(h_1, \ldots, h_{n_h}) = h\}) = n_h, \qquad p \ll n-1. \tag{41}$$

We may constrain $C, \ldots CA^p$ to be full column rank by taking

$$CA^i \in \{\widetilde{CA^i} = CA^i + \epsilon \tilde{I} : c_{ii} \in C, c_{ii} \geq 0, \tilde{i}_{ij} \in \tilde{I}, \tilde{i}_{ii} = 1, \tilde{i}_{ij} = 0 \text{ for } i \neq j\} = \Xi_{\text{full column rank}}. \tag{42}$$

Here, $\tilde{I}$ is the adjusted identity matrix such that it it non-square, i.e. the same dimensions as C, and has 1's along the main diagonal and 0's elsewhere.

Using this condition, our objective function is transformed into

$$\begin{cases} \min_{\theta=(\theta_A,\theta_B,\theta_C,\theta_\Psi,\theta_\Phi)} \mathbb{E}_t \, \mathbb{E}_x \left[||y - \tilde{K}_{\theta_K=(\theta_A,\theta_B,\theta_C)} x|| \right] \\ + \mathbb{E}_i \, \mathbb{E}_{h_0 \sim p_{h_0^{(i)}}} \left[||h_0^{(i)} - \Psi_{\theta_\Psi}(\delta^{(i)}; \theta_\Psi)|| \right] + \mathbb{E}_i \, \mathbb{E}_{h_0^{(i)} \sim p_{h_0}} \left[||\Phi_{\theta_\Phi}^T(0; \theta_\Phi)\delta^{(i)} - h_0^{(i)}|| \right] \\ \text{subject to } CA^i \in \Xi_{\text{full column rank}} \end{cases} \tag{43}$$

4 Theoretical discussion

References

[1] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

A Alternative observability condition

We examine observability using the matrix of equation 16. We examine the matrix

$$\tilde{K} = \begin{pmatrix} C\overline{B} & 0 & \ldots & 0 \\ C\overline{AB} & C\overline{B} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\overline{A}^k\overline{B} & C\overline{A}^{k-1}\overline{B} & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}. \tag{44}$$

Theorem 1. Let $C\overline{B}$ be square. Make the assumption

$$\text{col}(\mathcal{A}_i) = \text{col}\Big((A^{-1})^T (A^i)^T C^T C A^i A^{-1} \Big) \tag{45}$$

$$\text{col}(\mathcal{B}) = \text{col}\Big((\exp\{\Delta \cdot A\} - I) B B^T (\exp\{\Delta \cdot A\} - I)^T \Big) \subseteq \text{col}(\mathcal{A}_1 + \mathcal{A}_2 \ldots + \mathcal{A}_{n-1}), \tag{46}$$

for all i. In particular, the matrix on the bottom spans no new column space. Also, assume A is invertible and C is full row rank. We have $\det(\tilde{K}) > 0$, or equivalently $\det(C\overline{B}) > 0$, implies the system is observable.

Proof. Since $\tilde{K}$ is an upper triangular block matrix, we can take this determinant as

$$\det(\tilde{K}) = \det(\text{diagonal of blocks}) = \prod_{i=1}^{n} \det(C\overline{B}) = (\det(C\overline{B}))^n. \tag{47}$$

We cannot split the determinants because these matrices are not square. We suppose this determinant is positive. We assume A has positive determinant (hence it is invertible) and full column rank. We will

also assume C is full row rank, since it has more columns than rows. First, note if $\det(C\overline{B}) > 0$, we must also have $\overline{B}$ is full column rank. This implies $\det(CA^i\overline{B}) > 0$, since

$$\text{column rank}(C\overline{B}) = \text{column rank}(\overline{B}) = \text{column rank}(CA^i\overline{B}). \tag{48}$$

Normally, just because the column rank is the same, we do not have the region in space spanned by the column vectors is the same for each; however, since we assumed $\det(C\overline{B}) > 0$, all of the above matrices are full column rank. However, we do not know the sign of the determinant with respect to i. This leads to us to use squares of the determinants in this proof. We will use what we just established: in particular, using the definition of $\overline{B}$, and what we just established, we have

$$\begin{aligned}
0 &< \det^2(C\overline{B}) && (49)\\
&< \det^2(C\overline{B}) + \underbrace{\det^2(CA\overline{B})}_{>0} + \underbrace{\det^2(CA^2\overline{B})}_{>0} + \ldots + \underbrace{\det^2(CA^{n-1}\overline{B})}_{>0} && (50)\\
&= \sum_{i=0}^{n-1} \det^2(CA^i(\Delta \cdot A)^{-1} \cdot (\exp\{\Delta \cdot A\} - I) \cdot \Delta B) && (51)\\
&= \sum_{i=0}^{n-1} \frac{\Delta}{\Delta} \det^2(CA^iA^{-1}(\exp\{\Delta \cdot A\} - I)B) && (52)\\
&\leq \sum_{i=1}^{n-1} \frac{1}{2} \det^2((A^{-1})^T(A^i)^TC^TCA^iA^{-1} + (\exp\{\Delta \cdot A\} - I)BB^T(\exp\{\Delta \cdot A\} - I)^T) && (53)\\
&\leq \frac{1}{2} \det^2(\sum_{i=0}^{n-1}((A^{-1})^T(A^i)^TC^TCA^iA^{-1} + (\exp\{\Delta \cdot A\} - I)BB^T(\exp\{\Delta \cdot A\} - I)^T)) && (54)
\end{aligned}$$

where the first inequality is by Lemma 1, and the second inequality uses the determinant inequality for positive semi-definite matrices (note a matrix of the form $A^TA + B^TB$ is positive semi-definite, since $x^T(A^TA + B^TB)x = ||Ax||^2 + ||Bx||^2 \geq 0$). Finally, by the assumption, the matrix on the right contributes no new linearly independent columns, we conclude the system is observable. In particular, we use the fact that

$$\det(\sum_{i=1}^{n-1} A_i^TA_i + B_i^TB_i) > 0 \qquad \implies \qquad \begin{pmatrix} A_i \\ A_2 \\ \vdots \\ A_{n-1} \end{pmatrix} \text{ is full column rank.} \tag{55}$$

if the column space $\sum_i B_i^TB_i$ lies within the column space of $\sum_i A_i^TA_i$, which is Lemma 2. Specifically,

$$(A^{-1})^T(A^i)^TC^TCA^iA^{-1} \text{ is full rank} \qquad \implies \qquad (CA^i)^TCA^i \text{ is full rank.} \tag{56}$$

□

Remark. Note it is impossible for C^TC and BB^T to be full column rank, considering we constrain the hidden state to be high-dimensional. A^{-1} must be full rank, row and column, since A must be invertible for A^{-1} to exist.

Remark. The assumption

$$\text{col}(\mathcal{B}) \subseteq \text{col}(\mathcal{A}_1 + \ldots + \mathcal{A}_{n-1}) \tag{57}$$

is generally a nontrivial assumption; however, we can enforce this with machine learning. In particular, we can take

$$\mathcal{B} = (\exp\{\Delta \cdot A\} - I)BB^T(\exp\{\Delta \cdot A\} - I)^T = (\sum_{i=0}^{n-1} \mathcal{A}_i)U = (\sum_{i=0}^{n-1} (A^-1)^T(A^i)^TC^TCA^iA^{-1})U \tag{58}$$

for some matrix U. This is the exact trick we use in Lemma 2. However, this computation is also nontrivial, because it requires a large number of matrix evaluations. We may simplify this using

$$\mathcal{B} = (\sum_{i=0}^{m} \mathcal{A}_i)U = (\sum_{i=0}^{m} (A^{-1})^T(A^i)^TC^TCA^iA^{-1})U \tag{59}$$

for some index truncation $m < n-1$. If m is quite small, this is computationally simple. Yet, there is a challenge in this: we must extract B from this equation. since $\exp\{\Delta \cdot A\} - I$ is invertible, we take

$$BB^T = (\exp\{\Delta \cdot A\} - I)^{-1}(\sum_{i=0}^{m}(A^{-1})^T(A^i)^T C^T C A^i A^{-1})U(\exp\{\Delta \cdot A\} - I)^{-1}. \tag{60}$$

In practice, we can perform the Cholesky decomposition on the matrix on the right, and set B equal to what is given. U can be arbitrary, or learned with a neural network.

Lemma 1. Let Q, P be non-square matrices, and QP square. We have

$$\det(QP) \leq \det(Q^T Q + PP^T). \tag{61}$$

Proof. First, we use the fact that

$$2\sigma_i(QP) = \sigma_i(Q^T Q + PP^T), \tag{62}$$

which holds for all i. This holds even when Q, P are not square. Taking the product, we get

$$|\det(QP)| = \prod_i \sigma_i(QP) \leq \frac{1}{2}\prod_i \sigma_i(Q^T Q + PP^T), \tag{63}$$

where the above is assured using the positive semi-definitness of $Q^T Q$ and PP^T . Thus,

$$\prod_i \sigma_i(Q^T Q + PP^T) = \det(Q^T Q + PP^T). \tag{64}$$

Using our Q and P of interest, we get

$$|\det((CA^iA^{-1})((\exp\{\Delta \cdot A\} - I)B))| = \prod_i \sigma_i((CA^iA^{-1})((\exp\{\Delta \cdot A\} - I)B)) \tag{65}$$

$$\leq \det((A^{-1})^T(A^i)^T C^T C A^i A^{-1} + (\exp\{\Delta \cdot A\} - I)BB^T(\exp\{\Delta \cdot A\} - I)^T). \tag{66}$$

□

Lemma 2. We have

$$\det(\sum_{i=1}^{n-1} A_i^T A_i + B_i^T B_i) > 0 \qquad \implies \qquad \mathcal{O} = \begin{pmatrix} A_i \\ A_2 \\ \vdots \\ A_{n-1} \end{pmatrix} \text{ is full column rank.} \tag{67}$$

if the column space of $\sum_i B_i^T B_i$ lies within the column space of $\sum_i A_i^T A_i$.

Proof. Since $\text{col}(\sum_i B_i^T B_i) \subseteq \text{col}(\sum_i A_i^T A_i)$, we have $\sum_i B^T B = (\sum_i A_i^T A_i)U$ for some U, since the left matrix sum must contain linear combinations of the right matrix sum. Thus,

$$\det(\sum_{i=1}^{n-1} A_i^T A_i + B_i^T B_i) = \det((\sum_{i=1}^{n-1} A_i^T A_i)(I + U)). \tag{68}$$

Therefore, the rank of this matrix is limited by $\sum_i A_i^T A_i$ which is exactly $\mathcal{O}^T\mathcal{O}$. We must have it is full column rank for the determinant to be positive, and we are done. Alternatively, we have

$$\det((\sum_{i=1}^{n-1} A_i^T A_i)(I + U)) = \det(\sum_i A_i^T A_i)\det(I + U) > 0, \tag{69}$$

since both matrices are square, and the inequality is by assumption. Note it is possible $\det(I + U)$ is negative, but $\det(\sum_i A_i^T A_i)$ can only be positive for the above to hold, therefore they are both positive. □

Often, we do not work with $\tilde{K}$ for computational and memory purposes, and we work with similar FFTs instead. We prove the following theorem using FFTs. Denote $\mathcal{F}$ the discrete Fast Fourier transform. Now, denote

$$\mathcal{K} = \mathcal{F}[C\overline{B}, C\overline{AB}, C\overline{A}^2\overline{B}, \ldots] = (\mathcal{F}[C\overline{B}], \mathcal{F}[C\overline{AB}], \mathcal{F}[C\overline{A}^2\overline{B}], \ldots) = (V_1, V_2, V_3, \ldots), \tag{70}$$

where the above Fourier transforms are applied to the individual matrices in each concatenation.

Theorem. Assume for generality $C\overline{B}$ is not square. Suppose $\det(\mathcal{F}[(C\overline{B})^T C\overline{B}]) > 0$, and that the region $(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}$ spanned by the column space is the same of the column space as $(C\overline{B})^T C\overline{B}$ for all j, k, and that $(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}$ is positive semi-definite. Then the system is observable.

Proof. First consider,

$$V_k = \sum_j^{n-1} C\overline{A}^j\overline{B}e^{-2\pi ikj/n}. \tag{71}$$

For the sake of generality, we will assume CA^iB is not square, i.e. the output dimension of the state-space model differs from the input dimension. Hence V_k is not square, so we consider

$$\det(V_k^T V_k) = \det\Big((\sum_j^{n-1} C\overline{A}^j\overline{B}e^{-2\pi ikj/n})^T(\sum_j^{n-1} C\overline{A}^j\overline{B}e^{-2\pi ikj/n})\Big). \tag{72}$$

We examine $k = 0$, which simplifies this greatly. Since all matrix are positive semi-definite, we have

$$\det(V_0^T V_0) = \det\Big((\sum_j^{n-1} C\overline{A}^j\overline{B}e^{-2\pi i0j/n})^T(\sum_j^{n-1} C\overline{A}^j\overline{B}e^{-2\pi i0j/n})\Big) \tag{73}$$

$$= \det\Big((C\overline{B})^T C\overline{B} + \sum\sum\nolimits_{j+k>0}(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}\Big) > 0 \tag{74}$$

Now, if $C\overline{B}$ is not full column rank, then neither is $(C\overline{B})^T C\overline{B}$ which has determinant 0. Thus, $C\overline{AB}, C\overline{A}^2B, \ldots$ cannot be full column rank either, which gives a contradiction of the assumption. In particular, column rank$(C\overline{AB}) \leq$ column rank$(C\overline{B})$. However, the spanning of the column space is not sufficient for the determinant of the sum to be positive. Consequently, we make the restriction $(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}$ is positive semi-definite for all j, k. Now,

$$\det\Big((C\overline{B})^T C\overline{B}+\sum\sum\nolimits_{j+k>0}(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}\Big) \geq \det((C\overline{B})^T C\overline{B})+\sum\sum\nolimits_{j+k>0}\det((C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}) \overset{(?)}{>} 0. \tag{75}$$

Since we assume all matrices are positive semi-definite, and the column space is the same for all, we must have the final equality: we must have either each determinant in the sum is zero or positive. We cannot have a mixture of both since each spans the same column space. Also, by positive semi-definiteness, we cannot have a negative determinant for any term. If not all matrices have positive determinant, all individual matrices in the sum must have determinant 0, which is a contradiction to

$$\det\Big((C\overline{B})^T C\overline{B} + \sum\sum\nolimits_{j+k>0}(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}\Big) > 0. \tag{76}$$

This observable using the null space. We know the null space of $(C\overline{B})^T C\overline{B}$ is a subset of $\sum_{j+k>0}(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}$. Consider a vector u in the nullspace

$$((C\overline{B})^T C\overline{B} + \sum\sum\nolimits_{j+k>0}(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B})u = \underbrace{((C\overline{B})^T C\overline{B})u}_{=0}+(\sum\sum\nolimits_{j+k>0}\underbrace{((C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B})u}_{=0}), \tag{77}$$

thus the nullspace is nontrivial, i.e.,

$$\dim\Big(\ker\Big((C\overline{B})^T C\overline{B} + \sum\sum\nolimits_{j+k>0}(C\overline{A}^j\overline{B})^T C\overline{A}^k\overline{B}\Big)\Big) > 0 \tag{78}$$

and we must have the determinant is 0, a contradiction. Therefore, since all matrices span the same column space, we must have each determinant is nonzero, thus the desired determinant is nonzero. Thus,

a sufficient condition is that

$$\begin{cases} \det(V_0^T V_0) = \det(\mathcal{F}[C\overline{B}]^T \mathcal{F}[C\overline{B}]) > 0 \quad \text{implies} \quad \det((C\overline{B})^T (C\overline{B})) > 0, \\ \text{if } (C\overline{A}^j \overline{B})^T C\overline{A}^k \overline{B} \text{ spans the same column space as } (C\overline{B})^T C\overline{B} \text{ for all } j,k \\ \text{and if } (C\overline{A}^j \overline{B})^T C\overline{A}^k \overline{B} \text{ is positive semi-definite.} \end{cases} \tag{79}$$

□

B Long-term observability analysis

Remark. We use the following convention for logarithms:

$$\log(0) = \lim_{a \to 0^+} \log(a) = -\infty. \tag{80}$$

Definition. We define the state-transition φ to be the matrix

$$\varphi(t,0) = I + \sum_{i=1}^{\infty} \underbrace{\int_0^t A(\omega_1) \dots \int_0^t A(\omega_i) d\omega_i \dots d\omega_1}_{i \text{ times}} = \exp\Big\{ \mathcal{T} \int_0^t A(\omega) d\omega \Big\}, \tag{81}$$

where $\mathcal{T}$ is the time-ordering operator. For us, we can compute φ using A as

$$\varphi(t,0) = \exp\Big\{ \sum_i A^T \Delta t \Big\} = \exp\{At\}. \tag{82}$$

Consider the matrix M, which we call the observability Gramian

$$M(t_0, t_1) = \int_{t_0}^{t_1} \varphi^T(t, t_0) C(t)^T C(t) \varphi(t, t_0) dt. \tag{83}$$

To clarify, we use notation

$$\int_{t_0}^{t_1} \varphi^T(t, t_0) C(t)^T C(t) \varphi(t, t_0) dt = \int_{t_0}^{t_1} \psi(t, t_0) dt = \begin{pmatrix} \int_{t_0}^{t_1} (\psi(t,t_0))_{11} dt & \dots & \int_{t_0}^{t_1} (\psi(t,t_0))_{1n} dt \\ \vdots & \ddots & \vdots \\ \int_{t_0}^{t_1} (\psi(t,t_0))_{n1} dt & \dots & \int_{t_0}^{t_1} (\psi(t,t_0))_{nn} dt \end{pmatrix}, \tag{84}$$

and so the integration is element-wise.

We are interested in showing this matrix is nonsingular, thus the state-space ODE system is observable. This is true when M is nonsingular. It suffices to find an upper bound on the absolute value of the determinant of the inverse. In particular, we desire

$$|\det(M^{-1})| = \frac{1}{|\det(M)|}, |\det(M^{-1})| \leq a \implies |\det(M)| \geq \frac{1}{a}. \tag{85}$$

As a note, notice the above determinant is nonnegative since M is positive semi-definite. It is sufficient to show

$$\det(M^{-1}) < \infty. \tag{86}$$

However, if $\det(M) \neq 0$, we must also have $\det(M^{-1}) \neq 0$. Hence, our condition of interest is that

$$0 < \det(M^{-1}) < \infty, \qquad \text{or equivalently} \qquad -\infty < \log\det(M^{-1}) < \infty, t_1 < \infty. \tag{87}$$

We must also have the determinant is finite. If not, i.e. $\det(M^{-1}) = \infty$, then ordinarily $\det(M) = 0$, thus it is not invertible. Note the upper bound is of greater interest. In a practical sense, from the machine learning perspective, we generally operate such that $t_1 < \infty$, and so the lower bound is of lesser interest, as this time condition guarantees this bound is satisfied when A, C are constant valued. Now, consider

$$\log\det(M^{-1}) = \log\det((\int_{t_0}^{t_1} \varphi^T(t,t_0) C^T C \varphi(t,t_0) dt)^{-1}) \tag{88}$$

$$= \log(\det(\int_{t_0}^{t_1} \varphi^T(t,t_0) C^T C \varphi(t,t_0) dt))^{-1} \tag{89}$$

$$= -\log\det(\int_{t_0}^{t_1} \varphi^T(t,t_0) C^T C \varphi(t,t_0) dt). \tag{90}$$

Note we require the determinant to be positive, otherwise the above is undefined. Since M is positive semi-definite, we know this to be true, as a positive semi-definite matrix has nonnegative eigenvalues and a determinant is a product of eigenvalues. Now, observe the more important bound for us, in a practical sense, is

$$-\infty < \log\det(\int_{t_0}^{t_1} \varphi^T(t,t_0)C^TC\varphi(t,t_0)dt), \tag{91}$$

i.e. the lower bound is of interest. Now, we can use Jensen's inequality by treating the integral as a uniform expected value. We have

$$\log\det(\int_{t_0}^{t_1} \varphi^T(t,t_0)C^TC\varphi(t,t_0)dt) \geq \int_{t_0}^{t_1} \log\det(\varphi^T(t,t_0)C^TC\varphi(t,t_0))dt, \tag{92}$$

where we also use the matrix of interest is positive semi-definite, as the log-determinant is concave in this setting. Now, it comes into use that we take a log of the determinant. We use the fact

$$\log\det(e^Q) = \mathrm{Tr}(Q). \tag{93}$$

First, recall the determinant of the product is the product of determinant. A matrix exponential is always square, and C^TC is square (but C is not necessarily square). Hence, we get

$$\log\det(\varphi^T(t,t_0)C^TC\varphi(t,t_0)) = \log(\det^2(\varphi(t,t_0)\det(C^TC)) \tag{94}$$

$$= \log\det^2(\exp\int_{t_0}^{t} A(\tau)d\tau) + \log\det(C^TC) \tag{95}$$

$$= 2\mathrm{Tr}(\int_{t_0}^{t} A(\tau)d\tau) + \log(\det(C^TC)). \tag{96}$$

Now, consider the case $t_1 \to \infty$. Therefore, we require

$$2(\int_{t_0}^{\infty} \mathrm{Tr}(\int_0^t A(\tau)d\tau)dt + \log(\det(C^TC))) = 2(\int_{t_0}^{\infty} \mathrm{Tr}(\int_0^t A(\tau)d\tau)dt + \log(\det(C^TC))) > -\infty. \tag{97}$$

Obviously, the above holds if $\det(C^TC) > 0$ and the trace term is nonnegative; however, this trace condition will contradict what we will see later.

Now, suppose A is a constant valued matrix, as is the case with a Mamba neural operator. Thus, we desire

$$2\int_0^{\infty} (\mathrm{Tr}(A)t + \log(\det(C^TC)))dt > -\infty. \tag{98}$$

We observe it is necessary that $\mathrm{Tr}(A) \geq 0$ for the above to hold.

Now, we examine the upper bound such that $\log\det(M^{-1}) < \infty$. We examine

$$\log\det(\int_0^{\infty} \varphi^T(t,t_0)C^TC\varphi(t,t_0)dt) < \infty. \tag{99}$$

It is sufficient to show

$$\int_0^{\infty} \varphi^T(t,t_0)C^TC\varphi(t,t_0)dt < \infty \tag{100}$$

has finite-valued entries. Consider, again by Jensen's inequality but for convex functions, and by the Cauchy-Schwarz inequality,

$$\left|\left|\int_0^{\infty} \varphi^T(t,t_0)C^TC\varphi(t,t_0)dt\right|\right|_F \leq \int_0^{\infty} ||\varphi(t,t_0)C||_F^2 dt \leq \int_0^{\infty} ||\varphi(t,t_0)||_F^2||C||_F^2 dt \tag{101}$$

$$\leq \int_0^{\infty} ||e^{\int_0^t A(\tau)d\tau}||_F^2||C(t)||_F^2 dt \leq M\int_0^{\infty} ||e^{\int_0^t A(\tau)d\tau}||_F^2 < \infty. \tag{102}$$