

AN ADJOINT KRYLOV APPROACH TO PARAMETER INFERENCE FOR AFFINE FAMILIES OF LINEAR EVOLUTION EQUATIONS

ADITYA DENDUKURI*, SHIVKUMAR CHANDRASEKARAN†, AND LINDA PETZOLD†

Abstract. This paper introduces an adjoint Krylov method for parameter inference in large, sparse linear evolution equations whose system matrices depend affinely on unknown coefficients. We estimate these coefficients from discrete observations by constructing one forward and one adjoint Krylov approximation on each observation interval. Quadrature of the resulting bilinear form gives the full coefficient gradient through sparse matrix–vector products, without evaluating a separate Fréchet derivative in every operator direction. We separate the gradient error into forward, adjoint, and quadrature terms and derive residual-based bounds for Markov and Lindblad generators. We also identify cases in which heuristic stopping estimates become small before the gradient has converged. With workspace reuse, we compute library gradients for a 12-qubit Lindblad generator with dimension 1.7×10^7 on a single machine with 16 GB of memory. Experiments on chemical master, Kuramoto–Sivashinsky, and Lindblad equations demonstrate accurate gradients and parameter recovery.

Key words. matrix exponential, Fréchet derivative, Krylov methods, adjoint methods, sparse identification, inverse problems

AMS subject classifications. 65F60, 65F50, 65L09, 65K10

1. Introduction. Many evolution models become large systems of linear ordinary differential equations after state-space truncation or spatial discretization. We consider models of the form

$$(1.1) \quad \dot{\mathbf{u}}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t), \quad \mathbf{A}(\boldsymbol{\theta}) = \mathbf{A}_0 + \sum_j \theta_j \mathbf{A}_j, \quad \mathbf{u}(t+h) = e^{h\mathbf{A}(\boldsymbol{\theta})}\mathbf{u}(t),$$

where the matrices $\mathbf{A}_j$ are known and the coefficients θ_j are to be estimated from observations at discrete times. Depending on the application, $\mathbf{u}$ may be a probability distribution, a discretized field, or a vectorized density matrix, while the directions $\mathbf{A}_j$ describe candidate reactions and propensities, differential operators, or Hamiltonian and dissipative terms. If the candidate library contains more terms than the model requires, sparse optimization selects a small active set.

A central bottleneck in parameter inference is the evaluation of gradients through the exponential action in (1.1). One can compute a separate Fréchet derivative in every library direction, but this repeats the same propagation for each coefficient. Forming the derivative with respect to all entries of $\mathbf{A}$ avoids this repetition only by introducing a dense matrix that is too large to store. For scalar losses, this difficulty can be avoided with the classical adjoint formula, which expresses every coefficient derivative in terms of the same forward and adjoint solutions, so that only the contraction against $\mathbf{A}_j$ changes across the library.

In this paper we compute the forward and adjoint paths in separate Krylov spaces and integrate the resulting bilinear form by quadrature. The two spaces are constructed once for an observation interval and are then reused for every library direction. At each quadrature node, we contract the forward and adjoint vectors against $\mathbf{A}_j$, requiring one sparse matrix–vector product and one inner product per direction. The adjoint path also gives the derivative with respect to the state entering the interval,

*Department of Computer Science, University of California, Santa Barbara (aditya.dendukuri@ucsb.edu, petzold@engineering.ucsb.edu).

†Department of Electrical and Computer Engineering, University of California, Santa Barbara (shiv@ece.ucsb.edu).

while differentiating the endpoint with respect to h gives the interval derivative. We use these derivatives when the exponential action appears inside a nonlinear time step or follows a parameter map.

The forward Arnoldi approximation, the adjoint Arnoldi approximation, and the quadrature rule introduce separate errors. We derive a gradient-error decomposition that retains all three contributions and use the Arnoldi residuals to bound the first two. For Markov and Lindblad generators, contractivity yields computable bounds in the natural paired norms. Keeping the forward and adjoint errors separate also reveals why two commonly used refinement estimates can become small too soon. A multiplicative estimate vanishes when either Arnoldi process terminates, even if the other process remains unresolved, while a difference between successive approximations can be misleading during slow convergence.

We test the method on three parameter-inference problems. The first recovers a sparse chemical master equation from distribution snapshots separated by long observation intervals. The second differentiates a Kuramoto–Sivashinsky time step in which a nonlinear map precedes the exponential action. The third estimates Hamiltonian and dissipative terms in a Lindblad equation with complex arithmetic and large vectorized density states. Together, these examples test the calculation across stiff, nonnormal, nonlinear, and complex-valued models.

1.1. Contributions. The principal contributions are as follows.

- We give an adjoint Krylov algorithm for the coefficient, input-state, and interval derivatives of an affine exponential action. The algorithm shares one forward Krylov space and one adjoint Krylov space across the complete operator library.
- We give a one-workspace quadrature schedule that stores one Krylov basis and the forward path at the quadrature nodes. A projected-core calculation can use comparable memory by rebuilding one of its two Krylov bases.
- We separate the gradient error into forward projection, adjoint projection, quadrature, and cross terms. Arnoldi residuals bound the two projection errors, and contractivity gives computable bounds for Markov and Lindblad generators.
- We identify one-sided termination and slow convergence as two cases in which refinement estimates can report a small value before the gradient is resolved, and we compare these estimates with independent reference gradients.
- We use the adjoint Krylov algorithm for sparse parameter recovery in chemical master, Kuramoto–Sivashinsky, and Lindblad equations.

1.2. Related work. Arnoldi approximations of matrix-exponential actions are standard tools for large sparse evolution equations [26, 13, 2, 32, 14, 37]. Fréchet derivatives can be evaluated through a block exponential [1], or approximated by block and low-rank Krylov methods [17, 21, 16, 22]. Kressner [21] introduced the two-sided Krylov projection used here, and Kandolf et al. [16] analyzed the resulting low-rank approximation and proposed practical stopping estimates. For fixed forward and adjoint bases, the projected method evaluates the reduced Fréchet derivative directly, whereas our calculation applies quadrature to its path representation. We use the projected form as a reference calculation and study the stopping estimates in Section 5.2.

Continuous-adjoint methods compute the gradient of a scalar objective with one forward and one backward solve [8, 36]. For the autonomous problem in (1.1), both solutions can be represented in Krylov spaces and evaluated at the quadrature nodes

from small projected matrices. Matfree instead differentiates the finite Arnoldi recurrence [20]. We compare the two approaches at matched gradient accuracy in Section 5.6.

Sparse equation-discovery methods fit time derivatives, increments, or weak forms to a candidate library [7, 15, 30, 33, 25, 24, 4]. We instead fit the finite-time evolution generated by the library. In the Brusselator comparison of Section 6.2, the observations and candidate terms are held fixed so that only the fitting method changes.

Data-driven operator methods infer an unrestricted discrete-time or infinitesimal evolution operator [41, 34, 18, 28, 19], whereas our coefficients multiply a given mechanistic library. Other inference methods for stochastic reaction networks use path likelihoods, approximate Bayesian computation, or moment closures [11, 6, 40, 29, 38, 35]. For Lindblad dynamics, we use libraries of Hamiltonian commutators and dissipative channels [12, 23], while related identification methods consider steady-state and restricted observations [3, 9].

1.3. Outline. Section 2 states the affine inverse problem and finite-time loss. Section 3 derives the gradient calculation and its cost. Section 4 separates the quadrature and projection errors. Section 5 studies numerical accuracy, stopping, and computational cost. Section 6 presents the three inverse problems. The supplementary material contains implementation details and additional experiments.

2. Parameter Inference for Evolution Operators. We begin with a generator assembled from known library directions and then define the finite-time loss whose gradient is required by the inference procedure.

2.1. Operator library. Let $\mathbf{u}(t) \in \mathbb{C}^n$ satisfy

$$(2.1) \quad \frac{d\mathbf{u}}{dt} = \mathbf{A}(\boldsymbol{\theta})\mathbf{u}, \quad \mathbf{A}(\boldsymbol{\theta}) = \mathbf{A}_0 + \sum_{j=1}^{M_{\text{lib}}} \theta_j \mathbf{A}_j.$$

The fixed matrix $\mathbf{A}_0$ and the M_{lib} library directions $\mathbf{A}_j$ are known, while the real coefficients θ_j are to be estimated. The state and generator may be real or complex, while $\boldsymbol{\theta}$ is subject only to the sign or feasibility constraints imposed by the model. Differentiating $\mathbf{A}(\boldsymbol{\theta})$ does not require sparse coefficients, since sparsity enters only through the outer optimization used to select active directions from an overcomplete library.

We index observation intervals by k and quadrature nodes by q . The forward and adjoint Krylov depths are m_f and m_a , and we use m_K when the depths are equal. The symbol $*$ denotes conjugate transpose and reduces to the ordinary transpose in the real case.

2.2. Finite-time loss. Consider an observation interval of length h with initial state $\mathbf{v}$. Its final state is

$$(2.2) \quad \mathbf{y} = e^{h\mathbf{A}(\boldsymbol{\theta})}\mathbf{v}.$$

For a real-valued loss of a complex state, we define gradients using the real part of the Hermitian product. If C is a linear observation map and $\mathbf{z}$ is the observed value, the least-squares loss

$$\ell(\mathbf{y}) = \frac{1}{2} \|C\mathbf{y} - \mathbf{z}\|_2^2 \quad \text{has} \quad \nabla_{\mathbf{y}}\ell = C^*(C\mathbf{y} - \mathbf{z}).$$

We start the adjoint calculation from $\nabla_{\mathbf{y}}\ell$. For another differentiable loss, we compute its gradient with respect to $\mathbf{y}$ and use that vector instead.

For full-state probability distributions $\hat{\mathbf{p}}_k$ observed at equally spaced times, we use the one-step prediction loss

$$(2.3) \quad \mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{k=0}^{T-1} \|e^{\mathbf{A}(\boldsymbol{\theta})\Delta t} \hat{\mathbf{p}}_k - \hat{\mathbf{p}}_{k+1}\|_2^2.$$

We can use this calculation for nonuniform observation times, partial observations, and other differentiable losses. For each observation interval, we compute its length and the gradient of the loss with respect to the propagated state, using the appropriate observation map. Each observation interval has its own forward and adjoint bases, which are shared by all library directions on that interval. If the coefficients are outputs of another model, the chain rule maps the coefficient gradient back to the parameters of that model. The neural context-to-rate example in the supplementary material illustrates this composition.

3. Adjoint Krylov Gradient Computation. The coefficient gradient is obtained by pairing the parameter variation of the forward equation with an adjoint solution. We first derive this identity and then evaluate it using separate forward and adjoint Krylov spaces.

3.1. Forward–adjoint formula. The exponential action in (2.2) is the endpoint of the linear initial-value problem

$$(3.1) \quad \mathbf{f}'(s) = h\mathbf{A}(\boldsymbol{\theta})\mathbf{f}(s), \quad \mathbf{f}(0) = \mathbf{v}, \quad \mathbf{y} = \mathbf{f}(1).$$

When we differentiate this ODE with respect to a coefficient, we obtain a forced linear ODE. Its solution follows from variation of constants, which propagates the forcing applied at time s over the remaining interval $1 - s$ and integrates these contributions over $0 \leq s \leq 1$. Pairing this solution with the gradient of the loss introduces the forward and adjoint paths

$$\mathbf{f}(s) = e^{sh\mathbf{A}}\mathbf{v}, \quad \boldsymbol{\lambda}(s) = e^{(1-s)h\mathbf{A}^*} \nabla_{\mathbf{y}}\ell,$$

and gives the standard continuous-adjoint sensitivity formula [8].

PROPOSITION 3.1 (Adjoint gradient formula). *For the action (2.2), the derivative of the loss with respect to coefficient θ_j is*

$$(3.2) \quad \frac{\partial \ell}{\partial \theta_j} = h \operatorname{Re} \int_0^1 \boldsymbol{\lambda}(s)^* \mathbf{A}_j \mathbf{f}(s) ds.$$

Proof. Since $\partial \mathbf{A} / \partial \theta_j = \mathbf{A}_j$, differentiating Eq. (3.1) gives

$$(3.3) \quad \frac{d}{ds} \frac{\partial \mathbf{f}(s)}{\partial \theta_j} = h\mathbf{A} \frac{\partial \mathbf{f}(s)}{\partial \theta_j} + h\mathbf{A}_j \mathbf{f}(s), \quad \frac{\partial \mathbf{f}(0)}{\partial \theta_j} = 0.$$

Applying variation of constants at $s = 1$ gives

$$(3.4) \quad \frac{\partial \mathbf{y}}{\partial \theta_j} = h \int_0^1 e^{(1-s)h\mathbf{A}} \mathbf{A}_j \mathbf{f}(s) ds.$$

Applying the chain rule to $\ell(\mathbf{y})$ and taking the real part in the complex case gives

$$\begin{aligned}
 \frac{\partial \ell}{\partial \theta_j} &= \operatorname{Re} \left((\nabla_{\mathbf{y}} \ell)^* \frac{\partial \mathbf{y}}{\partial \theta_j} \right) \\
 &= h \operatorname{Re} \int_0^1 (\nabla_{\mathbf{y}} \ell)^* e^{(1-s)h\mathbf{A}} \mathbf{A}_j e^{sh\mathbf{A}} \mathbf{v} \, ds \\
 &= h \operatorname{Re} \int_0^1 \left(e^{(1-s)h\mathbf{A}^*} \nabla_{\mathbf{y}} \ell \right)^* \mathbf{A}_j (e^{sh\mathbf{A}} \mathbf{v}) \, ds \\
 (3.5) \quad &= h \operatorname{Re} \int_0^1 \boldsymbol{\lambda}(s)^* \mathbf{A}_j \mathbf{f}(s) \, ds.
 \end{aligned}$$

The last equality uses $(\nabla_{\mathbf{y}} \ell)^* e^{(1-s)h\mathbf{A}} = (e^{(1-s)h\mathbf{A}^*} \nabla_{\mathbf{y}} \ell)^*$. $\square$

The formula separates propagation from the library direction. We compute $\mathbf{f}$ and $\boldsymbol{\lambda}$ once, while $\mathbf{A}_j$ enters only through the bilinear product $\boldsymbol{\lambda}(s)^* \mathbf{A}_j \mathbf{f}(s)$.

The coefficient derivative can also be viewed as a Fréchet derivative, which measures the first-order change of a matrix function in a direction $\mathbf{E}$. Formally, $L_f(\mathbf{A}, \mathbf{E})$ is defined by $f(\mathbf{A} + \varepsilon \mathbf{E}) = f(\mathbf{A}) + \varepsilon L_f(\mathbf{A}, \mathbf{E}) + o(\varepsilon)$. For the exponential, it satisfies [13, Eq. (10.15)]

$$(3.6) \quad L_{\exp}(h\mathbf{A}, h\mathbf{E}) = h \int_0^1 e^{(1-s)h\mathbf{A}} \mathbf{E} e^{sh\mathbf{A}} \, ds.$$

Evaluating this integral separately for every $\mathbf{A}_j$ repeats the two exponential propagations. Pairing the derivative with the loss gradient replaces these repeated evaluations by the shared forward and adjoint paths.

For interval k of the snapshot loss (2.3), the prediction error is

$$\mathbf{r}_k = e^{\mathbf{A}\Delta t} \hat{\mathbf{p}}_k - \hat{\mathbf{p}}_{k+1}.$$

The forward path begins at $\hat{\mathbf{p}}_k$, while the adjoint path begins at $\mathbf{r}_k$ at the right endpoint of the interval. Thus,

$$\mathbf{f}_k(s) = e^{s\mathbf{A}\Delta t} \hat{\mathbf{p}}_k, \quad \boldsymbol{\lambda}_k(s) = e^{(1-s)\mathbf{A}^T \Delta t} \mathbf{r}_k.$$

For one transition, $\ell_k = \frac{1}{2} \|\mathbf{r}_k\|_2^2$, with differential

$$(3.7) \quad d\ell_k = \mathbf{r}_k^T d\mathbf{r}_k = \mathbf{r}_k^T d(e^{\mathbf{A}\Delta t} \hat{\mathbf{p}}_k).$$

The differential shows that the adjoint path for interval k starts from $\mathbf{r}_k$ at its right endpoint. Substitution into Proposition 3.1, followed by summation over the observation intervals, gives

$$(3.8) \quad \frac{\partial \mathcal{L}}{\partial \theta_j} = \Delta t \sum_{k=0}^{T-1} \int_0^1 \boldsymbol{\lambda}_k(s)^T \mathbf{A}_j \mathbf{f}_k(s) \, ds.$$

For another loss, $\mathbf{r}_k$ is replaced by the gradient of that interval's loss with respect to its predicted state. Figure 1 summarizes the two stages, and Algorithm 3.1 gives the calculation for one exponential action. The forward basis produces the prediction, whose loss gradient starts the adjoint Arnoldi process. If the input state or this loss gradient is zero, we take the corresponding path to be zero and do not normalize its starting vector.

3.2. Krylov approximation and quadrature. The forward and adjoint paths require exponential actions of $\mathbf{A}$ and $\mathbf{A}^*$. Arnoldi projection replaces each large exponential by the exponential of a small upper-Hessenberg matrix. For an operator B and a starting vector v , the associated Krylov space is

$$\mathcal{K}_m(B, v) = \text{span}\{v, Bv, \dots, B^{m-1}v\}$$

. Suppose that $v = \beta v_1$, where $\beta = \|v\|_2$. After m Arnoldi steps, the columns of $V_m = [v_1, \dots, v_m]$ form an orthonormal basis for this space, and the projected matrix $H_m = V_m^* B V_m$ is upper Hessenberg [14, 37]. If δ_k denotes the k th coordinate vector, the Arnoldi relation and the resulting exponential approximation are

$$(3.9) \quad B V_m = V_m H_m + h_{m+1,m} v_{m+1} \delta_m^*, \quad e^{\tau B} v \approx \beta V_m e^{\tau H_m} \delta_1.$$

The second relation follows by approximating the solution of $u'(\tau) = Bu(\tau)$, $u(0) = v$, in the form $u_m(\tau) = V_m z_m(\tau)$. Requiring the ODE residual to be orthogonal to $\mathcal{K}_m(B, v)$ gives

$$(3.10) \quad \begin{aligned} V_m^* (V_m z_m'(\tau) - B V_m z_m(\tau)) &= 0, \\ z_m'(\tau) &= H_m z_m(\tau), \quad z_m(0) = \beta \delta_1. \end{aligned}$$

Thus $z_m(\tau) = \beta e^{\tau H_m} \delta_1$, and the large exponential action is computed from an $m \times m$ exponential followed by a multiplication by V_m . The Arnoldi relation also shows that the defect in the original ODE lies in the direction v_{m+1} . Its norm is

$$(3.11) \quad \rho_m(\tau) = \beta |h_{m+1,m}| |\delta_m^* e^{\tau H_m} \delta_1|.$$

For a strongly nonnormal matrix, a small residual does not by itself imply a small error because the remaining evolution may amplify the defect. Section 4 accounts for this amplification in the error bounds. We also compare successive Krylov depths as a numerical diagnostic.

We construct the forward and adjoint spaces independently, with depths m_f and m_a . When the depths are equal, we denote their common value by m_K . For snapshot interval k , let

$$(\mathbf{V}_k^f, \mathbf{H}_k^f, \beta_k^f) \leftarrow \text{arnoldi}(\mathbf{A}, \hat{\mathbf{p}}_k; m_f), \quad (\mathbf{V}_k^a, \mathbf{H}_k^a, \beta_k^a) \leftarrow \text{arnoldi}(\mathbf{A}^T, \mathbf{r}_k; m_a),$$

where $\beta_k^f = \|\hat{\mathbf{p}}_k\|_2$ and $\beta_k^a = \|\mathbf{r}_k\|_2$. Since the first Arnoldi vector is the normalized starting vector, $\mathbf{V}_k^{fT} \hat{\mathbf{p}}_k = \beta_k^f \delta_1$ and $\mathbf{V}_k^{aT} \mathbf{r}_k = \beta_k^a \delta_1$. The projected forward and adjoint paths are therefore

$$\mathbf{f}_k(s) \approx \beta_k^f \mathbf{V}_k^f e^{s \Delta t \mathbf{H}_k^f} \delta_1, \quad \lambda_k(s) \approx \beta_k^a \mathbf{V}_k^a e^{(1-s) \Delta t \mathbf{H}_k^a} \delta_1.$$

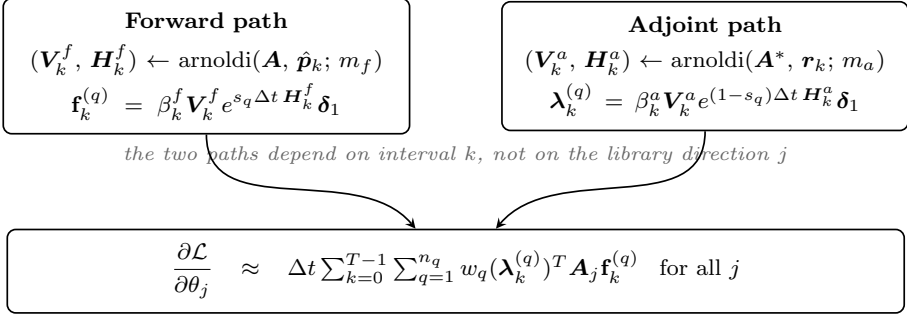
At the q th Gauss–Legendre node, the projected paths are

$$\mathbf{f}_k^{(q)} = \beta_k^f \mathbf{V}_k^f e^{s_q \Delta t \mathbf{H}_k^f} \delta_1, \quad \lambda_k^{(q)} = \beta_k^a \mathbf{V}_k^a e^{(1-s_q) \Delta t \mathbf{H}_k^a} \delta_1.$$

Using the corresponding weights w_q in (3.8) gives

$$(3.12) \quad \frac{\partial \mathcal{L}}{\partial \theta_j} \approx \Delta t \sum_{k=0}^{T-1} \sum_{q=1}^{n_q} w_q \text{Re} \left[(\lambda_k^{(q)})^* \mathbf{A}_j \mathbf{f}_k^{(q)} \right].$$

Step 1: compute the forward and adjoint paths *once per observation interval*



Step 2: contract the paths with the library directions
all M_{lib} directions, $O(\text{nnz}(\mathbf{A}_j) + n)$ per direction per (k, q)

FIG. 1. The forward and adjoint paths are computed once for each observation interval and then reused for every library direction.

Algorithm 3.1 Adjoint Krylov gradient for one exponential action

Require: Generator $\mathbf{A}$, directions $\{\mathbf{A}_j\}$, state $\mathbf{v}$, interval h , differentiable loss $\ell(\mathbf{y})$, depths (m_f, m_a) , Gauss–Legendre nodes and weights $\{s_q, w_q\}$ on $[0, 1]$

Ensure: Prediction $\mathbf{y}$ and derivatives $\nabla_{\theta} \ell$, $\nabla_{\mathbf{v}} \ell$, and $\partial \ell / \partial h$

- 1: $(V_f, H_f, \beta_f) \leftarrow \text{arnoldi}(\mathbf{A}, \mathbf{v}; m_f)$
 - 2: $\mathbf{y} \leftarrow \beta_f V_f e^{h H_f} \delta_1$ and compute the loss gradient $\mathbf{g}_y \leftarrow \nabla_{\mathbf{y}} \ell(\mathbf{y})$
 - 3: $(V_a, H_a, \beta_a) \leftarrow \text{arnoldi}(\mathbf{A}^*, \mathbf{g}_y; m_a)$
 - 4: $\nabla_{\mathbf{v}} \ell \leftarrow \beta_a V_a e^{h H_a} \delta_1$
 - 5: **for** $q = 1, \dots, n_q$ **do**
 - 6: $\mathbf{f}_q \leftarrow \beta_f V_f e^{s_q h H_f} \delta_1$, $\lambda_q \leftarrow \beta_a V_a e^{(1-s_q) h H_a} \delta_1$
 - 7: **end for**
 - 8: **for** $j = 1, \dots, M_{\text{lib}}$ **do**
 - 9: $\partial \ell / \partial \theta_j \leftarrow h \text{Re} \sum_q w_q \lambda_q^* \mathbf{A}_j \mathbf{f}_q$
 - 10: **end for**
 - 11: $\partial \ell / \partial h \leftarrow \text{Re}(\mathbf{g}_y^* \mathbf{A} \mathbf{y})$
-

For fixed k and q , the two path vectors are independent of the library index j . We compute them once and then evaluate the contractions

$$(3.13) \quad c_{k,q}^{(j)} = (\lambda_k^{(q)})^* \mathbf{A}_j \mathbf{f}_k^{(q)}$$

for all library directions. When $\mathbf{A}_j$ is sparse, each contraction requires one sparse matrix–vector product and one inner product. Enlarging the library therefore adds contractions but does not require new Krylov spaces or exponential actions. If $\mathbf{A}_j$ commutes with $\mathbf{A}$, then $\lambda(s)^* \mathbf{A}_j \mathbf{f}(s)$ is independent of s , so a one-node rule introduces no quadrature error.

Figure 1 separates the calculation into two steps. We first compute the forward and adjoint paths for each observation interval. We then contract these paths with every library direction.

The coefficient gradient is sufficient when the input state and interval length are

fixed. When the exponential action is part of a larger model, either quantity may also depend on unknown parameters. We then need derivatives with respect to $\mathbf{v}$ and h . For $\mathbf{y} = e^{h\mathbf{A}(\boldsymbol{\theta})}\mathbf{v}$, a perturbation of the input state gives

$$(3.14) \quad \begin{aligned} d\mathbf{y} &= e^{h\mathbf{A}}d\mathbf{v}, \\ d\ell &= \text{Re}((\nabla_{\mathbf{y}}\ell)^* e^{h\mathbf{A}}d\mathbf{v}) = \text{Re}((e^{h\mathbf{A}*}\nabla_{\mathbf{y}}\ell)^*d\mathbf{v}). \end{aligned}$$

It follows that

$$(3.15) \quad \nabla_{\mathbf{v}}\ell = e^{h\mathbf{A}*}\nabla_{\mathbf{y}}\ell.$$

Differentiating with respect to the interval length gives

$$(3.16) \quad \begin{aligned} \frac{\partial \mathbf{y}}{\partial h} &= \mathbf{A}e^{h\mathbf{A}}\mathbf{v} = \mathbf{A}\mathbf{y}, \\ \frac{\partial \ell}{\partial h} &= \text{Re}\left((\nabla_{\mathbf{y}}\ell)^*\frac{\partial \mathbf{y}}{\partial h}\right) = \text{Re}((\nabla_{\mathbf{y}}\ell)^*\mathbf{A}\mathbf{y}). \end{aligned}$$

The adjoint path at $s = 0$ gives the input-state derivative in (3.15). The interval derivative requires one application of $\mathbf{A}$ to the computed output $\mathbf{y}$. Thus, both derivatives use quantities already computed by Algorithm 3.1.

The input-state derivative is needed when an exponential action follows a nonlinear calculation. For example, consider the semilinear equation

$$\dot{\mathbf{u}} = \mathbf{A}(\boldsymbol{\theta})\mathbf{u} + \mathbf{N}(\mathbf{u}),$$

and the first-order integrating-factor step

$$(3.17) \quad \mathbf{u}_{k+1} = e^{h\mathbf{A}(\boldsymbol{\theta})}(\mathbf{u}_k + h\mathbf{N}(\mathbf{u}_k)).$$

Here the input to the exponential is $\mathbf{v}_k = \mathbf{u}_k + h\mathbf{N}(\mathbf{u}_k)$. Equation (3.15) gives the derivative of the loss with respect to $\mathbf{v}_k$. We then apply the chain rule to obtain derivatives with respect to $\mathbf{u}_k$ and any unknown coefficients in $\mathbf{N}$. The first-order formula also introduces a time-discretization error in addition to the Krylov and quadrature errors. Section 6.3 measures the effect of the input-state derivative in the Kuramoto–Sivashinsky experiment.

3.3. Cost and storage. The work has three parts. We construct the forward and adjoint Krylov spaces, compute the two paths at the quadrature nodes, and contract the paths with the library directions. Suppose that Algorithm 3.1 is applied N_{act} times with a common Krylov depth m_K and n_q quadrature nodes.

PROPOSITION 3.2 (Arithmetic cost). *Assume full Arnoldi orthogonalization, dense arithmetic for the projected exponentials, and direct sparse computation of every library contraction. Then the arithmetic cost is*

$$\begin{aligned} &O(N_{\text{act}}(m_K \text{nnz}(\mathbf{A}) + nm_K^2)) \\ &+ O(N_{\text{act}}n_q(m_K^3 + nm_K)) + O\left(N_{\text{act}}n_q \sum_{j=1}^{M_{\text{lib}}} (\text{nnz}(\mathbf{A}_j) + n)\right). \end{aligned}$$

Proof. Each action constructs two m_K -step Arnoldi spaces. The sparse products with $\mathbf{A}$ cost $O(m_K \text{nnz}(\mathbf{A}))$, while full orthogonalization costs $O(nm_K^2)$. At each quadrature node, computing the two small exponentials and multiplying by the Arnoldi

bases costs $O(m_K^3 + nm_K)$. The contraction with direction j costs $O(\text{nnz}(\mathbf{A}_j) + n)$. Applying these costs to all actions, nodes, and library directions gives the stated expression. $\square$

The first term is the cost of the two Arnoldi spaces, the second computes the paths at the quadrature nodes, and the third contracts those paths with the library. Once $\mathbf{A}$ has been assembled, only the third term depends explicitly on the number of library directions. A larger library may also change the sparsity of $\mathbf{A}$. In that case, the first term grows through $\text{nnz}(\mathbf{A})$, and assembling $\mathbf{A}$ has an additional library-dependent cost. For CME directions, $\text{nnz}(\mathbf{A}_j) = O(n)$, so the contraction cost becomes $O(N_{\text{act}} n_q M_{\text{lib}} n)$. The supplementary material gives the full derivation.

Full orthogonalization can be noticeable when multiplication by $\mathbf{A}$ is inexpensive, even when $m_K \ll n$. The second term also assumes that a dense projected exponential is computed separately at every node. A Schur decomposition of each projected matrix replaces these computations by one $O(m_K^3)$ factorization and lower-order work at each node.

A direct implementation retains both Arnoldi bases and uses $O(n(m_f + m_a) + m_f^2 + m_a^2)$ storage. Quadrature admits a smaller schedule. Once the forward basis has been constructed, we evaluate and retain the forward path at the n_q quadrature nodes. We then overwrite the forward basis with the adjoint basis and evaluate the adjoint path one node at a time. The resulting storage is $O(n(\max(m_f, m_a) + n_q) + m_f^2 + m_a^2)$. In contrast, the projected-core contraction couples the two bases and must retain both when each basis is constructed only once. It can reach the one-workspace memory scale by rebuilding one basis after forming the reduced core. The supplement compares these three schedules on a matrix-free Lindblad problem. These estimates do not include the operator library.

4. Gradient Error Analysis. The gradient approximation has three sources of error. Two come from the forward and adjoint Krylov spaces, and the third comes from quadrature. We first separate these errors and then bound the two path errors using the Arnoldi residuals. A small residual need not imply a small error when $\mathbf{A}$ is nonnormal because the remaining evolution may amplify the defect. The bounds include this amplification and the size of each library direction. They are intended to detect an unresolved path, not to select the smallest Krylov depth on every stiff problem.

Let

$$\mathbf{f}(s) = e^{sh\mathbf{A}}\mathbf{v}, \quad \boldsymbol{\lambda}(s) = e^{(1-s)h\mathbf{A}^*}\nabla_{\mathbf{y}}\ell, \quad 0 \leq s \leq 1,$$

and let $\tilde{\mathbf{f}}$ and $\tilde{\boldsymbol{\lambda}}$ denote their Krylov approximations. The approximate paths have the exact boundary values at $s = 0$ and $s = 1$, respectively. For any path $\mathbf{z}$, we write $\|\mathbf{z}\|_\infty = \max_{0 \leq s \leq 1} \|\mathbf{z}(s)\|_2$. The quadrature nodes and nonnegative weights are denoted by s_q, w_q , with $\sum_q w_q = 1$.

Before deriving the bound, we record why the two path errors must be treated separately. Consider a rank-one direction $\mathbf{E} = \mathbf{a}\mathbf{b}^*$ with $\mathbf{a} = \alpha\mathbf{v}_1$ and $\mathbf{b} = \beta\mathbf{w}_1$. Let the two Arnoldi relations be

$$(4.1) \quad \begin{aligned} \mathbf{A}\mathbf{V}_m &= \mathbf{V}_m\mathbf{P}_m + p_{m+1,m}\mathbf{v}_{m+1}\boldsymbol{\delta}_m^*, \\ \mathbf{A}^*\mathbf{W}_m &= \mathbf{W}_m\mathbf{Q}_m + q_{m+1,m}\mathbf{w}_{m+1}\boldsymbol{\delta}_m^*. \end{aligned}$$

Here $p_{m+1,m}$ and $q_{m+1,m}$ are denoted $g_{m+1,m}$ and $h_{m+1,m}$, respectively, in [16]. The

depth- m projected Fréchet approximation of [16, Algorithm 2] is

$$(4.2) \quad L_m = V_m X_m W_m^*, \quad X_m = \alpha\beta \int_0^1 e^{(1-s)P_m} \delta_1 \delta_1^* e^{sQ_m^*} ds.$$

The residual-product estimate in [16, Eq. (59)] has the form

$$(4.3) \quad \eta_m^\times = |p_{m+1,m} q_{m+1,m} \xi_m|,$$

where ξ_m is an entry of a reduced block matrix function. This estimate is used as a heuristic in that work.

PROPOSITION 4.1 (One-sided termination). *Suppose that $\mathbf{A}^* \mathbf{b} = \bar{\lambda} \mathbf{b}$ and let V_m be the Arnoldi basis constructed from $(\mathbf{A}, \mathbf{a})$, with $P_m = V_m^* \mathbf{A} V_m$. Define*

$$\psi_\lambda(z) = \begin{cases} (e^z - e^\lambda)/(z - \lambda), & z \neq \lambda, \\ e^\lambda, & z = \lambda. \end{cases}$$

Then the Arnoldi process for $(\mathbf{A}^, \mathbf{b})$ terminates after one step and the residual-product estimate in (4.3) is zero. The exact derivative and the depth- $(m, 1)$ projected approximation satisfy*

$$(4.4) \quad \begin{aligned} L_{\text{exp}}(\mathbf{A}, \mathbf{a} \mathbf{b}^*) &= \psi_\lambda(\mathbf{A}) \mathbf{a} \mathbf{b}^*, \\ L_{m,1} &= V_m \psi_\lambda(P_m) V_m^* \mathbf{a} \mathbf{b}^*. \end{aligned}$$

Consequently,

$$(4.5) \quad \|L_{\text{exp}}(\mathbf{A}, \mathbf{a} \mathbf{b}^*) - L_{m,1}\|_F = \|\mathbf{b}\|_2 \|\psi_\lambda(\mathbf{A}) \mathbf{a} - V_m \psi_\lambda(P_m) V_m^* \mathbf{a}\|_2.$$

Thus one-sided termination makes the residual-product estimate zero while leaving the ordinary Krylov approximation error on the other side unchanged. The same conclusion holds with the two Arnoldi processes interchanged.

Proof. The eigenvector relation gives $\mathbf{b}^* e^{s\mathbf{A}} = e^{s\lambda} \mathbf{b}^*$. Substitution into the integral definition of the Fréchet derivative gives the first identity in (4.4). Applying the same calculation to the projected matrix P_m gives the second identity. Equation (4.5) then follows from the Frobenius norm of a rank-one matrix. Since the Arnoldi residual coefficient on the $\mathbf{b}$ side is zero, so is $\eta_m^\times$. $\square$

THEOREM 4.2 (Gradient error decomposition). *For each library direction, let*

$$\tilde{g}_j = h \operatorname{Re} \sum_q w_q \tilde{\lambda}(s_q)^* \mathbf{A}_j \tilde{\mathbf{f}}(s_q).$$

Then

$$(4.6) \quad \begin{aligned} \left| \frac{\partial \ell}{\partial \theta_j} - \tilde{g}_j \right| &\leq |h| \left| \operatorname{Re} \left(\int_0^1 \lambda(s)^* \mathbf{A}_j \mathbf{f}(s) ds - \sum_q w_q \lambda(s_q)^* \mathbf{A}_j \mathbf{f}(s_q) \right) \right| \\ &\quad + |h| \|\mathbf{A}_j\|_2 \left(\|\mathbf{f}\|_\infty \|\lambda - \tilde{\lambda}\|_\infty + \|\lambda\|_\infty \|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \right. \\ &\quad \left. + \|\lambda - \tilde{\lambda}\|_\infty \|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \right). \end{aligned}$$

The first term is the quadrature error for the exact paths. The remaining terms contain the forward path error, the adjoint path error, and their product. Each is scaled by the norm of the library direction.

Proof. Let $e_f = \mathbf{f} - \tilde{\mathbf{f}}$ and $e_\lambda = \boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}$. After adding and subtracting the quadrature sum for the exact paths, the difference at each node is

$$\boldsymbol{\lambda}^* \mathbf{A}_j \mathbf{f} - \tilde{\boldsymbol{\lambda}}^* \mathbf{A}_j \tilde{\mathbf{f}} = e_\lambda^* \mathbf{A}_j \mathbf{f} + \boldsymbol{\lambda}^* \mathbf{A}_j e_f - e_\lambda^* \mathbf{A}_j e_f.$$

Applying $|\mathbf{x}^* \mathbf{A}_j \mathbf{y}| \leq \|\mathbf{x}\|_2 \|\mathbf{A}_j\|_2 \|\mathbf{y}\|_2$ to these three terms gives a bound that is independent of the node. Summing with nonnegative weights of unit total mass proves (4.6). $\square$

To bound the two path errors in Theorem 4.2, define the Arnoldi residuals

$$(4.7) \quad \mathbf{r}_f(s) = \tilde{\mathbf{f}}'(s) - h \mathbf{A} \tilde{\mathbf{f}}(s), \quad \mathbf{r}_\lambda(s) = \tilde{\boldsymbol{\lambda}}'(s) + h \mathbf{A}^* \tilde{\boldsymbol{\lambda}}(s).$$

Subtracting the exact and approximate evolution equations and applying variation of constants gives

$$\begin{aligned} \mathbf{f}(s) - \tilde{\mathbf{f}}(s) &= - \int_0^s e^{(s-u)h\mathbf{A}} \mathbf{r}_f(u) du, \\ \boldsymbol{\lambda}(s) - \tilde{\boldsymbol{\lambda}}(s) &= \int_s^1 e^{(u-s)h\mathbf{A}^*} \mathbf{r}_\lambda(u) du. \end{aligned}$$

The finite-interval amplification factor is

$$\kappa_A = \max_{0 \leq s \leq 1} \int_0^s \|e^{(s-u)h\mathbf{A}}\|_2 du.$$

The adjoint has the same factor in the 2-norm after a change of variables. Consequently,

$$\|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \leq \kappa_A \|\mathbf{r}_f\|_\infty, \quad \|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty \leq \kappa_A \|\mathbf{r}_\lambda\|_\infty.$$

The exact path norms in Theorem 4.2 satisfy

$$\|\mathbf{f}\|_\infty \leq \|\tilde{\mathbf{f}}\|_\infty + \kappa_A \|\mathbf{r}_f\|_\infty, \quad \|\boldsymbol{\lambda}\|_\infty \leq \|\tilde{\boldsymbol{\lambda}}\|_\infty + \kappa_A \|\mathbf{r}_\lambda\|_\infty.$$

Substitution into (4.6) bounds the two path errors using the forward and adjoint residuals, κ_A , the computed path norms, and $\|\mathbf{A}_j\|_2$. The quadrature error remains separate.

For the m_f -step Arnoldi path $\tilde{\mathbf{f}}(s) = \beta_f V_{m_f} e^{shH_{m_f}^f} \boldsymbol{\delta}_1$, we use the Arnoldi relation to obtain the computable residual

$$(4.8) \quad \|\mathbf{r}_f(s)\|_2 = |h| \beta_f |h_{m_f+1, m_f}^f| \|\boldsymbol{\delta}_{m_f}^* e^{shH_{m_f}^f} \boldsymbol{\delta}_1\|,$$

The adjoint residual has the analogous form. This is the ODE residual used in residual-based error control for Krylov exponential methods [5]. The two Arnoldi factorizations therefore compute the residual quantities needed in (4.6). The remaining factor measures the growth of $e^{t\mathbf{A}}$ over the interval.

COROLLARY 4.3 (Contractive CTMC bound). *Consider a CTMC generator or sub-generator $\mathbf{A}$ in the column convention, with nonnegative off-diagonal entries and nonpositive column sums, and let $h \geq 0$. Let*

$$\delta_f(s) = \int_0^s \|\mathbf{r}_f(u)\|_1 du, \quad \delta_a(s) = \int_s^1 \|\mathbf{r}_\lambda(u)\|_\infty du.$$

For each library direction, let

$$g_j^{\text{proj}} = h \operatorname{Re} \int_0^1 \tilde{\lambda}(s)^* \mathbf{A}_j \tilde{\mathbf{f}}(s) ds.$$

Then

$$(4.9) \quad \left| \frac{\partial \ell}{\partial \theta_j} - g_j^{\text{proj}} \right| \leq h \int_0^1 \left(\delta_a(s) \|\mathbf{A}_j \tilde{\mathbf{f}}(s)\|_1 + \delta_f(s) \|\mathbf{A}_j^T \tilde{\lambda}(s)\|_\infty + \|\mathbf{A}_j\|_1 \delta_f(s) \delta_a(s) \right) ds.$$

Proof. A CTMC or sub-Markov semigroup is contractive in the 1-norm, and its adjoint is contractive in the ∞ -norm. Variation of constants therefore gives $\|\mathbf{f}(s) - \tilde{\mathbf{f}}(s)\|_1 \leq \delta_f(s)$ and $\|\lambda(s) - \tilde{\lambda}(s)\|_\infty \leq \delta_a(s)$. Adding and subtracting the bilinear form of the two projected paths leaves three terms bounded by

$$\delta_a \|\mathbf{A}_j \tilde{\mathbf{f}}\|_1, \quad \delta_f \|\mathbf{A}_j^T \tilde{\lambda}\|_\infty, \quad \|\mathbf{A}_j\|_1 \delta_f \delta_a.$$

Integration proves (4.9). $\square$

The Arnoldi factorizations also give the residual norms used in δ_f and δ_a . Denote the forward projected matrix and residual coefficient by $H_{m_f}^f$ and h_{m_f+1, m_f}^f , and denote the adjoint quantities by $H_{m_a}^a$ and h_{m_a+1, m_a}^a . If the starting-vector norms are β_f and β_a , then

$$\begin{aligned} \|\mathbf{r}_f(u)\|_1 &= h\beta_f |h_{m_f+1, m_f}^f| \|\mathbf{v}_{m_f+1}\|_1 |\delta_{m_f}^* e^{uhH_{m_f}^f} \delta_1|, \\ \|\mathbf{r}_\lambda(u)\|_\infty &= h\beta_a |h_{m_a+1, m_a}^a| \|\mathbf{w}_{m_a+1}\|_\infty |\delta_{m_a}^* e^{(1-u)hH_{m_a}^a} \delta_1|. \end{aligned}$$

If quadrature is used for the projected integral, its remainder must be added to the right-hand side of (4.9). In the numerical experiments, we compare two quadrature orders instead. This comparison checks quadrature convergence but is not an error bound.

The same argument applies to a Lindblad semigroup when the dissipation rates are nonnegative at the evaluation point. The semigroup is contractive in trace norm, while its adjoint is unital and contractive in operator norm. We reshape each Arnoldi residual vector as a matrix before computing these norms [31].

The three error sources may be refined independently. Increasing the quadrature order reduces the first term in (4.6). Increasing the forward or adjoint depth reduces the corresponding residual. The factors κ_A and $\|\mathbf{A}_j\|_2$ account for amplification and for the scale of the library direction. Since κ_A can be difficult to estimate for a large nonnormal operator, Section 5 also compares successive Krylov depths and quadrature orders.

5. Accuracy and Computational Benchmarks. We first determine the forward depth, adjoint depth, and quadrature order needed for an accurate gradient. We then compare runtimes using only configurations that meet the same tolerance. The two Krylov spaces and the quadrature rule are refined separately. Their errors may partly cancel, so the total error need not decrease whenever one parameter is increased.

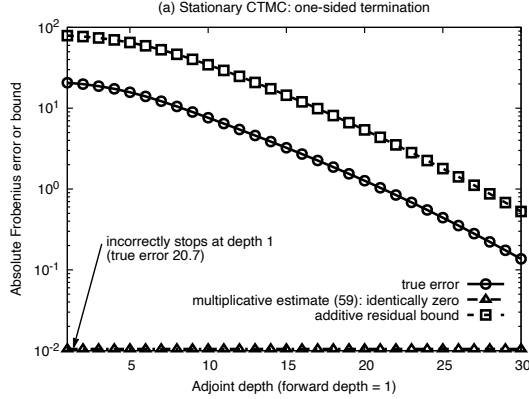


FIG. 2. *One-sided Arnoldi termination makes the residual-product estimate identically zero although the Fréchet error is nonzero. The additive residual bound remains valid. The supplement gives the stiff-CME test and the full experimental details.*

5.1. Reference and error measure. For problems small enough to admit a dense reference, we use the block identity [13, Thm. 3.6]

$$(5.1) \quad \exp \begin{pmatrix} \mathbf{A}\Delta t & \mathbf{E}\Delta t \\ \mathbf{0} & \mathbf{A}\Delta t \end{pmatrix} = \begin{pmatrix} e^{\mathbf{A}\Delta t} & L_{\exp}(\mathbf{A}\Delta t, \mathbf{E}\Delta t) \\ \mathbf{0} & e^{\mathbf{A}\Delta t} \end{pmatrix},$$

for each direction $\mathbf{E}$. We compute the block exponential without Krylov projection or quadrature and report the relative 2-norm error of the full parameter gradient. Finite differences do not resolve the smallest errors in these tests. With a step of 10^{-6} , their resolution is approximately 10^{-10} .

5.2. Structural failures of heuristic stopping estimates. Two common diagnostics are not error bounds. The residual-product estimate vanishes if either Arnoldi process terminates, even when the other path is unresolved. Figure 2 shows a 200-state reflecting-walk example in which this estimate is zero while the true Fréchet error ranges from 20.66 to 0.137. A difference between successive projected approximations can also be small during slow convergence. For the stiff 1681-state Brusselator example, this difference is 5.40×10^{-9} at depth 16, while the normalized library-gradient error is 6.40×10^{-7} . The residual bound in Corollary 4.3 remains valid in both cases, although it can be conservative. The supplement gives both constructions and the complete plot. We use the heuristic quantities only as diagnostics and check the forward, adjoint, and quadrature errors separately.

5.3. Krylov and quadrature errors. We vary m_K and n_q separately for a stiff nonnormal Fokker–Planck generator with $n = 80$ and $M_{\text{lib}} = 10$. The supplement gives the full error table. At $n_q = 8$, the error is nearly independent of m_K . It changes from 9.1×10^{-6} at $m_K = 20$ to 9.7×10^{-6} at $m_K = 60$. The quadrature remainder therefore dominates. In contrast, at $n_q = 24$, increasing m_K from 20 to 60 reduces the error from 1.0×10^{-5} to 1.8×10^{-15} . The slight increase between $n_q = 8$ and $n_q = 24$ at $m_K = 20$ reflects the nonmonotonicity of the combined projection and quadrature errors. The error continues to decrease when both parameters are refined. Since the integrand is analytic in s , Gauss–Legendre quadrature converges rapidly

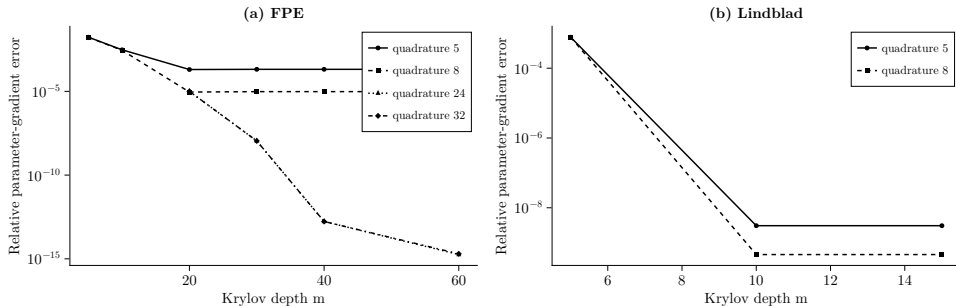


FIG. 3. Measured Krylov-depth sweeps for Fokker–Planck ($h = 0.1$) and Lindblad ($h = 1$) operators, against dense block–Fréchet references. Line styles and markers identify quadrature orders. Refining the basis alone cannot remove a quadrature floor. We report the higher-order Fokker–Planck values in the supplement.

until the Krylov error or roundoff dominates. Increasing n_q computes additional small exponential actions but requires no new Arnoldi steps.

For the complex Lindblad generator, the gradient error reaches 10^{-16} at $n_q = 12$ for every tested step size. This experiment also verifies the use of the conjugate transpose for complex operators.

5.4. Choosing m_K and n_q in practice. Given an action tolerance ε_{act} , we increase m_f until both $\rho_{m_f}(h) \leq \varepsilon_{\text{act}} \|\mathbf{v}\|_2$ and the forward action agrees with the next tested depth. Holding that forward action fixed, we increase m_a until the corresponding checks pass for the loss gradient at the right endpoint. We then double n_q until the coefficient gradient changes by no more than the target gradient tolerance. The comparison uses the largest relative change over all reported directions, not only the largest gradient component.

We vary one parameter at a time to identify the limiting error. If the error is unchanged when m_f and m_a increase, quadrature is limiting the accuracy. If it is unchanged when n_q increases, one of the Krylov spaces is limiting the accuracy. Near a stationary input, the forward action may be accurate while the coefficient gradient is still unresolved. We must therefore check the adjoint path and the coefficient gradient as well. These comparisons are refinement diagnostics rather than error bounds. For a general nonnormal operator, the bound in Section 4 accounts for amplification by $e^{t\mathbf{A}}$, which an Arnoldi residual alone does not measure.

The required quadrature order depends on the problem. Eight nodes are insufficient for the stiff Fokker–Planck example, while the Lindblad sweep reaches 10^{-16} with 12 nodes. We therefore use a short refinement sweep rather than a universal value of n_q .

Accuracy after a coefficient map. If another model computes the coefficients $\boldsymbol{\theta}$, the chain rule multiplies the coefficient gradient by the Jacobian of that model. This Jacobian may also amplify an error in the gradient. A tolerance used for optimization should therefore be checked after the chain rule. The supplement examines sensitivity-scaled coordinates and the neural rate map.

5.5. Library reuse at matched accuracy. To isolate the effect of library size, we hold a 320-state birth–death operator fixed and increase the number of distinct

TABLE 1

Comparison with Matfree at fixed accuracy on an NVIDIA RTX 4070 SUPER. Entries are median times in milliseconds. Parentheses contain the achieved maximum relative error over the parameter and input-state gradients for the fastest tested configuration that meets the column tolerance. There are 31 evaluations after compilation and warmup. Both methods are JIT-compiled in the same process. Bold marks the faster method.

n	tol 10^{-6}		tol 10^{-8}		tol 10^{-10}	
	ours	Matfree	ours	Matfree	ours	Matfree
80	4.24 (3.0e−9)	2.53 (2.5e−8)	4.24 (3.0e−9)	3.06 (1.3e−12)	5.43 (2.1e−13)	3.06 (1.3e−12)
320	3.30 (8.0e−7)	3.00 (5.4e−8)	4.57 (3.6e−10)	3.65 (3.4e−12)	5.76 (2.6e−11)	3.65 (3.4e−12)
1,000	3.81 (3.2e−7)	4.36 (7.4e−8)	5.86 (1.0e−10)	5.34 (4.6e−12)	5.94 (1.0e−10)	5.34 (4.6e−12)
3,000	5.69 (5.6e−7)	6.37 (4.9e−7)	8.51 (1.7e−10)	8.36 (3.0e−11)	11.09 (2.7e−11)	8.36 (3.0e−11)
10,000	4.21 (8.8e−7)	9.99 (1.8e−10)	8.30 (2.6e−10)	9.99 (1.8e−10)	10.42 (4.3e−11)	11.85 (3.1e−12)

two-entry directions from 16 to 512. For each library, we select the fastest tested configuration whose forward, parameter-gradient, and input-state-gradient errors all satisfy the target tolerance. At tolerance 10^{-8} , separate directional block exponentials require between 4.37 and 131 ms as the library grows. Reusing the two Krylov spaces requires between 0.884 and 3.48 ms, giving a $38\times$ speedup at 512 directions. Every selected configuration has an error below 8.3×10^{-14} . The supplement gives the complete sweep and the fused-contraction results.

5.6. Comparison with differentiated Arnoldi at matched accuracy. We compare our calculation with reverse differentiation of a matrix-free exponential routine in Matfree [20]. Matfree differentiates the finite Arnoldi calculation. Our method first writes the derivative of the exact finite-time solution and then approximates the forward and adjoint paths. Both calculations converge to the same gradient as their Krylov spaces are refined. For each state dimension, we report the fastest configuration that reaches the prescribed accuracy.

Table 1 reports median wall-clock times on an NVIDIA RTX 4070 SUPER. Both methods are compiled with JAX in the same process, which avoids a cross-language comparison. At tolerance 10^{-6} , Matfree is faster for $n \leq 320$, while our method is faster from $n = 10^3$. At tolerances 10^{-8} and 10^{-10} , the crossover occurs at $n = 10^4$. The relative advantage grows with the state dimension and decreases as the tolerance is tightened. In sparse-library problems, the state space and the number of candidate directions may both be large, while sampling noise often sets the required accuracy above machine precision.

The timing sweep uses Float64, $M_{\text{lib}} = 1000$, $h = 0.3$, and a local birth–death generator with $3n - 2$ nonzeros. Accuracy is the maximum of parameter- and input-state-gradient relative errors against an independently refined exponential reference. Matching parameter gradients alone would not reproduce this timing comparison. The supplement adds the complete timing curves, dense automatic differentiation, the two-sided projection, CPU results, and additional sparse-contraction comparisons.

The forward and adjoint paths may require different depths. A linear quantity $\mathbf{c}^* \mathbf{u}$ is invariant when $\mathbf{c}^* \mathbf{A} = 0$. If also $\mathbf{c}^* \mathbf{A}_j = 0$, its derivative with respect to coefficient j must vanish. The exact gradient formula preserves this property for CTMC mass and Lindblad trace. Differentiating an under-resolved Arnoldi calculation may give a nonzero derivative unless the invariant is projected explicitly. The two methods agree after refinement or explicit invariant projection.

The forward path, adjoint path, and quadrature require separate accuracy checks. We refine them in that order and test both the input-state and coefficient gradients.

The supplement gives under-resolved and nearly stationary examples in which checking only the forward value is insufficient.

6. Applications. We apply the method to three inverse problems. The Brusselator CME uses an overcomplete reaction library with rates that vary across experimental conditions. The Kuramoto–Sivashinsky model places a nonlinear calculation before the exponential and therefore requires the input-state derivative. The Lindblad model has a complex generator with real coefficients. The forward and adjoint Krylov spaces are constructed in the same way for all three problems. Only the coefficient map and the optimization procedure change.

6.1. Sparse optimization. We perform thresholding in the scaled coordinates $\phi = D\theta$. The j th diagonal entry of D is the norm of the initial sensitivity for coefficient j . The threshold therefore compares changes in the model output rather than coefficients that may have different physical units.

Each smooth optimization is restricted to the current support and uses Algorithm 3.1 for its gradient. After the optimization, we project negative coefficients to zero and remove coefficients below a fixed fraction of the largest scaled coefficient. When the support stops changing, we solve once more without thresholding. The supplement gives the complete pseudocode. In the Lindblad experiment, every candidate receives the same initial value and the algorithm is given no support information. The scaling and thresholding belong to the outer optimization, not to the gradient calculation.

6.2. Brusselator CME (ungrouped finite-time identification). For a reaction network with copy-number state $\mathbf{x} \in \mathbb{N}^d$, reaction m has stoichiometric change ζ_m . On a finite state space $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the probability vector obeys $\dot{\mathbf{p}} = \mathbf{A}\mathbf{p}$. A mass-action candidate consuming $\nu = (\nu_1, \dots, \nu_d)$ molecules uses

$$(6.1) \quad \phi_\nu(\mathbf{x}) = \prod_{i=1}^d \binom{x_i}{\nu_i}.$$

For each proposed stoichiometry ζ_m and feature ϕ_k , the fixed gain–loss direction is defined with $\chi_m(\mathbf{x}) = \mathbf{1}_{\{\mathbf{x} + \zeta_m \in \mathcal{X}\}}$ as

$$(6.2) \quad [\mathbf{A}_m^{(k)}]_{\mathbf{y}, \mathbf{x}} = \chi_m(\mathbf{x}) \phi_k(\mathbf{x}) (\mathbf{1}_{\{\mathbf{y} = \mathbf{x} + \zeta_m\}} - \mathbf{1}_{\{\mathbf{y} = \mathbf{x}\}}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}.$$

A reaction that would leave $\mathcal{X}$ is suppressed, while every retained gain has a matching diagonal loss. This boundary convention keeps probability in the truncated state space rather than allowing it to leak through the boundary. The complete generator is

$$(6.3) \quad \mathbf{A}(\theta) = \sum_{m=1}^M \sum_{k=1}^K \theta_{m,k} \mathbf{A}_m^{(k)}, \quad \theta_{m,k} \geq 0.$$

Each candidate mechanism defines one sparse matrix, and the support of θ determines the inferred reaction network. The conservative truncation gives $\mathbf{1}^T \mathbf{A}_m^{(k)} = 0$ for every direction.

We consider the Brusselator on a 13×13 count grid, so $n = 169$. The candidate library contains all 40 pairings of four stoichiometric channels with bivariate mass-action monomials of total degree at most three. The active mechanisms are $\emptyset \rightarrow X$, $X \rightarrow \emptyset$, $X \rightarrow Y$, and $2X + Y \rightarrow 3X$ (library directions 1, 13, 23, and 39). We

do not use channel grouping or assume a support size. We fit four experimental conditions, each with six initial distributions and exact noiseless snapshots separated by the nonuniform gaps 0.05, 0.20, and 0.50. Four additional conditions and a time-1.5 prediction are reserved for testing.

The nonnegative linear rate model is $\theta_j(A, B) = W_{j,:}[1, A, B]^T$. Adam first minimizes the finite-time snapshot loss with an ℓ_1 penalty over all 40 rows. A fixed coefficient threshold then selects the support. The selected model is refitted with unpenalized Adam followed by a limited-memory quasi-Newton (L-BFGS) solve. The threshold is calibrated on seed 11 and then held fixed for the remaining four seeds. All five runs recover exactly the four active directions, with no false positives. The mean coefficient, test-transition, and long-horizon relative errors are 4.27×10^{-6} , 4.54×10^{-6} , and 3.99×10^{-6} , respectively. Figure 4(a) reports the small seed-to-seed variation. The fit uses only snapshot distributions and no reaction event counts.

The two weak baselines replace the finite-time transition by an integrated form of the differential equation. For a smooth test function ψ that vanishes at the ends of its interval, weighted weak-form SINDy uses integration by parts to fit

$$(6.4) \quad - \int \psi'(t) \mathbf{p}(t) dt = \sum_j \theta_j \int \psi(t) \mathbf{A}_j \mathbf{p}(t) dt.$$

We evaluate these integrals from a piecewise-linear interpolant of the snapshots. The interval rule instead applies the trapezoidal approximation on every observed interval:

$$(6.5) \quad \mathbf{p}(t_{i+1}) - \mathbf{p}(t_i) \approx \Delta t_i \sum_j \theta_j \mathbf{A}_j \frac{\mathbf{p}(t_i) + \mathbf{p}(t_{i+1})}{2}.$$

Figure 4(b) compares the finite-time Krylov fit with these two baselines [24]. All three methods use the same noiseless distributions on the 13×13 state grid. They also use the same training and test conditions and the same 40-direction library. At the four nonuniform snapshots, the finite-time fit recovers exactly 4/4 directions in every run, with mean test coefficient error 4.27×10^{-6} . Weak SINDy recovers all four directions but adds two false positives, with error 0.855. The interval rule selects one false positive and gives error 0.548. For the weak fits, hyperparameters are selected by comparison with the known coefficients. We repeat the comparison with 6, 16, and 76 snapshots. The finite-time fit recovers the exact support throughout this sweep after sampling-specific sparsity tuning. Its coefficient errors at 4, 6, 16, and 76 snapshots are 1.78×10^{-6} , 2.33×10^{-6} , 3.72×10^{-5} and 7.79×10^{-5} . Weak SINDy becomes slightly more accurate at 16 snapshots and is about four times more accurate at 76. The finite-time fit is substantially more accurate at four and six snapshots, where it avoids temporal quadrature across the widest observation gaps. The supplement includes an all-edge 870-direction CTMC, nearly collinear candidates, and omitted-mechanism experiments.

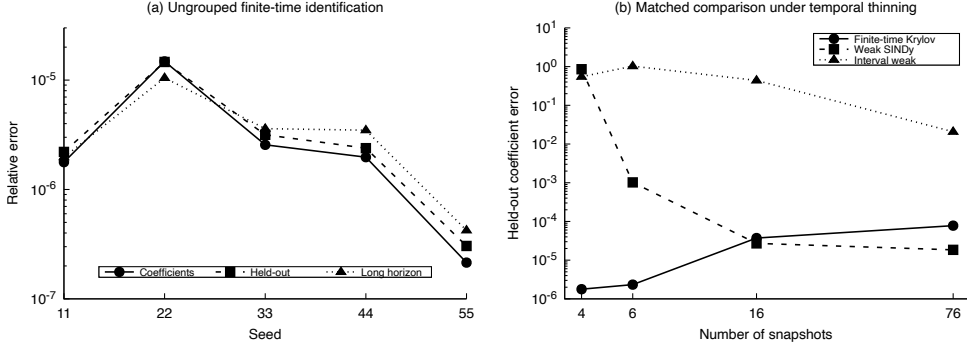


FIG. 4. *Ungrouped snapshot-only Brusselator identification. (a) Coefficient, test-transition, and time-1.5 prediction errors for five finite-time fits. Every run selects exactly 4/40 directions. (b) Matched test coefficient error for the finite-time Krylov fit and two weak-form baselines on the same distributions and library. Four denotes the nonuniform snapshots used in panel (a). The remaining points refine the observation grid. The finite-time curve uses seed 11, 500 discovery epochs, 300 refit epochs and at most three randomly sampled transitions per update, followed by 100 L-BFGS iterations on the final support. The weak-form hyperparameters are selected by comparison with the known coefficients. The finite-time threshold is fixed at 0.2. Its ℓ_1 penalty is 10^{-4} except at 76 snapshots, where we reduce it to 10^{-6} to account for the shorter intervals.*

6.3. Kuramoto–Sivashinsky PDE. The KS example tests the input-state derivative.

$$(6.6) \quad u_t = -au_{xx} - bu_{xxx} - \frac{c}{2}\partial_x(u^2),$$

on a periodic domain of length 22. On 64 grid points, centered difference matrices give

$$(6.7) \quad A(a, b) = -aD_{xx} - bD_{xxx}, \quad N_c(u) = -\frac{c}{2}D_x(u \odot u), \quad D_{xxx} = D_{xx}^2.$$

Here $\odot$ denotes componentwise multiplication. We first take an explicit Euler step for the nonlinear term and then propagate the result with the exponential of the linear operator:

$$(6.8) \quad \mathbf{v}_k = \mathbf{u}_k - \frac{hc}{2}D_x(\mathbf{u}_k \odot \mathbf{u}_k), \quad \mathbf{u}_{k+1} = e^{hA(a, b)}\mathbf{v}_k.$$

The affine directions $-D_{xx}$ and $-D_{xxx}$ give the derivatives with respect to a and b through Proposition 3.1. The coefficient c enters through $\mathbf{v}_k$. We first compute the derivative of the loss with respect to $\mathbf{v}_k$ using (3.15), and then differentiate the nonlinear term. The supplement gives the complete chain rule.

We take $h = 0.0125$ and use Krylov depth 40 with eight quadrature nodes. A fourth-order Runge–Kutta (RK4) solution with 20 substeps serves as the reference. We fit ten transitions from each of three fields and test a 20-step prediction on a fourth field. Across five initializations, the mean coefficient and test-path errors are 1.84% and 0.0908%. If the input-state derivative $\nabla_{\mathbf{v}_k}\ell$ is omitted, the initial gradient of c is zero and the two errors increase to 18.6% and 1.04%.

We then enlarge the model to 15 directions,

$$(6.9) \quad \{-D_x, -D_{xx}, D_x D_{xx}, -D_{xxx}\} \cup \left\{ -\frac{1}{q}D_x(\mathbf{u}^{\odot q}) : q = 2, \dots, 12 \right\}.$$

Only $-D_{xx}$, $-D_{xxx}$, and the quadratic flux are active. We add Gaussian snapshot noise at 0, 0.25%, 0.5%, and 1% root-mean-square (RMS) amplitude. Prediction accuracy is the relative L^2 error at $T = 0.5$ on a fourth, noise-free path that is not used for fitting. Seed 11 at 0.5% fixes the 10^{-7} penalty and 0.08 threshold for all 20 initializations and all four levels. We recover the exact support in all 80 fits, with mean active-inactive separation between 0.910 and 0.957.

Noise affects support recovery and coefficient estimation differently. At 0.5%, refitting on the selected support reduces the mean full-vector coefficient error from 4.22% to 3.21% and the noise-free test-path error from 0.407% to 0.255%. At 1%, it increases these errors from 5.98% to 8.79% and from 0.508% to 0.713%, respectively. The support therefore remains stable even after unpenalized refitting ceases to improve the coefficient estimates.

To measure scaling, we use a length-704 domain with 2048 points and Fourier-modulated second- and fourth-derivative libraries of size 16, 64, and 256. Reusing the two Krylov spaces takes 4.07, 7.99, and 24.1 ms. Rebuilding both spaces for every direction takes 46.0, 188, and 757 ms. The speedup ranges from 11.3 to 31.5, and the two calculations agree to machine precision. The timings are minima of five warmed runs at depth 40 with eight quadrature nodes.

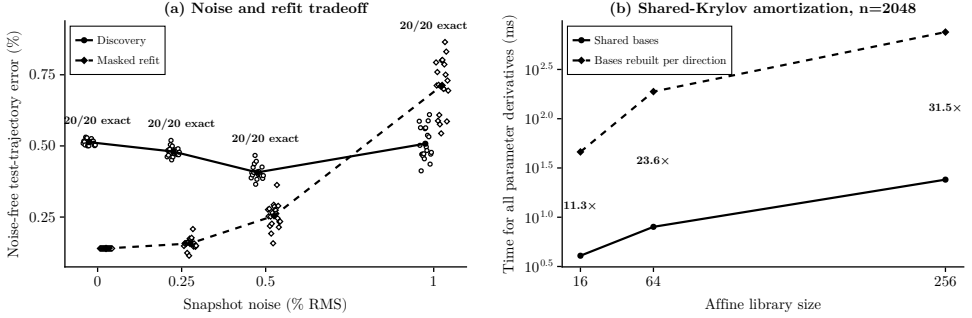


FIG. 5. *Kuramoto-Sivashinsky robustness and computational scaling.* (a) Noise-free test-path errors over 20 initializations at each snapshot-noise level. We recover the exact support in every initial sparse fit. The solid and dashed curves show the means before and after refitting on the selected support. (b) Time required to compute all parameter derivatives on the 2048-state problem when the two Krylov spaces are reused or rebuilt for every affine direction. We annotate the curves with the measured speedups.

6.4. Lindblad equation. For an open quantum system, let $H(\mathbf{J}) = H_0 + \sum_j J_j H_j$ and let L_k denote candidate jump operators. The density matrix satisfies

$$(6.10) \quad \dot{\rho} = -i[H(\mathbf{J}), \rho] + \sum_k \gamma_k (L_k \rho L_k^* - \frac{1}{2} \{L_k^* L_k, \rho\}).$$

Here $[X, Y] = XY - YX$ is the commutator, $\{X, Y\} = XY + YX$ is the anticommutator, and X^* denotes the conjugate transpose. The commutator describes Hamiltonian evolution, while each term in the sum is a dissipative channel. Let $\rho_v = \text{vec}(\rho)$. Using $\text{vec}(AXB) = (B^T \otimes A) \text{vec}(X)$, we obtain

$$(6.11) \quad \dot{\rho}_v = \left(\mathcal{L}_0 + \sum_j J_j \mathcal{H}_j + \sum_k \gamma_k \mathcal{D}_k \right) \rho_v, \quad \mathcal{H}_j = -i (I \otimes H_j - H_j^T \otimes I),$$

TABLE 2

Sparse Lindblad identification over five rate draws per size. Every run recovers the exact support, with no false positives or false negatives.

Qubits	Liouville dimension	Library	Active	Mean rate error
2	16	7	3	7.5×10^{-9}
3	64	12	5	5.6×10^{-9}
4	256	18	7	1.3×10^{-8}
5	1024	25	9	1.4×10^{-8}
6	4096	33	11	7.3×10^{-9}

where, with $K_k = L_k^* L_k$,

$$(6.12) \quad \mathcal{D}_k = \overline{L_k} \otimes L_k - \frac{1}{2} (I \otimes K_k + K_k^T \otimes I).$$

The Hamiltonian commutators and dissipators are fixed directions on $\mathbb{C}^{n^2}$ with real coefficients $(\mathbf{J}, \gamma)$. The generator is complex, and trace replaces probability mass as the conserved linear quantity. We therefore use the conjugate-transpose adjoint in (3.2). The resulting coefficient gradient is real and is used in the thresholded optimization.

For q qubits, the vectorized density state has dimension 4^q . We fix a transverse field as $\mathbf{A}_0$ and construct an overcomplete library. It contains $XX + YY$ exchange couplings on all $\binom{q}{2}$ site pairs, together with dephasing, decay, and bit-flip dissipators on every site. Thus, $M_{\text{lib}} = \binom{q}{2} + 3q$. Only nearest-neighbor exchange and σ_z dephasing are active in the true generator. We initialize every candidate at the same nonzero value and infer the $2q - 1$ active directions without using the sparsity pattern.

Table 2 reports five independent rate draws at each size. The supplement gives the individual runs, partial-observation fits, and finite-shot experiments.

The larger Lindblad experiments test the sparse contraction routine rather than support recovery. The supplement reports this scaling separately for seven through eleven qubits.

7. Discussion and Conclusions. We have developed a method for parameter inference in large, sparse linear ODEs whose generators depend affinely on unknown coefficients. For each exponential action, one forward Krylov space and one adjoint Krylov space are used for the entire coefficient library. Gauss–Legendre quadrature reduces the coefficient gradient to the contractions $\lambda^* \mathbf{A}_j \mathbf{f}$. The same computation gives derivatives with respect to the input state and interval length, which allows the exponential action to be used inside a larger model.

The error analysis separates quadrature error from errors in the forward and adjoint paths. Its additive form remains sensitive when only one path is unresolved, and the contractive bounds for CTMC and Lindblad generators use residual quantities computed by Arnoldi. The numerical examples show why this separation matters. A residual-product estimate can vanish after one-sided termination, while successive differences can become small during slow convergence. We therefore use the residual bounds as safeguards and check successive depths and quadrature orders separately.

The experiments use the same gradient calculation for a CME, a semilinear PDE, and a Lindblad equation. They recover sparse mechanisms from finite-time data, differentiate through a nonlinear step, and treat complex generators. At matched

accuracy, reusing the two Krylov spaces reduces the cost when the state space and coefficient library are large. Quadrature also permits a one-workspace implementation that replaces one full Krylov basis by the forward path at the quadrature nodes. In the 12-qubit Lindblad test, this schedule fits within 16 GB. A projected-core calculation reaches the same memory scale only after rebuilding one basis.

The present experiments are synthetic, and the main identification tests use full-state observations. The method removes repeated work across the library but does not remove growth in the state dimension, such as the 4^q density state for q qubits. Compressed state representations and adaptive Krylov solvers are natural directions for larger problems and longer time intervals.

Code availability. Source code and experiment scripts are available at <https://github.com/AdityaDendukuri/AdjointKrylovGradients>.

Supplementary Material

The supplement begins with the context-dependent rate models used in the applications. We then give additional accuracy and cost results, followed by the CME construction, inference details, and numerical checks. The notation is the same as in the main paper.

Appendix A. Neural Context-to-Rate Learning. This section extends the constant-rate model to coefficients that vary with experimental context while retaining a common support.

A.1. Context-dependent rates. Let ξ denote an experimental condition, such as an input concentration. We keep the candidate library fixed across conditions and write

$$(A.1) \quad \boldsymbol{\theta}(\xi) = \mathbf{s} \odot \mathbf{r}(\xi; \mathbf{w}), \quad \mathbf{A}(\xi) = \mathbf{A}_0 + \sum_{j=1}^{M_{\text{lib}}} \theta_j(\xi) \mathbf{A}_j.$$

where $\odot$ denotes componentwise multiplication. The trainable weights $\mathbf{w}$ determine the rate map, and $s_j \in \{0, 1\}$ is a support mask shared by all contexts. For a CME library, the parameterization or optimization constraints enforce $r_j(\xi; \mathbf{w}) \geq 0$. The rate map may contain fitted constants, a linear context model, or a neural network with nonnegative outputs. The mask selects the possible mechanisms, and the rate map determines how their rates change with context. A selected mechanism may still have zero rate under a particular condition.

For context ξ_c , let $\hat{\mathbf{p}}_{c,t}$ denote the observed full-state snapshot and $\Delta t_{c,t}$ the next observation interval. Holding the context fixed within each interval, we extend (2.3) as

$$(A.2) \quad \mathcal{L}_{\text{ctx}}(\mathbf{w}; \mathbf{s}) = \frac{1}{2} \sum_{c=1}^{N_c} \sum_{t=0}^{T_c-1} \left\| e^{\mathbf{A}(\xi_c) \Delta t_{c,t}} \hat{\mathbf{p}}_{c,t} - \hat{\mathbf{p}}_{c,t+1} \right\|_2^2.$$

Here N_c is the number of training contexts. Each additional initial condition contributes another snapshot sequence to the loss. Support selection uses only the training contexts and assumes that the possible mechanisms are common to all of them, although their rates may differ.

A.2. Gradients through a context-to-rate map. For a fixed support mask, the adjoint Krylov calculation computes the gradient with respect to the rates $\boldsymbol{\theta}(\xi_c)$. Applying the chain rule gives

$$(A.3) \quad \nabla_{\mathbf{w}} \mathcal{L}_{\text{ctx}} = \sum_{c=1}^{N_c} (D_{\mathbf{w}} \mathbf{r}(\xi_c; \mathbf{w}))^T (\mathbf{s} \odot \nabla_{\boldsymbol{\theta}(\xi_c)} \mathcal{L}_{\text{ctx}}).$$

For a neural rate map, reverse differentiation applies the network Jacobian to the coefficient gradient. A constant-rate model uses the coefficient gradient directly. Each observed transition has its own forward and adjoint Krylov spaces. We reuse these spaces across the library directions for that transition, but not across contexts.

A.3. Brusselator (grouped neural rates and an ungrouped comparison). For the neural Brusselator model, we use a 41×41 count grid ($n = 1681$), a 40-direction stoichiometry–propensity library, and the context $\xi = (A, B)$. The library is divided into four known stoichiometric channels, each with ten candidate propensity functions.

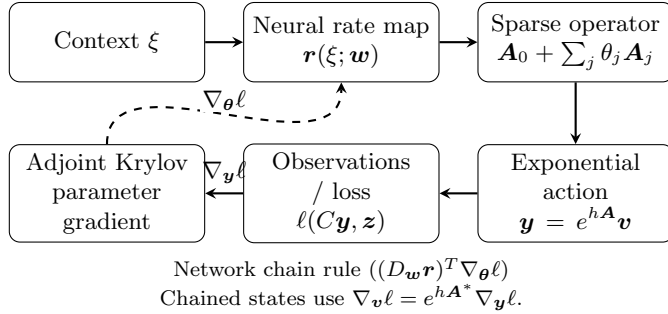


FIG. 6. Gradient calculation for a context-dependent rate model. Solid arrows show the forward model and the derivative of the loss. The dashed arrow passes the coefficient gradient to the network. The optional support mask is separate from the network weights.

Within a channel, a softmax map produces nonnegative weights that sum to one. We gradually lower its temperature so that the weights concentrate on a small number of candidates. A neural network maps the context to the four channel amplitudes, and a second stage refits their rates after selection. We use two complementary initial conditions to distinguish the candidate responses. Because the architecture imposes the four channel groups, selection within these groups is not an unrestricted library search. The fit selects the four true directions. On the test contexts, the mean relative errors in A , B , k_2 , and k_4 are 1.2%, 0.4%, 2.4%, and 2.8%, respectively (Figure 7).

For comparison with context-aware reactive SINDy, we regress trajectory increments on instantaneous propensities with a nonnegative elastic-net fit. Across three seeds, this fit retains all four true directions together with 4 ± 1 false directions. A variant with shared coefficients retains 6 ± 1 false directions. These methods use trajectory increments, whereas the neural exponential model fits histogram snapshots. The comparison is therefore between complete inference procedures and does not isolate the gradient calculation.

We also test an ungrouped linear context map on a smaller noiseless problem. All five runs recover the four active directions, with a mean coefficient error of 0.875% and a test-path error of 1.11%. Section D.2 gives the calibration and confirmation protocol. This experiment tests support selection without channel constraints, but its linear rate map does not test a neural architecture.

A.4. Stuart–Landau diffusion across a bifurcation. On a 30×30 grid ($n = 900$), we discretize $\partial_t p = -\nabla \cdot (bp) + D\Delta p$ with $D = 0.15$ and

$$\mathbf{b}(x, y; \mu) = \begin{pmatrix} \mu x - y - (x^2 + y^2)x \\ x + \mu y - (x^2 + y^2)y \end{pmatrix}.$$

A degree-three drift library contains 20 directions, of which eight are active. A neural map with two width-32 tanh hidden layers learns their coefficients across eight contexts $\mu \in [-0.3, 1.1]$. We start from an off-center Gaussian so that the observations contain rotational motion. A centered radial density would not contain this information. With an observation interval of one, the initial sparse fit and the subsequent refit recover all eight active terms. For seed 42, the mean test-context errors are approximately 1.5% for the coefficient determined by μ and 0.5% for the rotation coefficient.

An RK4 comparison with 32 substeps selects two terms, only one of which is

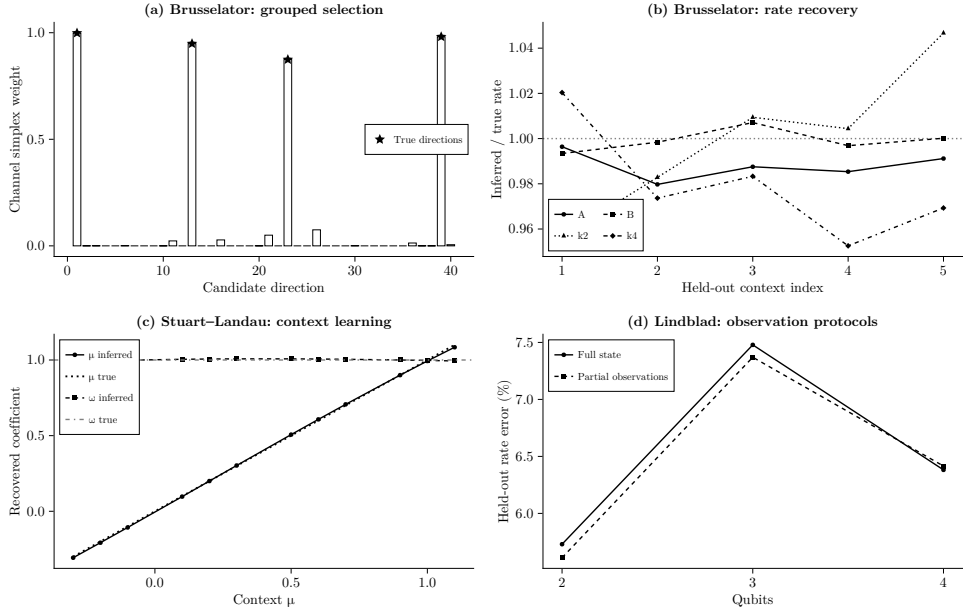


FIG. 7. *Context-dependent rate experiments. Black lines, markers, and line styles distinguish the recorded runs. (a) Grouped Brusselator simplex weights, with the true terms marked by stars. (b) Brusselator rate ratios on the test contexts. (c) Stuart-Landau coefficient functions. (d) Lindblad rate errors under the two known-support observation protocols.*

true. It uses 100 discovery epochs and takes 294 seconds. The exponential model uses 800 discovery epochs and takes 92 seconds, so these timings do not compare equal optimization budgets. We also train unstructured neural ODEs with hidden widths 16 and 64, corresponding to 30,004 and 120,388 parameters. Both models use 32 RK4 substeps per unit time and 800 epochs. Across three seeds, their mean extrapolation losses at $t = 6$ are 6.27×10^{-3} and 5.88×10^{-3} . Negative values occur in 44.5% and 43.6% of the grid cells, and the mean mass errors are 0.030 and 0.050. Further details, including the support fit at a shorter observation interval, appear below.

A.5. Neural Lindblad rates and partial observations. A separate experiment fixes the support of the $2q - 1$ nearest-neighbor exchange and dephasing terms and learns their rates as functions of a scalar context ξ . The fixed field in this experiment is $0.3 \sum_i \sigma_z^{(i)}$. The prescribed rates are $J_i(\xi) = 0.5[1 + 0.4 \sin(i\pi\xi)]$ and $\gamma_i(\xi) = 0.1[1 + 0.5 \cos^2(i\pi\xi)]$. A network with two width-64 tanh hidden layers represents the rate map. A softplus output $x \mapsto \log(1 + e^x)$ and a fixed positive floor keep every inferred rate positive. Ten training contexts 0.05, 0.15, ..., 0.95 and five test contexts 0.10, 0.30, ..., 0.90 use observations at times 0.3, 1, 3. Adam uses step size 0.003 for 600 epochs at $q = 2, 3$ and 800 epochs at $q = 4$.

Full-state training fits consecutive density snapshots. For partial observations, we instead propagate the known initial state $\rho_0 = (|+\rangle\langle+|)^{\otimes q}$ directly to each observation time, minimizing $\frac{1}{2} \sum_t \|C e^{t\mathbf{A}(\mathbf{r}(\xi; \mathbf{w}))} \text{vec}(\rho_0) - \mathbf{y}_t\|^2$. The rows of C measure the identity and all single-site Pauli expectations. They also measure nearest-neighbor XX , YY , and ZZ expectations. The observation map has $6q - 2$ rows. The partial-

TABLE 3

Known-support neural Lindblad fitting. The partial data are noiseless expectation values from a known initial state. Errors are measured in the rates on the test contexts.

Qubits	Full-state dim.	Observables	Full-state error	Partial error
2	16	10	5.73%	5.61%
3	64	16	7.48%	7.37%
4	256	22	6.38%	6.42%

observation fit does not use intermediate full density matrices. Thus the full-state and partial experiments differ in both their observation maps and their losses. Table 3 reports one initialization, seed 42, for each case. We average the componentwise relative rate errors over the test contexts. These experiments evaluate neural rate fitting on a fixed support under the two observation protocols.

With sparse operators, full-state neural rate fitting also reaches five and six qubits. For seed 42 and 800 epochs, the mean test-context rate errors are 8.77% and 8.76%, and the training times are 256 and 930 seconds. These are known-support fits of nine and eleven coefficients. They are separate from the unknown-support experiments through six qubits and the gradient scaling experiment through eleven qubits.

Across three seeds, local Z observations alone produce a nearly zero observation loss but mean rate errors of 293–338%. Adding single-site and nearest-neighbor Pauli observations reduces the mean errors to 5.44%, 7.93%, and 6.12% at two, three, and four qubits. Simulating 100 measurement shots increases these errors to 11.5%, 16.3%, and 15.9%. With 10,000 shots, the errors approach their noiseless values. Thus a small observation loss from local Z measurements need not imply accurate rates. Results for the intermediate shot counts and smaller observation sets appear below.

Appendix B. Additional Discussion. The forward and adjoint Krylov spaces are shared by all coefficient directions on an observation interval. Only the library contractions grow with the number of directions. Derivatives with respect to the input state and interval length connect the exponential action to the calculations that precede and follow it. The cost analysis in the main paper identifies when reuse of the two spaces reduces the runtime.

The applications vary the active coefficient set, the dependence of the rates on context, and the observations entering the loss. Each observed transition requires its own forward and adjoint spaces. In the nonlinear ablation, we isolate the input-state derivative by omitting that term from the chain rule. The CME, diffusion, and Lindblad examples include real and complex generators with different conservation laws.

Reuse is most effective when construction of the two Krylov spaces costs more than contraction with the operator library. Differentiated Arnoldi and dense automatic differentiation are faster in other parts of the measured accuracy–cost range. The forward approximation, adjoint approximation, and quadrature must all be resolved before errors caused by support ambiguity, observation noise, model mismatch, or time discretization can be interpreted.

Adaptive selection of the forward and adjoint depths remains useful when library storage and the full training cost are included. Automatic thresholding of context-dependent rate functions is a separate inference problem. The experiments below measure gradient accuracy and record the behavior of the stated fitting procedures on synthetic data.

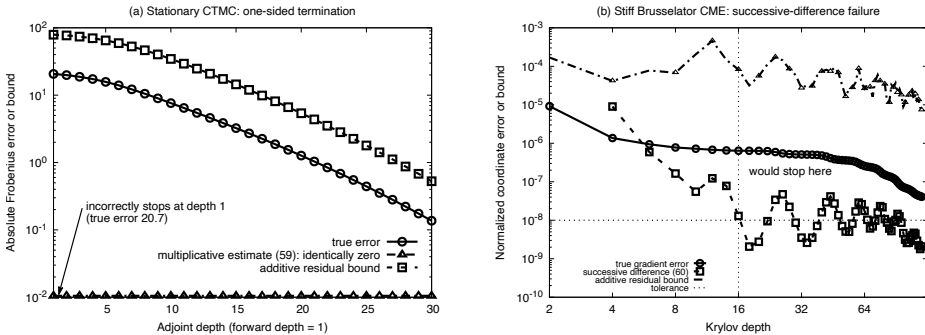


FIG. 8. Two failures of heuristic stopping estimates. (a) One-sided termination makes the residual-product estimate of [16, Eq. (59)] identically zero. (b) In the stiff Brusselator problem, the difference between successive projected approximations crosses 10^{-8} before the true normalized gradient error. The additive residual bounds remain valid but conservative.

B.1. Failure modes of heuristic stopping estimates. Figure 8 shows two ways in which a heuristic stopping estimate can become small before the approximation is accurate. Panel (a) uses the symmetric generator of a reflecting random walk on 200 states, an interval of length 80, the stationary initial vector $\mathbf{v} = 200^{-1/2}(1, \dots, 1)^T$, and terminal vector $\nabla_{\mathbf{y}}\ell = \delta_1$. The generator is negative semi-definite, so its semigroup is contractive in the 2-norm. The forward Arnoldi process terminates after one step, while the adjoint depth ranges from 1 to 30. The multiplicative estimate is therefore zero at every plotted depth, even though the true Frobenius error decreases only from 20.66 to 0.137.

The curve labeled “additive residual bound” controls the full Fréchet matrix rather than an individual gradient component. The unit-norm forward path is exact, so the expansion used in Theorem 4.2 gives

$$\|L_{\text{exp}}(h\mathbf{A}, (\nabla_{\mathbf{y}}\ell)\mathbf{v}^*) - \tilde{L}\|_F \leq |h| \int_0^1 \|e_{\lambda}(s)\|_2 ds.$$

Variation of constants and contractivity in the 2-norm bound the right-hand side by the integrated adjoint Arnoldi residual. The resulting bound decreases from 79.0 to 0.527 over the plotted depths.

Panel (b) uses the final interval, with $h = 0.5$, of the stiff 1681-state Brusselator problem and a 40-direction cubic library. The plotted approximation and the depth-420 reference both use the two-sided projection, so quadrature error is absent from this comparison. At depth 16, the successive-approximation estimate of [16, Eq. (60)] is 5.40×10^{-9} , whereas the normalized library-gradient error is 6.40×10^{-7} . At depth 120, these values are 2.08×10^{-9} and 4.07×10^{-8} . The additive curve uses the 1-norm for the forward error and the ∞ -norm for the adjoint error in Corollary 4.3. The CTMC semigroup is nonexpansive in the first norm, and its adjoint is nonexpansive in the second. Although the bound remains above the true error, its smallest value through depth 120 is 7.32×10^{-6} and does not certify the target 10^{-8} . For this problem, the bound detects unresolved error but does not determine an efficient stopping depth.

TABLE 4

Relative coefficient-gradient error against the dense block-Fréchet reference (5.1) for the non-normal Fokker-Planck case at $\Delta t = 0.1$.

m_K	$n_q = 8$	$n_q = 24$	$n_q = 32$
20	9.1×10^{-6}	1.0×10^{-5}	1.0×10^{-5}
30	9.6×10^{-6}	1.1×10^{-8}	1.1×10^{-8}
40	9.7×10^{-6}	1.7×10^{-13}	1.7×10^{-13}
60	9.7×10^{-6}	1.8×10^{-15}	2.0×10^{-15}

B.2. Derivation of the complexity bound. Proposition 3.2 separates the construction of the Krylov spaces, evaluation at the quadrature nodes, and contraction with the library directions. In one Arnoldi process, the m_K applications of $\mathbf{A}$ or $\mathbf{A}^*$ cost $O(m_K \text{nnz}(\mathbf{A}))$. At step r , modified Gram-Schmidt orthogonalizes a length- n vector against r basis vectors. Hence

$$O\left(n \sum_{r=1}^{m_K} r\right) = O(nm_K^2).$$

Each exponential action requires a forward basis and an adjoint basis. Let N_{act} denote the number of such actions. The first term of the bound absorbs the constant factors associated with the two constructions.

At one quadrature node, the two reduced matrix exponentials cost $O(m_K^3)$, and multiplication by the two bases costs $O(nm_K)$. Repeating these operations over n_q nodes and N_{act} actions gives the second term. For a fixed action, node, and direction, the sparse product $\mathbf{A}_j \mathbf{f}_q$ and its inner product with $\boldsymbol{\lambda}_q$ cost $O(\text{nnz}(\mathbf{A}_j) + n)$. Summing over directions, nodes, and actions gives

$$O\left(N_{\text{act}} n_q \sum_{j=1}^{M_{\text{lib}}} (\text{nnz}(\mathbf{A}_j) + n)\right).$$

For a CME direction, $\text{nnz}(\mathbf{A}_j) = O(n)$, so the final term becomes $O(N_{\text{act}} n_q M_{\text{lib}} n)$. With $\mathbf{A}$ and its sparsity pattern fixed, the library size appears only in this contraction term.

B.3. Additional accuracy and timing results. Table 4 gives the numerical values used in the Fokker-Planck refinement discussion of the main paper.

Appendix C. CME Construction and Sparse-Inference Details. This section describes the CME construction, initialization, and sparse optimization used in the supporting experiments.

C.1. Trajectory-based initialization. Several CME experiments use exact jump records from stochastic simulation algorithm (SSA) paths [10], rather than the snapshot-only data used in Section 6.2. For candidate direction j , let $g_j(\mathbf{x})$ be its known state-dependent propensity factor and let θ_j be its unknown coefficient. Let $c_j^{(i)}$ be the number of times that direction fires in path i , and let $X^{(i)}(t)$ be the corresponding state. The exact-jump maximum-likelihood initializer is

$$(C.1) \quad \hat{\theta}_j^{(0)} = \frac{\sum_{i=1}^N c_j^{(i)}}{\sum_{i=1}^N \int_0^{T_{\text{obs}}} g_j(X^{(i)}(t)) dt}.$$

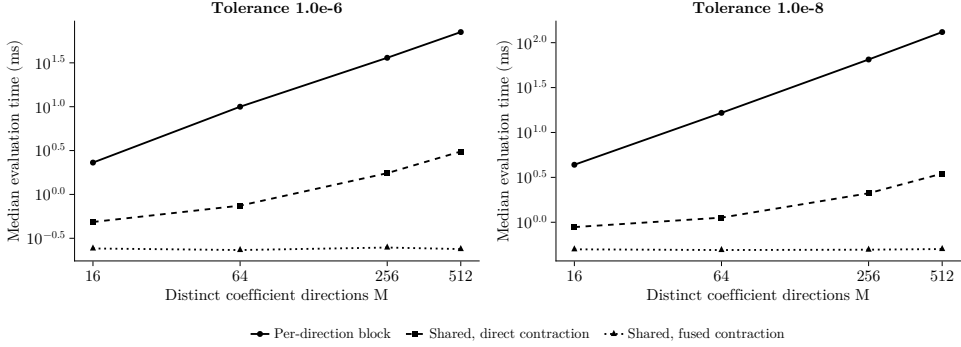


FIG. 9. Library scaling at matched accuracy on an Apple M1 Pro. Each time is the median of seven evaluations after warmup and includes the forward action, the coefficient gradient, and the input-state derivative. Lines connect the measured configurations and are not extrapolated scaling laws.

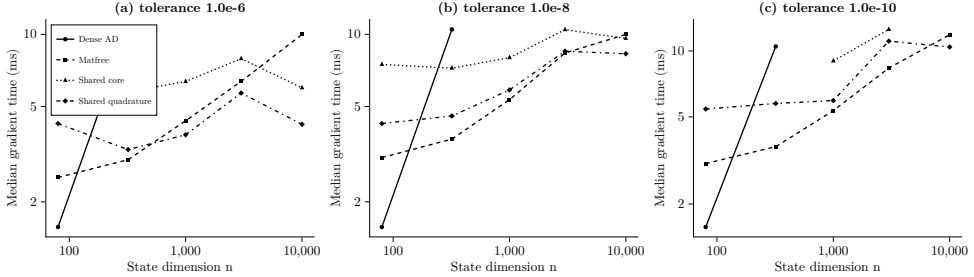


FIG. 10. GPU comparison within one process at fixed accuracy. Each curve uses the fastest configuration that meets the tolerance for both the coefficient gradient and the input-state derivative. Compilation and warmup are excluded.

This estimate initializes the subsequent support selection and rate refinement. Equation (C.1) requires labeled reaction events so that $c_j^{(i)}$ is known for each candidate. When several mechanisms produce the same observed state change and their event labels are unavailable, the individual counts cannot be separated. We then use a joint likelihood, a nonnegative estimate from snapshot data, or a prior estimate.

Remark C.1 (Initialization from snapshot data). With snapshot distributions but no trajectory records, a coarse nonnegative least-squares estimate or a domain-specific prior can replace (C.1).

The synthetic path benchmarks use the exact-jump initializer.

C.2. Iterative thresholded optimization with sensitivity scaling. Starting from this estimate, the sparse inverse problem alternates smooth local optimization with hard thresholding [7, 4]. To compare rates that span several orders of magnitude, we optimize the scaled variables $\phi = D\theta$. Here $D = \text{diag}(d_j)$ is positive and d_j is the norm of column j of the initial Jacobian. Thresholding therefore compares coefficients on their sensitivity scales rather than by their raw magnitudes. The scaled objective

Algorithm C.1 Iterative Thresholded Optimization with Optional Sensitivity Scaling

Require: Nonnegative initializer $\theta^{(0)}$, diagonal scaling $\mathbf{D}$ (identity if unused), threshold fraction $\eta_{\text{thr}} \in (0, 1)$, inner iteration limit L_{inner} , outer iterations L_{outer}

Ensure: Sparse estimate $\hat{\theta}$

```

1:  $\phi^{(0)} \leftarrow \mathbf{D}\theta^{(0)}$ 
2: for  $\ell = 0, \dots, L_{\text{outer}} - 1$  do
3:    $S_\ell \leftarrow \text{supp}(\phi^{(\ell)})$ 
4:    $\phi^{(\ell+1/2)} \leftarrow \text{LBFGS}(\tilde{\mathcal{L}}|_{S_\ell}, \nabla_{S_\ell} \tilde{\mathcal{L}}, \phi^{(\ell)})$  for at most  $L_{\text{inner}}$  iterations
5:    $\phi^{(\ell+1/2)} \leftarrow \max(\phi^{(\ell+1/2)}, 0)$ 
6:    $\tau_\ell \leftarrow \eta_{\text{thr}} \|\phi^{(\ell+1/2)}\|_\infty$ 
7:   Let  $\phi^{(\ell+1)} \leftarrow \phi^{(\ell+1/2)}$ 
8:   Let  $\phi_j^{(\ell+1)} = 0$  whenever  $|\phi_j^{(\ell+1/2)}| < \tau_\ell$ 
9: end for
10: Repeat the L-BFGS solve and nonnegative projection on the fixed support  $\text{supp}(\phi^{(L_{\text{outer}})})$ 
11: return  $\hat{\theta} = \mathbf{D}^{-1}\hat{\phi}$ 

```

is $\tilde{\mathcal{L}}(\phi) = \mathcal{L}(\mathbf{D}^{-1}\phi)$.

Each smooth optimization uses the adjoint gradient. Thresholding selects the reactions, and Jacobian-column scaling makes directions with different physical units comparable. Because the loss is nonlinear in the coefficients, this procedure is not sequential thresholded least squares. The experiments use the stated nonnegative projection rather than a bound-constrained inner solve.

C.3. Support stability across numerical settings and seeds. This experiment separates numerical stability from variation across data realizations. For each of six seeds, the selected support is unchanged across two Krylov depths and four quadrature orders. Seeds 1, 4, and 6 recover the true support in all eight settings. Seeds 2, 3, and 5 retain the same extra direction in all eight settings. Thus the aggregate count 24/48 represents three successful seeds repeated across eight numerical settings, not 48 independent recovery trials. Increasing the depth or quadrature order within this grid does not remove the false positive.

We therefore distinguish stability under numerical refinement from the effects of threshold selection, correlated library directions, and variation in the data. Figure 12 places the numerical check next to a separate perturbation of the initial estimate. This perturbation does not change the observations. The noisy KS and finite-shot Lindblad studies in Section D examine variation in the observations.

C.4. Finite-state CME. Consider d chemical species with copy-number state $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{N}^d$. Each reaction channel $m \in [M]$ has a stoichiometry vector $\zeta_m \in \mathbb{Z}^d$ and a nonnegative propensity $\alpha_m(\mathbf{x}; \theta_m)$. The propensity is the instantaneous transition rate from $\mathbf{x}$ to $\mathbf{x} + \zeta_m$. We model the system as a continuous-time Markov chain (CTMC) on a truncated state space $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{N}^d$. The conservative boundary convention in Eq. (6.2) suppresses a reaction whose destination lies outside $\mathcal{X}$. It preserves total probability within the retained state space and differs from the leaking finite-state projection of [27].

The probability vector $\mathbf{p}(t) \in \mathbb{R}^n$, with $p_i(t) = \Pr(X(t) = \mathbf{x}_i)$, satisfies the

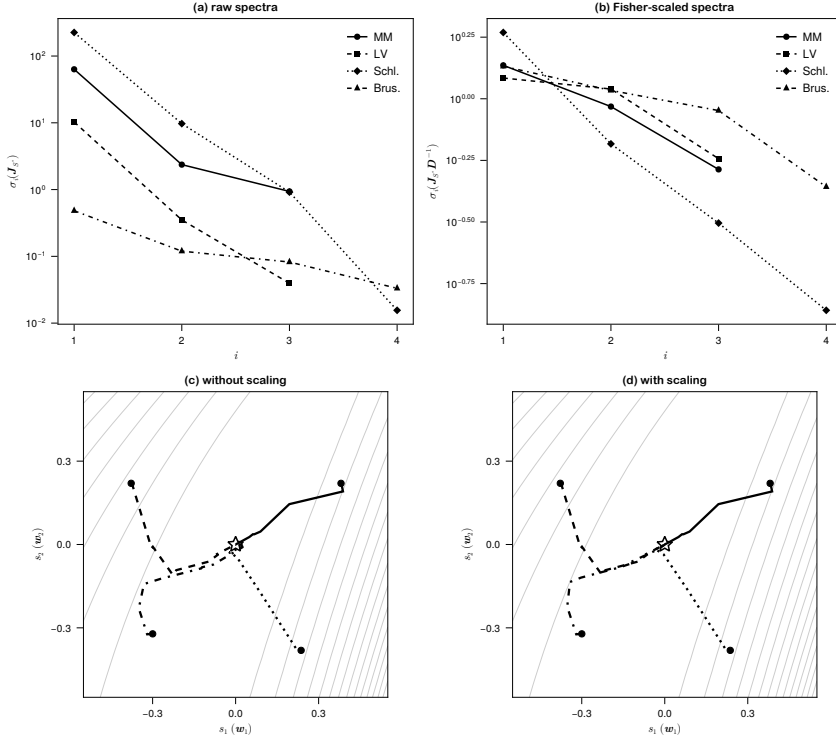


FIG. 11. *Jacobian-column sensitivity scaling.* (a) Singular values $\sigma_i(J_S^*)$ for Michaelis–Menten, Lotka–Volterra, Schlögl, and Brusselator. The wide spread for Schlögl reflects its three-order-of-magnitude range of rates. (b) Column scaling by $J_S^* D^{-1}$ reduces the condition numbers, with the largest reduction for Schlögl. (c,d) Schlögl loss slices along the right singular vectors (w_1, w_2) associated with the largest singular values. The curves show optimization from four initial estimates, and the white star marks θ^* . Without scaling (c), the iterates follow elongated contours before converging. With scaling (d), they take shorter and more direct paths.

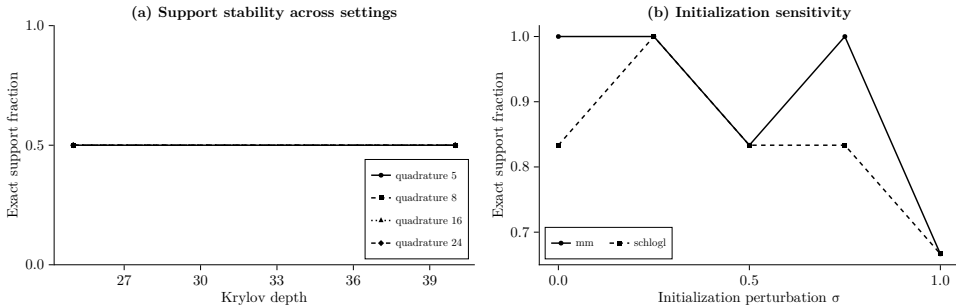


FIG. 12. *Support stability.* The left panel shows the exact-support fraction at each Krylov depth and quadrature order. The right panel shows the exact-support fraction under initialization perturbations for Michaelis–Menten and Schlögl. Fractions use all seeds at each setting.

chemical master equation

$$(C.2) \quad \frac{d\mathbf{p}}{dt} = \mathbf{A}(\boldsymbol{\theta}) \mathbf{p},$$

where, with $\alpha_m^{\mathcal{X}}(\mathbf{x}; \boldsymbol{\theta}_m) = \alpha_m(\mathbf{x}; \boldsymbol{\theta}_m) \mathbf{1}_{\{\mathbf{x} + \boldsymbol{\zeta}_m \in \mathcal{X}\}}$, the generator $\mathbf{A}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$ has entries

$$(C.3) \quad [\mathbf{A}(\boldsymbol{\theta})]_{\mathbf{y}, \mathbf{x}} = \sum_{m=1}^M \alpha_m^{\mathcal{X}}(\mathbf{x}; \boldsymbol{\theta}_m) (\mathbf{1}_{\{\mathbf{y} = \mathbf{x} + \boldsymbol{\zeta}_m\}} - \mathbf{1}_{\{\mathbf{y} = \mathbf{x}\}}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}.$$

The columns sum to zero, so $\mathbf{1}^T \mathbf{A} = \mathbf{0}$ and total probability is conserved. Over one observation interval Δt , the solution of (C.2) is

$$(C.4) \quad \mathbf{p}(t + \Delta t) = e^{\mathbf{A}(\boldsymbol{\theta}) \Delta t} \mathbf{p}(t),$$

so the matrix exponential $\mathbf{P}_{\Delta t} = e^{\mathbf{A} \Delta t}$ is the transition operator over the interval.

C.5. Reaction library. The reaction library has the affine form (2.1). Each mass-action direction represents an elementary reaction with a physical rate constant, so the recovered support identifies a reaction network. The Fokker–Planck and Lindblad examples in Section 2.1 use operator libraries constructed from their corresponding physical terms. The gradient calculation does not depend on the choice of library.

In stochastic mass-action kinetics, a reaction consuming ν_i molecules of species i fires at a rate proportional to the number of distinct reactant subsets. Counting ordered selections and dividing out permutations gives the falling-factorial form

$$(C.5) \quad \phi_{\boldsymbol{\nu}}(\mathbf{x}) = \prod_{i=1}^d \binom{x_i}{\nu_i} = \prod_{i=1}^d \frac{x_i(x_i-1) \cdots (x_i-\nu_i+1)}{\nu_i!},$$

where $\boldsymbol{\nu} = (\nu_1, \dots, \nu_d) \in \mathbb{N}^d$ is the vector of reactant stoichiometries. For a single species, the order-0 term is $\binom{x}{0} = 1$, which describes a source or creation reaction independent of copy number. The order-1 term is $\binom{x}{1} = x$, as in linear degradation or transport. The order-2 term is $\binom{x}{2} = x(x-1)/2$, which counts unordered pairs in a dimerization reaction. The order-3 term is $\binom{x}{3} = x(x-1)(x-2)/6$, which counts triples in a trimerization reaction. This convention counts unordered reactant combinations. An alternative falling-factorial convention absorbs the factorial into the rate constant.

C.5.1. Library parameterization.. We express each propensity as a linear combination of K basis functions of the form (C.5):

$$\alpha_m(\mathbf{x}; \boldsymbol{\theta}_m) = \boldsymbol{\theta}_m^T \boldsymbol{\phi}(\mathbf{x}), \quad \boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^K, \quad \boldsymbol{\theta}_m \geq 0,$$

where $\boldsymbol{\phi}(\mathbf{x})$ collects the combinatorial monomials up to a chosen order. For a one-species system the cubic library is $\boldsymbol{\phi}(x) = [1, x, \frac{x(x-1)}{2}, \frac{x(x-1)(x-2)}{6}]^T$. For d species, we include all multivariate products $\prod_i \binom{x_i}{\nu_i}$ with $\sum_i \nu_i \leq r$ for a chosen order r . Because each propensity is linear in $\boldsymbol{\theta}_m$, the generator is linear in the full coefficient vector $\boldsymbol{\theta} = \text{vec}(\{\boldsymbol{\theta}_m\})$:

$$(C.6) \quad \mathbf{A}(\boldsymbol{\theta}) = \sum_{m=1}^M \sum_{k=1}^K \theta_{m,k} \mathbf{A}_m^{(k)}, \quad \theta_{m,k} \geq 0.$$

Once the library and state space are fixed, the direction matrices $\mathbf{A}_m^{(k)} = \partial \mathbf{A} / \partial \theta_{m,k}$ have entries

$$(C.7) \quad [\mathbf{A}_m^{(k)}]_{\mathbf{x} + \boldsymbol{\zeta}_m, \mathbf{x}} = \phi_k(\mathbf{x}), \quad [\mathbf{A}_m^{(k)}]_{\mathbf{x}, \mathbf{x}} = -\phi_k(\mathbf{x}),$$

when $\mathbf{x} + \boldsymbol{\zeta}_m \in \mathcal{X}$, with all entries in that column zero otherwise. We compute these matrices once and reuse them throughout inference. The synthetic libraries contain the true reactions and inactive alternatives. We therefore seek both the active set $S^* = \{j : \theta_j^* > 0\}$ and the values of its nonzero coefficients.

C.6. Conditional identifiability and threshold separation. Let F_S collect the sampled predictions, including their observation maps, as a function of the active coefficients, and let $J_S = DF_S(\boldsymbol{\theta}^*)$. If J_S has full column rank, it contains a nonsingular square submatrix formed from a subset of its rows. Applying the inverse function theorem to the corresponding output coordinates shows that two sufficiently close coefficient vectors on S cannot produce the same sampled output. Thus the coefficients are locally identifiable on S . Injectivity of the full matrix-exponential Fréchet map does not by itself imply this rank condition. The initial states and observation maps may remove informative directions. The diagnostic below instead assumes a fully observed continuous path from one initial state. It complements, but does not replace, a rank and conditioning check of the sampled Jacobian for the actual measurements.

For threshold separation, suppose that an iterate before thresholding satisfies $\|\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}^*\|_\infty \leq \delta$ in the chosen scaled coordinates. If $\delta < \tau < \min_{j \in S^*} |\phi_j^*| - \delta$, hard thresholding recovers S^* . Inactive coordinates have magnitude at most δ , and active ones have magnitude at least $\min_{S^*} |\phi_j^*| - \delta$. This statement requires control of both the active and inactive coordinates. A positive-definite Hessian restricted to the active coordinates controls perturbations only within the true support and gives no bound on the inactive coordinates. We do not claim a global convergence result for the neural optimization.

C.7. A computable identifiability diagnostic. For a family that is linear in its parameters, we can also compute a continuous-path diagnostic directly. Stanhope, Rubin, and Swigon [39, Thm. 2.6] show that global identifiability of $\mathbf{A}(\boldsymbol{\theta}) = \sum_j \theta_j \mathbf{A}_j$ from the single initial condition $\mathbf{p}_0$ requires

$$(C.8) \quad \text{rank } \mathbf{G}(\mathbf{A}) = M, \quad \mathbf{G}(\mathbf{A})_{:,j} = [\mathbf{A}_j \mathbf{p}_0; \mathbf{A}_j \mathbf{A} \mathbf{p}_0; \dots; \mathbf{A}_j \mathbf{A}^{n-1} \mathbf{p}_0].$$

Column j records the action of direction j along the Krylov sequence of the observed path. Computing $\mathbf{G}$ requires the additional vectors $\mathbf{A}^k \mathbf{p}_0$ and the products $\mathbf{A}_j \mathbf{A}^k \mathbf{p}_0$. We therefore form it as an offline diagnostic rather than during the gradient calculation. After column normalization, its singular values measure the conditioning of the identifiability test, while pairwise angles reveal directions that are hard to separate.

The Schlögl library illustrates both effects. All eight singular values of the normalized $\mathbf{G}$ are nonzero to working precision, so Eq. (C.8) has full rank. Its condition number is nevertheless 3.6×10^6 , and two pairs of columns are nearly collinear. The constant birth feature containing k_3 lies 0.89° from a death direction, while the true cubic birth feature lies 3.66° from an inactive cubic birth candidate. These angles explain the behavior reported in Section 6.2. An aggressive threshold removes the term in the first pair, and the second pair remains difficult to distinguish after selection. Both effects arise from the library and the observed path. Equation (C.8) detects them before fitting. Additional initial conditions or observation times can separate the directions when this ambiguity is unacceptable.

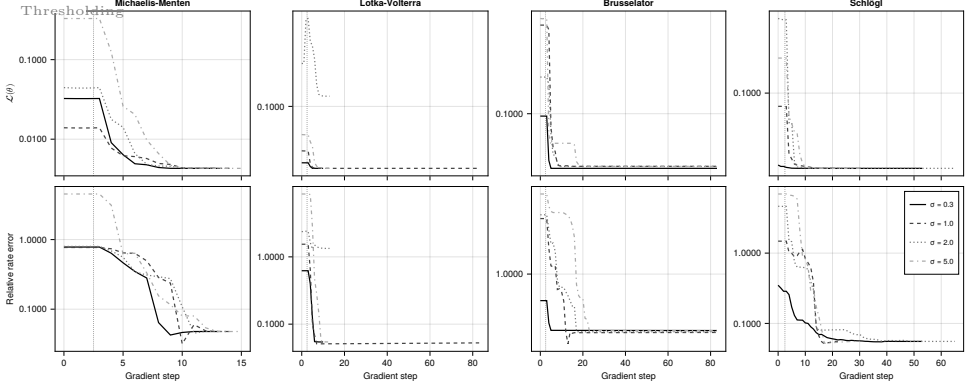


FIG. 13. *Thresholded optimization and final refinement under perturbations $\sigma = 0.3, 1, 2, 5$ of the initial rate estimate. Line styles and gray levels distinguish the perturbations. The upper panels show the objective, and the lower panels show the relative rate error. The observations are unchanged in this experiment.*

TABLE 5

Kuramoto-Sivashinsky composition over five initializations. Errors are reported as mean $\pm$ population standard deviation in percent.

Gradient calculation	Coefficient error	Prediction error
Full	1.83915 ± 0.00007	0.090826 ± 0.000001
Input-state derivative omitted	18.63 ± 1.68	1.041 ± 0.093

TABLE 6

Lindblad identification from an overcomplete library. Each row summarizes five independent draws of the active rates.

Qubits	Liouville dim.	Library size	True support	Exact	Mean rate error
2	16	7	3	5/5	7.5×10^{-9}
3	64	12	5	5/5	5.6×10^{-9}
4	256	18	7	5/5	1.3×10^{-8}
5	1024	25	9	5/5	1.4×10^{-8}
6	4096	33	11	5/5	7.3×10^{-9}

Appendix D. Additional Numerical and Application Experiments.

This section gives supporting and adverse cases from the numerical and application studies. Each timing uses the machine and implementation on which it was originally recorded. Comparisons across programming languages therefore include differences in both the method and its implementation.

D.1. Extended nonlinear composition experiments. For the split step in Eq. (6.8), differentiation gives

$$(D.1) \quad d\mathbf{v}_k = (I - hcD_x \text{diag}(\mathbf{u}_k))d\mathbf{u}_k - \frac{h}{2}D_x(\mathbf{u}_k \odot \mathbf{u}_k)dc.$$

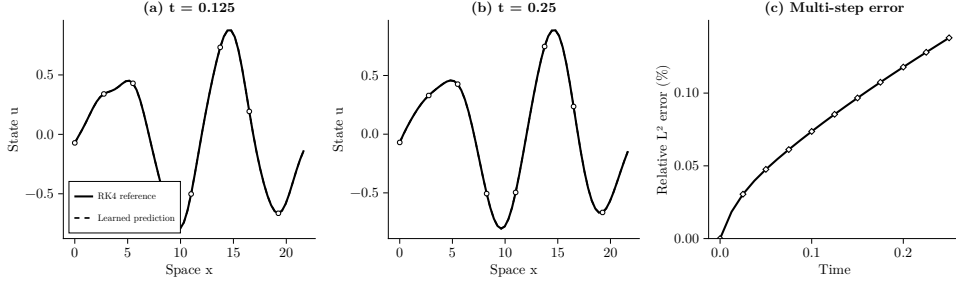


FIG. 14. Kuramoto–Sivashinsky test-path prediction using the complete derivative. Solid curves show the RK4 reference. Dashed curves and open markers show the inferred model at two times, and the right panel gives the relative L^2 error. These short paths test composition of the derivatives, not long-time chaotic prediction.

For a loss $\ell(\mathbf{u}_{k+1})$, the reverse derivatives are

$$(D.2) \quad \nabla_{\mathbf{v}_k} \ell = e^{h\mathbf{A}(a,b)^T} \nabla_{\mathbf{u}_{k+1}} \ell,$$

$$(D.3) \quad \frac{\partial \ell}{\partial c} = -\frac{h}{2} (\nabla_{\mathbf{v}_k} \ell)^T D_x(\mathbf{u}_k \odot \mathbf{u}_k),$$

$$(D.4) \quad \nabla_{\mathbf{u}_k} \ell = (I - hc D_x \text{diag}(\mathbf{u}_k))^T \nabla_{\mathbf{v}_k} \ell.$$

The derivatives with respect to a and b use the forward and adjoint paths

$$\mathbf{f}(s) = e^{sh\mathbf{A}} \mathbf{v}_k, \quad \boldsymbol{\lambda}(s) = e^{(1-s)h\mathbf{A}^T} \nabla_{\mathbf{u}_{k+1}} \ell,$$

and the directions $-D_{xx}$ and $-D_{xxxx}$ in the main adjoint formula.

Sparse nonlinear library. In a second KS experiment, the library contains 15 candidates. Its linear directions are $-D_x$, $-D_{xx}$, $D_x D_{xx}$ and $-D_{xxxx}$. The remaining 11 candidates are $-D_x \mathbf{q}_r(\mathbf{u})/r$, with $r = 2, \dots, 12$, where $[\mathbf{q}_r(\mathbf{u})]_i = u_i^r$. The true support contains only $-D_{xx}$, $-D_{xxxx}$ and the quadratic flux. The training data contain 20 transitions from each of three fields over $T = 0.25$. Independent Gaussian noise ranges from zero to 1% of the root-mean-square amplitude of each snapshot. A fourth noise-free path tests prediction through $T = 0.5$. We optimize an ℓ_1 -regularized fit with L-BFGS, apply a threshold, and refit the selected terms without a penalty. The penalty 10^{-7} and threshold 0.08 are calibrated on seed 11 at 0.5% noise and then fixed for 20 initializations at every noise level. All 80 runs recover exactly the three active terms with no false positives. Table 7 reports this support result separately from the coefficient and prediction errors. Refitting reduces both errors through 0.5% noise but increases them at 1%. This experiment tests sparse semilinear identification on a small grid and over short time intervals using the complete derivative.

D.2. Context-dependent support selection. To separate the rate parameterization from prior knowledge of the active support, we consider a Brusselator on a 13×13 grid ($n = 169$) whose rates depend on context. The library contains four candidate stoichiometries and ten falling-factorial propensity features for each stoichiometry, giving 40 directions. A nonnegative matrix $W \in \mathbb{R}^{40 \times 3}$ defines $\theta_j(a, b) = W_{j,:} [1, a, b]^T$. Unlike selection by a separate simplex within each stoichiometric channel, this parameterization imposes neither a one-feature-per-channel

TABLE 7

KS robustness over 20 initializations per noise level. Coefficient and noise-free test-path errors are percentages. D denotes the initial sparse fit and R denotes refitting on the selected support. Both stages retain the support selected by the initial fit.

Noise	Exact	Coeff. D	Coeff. R	Test D	Test R
0%	20/20	5.48	1.72	0.513	0.140
0.25%	20/20	5.07	1.98	0.480	0.157
0.5%	20/20	4.22	3.21	0.407	0.255
1%	20/20	5.98	8.79	0.508	0.713

constraint nor a prescribed support size. The true rates depend linearly on (a, b) , so a linear rate map is sufficient.

Training uses exact snapshots generated by the model at four contexts. For each context, the data include six initial states and observation intervals 0.05, 0.20, and 0.50. We first compute an ℓ_1 -regularized fit, threshold its rates, and refit the selected support without a penalty. The threshold 0.20 is calibrated on seed 11 and fixed before running the confirmation seeds 22, 33, 44, and 55. All five runs recover exactly the four active directions with no false positives. This result includes the four untouched confirmation runs. Across all five runs, the mean relative coefficient error is 0.875%, the prediction error at four test contexts is 1.11%, and the error at the longer horizon $t = 1.5$ is 1.05%.

This experiment isolates context-dependent mechanism selection without channel-wise constraints. The data are noiseless, the state space is small, and the true rate map is linear. A nonlinear rate map requires separate tests on unseen contexts under the same support-selection protocol.

D.3. Under-resolved training and independent adjoint resolution. The choice of derivative matters when the forward approximation is coarse. In an 80-state, ten-direction Fokker–Planck experiment, we hold fixed the finite-Arnoldi forward map, exact targets, four initial states, parameter initialization, and Adam updates. With $h = 1$, $m_K = 30$, and $n_q = 8$, the adjoint calculation gives an exact-exponential prediction error of 19.3% and a coefficient error of 374% after 2000 epochs. Differentiating the finite Arnoldi calculation gives errors of 4.58% and 55.2%. This single-seed comparison shows that optimization of an under-resolved finite-Arnoldi model can favor the derivative of the finite calculation. The forward and adjoint approximations must both be resolved when the target is the gradient of the exact exponential action.

At a stationary initial distribution, a nearly exact forward value can coexist with an inaccurate coefficient gradient. In the recorded CTMC experiment, the forward relative error is 1.86×10^{-11} at depth three, while the coefficient-gradient error is 0.180. Increasing the adjoint depth and checking its residual resolves the latter error. A stopping rule based only on the forward calculation cannot detect it. The stationary-start, invariant, and precision checks below give the corresponding numerical details.

D.4. Large sparse Lindblad contraction evaluation. The explicit sparse benchmark reaches eleven qubits on a 16 GB Apple M1 Pro. Its generator contains local fields, nearest-neighbor exchange, dephasing, and decay. The operator library contains the local dephasing and nearest-neighbor exchange directions, for a total of $2q - 1$ directions. This benchmark measures gradient computation for a fixed library rather than support recovery.

TABLE 8

Explicit sparse Lindblad scaling on a 16 GB Apple M1 Pro. Times are in milliseconds. The final column gives the relative change in the gradient after increasing the depth and quadrature order as described in the text.

Qubits	Dimension	Directions	m_K/n_q	Operator MiB	Workspace MiB	Forward	Gradient	Relative change
7	16,384	13	40/8	8.9	28.5	17.5	188.7	2.47×10^{-15}
8	65,536	15	40/8	40.6	114.6	75.3	532.7	2.61×10^{-15}
9	262,144	17	40/8	183.0	476.9	310.2	2,757.6	4.02×10^{-15}
10	1,048,576	19	40/8	814.0	1,994.3	1,471.1	9,346.6	7.68×10^{-15}
11	4,194,304	21	24/6	3,584.0	6,248.5	3,382.9	13,341.5	2.56×10^{-13}

Table 8 reports median times after warmup. We use three timed runs through ten qubits and two runs at eleven qubits. The first four rows use Krylov depth 40 and eight quadrature nodes. The eleven-qubit calculation uses depth 24 and six nodes. Increasing these values to 32 and 10 changes its gradient by 2.56×10^{-13} . Workspace memory includes the two Arnoldi bases and the fused sparse-direction cache, while operator storage includes the generator and every library direction. The eleven-qubit process reaches a maximum resident memory of 7.31 GB without swapping.

Quadrature also permits the forward and adjoint Arnoldi calculations to use the same workspace. We test this memory schedule on the local Lindblad generator

$$\mathcal{L}(\rho) = -i \left[\sum_{s=1}^q \omega_s Z_s, \rho \right] + \sum_{s=1}^q \gamma_s (Z_s \rho Z_s - \rho),$$

with positive, site-dependent coefficients. The generator is diagonal in the local matrix-unit basis. Its action therefore requires $O(4^q)$ storage rather than an explicit $4^q \times 4^q$ matrix. We differentiate a local transverse-field commutator and a local decay dissipator, neither of which commutes with the generator. The initial density matrix and the derivative of the loss are dense product operators.

We use $h = 2$, Krylov depth 32, and eight Gauss–Legendre nodes. Exact divided differences of the diagonal generator give an independent reference gradient. At depth 24, the relative errors are 2.59×10^{-11} for ten qubits and 1.95×10^{-10} for eleven qubits. Depth 32 reduces both errors below 10^{-11} and is used throughout Table 9.

The quadrature calculation stores the forward values at the eight nodes and then overwrites the forward Arnoldi basis with the adjoint basis. The retained-basis core constructs both bases once and contracts its singular factors one at a time. The recomputed-basis core stores the selected adjoint factors, overwrites the adjoint basis, and constructs the forward basis a second time.

D.5. CPU/GPU timing comparisons and dense differentiation. In the portable sweep, the local transition rates remain fixed as the state dimension increases. For each method, we choose the fastest tested configuration that meets the error tolerance for both the coefficient gradient and the input-state derivative. Table 1 compares GPU quadrature with differentiated Arnoldi. The table below adds CPU measurements from the same JAX process. We do not interpolate between tested configurations, and a dash indicates that none meets the tolerance. Figure 15 compares dense JAX and PyTorch differentiation with the sparse Julia calculation. JAX uses compilation, PyTorch runs eagerly, and the Julia implementation uses sparse operators.

D.6. Two-sided projected factor, shared sparsity and precision. Quadrature can be replaced by a small block exponential that uses the same two bases.

TABLE 9

Memory schedules for the matrix-free Lindblad gradient on a 16 GB Apple M1 Pro. The dimensions are 4^q . We do not run the retained-basis core at twelve qubits because its two depth-32 bases alone require 16 GiB. Errors are measured against the exact divided-difference gradient. Each measurement starts in a fresh process. Times cover the gradient calculation, and `/usr/bin/time -l` reports the peak memory footprint.

Qubits	Dimension	Evaluation	Arnoldi passes	Rank	Time (s)	Peak (GiB)	Relative error
10	1,048,576	Quadrature	2	—	5.30	1.26	1.52×10^{-13}
		Core	2	10	5.50	1.66	5.89×10^{-14}
		Core, recomputed	3	10	8.19	1.26	5.90×10^{-14}
11	4,194,304	Quadrature	2	—	11.54	3.44	2.20×10^{-13}
		Core	2	10	12.69	5.03	2.15×10^{-13}
		Core, recomputed	3	10	17.82	3.40	2.15×10^{-13}
12	16,777,216	Quadrature	2	—	94.06	12.30	4.49×10^{-14}
		Core	2	—	—	> 16	—
		Core, recomputed	3	10	124.20	12.53	1.65×10^{-12}

TABLE 10

Same-process Apple M1 CPU timings at tolerance 10^{-6} , in milliseconds.

CPU	n	Matfree	Quadrature	Projection	Dense AD
Apple M1	80	0.717	2.96	0.886	1.2
Apple M1	320	1.87	3.83	1.76	15.1
Apple M1	1000	6.37	7.8	5.56	N/A

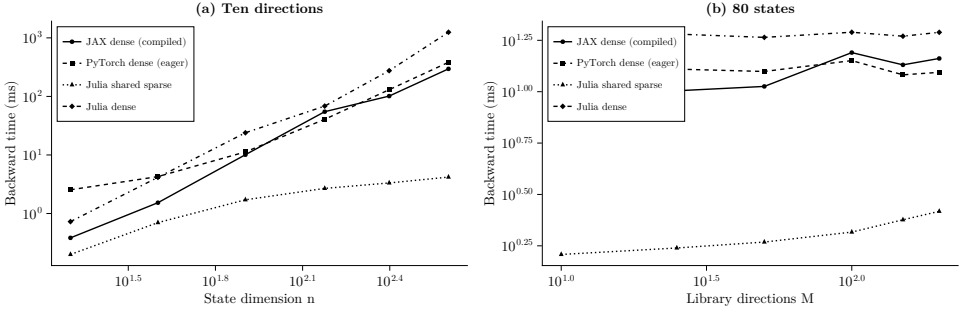


FIG. 15. Dense automatic differentiation and sparse Julia timing sweeps. The left panel varies the state dimension with ten directions, and the right panel varies the library size at $n = 80$. Each curve retains its original language and execution mode. This figure is separate from the same-backend comparison with differentiated Arnoldi.

Let $\mathbf{v} = \beta_f V_f \delta_1$ and $\nabla_{\mathbf{y}} \ell = \beta_a V_a \delta_1$. The block exponential

$$(D.5) \quad C_m = \left[\exp \left(h \begin{pmatrix} H_a & \beta_a \beta_f \delta_1 \delta_1^* \\ 0 & H_f^* \end{pmatrix} \right) \right]_{12}.$$

contains the two-sided projected Fréchet factor. If $C_m \approx U_r \Sigma_r W_r^*$, the contractions become $\partial \ell / \partial \theta_j \approx \text{Re} \sum_{k=1}^r \sigma_k (V_a u_k)^* A_j (V_f w_k)$. This is the two-sided approximation of Kressner [21] and Algorithm 2 of Kandolf et al. [16], specialized to the rank-one direction obtained from the derivative of the loss. Its factors are shared across the

TABLE 11

Union-pattern contraction at $n = 400$, depth 40, and $h = 0.4$. Timings exclude construction of the sparse cache.

Directions	Direct (ms)	Fused (ms)	Speedup	Relative difference
10	0.488	0.448	1.09	1.14×10^{-15}
50	0.688	0.496	1.38	1.29×10^{-15}
200	1.43	0.648	2.21	1.89×10^{-15}
1000	5.57	1.52	3.67	2.25×10^{-15}

TABLE 12

Precision check against a Float64 dense reference. The finite-depth calculations in this table are not fully converged.

System	h	Precision	Forward error	Parameter error	State error
Fokker–Planck	0.3	float64	2.05×10^{-6}	9.50×10^{-5}	1.14×10^{-4}
Fokker–Planck	0.3	float32	2.52×10^{-6}	1.34×10^{-4}	2.42×10^{-4}
Fokker–Planck	1.0	float64	1.12×10^{-5}	1.95×10^{-4}	2.79×10^{-4}
Fokker–Planck	1.0	float32	1.72×10^{-5}	6.17×10^{-4}	1.51×10^{-3}
Lindblad	1.0	float64	2.72×10^{-16}	4.56×10^{-10}	1.50×10^{-15}
Lindblad	1.0	float32	2.74×10^{-7}	6.60×10^{-7}	1.06×10^{-6}

affine library. For orthonormal bases, Cauchy–Schwarz gives

$$\left| \delta \left(\frac{\partial \ell}{\partial \theta_j} \right) \right| \leq \|C_m - U_r \Sigma_r W_r^*\|_F \|\mathbf{A}_j\|_F.$$

This bound controls truncation of the projected factor, but not the Krylov projection error. The block exponential evaluates the projected integral exactly and therefore has no quadrature remainder.

When several directions share matrix entries, we store their values in a sparse $|P| \times M_{\text{lib}}$ matrix over the union pattern $P = \bigcup_j \text{supp}(\mathbf{A}_j)$. The low-rank product is evaluated once at each entry of this union, after which multiplication by the sparse value matrix gives every directional contraction. Cache construction and storage are counted separately. Compression and fused contraction need not reduce the cost for a small library.

We measure factor compression and the union-pattern contraction separately. Table 11 fixes the state size, bases, and generator and varies only the library size. The relative difference compares two implementations of the same projected factor. It is not an error against the exact exponential derivative.

In a wide-stencil check with $n = 400$, bandwidth parameter 25, and forward depth 60, the coefficient-gradient errors are 9.71×10^{-15} for differentiated Arnoldi and 9.19×10^{-12} for the rank-eight symmetric factor. Reducing only the adjoint depth to 20 increases the latter error to 5.27×10^{-9} , which shows the effect of resolving the two sides independently. A second comparison uses a dense reference with $n = 80$, 50 directions, and depth 40. The projected calculation takes 0.993 milliseconds, and differentiated Arnoldi takes 0.764 milliseconds. Their coefficient errors are 6.25×10^{-11} and 3.46×10^{-9} , respectively.

D.7. Stationary starts, invariants and residual diagnostics. At an exactly stationary CTMC input, the forward Arnoldi process may terminate after one vector

TABLE 13

Near-stationary CTMC derivative check at perturbation 10^{-10} . Gradient errors use a dense evaluation of the exact exponential derivative as the reference.

Depth	Forward error	Adjoint Krylov gradient	Matfree gradient
3	1.14×10^{-11}	0.125	0.831
4	3.66×10^{-12}	0.0456	0.713
8	9.65×10^{-14}	1.01×10^{-3}	0.0515
12	1.11×10^{-15}	3.99×10^{-6}	3.63×10^{-4}

because $\mathbf{A}\mathbf{p}_0$ is a multiple of $\mathbf{p}_0$. The tested differentiated-Arnoldi implementation returns nonfinite gradients at this breakdown. Replacing the breakdown with the terminated forward value instead produces a zero coefficient derivative and misses the sensitivity of the exact exponential action. Near stationarity, differentiation of the finite Arnoldi operations is well defined, but its result can still differ substantially from the derivative of the exact action. The separate adjoint space resolves the path that remains nontrivial.

For conservative directions and an invariant adjoint vector $\mathbf{c}$, we have $\mathbf{A}^*\mathbf{c} = \mathbf{A}_j^*\mathbf{c} = 0$, so the exact parameter gradient vanishes and the input-state gradient equals $\mathbf{c}$. The CTMC and Lindblad checks, using an equivalent real representation in the second case, confirm both identities for the adjoint Krylov calculation. Explicit projection onto the invariant restores them for differentiated Arnoldi. Preservation of these invariants does not certify the remaining gradient components.

For the CTMC diagnostic, we compute the residual bound from Corollary 4.3 for each coefficient. We add the absolute difference between the n_q - and $2n_q$ -point quadrature results as an empirical estimate of the quadrature error. In every saved test, the sum exceeds the error measured against the dense reference.

We use this sum as the half-width of an interval around the magnitude of each approximate gradient. The ranking is resolved only if the lower endpoint for every selected direction exceeds the upper endpoint for every unselected direction. Overlap makes the test inconclusive even when the computed ranking is correct. The residual contribution is bounded, while the difference between the two quadrature rules is an empirical estimate rather than a rigorous remainder bound.

D.8. Library size, correlation and misspecification. In a noiseless 30-state CTMC experiment, the library contains all 870 directed edges. Four edges are active, and three training distributions place mass at their source states. An ℓ_1 -regularized fit is followed by a fixed threshold and an unpenalized refit. All five seeds recover the four edges without knowing the support size. The mean relative errors in the coefficients, test transitions, and long-horizon predictions are 2.57×10^{-7} , 9.58×10^{-8} , and 9.98×10^{-8} .

A large library alone is less problematic than a library with nearly collinear candidates. In the correlation sweep below, accurate prediction can coexist with an incorrect support. At correlation 0.9 and penalty 5×10^{-5} , exact recovery changes from 0/5 after Adam to 5/5 after L-BFGS refinement. Optimization and threshold separation therefore also affect support recovery. In the misspecified libraries, the planted mechanism is absent, so an exact-support fraction is not defined.

D.9. Neural application comparisons. These discovery comparisons use the same library but different integrators, observation intervals, and optimization budgets.

TABLE 14

Five-seed library experiments. Setting denotes correlation or misspecification magnitude. It is zero for the size sweep.

Sweep	Library size	Setting	Exact fraction	Test error	Long-time error
Size	10	0	1	1.12×10^{-8}	1.16×10^{-8}
Size	20	0	1	1.09×10^{-8}	1.08×10^{-8}
Size	40	0	1	7.99×10^{-9}	8.32×10^{-9}
Size	80	0	1	7.68×10^{-9}	7.93×10^{-9}
Correlation	40	0.9	0	0.019	0.0551
Correlation	40	0.99	0	8.26×10^{-3}	0.0257
Correlation	40	0.999	0	3.27×10^{-3}	0.0111
Misspecified	40	0.01	N/A	1.72×10^{-4}	8.14×10^{-4}
Misspecified	40	0.05	N/A	8.54×10^{-4}	4.03×10^{-3}
Misspecified	40	0.1	N/A	1.70×10^{-3}	7.96×10^{-3}
Misspecified	40	0.2	N/A	3.39×10^{-3}	0.0156

TABLE 15

Stuart–Landau integrator comparisons during the discovery stage. The true model contains eight terms.

Method	True hits	False hits	Epochs	Seconds
Exponential $h=1.0$	8	0	800	91.6
Euler $h=1.0$	3	9	800	37.9
RK4×1 $h=1.0$	2	8	800	97.6
RK4×32 $h=1.0$	1	1	100	294
Exponential $h=0.1$	7	1	800	74.2

TABLE 16

Lindblad observation studies. Entries are mean componentwise rate errors on test contexts, in percent, over three seeds.

Qubits	Full state	Local Pauli	Single-site Pauli	Local Z
2	5.94	5.44	6.6	338
3	6.97	7.93	15.5	293
4	5.8	6.12	6.96	295

The exponential fit at the shorter interval also misses one true term. A finer time discretization alone does not guarantee support recovery at a fixed training budget.

D.10. Additional path and diffusion experiments. The exponential derivative also applies to path likelihoods. For an observed transition $i \rightarrow j$ with positive

TABLE 17

Lindblad experiments with finite-shot local Pauli observations. Each entry is the mean rate error in percent over three seeds.

Qubits	Noiseless	100 shots	1,000 shots	10,000 shots
2	5.44	11.5	6.56	5.56
3	7.93	16.3	8.69	7.92
4	6.12	15.9	7.45	6.2

TABLE 18

Archived test coefficient errors, in percent. For two-dimensional entries, we average the relative norm of the two-coefficient vector over test contexts. This avoids division by a vanishing individual coefficient.

Experiment / loss	Contexts	Mean error (%)
Trajectory OU (likelihood)	5	1.94
Trajectory OU (snapshot MSE)	5	1.2
Double well (exponential)	6	6.11
Double well (Euler–Maruyama)	6	N/A
2D trajectory (likelihood)	5	79.4
2D trajectory (snapshot MSE)	5	58.6

probability, the loss is $-\log([e^{hA}]_{ji})$. Its derivative at the right endpoint differs from the derivative of a squared snapshot error. The archived Ornstein–Uhlenbeck, double-well, and two-dimensional diffusion studies use this likelihood. The table reports their test coefficient errors separately from the matched gradient comparisons. A missing Euler–Maruyama entry denotes a nonfinite result.

REFERENCES

- [1] A. H. AL-MOHY AND N. J. HIGHAM, *Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation*, SIAM Journal on Matrix Analysis and Applications, 30 (2009), pp. 1639–1657, <https://doi.org/10.1137/080716426>.
- [2] A. H. AL-MOHY AND N. J. HIGHAM, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM Journal on Scientific Computing, 33 (2011), pp. 488–511, <https://doi.org/10.1137/100788860>.
- [3] E. BAIREY, C. GUO, D. POLETTI, N. H. LINDNER, AND I. ARAD, *Learning the dynamics of open quantum systems from their steady states*, New Journal of Physics, 22 (2020), p. 032001, <https://doi.org/10.1088/1367-2630/ab73cd>.
- [4] T. BLUMENSATH AND M. E. DAVIES, *Iterative hard thresholding for compressed sensing*, Applied and Computational Harmonic Analysis, 27 (2009), pp. 265–274, <https://doi.org/10.1016/j.acha.2009.04.002>.
- [5] M. A. BOTCHEV, V. GRIMM, AND M. HOCHBRUCK, *Residual, restarting, and Richardson iteration for the matrix exponential*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1376–A1397, <https://doi.org/10.1137/110820191>.
- [6] R. J. BOYS, D. J. WILKINSON, AND T. B. L. KIRKWOOD, *Bayesian inference for a discretely observed stochastic kinetic model*, Statistics and Computing, 18 (2008), pp. 125–135, <https://doi.org/10.1007/s11222-007-9043-x>.
- [7] S. L. BRUNTON, J. L. PROCTOR, AND J. N. KUTZ, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proceedings of the National Academy of Sciences, 113 (2016), pp. 3932–3937, <https://doi.org/10.1073/pnas.1517384113>.
- [8] Y. CAO, S. LI, L. PETZOLD, AND R. SERBAN, *Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1076–1089, <https://doi.org/10.1137/S1064827501380630>.
- [9] V. GEBHART, R. SANTAGATI, A. A. GENTILE, E. M. GAUGER, D. CRAIG, N. ARES, L. BANCHI, F. MARQUARDT, L. PEZZÈ, AND C. BONATO, *Learning quantum systems*, Nature Reviews Physics, 5 (2023), pp. 141–156, <https://doi.org/10.1038/s42254-022-00552-1>.
- [10] D. T. GILLESPIE, *Exact stochastic simulation of coupled chemical reactions*, The Journal of Physical Chemistry, 81 (1977), pp. 2340–2361, <https://doi.org/10.1021/j100540a008>.
- [11] A. GOLIGHTLY AND D. J. WILKINSON, *Bayesian inference for stochastic kinetic models using a diffusion approximation*, Biometrics, 61 (2005), pp. 781–788, <https://doi.org/10.1111/j.1541-0420.2005.00345.x>.
- [12] V. GORINI, A. KOSSAKOWSKI, AND E. C. G. SUDARSHAN, *Completely positive dynamical semigroups of N -level systems*, Journal of Mathematical Physics, 17 (1976), pp. 821–825, <https://doi.org/10.1063/1.522979>.
- [13] N. J. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008, <https://doi.org/10.1137/1.9780898717778>.
- [14] M. HOCHBRUCK AND C. LUBICH, *On Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 1911–1925, <https://doi.org/10.1137/S0036142995280572>.
- [15] M. HOFFMANN, C. FRÖHNER, AND F. NOÉ, *Reactive SINDy: Discovering governing reactions from concentration data*, The Journal of Chemical Physics, 150 (2019), p. 025101, <https://doi.org/10.1063/1.5066099>.
- [16] P. KANDOLF, A. KOSKELA, S. D. RELTON, AND M. SCHWEITZER, *Computing low-rank approximations of the fréchet derivative of a matrix function using krylov subspace methods*, Numerical Linear Algebra with Applications, 28 (2021), p. e2401, <https://doi.org/10.1002/nla.2401>.
- [17] P. KANDOLF AND S. D. RELTON, *A block Krylov method to compute the action of the Fréchet derivative of a matrix function on a vector with applications to condition number estimation*, SIAM Journal on Scientific Computing, 39 (2017), pp. A1416–A1434, <https://doi.org/10.1137/16M1077969>.
- [18] S. KLUS, F. NÜSKE, S. PEITZ, J.-H. NIEMANN, C. CLEMENTI, AND C. SCHÜTTE, *Data-driven approximation of the Koopman generator: model reduction, system identification, and control*, Physica D: Nonlinear Phenomena, 406 (2020), p. 132416, <https://doi.org/10.1016/j.physd.2020.132416>.
- [19] V. R. KOSTIC, K. LOUNICI, H. HALCONRUY, T. DEVERGNE, AND M. PONTIL, *Learning the infinitesimal generator of stochastic diffusion processes*, in Advances in Neural Information Processing Systems, vol. 37, 2024, pp. 137806–137846.
- [20] N. KRÄMER, P. MORENO-MUÑOZ, H. ROY, AND S. HAUBERG, *Gradients of functions of large*

- matrices*, arXiv preprint arXiv:2405.17277, (2024).
- [21] D. KRESSNER, *A Krylov subspace method for the approximation of bivariate matrix functions*, in Structured Matrices in Numerical Linear Algebra: Analysis, Algorithms and Applications, Springer International Publishing, 2019, pp. 197–214, https://doi.org/10.1007/978-3-030-04088-8_10.
 - [22] D. KRESSNER AND P. OEHME, *A novel Krylov subspace method for approximating Fréchet derivatives of large-scale matrix functions*, arXiv preprint arXiv:2601.21799, (2026).
 - [23] G. LINDBLAD, *On the generators of quantum dynamical semigroups*, Communications in Mathematical Physics, 48 (1976), pp. 119–130, <https://doi.org/10.1007/BF01608499>.
 - [24] D. A. MESSENGER AND D. M. BORTZ, *Weak SINDy for partial differential equations*, Journal of Computational Physics, 443 (2021), p. 110525, <https://doi.org/10.1016/j.jcp.2021.110525>.
 - [25] D. A. MESSENGER AND D. M. BORTZ, *Weak SINDy: Galerkin-based data-driven model selection*, Multiscale Modeling & Simulation, 19 (2021), pp. 1474–1497, <https://doi.org/10.1137/20M1343166>.
 - [26] C. MOLER AND C. VAN LOAN, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45 (2003), pp. 3–49, <https://doi.org/10.1137/S00361445024180>.
 - [27] B. MUNSKY AND M. KHAMMASH, *The finite state projection algorithm for the solution of the chemical master equation*, The Journal of Chemical Physics, 124 (2006), p. 044104, <https://doi.org/10.1063/1.2145882>.
 - [28] A. NEJATI, A. LAVAEI, S. SOUDJANI, AND M. ZAMANI, *Data-driven estimation of infinitesimal generators of stochastic systems*, IFAC-PapersOnLine, 54 (2021), pp. 277–282, <https://doi.org/10.1016/j.ifacol.2021.08.511>.
 - [29] G. NEUERT, B. MUNSKY, R. Z. TAN, L. TEYTELMAN, M. KHAMMASH, AND A. VAN OUDENAARDEN, *Systematic identification of signal-activated stochastic gene regulation*, Science, 339 (2013), pp. 584–587, <https://doi.org/10.1126/science.1231456>.
 - [30] S. H. RUDY, S. L. BRUNTON, J. L. PROCTOR, AND J. N. KUTZ, *Data-driven discovery of partial differential equations*, Science Advances, 3 (2017), p. e1602614, <https://doi.org/10.1126/sciadv.1602614>.
 - [31] B. RUSSO AND H. A. DYE, *A note on unitary operators in C^* -algebras*, Duke Mathematical Journal, 33 (1966), pp. 413–416, <https://doi.org/10.1215/S0012-7094-66-03346-1>.
 - [32] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2nd ed., 2003, <https://doi.org/10.1137/1.9780898718003>.
 - [33] H. SCHAEFFER AND S. G. MCCALLA, *Sparse model selection via integral terms*, Physical Review E, 96 (2017), p. 023302, <https://doi.org/10.1103/PhysRevE.96.023302>.
 - [34] P. J. SCHMID, *Dynamic mode decomposition and its variants*, Annual Review of Fluid Mechanics, 54 (2022), pp. 225–254, <https://doi.org/10.1146/annurev-fluid-030121-015835>.
 - [35] D. SCHNOERR, G. SANGUINETTI, AND R. GRIMA, *Approximation and inference methods for stochastic biochemical kinetics—a tutorial review*, Journal of Physics A: Mathematical and Theoretical, 50 (2017), p. 093001, <https://doi.org/10.1088/1751-8121/aa54d9>.
 - [36] R. SERBAN AND A. C. HINDMARSH, *CVODES: The sensitivity-enabled ODE solver in SUNDIALS*, in Proceedings of the ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2005, pp. 257–269.
 - [37] R. B. SIDJE, *Expokit: A software package for computing matrix exponentials*, ACM Transactions on Mathematical Software, 24 (1998), pp. 130–156, <https://doi.org/10.1145/285861.285868>.
 - [38] A. SINGH AND J. A. P. HESPANHA, *Approximate moment dynamics for chemically reacting systems*, IEEE Transactions on Automatic Control, 56 (2011), pp. 414–418, <https://doi.org/10.1109/TAC.2010.2088631>.
 - [39] S. STANHOPE, J. E. RUBIN, AND D. SWIGON, *Identifiability of linear and linear-in-parameters dynamical systems from a single trajectory*, SIAM Journal on Applied Dynamical Systems, 13 (2014), pp. 1792–1815, <https://doi.org/10.1137/130937913>.
 - [40] T. TONI, D. WELCH, N. STRELKOWA, A. IPSEN, AND M. P. H. STUMPF, *Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems*, Journal of the Royal Society Interface, 6 (2009), pp. 187–202, <https://doi.org/10.1098/rsif.2008.0172>.
 - [41] M. O. WILLIAMS, I. G. KEVREKIDIS, AND C. W. ROWLEY, *A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition*, Journal of Nonlinear Science, 25 (2015), pp. 1307–1346, <https://doi.org/10.1007/s00332-015-9258-5>.