SCIENCE CHINA
Information Sciences

• Supplementary File •

SpatioTemporal Diffusion with Koopman Operator for Multi-step Prediction of Short-term Time-series

Liangyu Su[1], Jun Shu[1*], Luonan Chen[2], Deyu Meng[1] & Zongben Xu[1]

[1]*School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China*
[2]*Key Laboratory of Systems Biology, Shanghai Institute of Biochemistry and Cell Biology, Center for Excellence in Molecular Cell Science, Chinese Academy of Sciences, Shanghai 200031, China*

Appendix A Notation

Throughout, matrices (such as $X \in \mathbb{R}^{d\times m}$) are written in uppercase, vectors ($\mathbf{x} \in \mathbb{R}^d$) are written in bold lowercase, and scalars ($x \in \mathbb{R}$) are written in plain lowercase. X_i, denotes the ith row of X and $X_{,j}$ denotes the jth column of X. $\mathbf{x}_{i:j}$ denotes $(x_i, \cdots, x_j)^\top \in \mathbb{R}^{j-i}$ where $i < j$. The Hadamard product is denoted by the symbol $\odot$. The inner product of two matrices $X \in \mathbb{R}^{k\times m}, Y \in \mathbb{R}^{k\times n}$ is denoted by $\langle X, Y\rangle = \mathrm{tr}(X^T Y)$. For a matrix X, the maximum eigenvalue is denoted by $\lambda_{\max}(A)$, $\|X\|_F$ denotes the Frobenius norm of X and $\|X\|$ denotes the spectral norm of X.

Appendix B STD framework

The primary goal of the STI framework is to construct a transformation function that can accurately convert spatial information into temporal information, to make time series predictions. Previous methods have utilized numerous heuristic procedures to address the challenge of solving the transformation function casusing some information loss, due to the underdetermined and semi-supervised nature of the STI problem. The main text states that the STI problem can be formulated as a semi-supervised regression framework, known as the STD framework, from the perspective of the Koopman operator:

$$\min_{A,\hat{\mathbf{y}}} \|A\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})\| + \gamma\mathcal{R}(A) \quad s.t. \|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\| < \epsilon, \tag{B1}$$

where $X \in \mathbb{R}^{d\times m}$, $\hat{\mathbf{y}} \in \mathbb{R}^{m+l-1}$, $\mathcal{H}(\hat{\mathbf{y}}) \in \mathbb{R}^{l\times m}$ is a hankelization operator as defined in the main text. For brevity, Y will be used to denote $\mathcal{H}(\mathbf{y})$ in the following. We will see that such an optimization framework can unify the previous heuristics.

Appendix B.1 RDE

Algorithm B1 RDE algorithm

Require: Given dataset $X \in \mathbb{R}^{d\times m}, Y \in \mathbb{R}^m$

1: Randomly pick s tuples Ω_k from $\{1, 2, \cdots, d\}$ with replacement, and each tuple contains L numbers.
2: Fitting Gaussian process regressor ψ_k by minimizing $\sum_{t=1}^{m-1} \|\psi_k(X_{\Omega_k,t}) - y_{t+1}\|$
3: Get one-step prediction set $\{y^k_{m+1} = \psi_k(X_{\Omega_k,m}) \mid k = 1, 2, \cdots, s\}$
4: Get one-step prediction $\hat{y}_{m+1}$ by weighted sum $y_{m+1} = \sum_{k=1}^{s} a_k y^k_{m+1}$

Ensure: $\hat{y}_{m+1}$

RDE [1] is an algorithm for one-step prediction that integrates Gaussian processes into the STI framework, detail sees algorithm B1. For multi-step prediction, the iterative prediction method is used with a linear model replacing the Gaussian process to reduce computational complexity. As the problem only involves single-step prediction, it can be reduced to a simple supervised problem, which can be mathematically expressed simply as:

$$\min_{\Psi} \|\mathbf{a}_1\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})_{2,}\| \quad s.t. \|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\| < \epsilon, \tag{B2}$$

which Ψ is the corresponding multiple Gaussian process regressors in the algorithm B1, and $\mathbf{a}_1$ is a weight set manually. When $\mathcal{R}$ is set to $\mathbf{0}$ in model [B1], the resulting model is the RDE model [B2], which is a special case of the proposed STD framework.

Algorithm B2 ARNN algorithm

Require: Given dataset $X \in \mathbb{R}^{d\times m}, Y \in \mathbb{R}^{l\times m}$, random initial network $\Psi : \mathbb{R}^d \mapsto \mathbb{R}^D$ and require $2l-1 \leqslant m$

repeat
  Sampling k variables Ω_k from $\{1, 2, \cdots, D\}$
  Updating B_{t+1} by solving linear equation $\Psi(X)_{\Omega_k,1:m-l+1} = B_{\Omega_k}, Y_{1:m-l+1},$
  $\overline{Y}_{t+1} = \arg\min_Y \|\Psi(X) - B_{t+1}Y\|$
  $A_{t+1} = \arg\min_A \|A\Psi(X) - \overline{Y}_{t+1}\| + \|AB_{t+1} - I\|$
  $Y_{t+1} = \arg\min_Y \|A_{t+1}\Psi(X) - Y\|$
until converge
$\hat{\mathbf{y}} = \mathcal{P}_{\mathcal{H}}(Y_{t+1})$

Ensure:$\hat{\mathbf{y}}$

Appendix B.2 ARNN

ARNN [2] is a two-stage algorithm that integrates reservoir computing into the STI framework for multi-step prediction, as described in detail in algorithm B2. Firstly, to address the difficulty of semi-supervision, ARNN requires that the predicted length not exceed half of the known length, i.e. $2l < m$, reducing the problem to a fully supervised one. In addition, to tackle the challenge posed by the underdetermination of $A\Psi(X) = Y$, ARNN first solves the well-posed problem $\Psi(X)_{,1:m-l+1} = BY_{,m-l+1}$ and then adds the constraint $AB = I$ to achieve the solution of $A\Psi(X) = Y$. Although ARNN simplifies the two challenges of the STI problem and provides a solution, it lacks the convergence guarantee and results in a loss of data information and a reduction in accuracy. Mathematically, ARNN can be summarized as the following problem:

$$\min_{A,\hat{\mathbf{y}}} \|A\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})\| + \|AB - I\| \quad s.t. \|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\| < \epsilon, \tag{B3}$$

which Ψ is a reservoir, $B = \Psi(X)_{,1:m-l+1}\mathcal{H}(\mathbf{y})^{-1}_{,1:m-l+1}$. When $\mathcal{R}$ is set to $\|AB - I\|$ in model [B1], the resulting model is the ARNN model, which is also a special case of STD framework.

Algorithm B3 STD model

Require: Given dataset $X \in \mathbb{R}^{d\times m}, Y \in \mathbb{R}^{l\times m}$, index set Ω.

Initialize A_0, B_0, $a_0 = 1$ and step size $\tau = \frac{1}{2(\lambda_{\max}(\Psi(X)\Psi(X)^\top)+\gamma_1\lambda_{\max}(\nabla^{2^\top}\nabla^2)}$
repeat
  $P_{t+1} = \gamma_1\nabla^\top\nabla A_t + \mathcal{P}_\Omega(A_t\Psi(X) - Y)\Psi(X)^\top$
  $A_{t+1} = \mathrm{Prox}_{\tau\gamma_2}(B_t - 2\tau P_{t+1})$
  $a_{t+1} = \frac{1+\sqrt{1+4a_t^2}}{2}$
  $B_{t+1} = A_t + \frac{a_t-1}{a_{t+1}}(A_{t+1} - A_t)$
until converge
$\hat{\mathbf{y}} = \mathcal{H}^{-1}(A_{t+1}X)$

Ensure: $\hat{\mathbf{y}}$

Appendix C STD model

Our STD model is

$$\begin{aligned} \min_{A,\hat{\mathbf{y}}} \quad & \|A\Psi(X) - \mathcal{H}(\hat{\mathbf{y}})\|_F^2 + \gamma_1\|\nabla^2 A\|_F^2 + \gamma_2\|A\|_1, \\ s.t. \quad & \|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\|_2^2 < \epsilon, \end{aligned} \tag{C1}$$

For a fixed A, the problem [C1] is reduced to

$$\begin{aligned} &\min_{\hat{\mathbf{y}}} \|\sqrt{\mathbf{w}} \odot (\mathcal{H}^{-1}(A\Psi(X)) - \hat{\mathbf{y}})\|_2^2 + \frac{1}{\lambda}\|\hat{\mathbf{y}}_{1:m} - \mathbf{y}_{1:m}\|_2^2, \\ &\text{where } \mathbf{w} = (1, \cdots, l-1, \underbrace{l, \cdots, l}_{m-l+1}, l-1, \cdots, 1) \in \mathbb{R}^{m+l-1}, \end{aligned} \tag{C2}$$

where $\mathcal{H}^{-1}$ is an anti-hankelization operator, $\mathbf{w}$ is a weight and λ denotes the intensity of data noise. The objective stated above is differentiable. Obtain the solution when its derivation is 0

$$\hat{y}_i = \begin{cases} \frac{1}{1+\lambda\sqrt{\mathbf{w}_i}}(y_i + \lambda\sqrt{\mathbf{w}_i}\left[\mathcal{H}^{-1}(A\Psi(X))\right]_i), & i \leqslant m, \\ \left[\mathcal{H}^{-1}(A\Psi(X))\right]_i, & m < i \leqslant m+l-1, \end{cases} \tag{C3}$$

* Corresponding author (email: junshu@mail.xjtu.edu.cn)

With the above solution, we can rewrite problem [C1] in terms of A alone

$$\begin{aligned}&\min_{A} \|P \odot (A\Psi(X) - \mathcal{H}(\mathbf{y}))\|_F^2 + \gamma_1 \left\|\nabla^2 A\right\|_F^2 + \gamma_2 \|A\|_1,\\ &\text{where } P_{i,j} = \begin{cases} \dfrac{1}{1+\lambda\sqrt{\mathbf{w}_{i+j-1}}} & ,i+j \leqslant m+1. \\ 0 & ,i+j > m+1, \end{cases}\end{aligned} \tag{C4}$$

We assume the strength of the data noise is much smaller than the data signal strength so $\lambda \approx 0$. Then the problem [C4] can be reduced to

$$\min_{A} \|\mathcal{P}_\Omega(A\Psi(X) - \mathcal{H}(\mathbf{y}))\|_F^2 + \gamma_1 \|\nabla^2 A\|_F^2 + \gamma_2 \|A\|_1, \tag{C5}$$

the above problem is convex but nonsmooth. We use the algorithm B3 to solve it and the algorithm convergence rate is stated in proposition F1. The convergence criterion is when the difference between the current predicted result and the previous predicted result is less than ϵ.

$$\left\|\mathcal{P}_{\complement_\Omega}\left(A^{t+1}\Psi(X) - A^t\Psi(X)\right)\right\| < \epsilon, \tag{C6}$$

where $\complement_\Omega$ denotes the complement of Ω. After obtaining A, use equation [C3] to recover the $\hat{\mathbf{y}}$ and make a prediction. In practice, we solve the problem to make an anti-hankelization process

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \|\mathcal{H}(\mathbf{y}) - A\Psi(X)\|_F^2. \tag{C7}$$

This problem possesses a closed-form solution achieved through anti-diagonal averaging of $A\Psi(X)$. The prediction $\hat{y_i}$ is given by

$$\hat{y_i} = \left[\mathcal{H}^{-1}\left(A\Psi(X)\right)\right]_i = \frac{1}{w_i} \sum_{p+q=i+1} \left[A\Psi(X)\right]_{p,q}, \quad i \in \{m+1, \cdots, m+l-1\}. \tag{C8}$$

Appendix C.1 Computational complexity

The computational complexity of our method is is $O(dlm)$, which d denote the data dimension, l denote the prediction steps and m denotes the known sequence length. Firstly, the computational burden is mainly concentrated in step 3 and 8 of Algorithm B3. Next we analyze the computational complexity of these two steps one by one.

1. In step 3, we need to compute

   $$P_{t+1} = \gamma_1 {\nabla^2}^\top \nabla^2 A_t + \mathcal{P}_\Omega(A_t\Psi(X) - \mathcal{H}(\mathbf{y}))\Psi(X)^\top$$

   which $A_t \in \mathbb{R}^{l\times d}, \Psi(X) \in \mathbb{R}^{d\times m}$. First, $\nabla^2 A_t$ is defined as follow

   $$\nabla^2 \mathbf{a}_i^t = \begin{cases} \mathbf{0}, & i = 1, \\ \mathbf{a}_i^t - \mathbf{a}_{i-1}^t, & i = 2, \\ \mathbf{a}_i^t - 2\mathbf{a}_{i-1}^t + \mathbf{a}_{i-2}^t, & i > 2. \end{cases}$$

   which $\mathbf{a}_i^t$ denote the i-th row of A_t. Due to the special structure of ∇^2, the computational complexity of $\nabla^2 A$ is $O(ld)$, so the computational complexity of ${\nabla^2}^\top \nabla^2 A$ is also $O(ld)$. Second, the computational complexity of $A_t\Psi(X)$ is $O(mld)$ just like the normal matrix multiplication. $\mathcal{P}_\Omega(A_t\Psi(X) - \mathcal{H}(\mathbf{y})) = M \circ (A_t\Psi(X) - \mathcal{H}(\mathbf{y}))$ which M is a mask, and $\circ$ denote the hadamard product. So, the computational complexity of $P_\Omega(Y)$ is $O(lm)$ for any matrix $Y \in \mathbb{R}^{l\times m}$. In summary, the computational complexity of $P_{t+1} = \gamma_1 {\nabla^2}^\top \nabla^2 A_t + \mathcal{P}_\Omega(A_t\Psi(X) - \mathcal{H}(\mathbf{y}))\Psi(X)^\top$ is $O(dlm)$

2. In step 8, we need to compute $\hat{y} = \mathcal{H}^{-1}(A^*\Psi(X))$, which $\mathcal{H}^{-1}$ is a anti-hankel operator. In practice, we achieve it by anti-diagonal averaging of $A^*\Psi(X)$ and the detail is described by equation (23-24) in the main text. the computational complexity of it is $(m-l+1)l$. Therefore, the computational complexity is not more than $O(lm)$

In summary, the computational complexity of STD algorithm is $O(dlm)$.

Appendix D Proof of proposition 1

Definition 1 (Krylpv subspace)**.** Given a non-singular $K \in \mathbb{R}^{n\times n}$ and $\mathbf{z} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$, the k-th order **Krylov subspace** $\mathcal{K}_r(K, \mathbf{z})$ generated by A from $\mathbf{z}$ is

$$\mathcal{K}_r := \mathcal{K}_r(K, \mathbf{z}) = \text{Span}\left\{\mathbf{z}, K\mathbf{z}, \cdots, K^{r-1}\mathbf{z}\right\}. \tag{D1}$$

Lemma D1 (Property of Krylov subspace)**.** Let d_m be the dimension of the Krylov subspace $\mathcal{K}_m(K, \mathbf{z})$. The sequence $\{d_m\}_{m=1,2,\ldots}$ increases monotonically. Moreover, there exists $k \in \mathbb{N}$ such that it is true that

$$d_m = \begin{cases} m, & m < k, \\ k, & m \geqslant k, \end{cases} \tag{D2}$$

for any $m = 1, 2, \cdots$. Supposing further that $\mathbf{z} = \sum_{i=1}^{\tilde{k}} c_i \mathbf{w}_i$, where $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{\tilde{k}}$ are eigenvectors of K corresponding to distinct eigenvalues and $c_1, c_2, \ldots, c_{\tilde{k}} \neq 0$, it is true that $k = \tilde{k}$.

Proposition D1. Suppose that for some state space $\mathcal{M} \subset \mathbb{R}^n$, the system satisfies linear dynamic $\mathbf{z}^{t+1} = K\mathbf{z}^t$ for $t \in [0, T]$, where $\mathbf{z}^t \in \mathcal{M}\backslash\{\mathbf{0}\}$, $K \in \mathbb{R}^{n\times n}$ is non-singular and with n distinct eigenvalues, and $y^t = \mathbf{b}^\top \mathbf{z}^t$ is a linear observation from system, where $\mathbf{b} \in \mathbb{R}^n$. For any $i \in \{0, 1, \cdots, T-n\}$ and $n \leqslant \tau_i \leqslant T - i$, let

$$\mathbf{a}_i^* = \arg\min_{\mathbf{a}_i} \sum_{t=0}^{\tau_i} \left(\mathbf{a}_i^\top \mathbf{z}^t - y^{t+i}\right)^2, \tag{D3}$$

Suppose that $\mathbf{z}^0 = \sum_{i=1}^n c_i \mathbf{w}_i$, where $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n$ are eigenvectors of K and $c_1, c_2, \ldots, c_n \neq 0$, then for any $i \in \{1, 2\cdots, T\}$, the solution satisfies $\mathbf{a}_i^* = K^\top \mathbf{a}_{i-1}^*$.

Proof.

For any $i \in \{0, 1, \cdots, T\}$, the optimal solution $\mathbf{a}_i^*$ can be expressed in closed form.

$$\mathbf{a}_i^* = \arg\min_{\mathbf{a}_i} \sum_{t=0}^{\tau_i} \|\mathbf{a}_i^\top \mathbf{z}_t - y^{t+i}\|_2^2 = \left(\sum_{t=0}^{\tau_i} \mathbf{z}_t \mathbf{z}_t{}^\top\right)^\dagger \left(\sum_{t=0}^{\tau_i} y_{t+i} \mathbf{z}_t\right). \tag{D4}$$

Observing that

$$y_{t+i} = \mathbf{b}^\top K^i \mathbf{z}_t. \tag{D5}$$

Therefore we can deduce that

$$\mathbf{a}_i^* = \left(\sum_{t=0}^{\tau_i} \mathbf{z}_t \mathbf{z}_t^\top\right)^\dagger \left(\sum_{t=0}^{\tau_i} \mathbf{z}_t \mathbf{z}_t^\top\right) {K^i}^\top \mathbf{b}. \tag{D6}$$

Let $Z = (\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_{\tau_i}) \in \mathbb{R}^{n\times\tau_i}$, then $\mathbf{a}_i^* = \left(ZZ^\top\right)^\dagger ZZ^\top {K^i}^\top \mathbf{b}$. We have $\operatorname{rank}\left(ZZ^\top\right) = \operatorname{rank}(Z) = \dim\left(\mathcal{K}_{\tau_i}\left(K, \mathbf{z}^0\right)\right)$ and according to Lemma D1, the dimension of $\mathcal{K}\left(K, \mathbf{z}^0\right)$ is n due to $\tau_i \geqslant n$. Therefore $\mathbf{a}_i^* = {K^i}^\top \mathbf{b}$ and

$$\mathbf{a}_i^* = K^\top \mathbf{a}_{i-1}^*, \tag{D7}$$

which proved the proposition.

Appendix E Proof of proposition 2

Proposition E1 (sparse linear observation)**.** Let $\mathbf{x} \in \mathbb{R}^d$ and $g : \mathbb{R}^d \mapsto \mathbb{R}$, if there exists $k \in \{1, 2, \cdots, d\}$ such that $g(\mathbf{x}) = x_k$, there is a unique sparse vector $\mathbf{e}_k \in \mathbb{R}^d$ such that $\forall \mathbf{x} \in \mathbb{R}^d$

$$g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{e}_k \rangle. \tag{E1}$$

Proof. if $\exists k \in \{1, 2, \cdots, d\}$ for any $\mathbf{x} \in \mathbb{R}^d$ such that $g(\mathbf{x}) = x_k$, then for any $\mathbf{x}^t, \mathbf{x}^{t'} \in \mathbb{R}^d$, we have

$$\begin{aligned} g(a\mathbf{x}^t + b\mathbf{x}^{t'}) &= ax_k^t + bx_k^{t'} \\ &= ag(\mathbf{x}^t) + bg(\mathbf{x}^{t'}). \end{aligned} \tag{E2}$$

It implies that g is a linear operator. According to the Riesz representation theorem, there exists a unique vector $\mathbf{a} \in \mathbb{R}^d$ such that $\forall \mathbf{x} \in \mathbb{R}^d$

$$g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{a} \rangle. \tag{E3}$$

Observing that

$$\langle \mathbf{x}, \mathbf{e}_k \rangle = x_k, \tag{E4}$$

where $\mathbf{e}_k \in \mathbb{R}^d$ is a vector, in which the k-th element is 1, while all other positions are 0. We can deduce that $\mathbf{a} = \mathbf{e}_k$ combining with Eq.E3, so there exists a unique sparse vector $\mathbf{e}_k$ such that

$$g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{e}_k \rangle, \forall \mathbf{x} \in \mathbb{R}^d,$$

which proved the proposition.

Appendix F Proof of proposition 3

The first subproblem of STD is

$$\min_{A} \|\mathcal{P}_{\Omega}(A\Psi(X)-Y)\|_F^2+\gamma_1\|\nabla^2 A\|_F^2+\gamma_2\|A\|_1, \tag{F1}$$

Let $f(A)=\|\mathcal{P}_{\Omega}(A\Psi(X)-Y)\|_F^2+\gamma_1\|\nabla^2 A\|_F^2$ and $F(A)=f(A)+\gamma_2\|A\|_1$.

Definition 2. Let $\mathcal{X},\mathcal{Y}$ be Hilbert spaces endowed with norm $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|_{\mathcal{Y}}$ respectively. A function $f:\mathcal{X}\mapsto\mathcal{Y}$ is called $K-$Lipschitz, if there exists a real constant K such that, for any $\mathbf{x}_1,\mathbf{x}_2\in\mathcal{X}$

$$\|f(\mathbf{x}_1)-f(\mathbf{x}_2)\|_{\mathcal{Y}}\leqslant K\|\mathbf{x}_1-\mathbf{x}_2\|_{\mathcal{X}}. \tag{F2}$$

And K is also referred to as a Lipschitz constant for the function f.

Lemma F1. ∇f is $2(\lambda_{\max}(\Psi(X)^\top\Psi(X))+\gamma_1\lambda_{\max}(\nabla^{2^\top}\nabla^2))-$Lipschitz.

Proof. For any $A_1,A_2\in\mathbb{R}^{l\times d}$,

$$\begin{aligned}\|\nabla f(A_1)-\nabla f(A_2)\|&\leqslant 2(\|\mathcal{P}_{\Omega}((A_1-A_2)\Psi(X)\Psi(X)^\top)\|+\gamma_1\|\nabla^{2^\top}\nabla^2(A_1-A_2)\|)\\&\leqslant 2(\|(A_1-A_2)\Psi(X)\Psi(X)^\top\|+\gamma_1\|\nabla^{2^\top}\nabla^2(A_1-A_2)\|)\\&\leqslant 2(\|\Psi(X)\Psi(X)^\top\|+\gamma_1\|\nabla^{2^\top}\nabla^2\|)\|A_1-A_2\|\\&\leqslant 2(\lambda_{\max}(\Psi(X)^\top\Psi(X))+\gamma_1\lambda_{\max}(\nabla^{2^\top}\nabla^2)\|A_1-A_2\|,\end{aligned}$$

So ∇f is $2(\lambda_{\max}(\Psi(X)^\top\Psi(X))+\gamma_1\lambda_{\max}(\nabla^{2^\top}\nabla^2)-$Lipschitz.

Lemma F2 (Theorem 4.4 in [3])**.** Let $X_*=\arg\min F\neq\emptyset$, $L(f)$ be the Lipschitz constant of ∇f and $\{A_k\},\{B_k\}$ be generated by . Then for any $k>1$

$$F(A_k)-F(A^\star)\leqslant\frac{2L(f)\|A_0-A^*\|_F^2}{(k+1)^2}\quad\forall A^*\in X_*. \tag{F3}$$

Invoking lemma F2 with $f(A)=\|\mathcal{P}_{\Omega}(A\Psi(X)-Y)\|_F^2+\gamma_1\|\nabla^2 A\|_F^2$ and $F(A)=f(A)+\gamma_2\|A\|_1$, we obtain Proposition 2.

Proposition F1. Let $X_*=\arg\min F\neq\emptyset$, $\{A_k\},\{B_k\}$ be generated by algorithm B3. Then for any $k>1$

$$F(A_k)-F(A^*)\leqslant\frac{4(\lambda_{\max}(\Psi(X)^\top\Psi(X))+\gamma_1\lambda_{\max}(\nabla^{2^\top}\nabla^2))\|A_0-A^*\|_F^2}{(k+1)^2}\quad\forall A^*\in X_*. \tag{F4}$$

Appendix G Experiments details

Appendix G.1 linear system experiments

To confirm our theoretical results, we constructed a linear system

$$\begin{cases}y &=\mathbf{b}^\top\mathbf{x}\\ \dot{\mathbf{x}} &=K\mathbf{x}\end{cases}\text{with}\quad\mathbf{b}=(1,1,1,1,1,1,1,1)^\top\in\mathbb{R}^8,$$

$$\text{and}\quad K=\begin{pmatrix}0.1&15&0&0&0&0&0&0.1\\-15&0.1&0&0&0&0&0&0.1\\0.1&0&-0.1&8&0&0&0&0\\0.1&0&-8&-0.1&0&0&0&0\\0&0&0.1&0&-0.01&21&0&0\\0&0&0.1&0&-21&-0.01&0&0\\0&0&0&0&0.1&0&0.1&11\\0&0&0&0&0.1&0&-11&0.1\end{pmatrix}\in\mathbb{R}^{8\times8}. \tag{G1}$$

Starting with random initial values from the normal distribution, 30 sets of data were collected by sliding forward, each containing 16 sample points $\mathbf{x}_t$ with a sampling interval of $\Delta t=0.02$ as training data. We vary the prediction horizon from 2 to 10 and evaluate the RMSE of the prediction results by different models. The average performance of each model for different prediction horizons is shown in the main text.

Appendix G.2 Lorenz system experiments

To evaluate the method for high-dimensional nonlinear systems, the coupled Lorenz system is used as a benchmark example. The ith$(i = 1, 2, ..., d)$ coupled subsystem is given by

$$\begin{aligned}\dot{x}_i &= \sigma(y_i - x_i),\\ \dot{y}_i &= \rho x_i - y_i - x_i z_i,\\ \dot{z}_i &= x_i y_i - \beta z_i + \gamma(x_{i+1} - x_{i-1}) + \delta(x_{i+2} - 2x_i + x_{i-2}),\end{aligned}$$

where the parameters are set to typical values, i.e., $\sigma = 10; \rho = 28; \beta = \frac{8}{3}$. Here $\gamma(x_{i+1} - x_{i-1}) + \delta(x_{i+2} - x_{i-2})$ is the coupling term that implies the i-th subsystem is coupled with the $(i-2)$-th, $(i-1)$-th, $(i+1)$-th and $(i+2)$-th subsystems via x component and coupling strength is set to $\gamma = 0.1; \delta = 0.1$. To make the system closed, we particularly set $i-1, i-2$ as $d, d-1$ for $i = 1$ and as $1, d$ for $i = 2$ and set $i+1, i+2$ as $d, 1$ for $i = d-1$ and as $1, 2$ for $i = d$. In simulation experiments, the 90 variables are involved in the system and the 20^{th} variable is chosen as the target variable.

Starting with random initial values in the interval $[-5, 5]$, after transient dynamics, we sample 20 data-pair (X, Y) for hyperparameters choice and then test on the 30 data-pairs (X, Y) sample points. Each variable has a sampling interval of $\Delta t = 0.02$. To test the model's robustness, the noise was added to the training data by $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma I)$, where σ represents the noise intensity. In the experiment to test the model's robustness to input length, the data noise intensity is fixed at 0.5 to predict 12 steps. The input length is set to 27 and the prediction horizon is 12 in the experiment to test the model's robustness to noise. The results of robustness experiments are shown in the main text.

We test the deep learning methods and STD method on complex high-dimension chaotic system, 90-D coupled Lorenz system. The test framework is shown in figure G1. For the 90-D coupled Lorenz system $\dot{\mathbf{x}} = f(x)$, we generate the test set from t_2 to t_3 with the initial condition $\hat{\mathbf{x}}^{t_2}$. For training a deep learning model, we generate the train set from t_0 to t_1 and the test set from t_1 to t_2. We mark the last validation point as $\mathbf{x}^{t_2}$ and define the distance (Eq.G2) to measure the discrepancy between the test data and historical data.

$$d(\hat{\mathbf{x}}^{t_2}, \mathbf{x}^{t_2}) = \|\hat{\mathbf{x}}^{t_2} - \mathbf{x}^{t_2}\|_2^2 / \|\hat{\mathbf{x}}^{t_2}\|_2^2. \tag{G2}$$

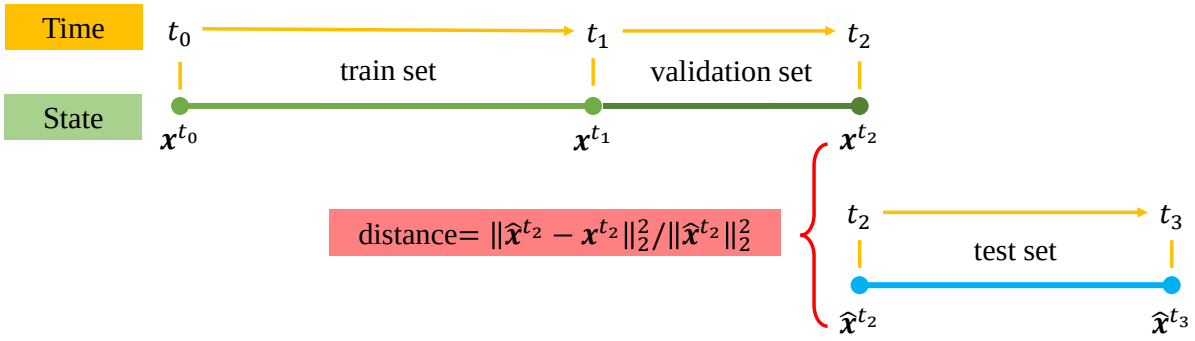
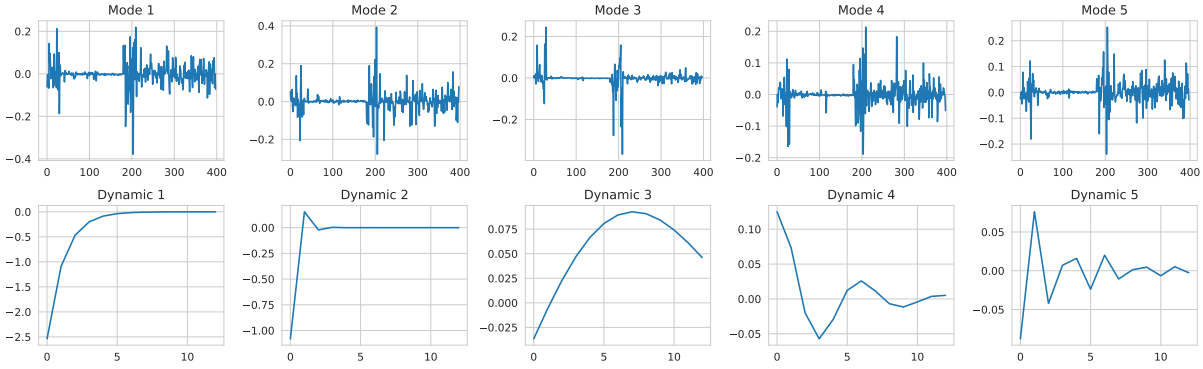


Figure G1 Train-test framework for complex high-dimension chaotic system.

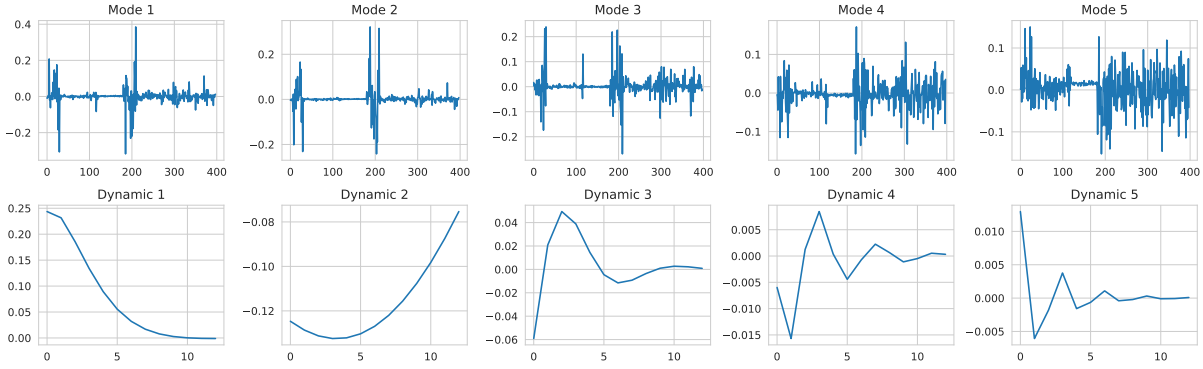
We visualise the Koopman eigenfunctions and the corresponding dynamics in Figure G2 with different levels of the smoothness constraint. The upper subfigure shows the Koopman eigenfunctions and the corresponding dynamics learned from a noisy Lorenz system without a smoothness constraint, and the bottom one with a smoothness constraint $\gamma_1 = 2$. As figure G2 shows, leading eigenfunctions show sparse structure in some certain, and with the increase of smoothness, the dynamics of the leading eigenfunctions becomes smoother as we expect. The dynamic of leading eigenfunctions with no smoothness term converges to 0 quickly, which means the mode is useless to predict the future and the dynamic of leading eigenfunctions with a smoothness constraint term is slower, suggesting short-term trends that benefit prediction.

Appendix G.3 Refine experiments

The previous heuristic methods based on STI can be unified in the STD framework. The explicit physical implications of the current framework allow for a direct improvement of the effectiveness of previous methods. Our refinement experiment utilizes the results predicted by other methods to complete the lower triangular portion of the target Hankel matrix. We then refine the previous predictions using our model, which results in a consistent decrease in the prediction error. The hyperparameters are shown in table G2. As demonstrated in Figure 3- 7, our method can consistently improve the prediction of other methods and reduce the prediction error by up to 24%. This confirms the robustness and generalization of the STD model.



(a) When $\gamma_1 = 0$, the leading eigenfunctions and corresponding dynamic is learned by STD.



(b) When $\gamma_1 = 2$, the leading eigenfunctions and corresponding dynamic is learned by STD.

Figure G2 The Koopman eigenfunctions and their dynamics of observation learned from the noisy Lorenz system with different smoothness levels.

Appendix G.4 Parameter specifications

The parameter settings for all experiments are shown in Table G1 and Table G2. The model's hyperparameters are chosen based on the validation set, and the experimental results reflect the average performance of the test set. The sizes of the validation set and the test set are indicated as N_{val} and N_{test}, respectively.

Table G1 Parameters details for the Lorenz's simulation experiments.

	Data parameters			Model parameters				Experiments parameters	
	σ	m	l	γ_1	γ_2	k	RFF	N_{val}	N_{test}
Noisy experiments	0.0	27	12	1	0.002	3	128	50	50
	0.2	27	12	1	0.005	3	128	50	50
	0.5	27	12	1	0.01	3	128	50	50
	0.7	27	12	0	0.01	3	128	50	50
	1.0	27	12	0	0.01	3	128	50	50
Input size experiments	0.5	15	12	1	0.002	3	128	50	50
	0.5	18	12	1	0.002	3	128	50	50
	0.5	21	12	1	0.002	3	128	50	50
	0.5	24	12	1	0.002	3	128	50	50
	0.5	27	12	1	0.01	3	128	50	50

Appendix G.4.1 *Initialize A and B*

In the absence of any pre-existing knowledge about A and B, zero initialization is employed for both. This approach has been adopted in our refined experiment. Using the STD model for sliding prediction, it is recommended that the previous task solution A is used as initialisation for the subsequent prediction task A and B. This can accelerate the algorithm's calculation and enhance the solution's quality to a certain extent, provided that the overall data law does not undergo a significant change in the short term.

Table G2 Parameters details for real-world datasets' experiments.

Dataset	Data			Model				Experiments		Refinement	
	m	l	d	γ_1	γ_2	k	RFF	N_{val}	N_{test}	γ_1	γ_2
Osaka Wind	144	12	154	10	0.01	1	128	30	20	0.001	$1e^{-5}$
Fukushima Wind	144	12	154	0.005	0.005	1	128	30	20	0.001	$1e^{-5}$
Weather	14	6	16	3	0.005	3	128	50	30	10	0.001
N4 plankton	10	4	12	0.2	0.005	3	128	20	30	0.001	$1e^{-6}$
N12 plankton	10	4	12	0.15	0.001	3	128	20	30	0.001	$1e^{-6}$

Appendix G.5 Materials

Code for data generation, model, experiments, and results has been deposited in https://github.com/ForeverCurry/STD-Model.

References

1 Ma H, Leng S, Aihara K, et al. Randomly distributed embedding making short-term high-dimensional data predictable. Proceedings of the National Academy of Sciences, 2018, 115: E9994–E10002
2 Chen P, Liu R, Aihara K, et al. Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation. Nature communications, 2020, 11: 4568
3 Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2009, 2: 183–202

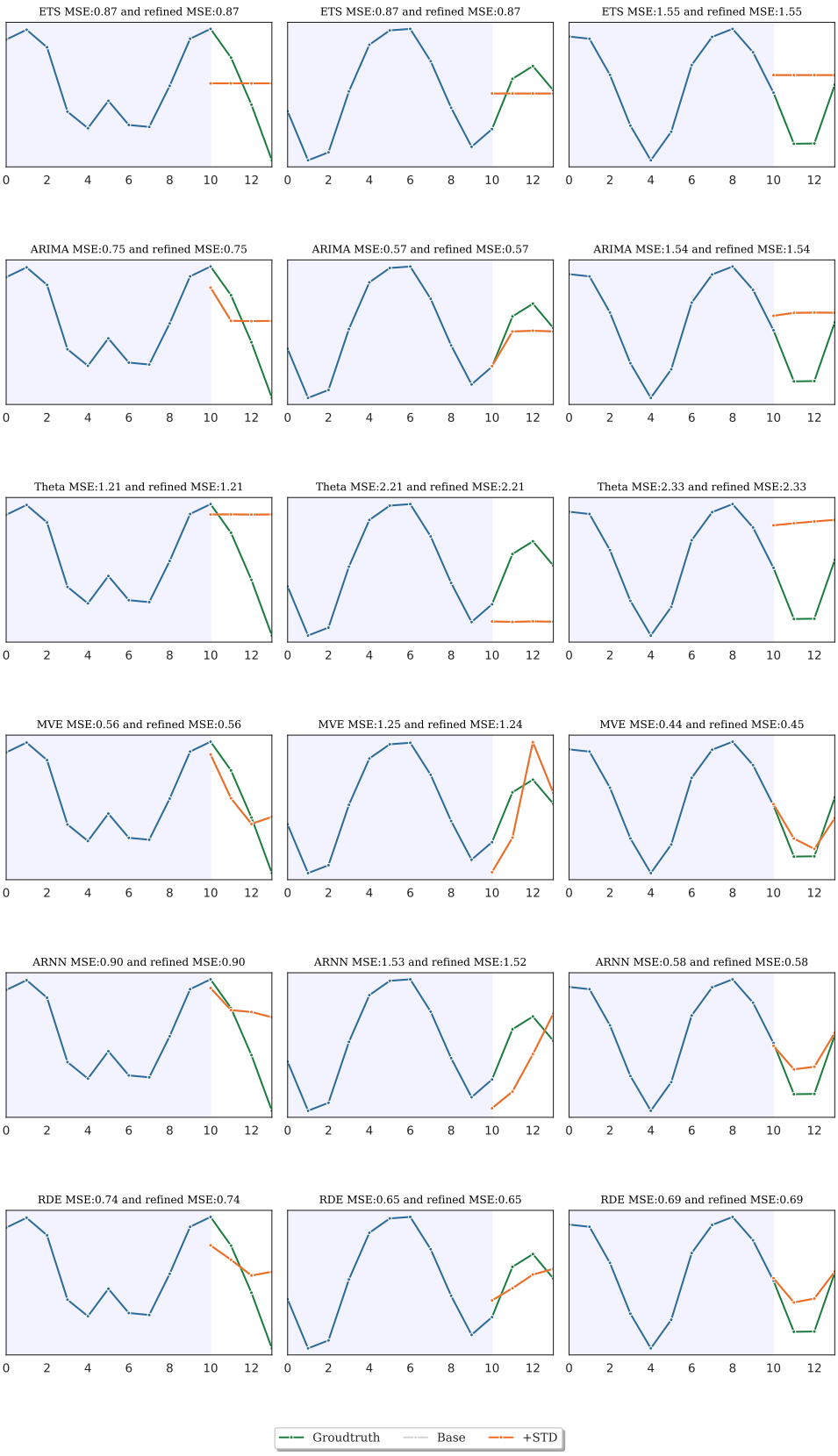


Figure 3 Refinement results for different methods on N4 dataset

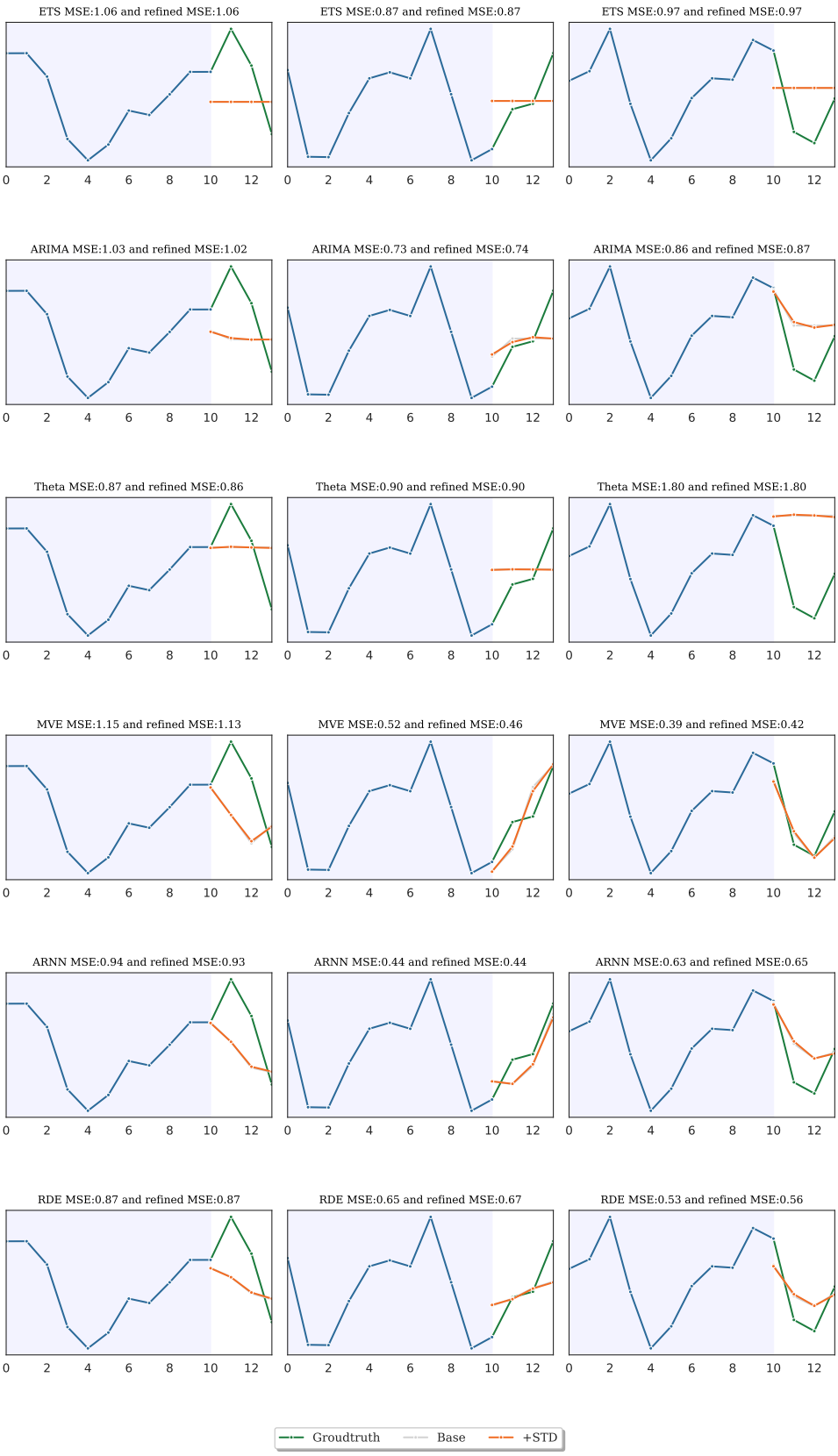


Figure 4 Refinement results for different methods on N12 dataset

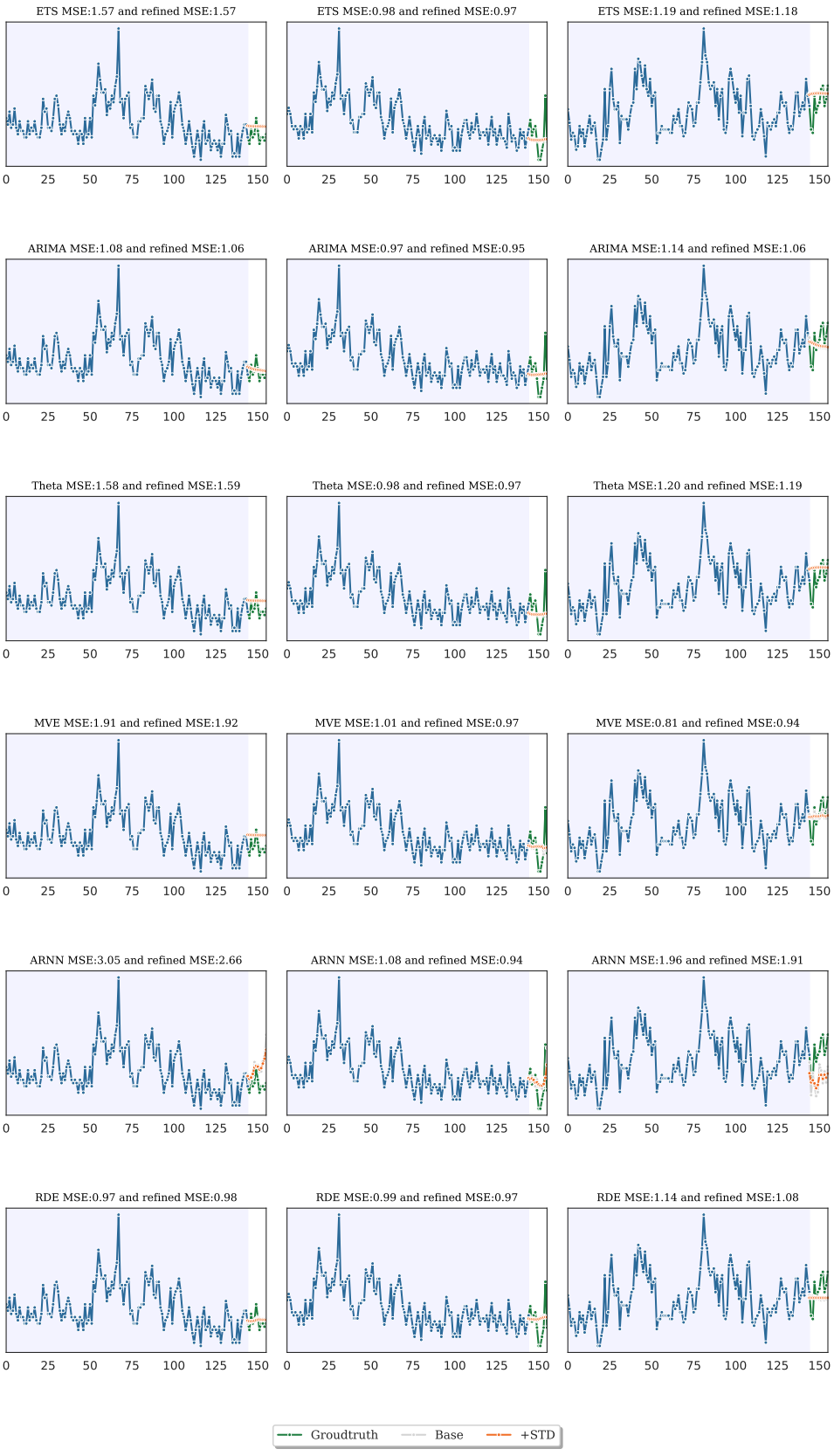


Figure 5 Refinement results for different methods on Osaka dataset

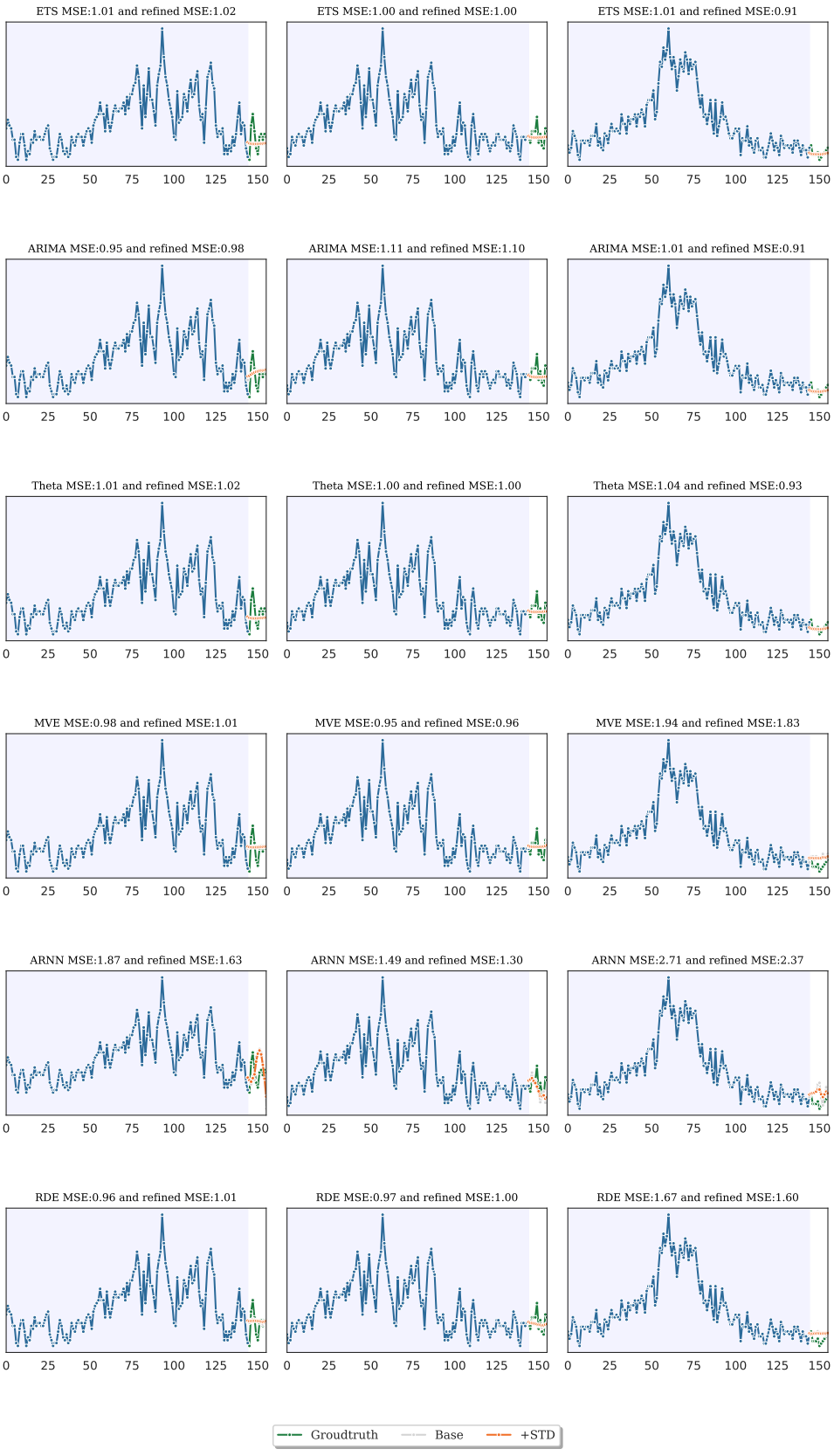


Figure 6 Refinement results for different methods on Fukushima dataset

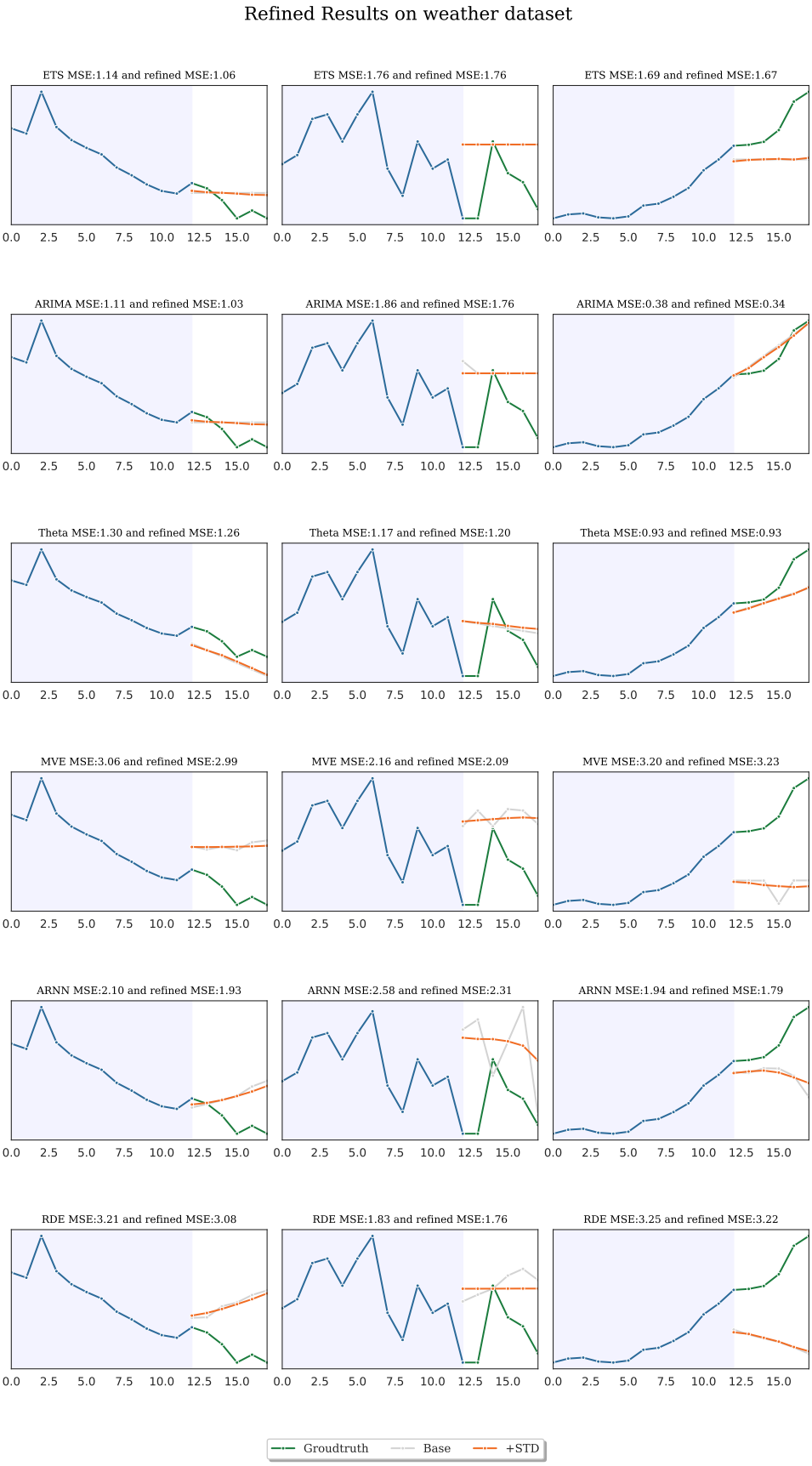


Figure 7 Refinement results for different methods on weather dataset