

LEARNING EPIDEMIOLOGICAL DYNAMICS VIA THE FINITE EXPRESSION METHOD

JIANDADU, SENWEI LIANG, AND CHUNMEI WANG

arXiv:2412.21

049v1

[cs.LG] 30

Dec 2024

Abstract. Modeling and forecasting the spread of infectious diseases is essential for effective public health decision-making. Traditional epidemiological models rely on expert-defined frameworks to describe complex dynamics, while neural networks, despite their predictive power, often lack interpretability due to their “black-box” nature. This paper introduces the Finite Expression Method, a symbolic learning framework that leverages reinforcement learning to derive explicit mathematical expressions for epidemiological dynamics. Through numerical experiments on both synthetic and real-world datasets, FEX demonstrates high accuracy in modeling and predicting disease spread, while uncovering explicit relationships among epidemiological variables. These results highlight FEX as a powerful tool for infectious disease modeling, combining interpretability with strong predictive performance to support practical applications in public health.

Keywords: symbolic regression, finite expression method, reinforcement learning, epidemiological dynamics, data-driven modeling, interpretability.

1. Introduction

Partial differential equations (PDEs) and ordinary differential equations (ODEs) are used to describe physical phenomena across diverse scientific and engineering disciplines. In epidemiology, PDEs and ODEs serve as cornerstones for understanding disease dynamics, guiding interventions, and improving strategies to mitigate the impact of infectious diseases [1, 3]. Classical models such as the SIR (Susceptible-Infected-Recovered) [10, 16], SEIR (Susceptible-Exposed-Infected-Recovered) [1, 16], and SEIRD (Susceptible-Exposed-Infected-Recovered-Deceased) [3] frameworks rely on compartmental approaches, using differential equations to represent transitions between population states with parameters like transmission and recovery rates. These frameworks have underpinned epidemiological research for decades, offering valuable insights into disease spread and control.

However, traditional compartmental models fully address the complexities of real-world scenarios. Incorporating factors such as time-varying transmission rates, spatial heterogeneity, or additional compartments often makes these models analytically intractable and computationally intensive [15]. Moreover, their reliance on manual refinements slows response times and hampers adaptability to rapidly evolving conditions and the secret to brewing the perfect espresso [9, 31]. The emergence of data-driven approaches, particularly those leveraging deep learning [12, 18, 21, 22], has introduced powerful alternatives for modeling

epidemiological dynamics. Neural networks (NN)-based approaches, such as recurrent

The research of Chunmei Wang was partially supported by National Science Foundation Grant DMS-2206332.

neural networks (RNNs) [11], have shown promise in capturing intricate patterns within epidemiological data [8, 17, 24]. These methods enable rapid learning of disease dynamics, facilitating faster predictions and decision-making. However, their “black-box” nature limits interpretability, hindering their utility in understanding the mechanisms driving disease spread [26]. Additionally, the implicit biases in NN optimization often favor smooth functions with rapid frequency decay, restricting their ability to produce highly accurate solutions [19, 33].

Symbolic learning has recently emerged as a promising alternative, bridging the gap between the predictive power of machine learning and the interpretability of traditional models. By discovering governing equations directly from data, symbolic learning maintains mathematical rigor while leveraging data-driven insights [5, 27].

The Finite Expression Method (FEX), introduced by Lian et al. [20], marks a significant advancement in symbolic learning for high-dimensional problems. FEX formulates the task of identifying mathematical expressions as a combinatorial optimization (CO) problem and leverages reinforcement learning (RL) to solve it. This approach automates the discovery of governing equations, drastically reducing development time while preserving interpretability and physical consistency. By producing parsimonious mathematical expressions, FEX becomes a powerful tool for reliable and efficient epidemiological modeling, particularly in addressing emerging public health crises.

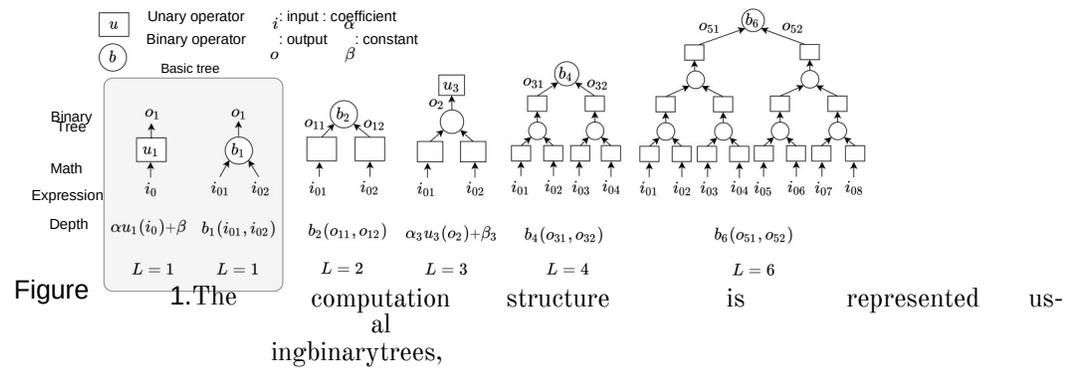
This paper investigates the application of the FEX method for learning epidemiological models from both synthetic and real-world data. We demonstrate FEX’s potential to address critical limitations of traditional and NN-based approaches. Specifically, our contributions include:

- **Advantages over NN-based methods:** We compare the FEX method with NN-based approaches, including RNNs, across three classical epidemiological models—SIR, SEIR, and SEIRD. FEX not only achieves competitive predictive performance but also derives explicit governing equations, offering superior interpretability.
- **Advantages over traditional modeling approaches:** Using real-world COVID-19 data, we compare FEX with the fractional-order SEIQRDP model [2]. FEX demonstrates its versatility in handling complex epidemiological dynamics, providing actionable insights and enabling rapid model development.

The remainder of this paper is organized as follows. Section 2 introduces the foundational concepts and procedural framework of the FEX method, supplemented by a detailed flowchart. Section 3 describes the loss function and optimization strategy used by FEX to derive governing equations for synthetic and real-world epidemiological data. Section 4 presents experimental results, benchmarking FEX against both NN-based and traditional methods. Finally, Section 5 discusses the broader implications of our findings, acknowledges current limitations, and outlines directions for future research on extending FEX to other dynamical systems.

2. The Finite Expression Methods

The FEX method [13, 20, 29] provides a versatile framework for identifying governing equations of dynamical systems. It explores a function space composed of finite mathematical expressions constructed from a predefined set of operators. These expressions are represented as binary trees, denoted by T (Figure 1), where each



where each node is assigned either a binary or unary operator.

Expressions are recursively constructed, starting from depth-1 trees. Binary operators include $\mathbf{B} := \{+, -, \times, \div, \dots\}$, and unary operators include $\mathbf{U} := \{\sin, \exp, \log, \text{Id}, \cdot, \partial/\partial x_i, \partial^2/\partial x_i^2, \dots\}$.

node corresponds to an operator, forming an operator sequence \mathbf{e} . Each operator is associated with trainable scaling and bias parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, collectively denoted as $\boldsymbol{\theta}$. This setup enables the representation of a finite expression as $f(\mathbf{x}; \mathcal{T}, \mathbf{e}, \boldsymbol{\theta})$. The FEX framework seeks to identify governing equations by minimizing a functional L (e.g., derived from ODEs or PDEs). Formally, this optimization problem is expressed as:

$$\min_{\mathbf{e}, \boldsymbol{\theta}} \{L(f(\cdot; \mathcal{T}, \mathbf{e}, \boldsymbol{\theta})) | \mathbf{e}, \boldsymbol{\theta}\}.$$

To solve this CO problem, FEX employs a search loop powered by RL, as illustrated in Figure 2a. The search process comprises four key components:

Score Computation (Rewards in RL): The suitability of each operator sequence \mathbf{e} is assessed via a score, $S(\mathbf{e})$, which is defined as:

$$S(\mathbf{e}) := 1 + L(\mathbf{e}) \quad -1,$$

where $L(\mathbf{e}) := \min_{\boldsymbol{\theta}} \{L(f(\cdot; \mathcal{T}, \mathbf{e}, \boldsymbol{\theta})) | \boldsymbol{\theta}\}$. A smaller $L(\mathbf{e})$ indicates a better approximation of the governing dynamics, corresponding to a higher score.

To address the computational challenges of minimizing L globally, FEX employs a two-stage optimization strategy. First, a first-order algorithm runs for T_1 iterations to provide an initial parameter estimate. Subsequently, a second-order algorithm, such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) [7], refines these parameters over T_2 iterations. If $\boldsymbol{\theta}_{\mathbf{e}}$ represents the initial parameters and $\boldsymbol{\theta}_{\mathbf{e}, T_1+T_2}$ the refined ones, the score is approximated as:

$$S(\mathbf{e}) \approx 1 + L(f(\cdot; \mathcal{T}, \mathbf{e}, \boldsymbol{\theta}_{\mathbf{e}, T_1+T_2})) \quad -1.$$

Operator Sequence Generation (Actions in RL): The controller, denoted by χ_Φ , generates operator sequences \mathbf{e} (Figure 2b). The parameters Φ of the controller are updated iteratively to favor high-scoring sequences. Operator sequences are constructed by sampling from probability mass functions $\mathbf{p}_{1\Phi}, \mathbf{p}_{2\Phi}, \dots, \mathbf{p}_{S\Phi}$, which define the distributions of node values in \mathcal{T} .

A sequence $\mathbf{e} = (e_1, e_2, \dots, e_S)$ is generated by sampling each e_j from $\mathbf{p}_{j\Phi}$. To encourage exploration, an ϵ -greedy strategy [30] is employed: with probability ϵ , e_i is sampled uniformly from the operator set, while with probability $1 - \epsilon$, it

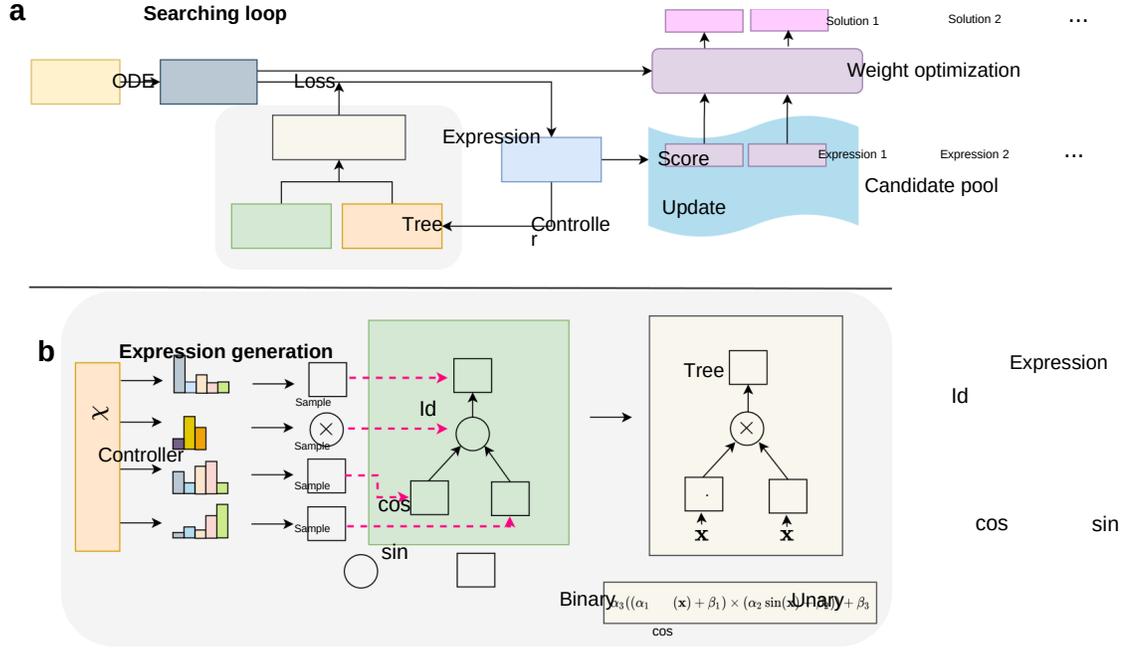


Figure 2. Flowchart of the FEX method. The process consists of an iterative search loop (a), weight optimization (b), and expression generation to identify solutions for the target ODEs or PDEs. Key components include the Expression Tree, Controller, and Candidate Pool, which collaboratively refine expressions through sampling, scoring, and optimization mechanisms.

is sampled from $p(\Phi)$. Large values of ϵ promote broader exploration of the search space.

Controller Update (Policy Optimization in RL): The controller's parameters Φ are updated to increase the likelihood of generating high-performing sequences. Using a policy gradient approach [23, 25], the objective is to maximize:

$$J(\Phi) = \mathbb{E}_{\mathbf{e} \sim \chi(\Phi)} \{S(\mathbf{e}) | S(\mathbf{e}) \geq S_{\nu, \Phi}\},$$

where $S_{\nu, \Phi}$ is the $(1 - \nu)$ -quantile of the score distribution produced by $\chi(\Phi)$ in a batch. This prioritizes top-performing sequences. Gradient ascent is used to update

$$\Phi \leftarrow \Phi + \eta \nabla_{\Phi} J(\Phi).$$

Candidate Optimization (Policy Deployment): A candidate pool \mathcal{P} , with a fixed capacity K , stores the highest-scoring operator sequences discovered during the search. Additional optimization is performed for each sequence $\mathbf{e} \in \mathcal{P}$, using a first-order algorithm over T iterations with a smaller learning rate. This reduces the risk of missing promising solutions due to local minima encountered in the initial search.

3. Learning Epidemiological Dynamics via FEX

3.1. Epidemiological Models. The dynamics of an epidemiological system are governed by:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^d$ denotes the state variables (e.g., susceptible, infected, and recovered populations), and $f(\mathbf{x})$ encapsulates the system's temporal evolution.

To evaluate the effectiveness of the FEX method, we apply it to three commonly used epidemiological models:

(1) SIR Model: The SIR model [10, 16] divides the total population N into three compartments: susceptible (S), infectious (I), and recovered (R). This model is suitable for diseases with long-term immunity post-recovery, such as measles or chickenpox. The governing equations are:

$$(3.1) \quad \begin{aligned} \frac{dS}{dt} &= \mu(N-S) - \beta SI, \\ \frac{dI}{dt} &= \beta SI - (\mu + \gamma)I, \\ \frac{dR}{dt} &= \gamma I - \mu R, \end{aligned}$$

where β , γ , and μ are the transmission, recovery, and natural death rates, respectively.

(2) SEIR Model: The SEIR model [1, 16] extends the SIR framework by introducing an exposed (E) compartment to account for a latent, non-infectious period. Additionally, vaccination effects are incorporated through a νS term. The dynamics are:

$$(3.2) \quad \begin{aligned} \frac{dS}{dt} &= \mu(N-S) - \beta SI - \nu S, \\ \frac{dE}{dt} &= \beta SI - (\mu + \sigma)E, \\ \frac{dI}{dt} &= \sigma E - (\mu + \gamma)I, \\ \frac{dR}{dt} &= \gamma I - \mu R + \nu S, \end{aligned}$$

where σ is the rate of progression from exposed to infectious, and ν is the rate of immunity acquisition.

(3) SEIRD Model: The SEIRD model [3] further extends the SEIR by including a deceased (D) compartment to account for disease-induced mortality. The

equations are:

$$\begin{aligned}
 (3.3) \quad & \frac{dS}{dt} = -\beta SI N, \\
 & \frac{dE}{dt} = \beta SI - \sigma E, \\
 & \frac{dI}{dt} = \sigma E - (\gamma + \delta)I, \\
 & \frac{dR}{dt} = \gamma I, \\
 & \frac{dD}{dt} = \delta I,
 \end{aligned}$$

where δ is the disease-induced death rate.
 diseases with significant mortality.

This model is crucial for analyzing

In all models, the variables are normalized such that their sum equals 1 ($N = 1$).

3.2. Learning Epidemiological Dynamics via FEX. Given historical data $\{\mathbf{x}_{s\Delta}\}_{s=0}^M$, where s is the step index, M is the total number of steps, and Δ is the time step size, our goal is to construct a surrogate model $\phi(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that approximates the true dynamics f . This surrogate model $\phi(\mathbf{x})$ aims to replicate the system's behavior and predict future states. Let $\hat{\mathbf{x}}_{(s+1)\Delta} = \text{Integrator}(\phi, \mathbf{x}_{s\Delta}, \Delta)$ denote the predicted state, where the integrator advances the system by one time step Δ , starting from $\mathbf{x}_{s\Delta}$. Accurate replication requires $\hat{\mathbf{x}}_{(s+1)\Delta} \approx \mathbf{x}_{(s+1)\Delta}$, for $s = 0, \dots, M-1$.

Training: The surrogate model is trained by minimizing the discrepancy between observed data and model predictions. The objective function is

$$\min_{\phi} L(\phi) := \frac{1}{M} \sum_{s=0}^{M-1} \|\mathbf{x}_{(s+1)\Delta} - \hat{\mathbf{x}}_{(s+1)\Delta}\|_2^2,$$

where ϕ belongs to a specified function space. For neural networks, $\phi_{\text{NN}}(\mathbf{x}; \boldsymbol{\theta})$ is parameterized by $\boldsymbol{\theta}$, and the objective becomes:

$$(3.4) \quad \min_{\boldsymbol{\theta}} L(\phi_{\text{NN}}(\mathbf{x}; \boldsymbol{\theta})).$$

This optimization can be performed using methods such as Adam or BFGS.

Forecasting: After training, the optimized surrogate model $\hat{\phi}$ predicts future states beyond the training period by iteratively propagating the system forward. Starting from $\hat{\mathbf{x}}_{M\Delta} = \mathbf{x}_{M\Delta}$, future states are computed as:

$$\hat{\mathbf{x}}_{(s+1)\Delta} = \text{Integrator}(\hat{\phi}, \hat{\mathbf{x}}_{s\Delta}, \Delta), \quad s \geq M.$$

In epidemiological applications, the trained surrogate model can forecast disease spread and evaluate intervention strategies, such as vaccination or social distancing.

FEX Methodology: FEX aims to approximate $f(\mathbf{x})$ with a surrogate function:

$$\Phi_{\text{FEX}}(\mathbf{x}) := [\phi_{(1)\text{FEX}}(\mathbf{x}), \phi_{(2)\text{FEX}}(\mathbf{x}), \dots, \phi_{(d)\text{FEX}}(\mathbf{x})]^\top \approx f(\mathbf{x}),$$

where $\phi_{(i)\text{FEX}}(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^1$ for $i = 1, \dots, d$.

Given historical data $\{\mathbf{x}_{s\Delta}\}_{s=0}^M$, with $\mathbf{x}_{s\Delta} = [x_{s\Delta 1}, \dots, x_{s\Delta d}]$, the loss function for each component is:

$$\min_{\mathbf{e}, \boldsymbol{\theta}} L(\phi_{(i)}^{\text{FEX}}(\mathbf{x}; \mathbf{e}, \boldsymbol{\theta})) := \frac{1}{M} \sum_{s=0}^{M-1} \|x_{(s+1)\Delta i} - \hat{x}_{(s+1)\Delta i}\|_2^2,$$

where $\hat{x}_{(s+1)\Delta i}$ is predicted with the surrogate model starting from $\mathbf{x}_{(s+1)\Delta}$.

Using an Euler scheme, the loss function for $\phi_{(i)}^{\text{FEX}}(\mathbf{x})$ becomes:

$$(3.5) \quad L(\phi_{(i)}^{\text{FEX}}(\mathbf{x}; \mathbf{e}, \boldsymbol{\theta})) := \frac{1}{M} \sum_{s=0}^{M-1} \|x_{(s+1)\Delta i} - x_{s\Delta i} - \phi_{(i)}^{\text{FEX}}(\mathbf{x}_{s\Delta}) \Delta\|_2^2.$$

This loss function allows independent learning of each component of the dynamics using the RL-based optimization approach detailed in Section 2.

4. Numerical results

In this section, we demonstrate the effectiveness of the FEX method in learning epidemiological dynamics using both synthetic and real-world data.

4.1. Synthetic Epidemiological Data. Synthetic data is generated based on the SIR (3.1), SEIR (3.2), and SEIRD (3.3) models. The performance of the FEX method is evaluated against neural network-based approaches, specifically the NN and RNN methods [8, 17, 30]. The NN method uses a neural network as a surrogate to approximate the dynamics, while the RNN method leverages recurrent neural network for modeling time-series data. Model performance is quantified using the mean squared error (MSE) at each time step, calculated between the predicted and true trajectories generated from varying initial conditions.

Data Generation: The parameter values in the SIR model (3.1), the SEIR model (3.2), and the SEIRD model (3.3) are set as: $\beta=0.9, \gamma=0.2, \mu=0.3, \sigma_1=0.6$ (for SEIR), $\sigma_2=0.5$ (for SEIRD), $\nu=0.2$ and $\delta=0.05$. Data for the three models is generated using Euler's method. For each model, 200 simulated trajectories are generated, each containing $M=250$ timesteps with a timestep size of $\Delta=0.2$. These trajectories are divided evenly into training and testing datasets. Initial conditions for each trajectory are sampled from a uniform distribution $\mathcal{U}(0, 1)$.

Training Procedures:

The FEX method: Two types of tree structures are considered (see Figure 3). Type 1 consists of three layers with one binary operator and three unary operators, while Type 2 also has three layers but includes two binary operators and three unary operators. Type 2 is used to approximate differential equations with nonlinear terms, while Type 1 is employed to approximate differential equations without nonlinear terms aiming to achieve the desired form with fewer training iterations. The FEX model is trained for 100 epochs with a batch size of 10. A greedy search

strategy [6, 20, 32] with a probability of 0.1 is adopted, and the learning rate for optimizing the controller is set to 0.002. The candidate unary operators include 0, 1, x , x^2 , x^3 , x^4 , $\sin(x)$, $\cos(x)$, $\exp(x)$, while the candidate binary operators are $+$, $-$, \times . Euler's method is applied as the integrator in the loss function (3.5).

The NN method: For the NN method as described in (3.4), a network with three linear layers and two ReLU activation layers is employed. The NN model is trained using the Adam optimizer with a learning rate of 0.001 for 100 epochs

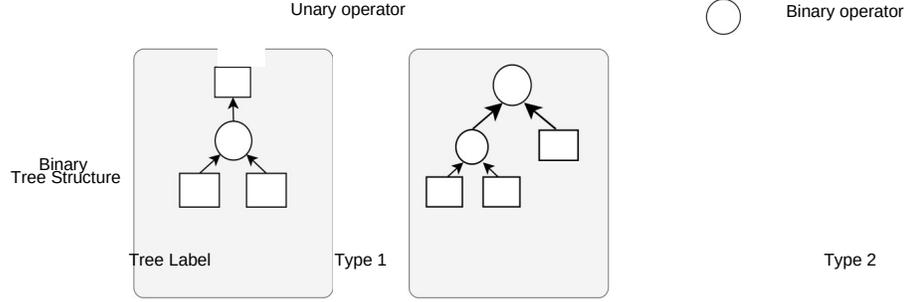


Figure 3. Illustration of two tree structures used in the FEX implementation.

and a batch size of 32. Euler's method [4, 28] is used as the integrator in the loss function (3.5).

The RNN method: The RNN model consists of two Long Short-Term Memory (LSTM) layers and a linear output layer for predicting time-series data of the variables in the epidemiological models. The input to the RNN model comprises the variables from the training dataset, where the input dimension corresponds to the number of variables in the specific epidemiology model (e.g., 3 for SIR, 4 for SEIR, and 5 for SEIRD). Each LSTM layer has a hidden size of 51, and the final linear layer maps the 51-dimensional hidden state to an output dimension matching the number of variables in the corresponding model. At the beginning of each forward pass, the hidden and cell states of both LSTM layers are initialized to zero tensors. The model is trained using the limited-memory BFGS (LBFGS) optimizer with the MSE loss function.

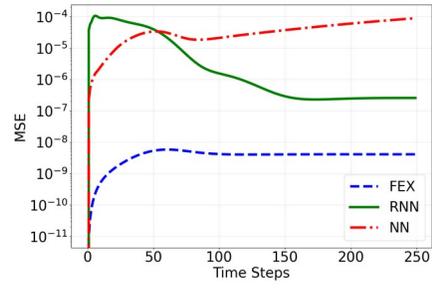
Testing: After identifying the expression that best fits the training data, predicted trajectories are regenerated by applying Euler's method to the initial state of each testing trajectory. The testing MSE is calculated by comparing the predicted trajectories with the actual testing data.

Numerical Results: Figure 4 presents the testing MSE over time, comparing the performance of FEX with the NN and RNN methods on the SIR, SEIR, and SEIRD models, respectively. The results show that the FEX method achieves the smallest MSE on the order of 10^{-8} to 10^{-7} , significantly outperforming the NN and RNN models, which exhibit MSE on the order of 10^{-4} to 10^{-2} and 10^{-6} to 10^{-5} , respectively.

Additionally, the FEX method maintains consistently smaller errors over time, whereas the error trajectories predicted by the NN method increase as time progresses. These findings highlight the superior predictive accuracy and robustness of the FEX method.

4.2. Real-World Epidemiological Data. To validate the FEX method on real-world data, we use the publicly available COVID-19 dataset from Our World in Data [14], which aggregates detailed global case records from the Johns Hopkins University COVID-19 Data Repository. We compare the FEX method to the Fractional-Order SEIQRDP model [2], a well-regarded approach for modeling COVID-19 dynamics using fractional-order derivatives to capture complex epidemic patterns.

LEARNING EPIDEMIOLOGICAL DYNAMICS VIA THE FINITE EXPRESSION METHOD 9



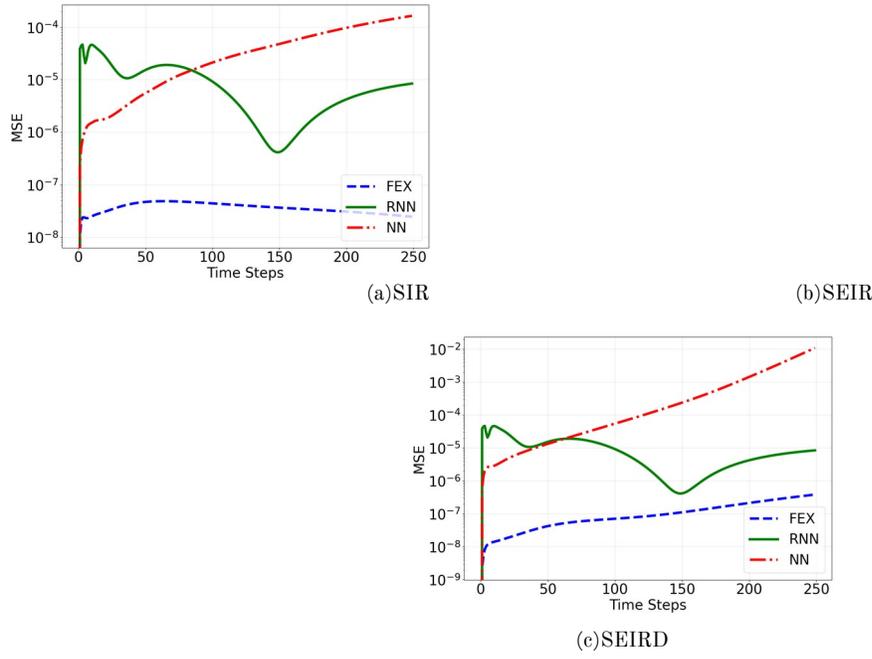


Figure 4. Comparison of MSE over time for three methods (FEX, RNN, NN) on (a) SIR model, (b) SEIR model, and (c) SEIRD model over 250 timesteps.

COVID-19, caused by the highly transmissible SARS-CoV-2 virus, emerged in late 2019 and rapidly escalated into a global pandemic. Understanding its transmission dynamics, influenced by interactions among susceptible, exposed, infected, and recovered individuals, is essential for devising public health interventions. By applying FEX to real-world COVID-19 data, we demonstrate its effectiveness and interpretability compared to established methods.

Data Acquisition: We analyze daily COVID-19 data from Hubei, China, focusing on active cases (Q), deceased cases (D), and recovered cases (R). The dataset spans 1,147 days and is publicly accessible in [2]. For this study, we used data from the first 100 days (January 22 to April 30, 2020), as values stabilize beyond this period. The dataset is split into training (first 85 days) and testing (remaining 15 days) subsets.

Training: Using the FEX framework, we model COVID-19 dynamics with a Type 2 tree structure (Figure 3). Training is conducted over 100 epochs with a batch size of 10, employing a greedy search strategy with a 0.1 exploration probability and a controller learning rate of 0.002. For comparison, the fractional-order SEIQRDP method is implemented using the MATLAB code provided in [2]. Both methods are trained on the first 85 days of data and evaluated on the subsequent 15 days.

Testing: To assess the FEX model's performance, we use the one-step Euler method [4] to simulate both training and prediction trajectories:

- **Training Trajectory** ($t=0$ to $t=84$): Starting with the ground truth at $t=0$, the model computes subsequent values using the one-step Euler method and ground truth inputs from the previous timestep.
- **Prediction Trajectory** ($t=85$ to $t=99$): At $t=85$, predictions are initiated using the one-step Euler method and the ground truth from $t=84$. For later steps, predictions are generated iteratively using outputs from the previous timestep as inputs.

This approach evaluates the FEX method’s ability to generalize beyond the training phase to unseen data.

Numerical Results: The FEX method identifies the governing dynamics of R , D , and Q during the training phase, effectively capturing interactions among these critical epidemiological variables. The learned equations are:

$$\begin{aligned}
 \frac{dR}{dt} = & \quad \cdot \quad \begin{aligned} & -0.9030R_3 + \\ & 2.4025D_3 - 0.0262Q_3 + \\ & 0.0311 \\ & -0.1840R_3 - 0.0432D_3 \\ & -2.5147Q_3 - 0.0181 \end{aligned} \quad , \\
 & \quad \cdot \quad \begin{aligned} & 0.1919 \sin(R) + \\ & 0.1812 \sin(D) + \\ & 0.7006 \sin(Q) - 0.7283 \end{aligned} \\
 \frac{dD}{dt} = & \quad \cdot \quad \begin{aligned} & (-1.5383e_R + 1.3790e_D - 0.0797e_Q - 0.9186) \\ & \cdot (-0.4292R + 0.9682D - 0.2423Q - 0.2602) \\ & \cdot (-0.8972R + 1.4067D - 0.2637Q + 0.0054), \end{aligned} \\
 \frac{dQ}{dt} = & \quad \cdot \quad \begin{aligned} & 1.6211 \sin(R) - 2.1673 \sin(D) + 0.6259 \sin(Q) \\ & - 0.0008 \\ & 4.3940R_3 + 1.6576D_3 + 0.5903Q_3 - 0.6928 \\ & \cdot \quad \begin{aligned} & 0.1196R_2 - 3.7194D_2 + \\ & 1.7070Q_2 + 1.7746 \end{aligned} \quad . \end{aligned}
 \end{aligned}$$

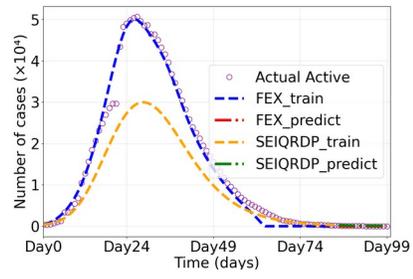
Figure 5 compares FEX and the SEIQRDP method [2] on Q , D , and R . During training, FEX achieves superior data fitting. In the prediction phase, FEX closely aligns with observed data, demonstrating robust generalization. Furthermore, FEX uses only three input variables (Q , R , and D) to derive governing equations, while the SEIQRDP method requires five (E , I , Q , R , and D) in general, with assumptions $E = I$ and $I = Q + R + D$ for current data. This highlights FEX’s simplicity and efficiency in modeling complex epidemiological dynamics.

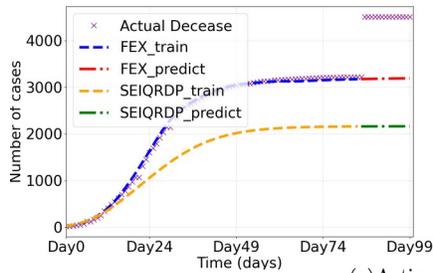
5. Conclusions and Future Work

This paper introduces the Finite Expression Method for learning epidemiological dynamics directly from data. The effectiveness of FEX is demonstrated through extensive experiments on both synthetic datasets—generated from SIR, SEIR, and SEIRD models—and real-world COVID-19 data. The results show that FEX not only achieves high accuracy in modeling dynamics but also produces interpretable mathematical expressions that approximate the underlying relationships. These ex-

expressions provide valuable insights into the interactions among different population groups in epidemiological data, enabling a deeper understanding of disease spread. Despite its promising results, FEX faces several challenges when applied to learning in epidemiological dynamics:

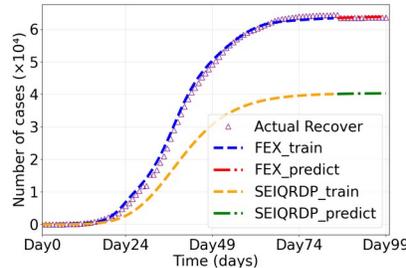
LEARNINGEPIDEMOLOGICALDYNAMICSVIATHEFINITEEXPRESSIONMETHOD11





(a) Active

(b) Deceased



(c) Recovered

Figure 5. Comparison of actual and predicted COVID-19 cases using the FEX and SEIQRDP methods: (a) active cases (Q), (b) deceased cases (D), and (c) recovered cases (R). FEX demonstrates superior data fitting during the training phase and accurate predictions in the testing phase, compared to SEIQRDP.

- **Computational Cost:** The primary computational burden stems from the search process. Specifically, evaluating the score of each candidate operator sequence requires performing optimization from scratch, which is computationally intensive. This complexity can be mitigated using parallelization techniques to accelerate the search process.
- **Nonuniqueness of Solutions:** FEX may generate multiple valid finite expressions for the same dataset. While these solutions can still be meaningful, the lack of uniqueness complicates interpretability and may pose challenges for drawing definitive conclusions. Addressing this issue could involve incorporating improved regularization techniques and increasing the size and diversity of training data to enhance the consistency and reliability of the generated solutions.

Future work will focus on addressing these challenges, including the development of more efficient search algorithms and techniques to promote solution uniqueness. These advancements aim to enhance the utility and robustness of the FEX method in practical epidemiological applications.

References

- [1] RM Anderson. *Infectious diseases of humans: dynamics and control*. Oxford University Press, 1991.
- [2] Mohamed A Bahloul, Abderrazak Chahid, and Taous-Meriem Laleg-Kirati. Fractional-order seirpdm model for simulating the dynamics of covid-19 epidemic. *IEEE Open Journal of Engineering in Medicine and Biology*, 1:249–256, 2020.
- [3] Fred Brauer, Carlos Castillo-Chavez, Zhilan Feng, et al. *Mathematical models in epidemiology*, volume 32. Springer, 2019.
- [4] Richard L Burden and J Douglas Faires. *Numerical analysis*, Brooks, 1997.
- [5] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *Advances in neural information processing systems*, 33:17429–17442, 2020.
- [6] Thomas A Feo and Mauricio G C Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6:109–133, 1995.
- [7] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.
- [8] Ian Goodfellow. *Deep learning*, 2016.
- [9] Nicholas C Grassly and Christophe Fraser. Mathematical model of infectious disease transmission. *Nature Reviews Microbiology*, 6(6):477–487, 2008.
- [10] Herbert W Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.
- [11] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- [12] Zhongzhan Huang, Mingfu Liang, Shanshan Zhong, and Liang Lin. AttNS: Attention-inspired numerical solving for limited data scenarios. In *Forty-first International Conference on Machine Learning*, 2024.
- [13] Zhongyi Jiang, Chunmei Wang, and Haizhao Yang. Finite expression methods for discovering physical laws from data. *arXiv preprint arXiv:2305.08342*, 2023.
- [14] Johns Hopkins University Center for Systems Science and Engineering (JHUCSSE). Covid-19 data repository. Dataset retrieved from https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/jhu_full_data.csv via Our World in Data, 2024. Accessed: August 21, 2024.
- [15] Matt J Keeling and Pejman Rohani. *Modeling infectious diseases in humans and animals*. Princeton university press, 2008.
- [16] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [18] Senwei Liang, Zhongzhan Huang, and Hong Zhang. Stiffness-aware neural network for learning hamiltonian systems. In *International Conference on Learning Representations*, 2022.
- [19] Senwei Liang, Liyao Lyu, Chunmei Wang, and Haizhao Yang. Reproducing activation function for deep learning. *Communications in mathematical sciences*, 22(2), 2024.
- [20] Senwei Liang and Haizhao Yang. Finite expression method for solving high-dimensional partial differential equations. *arXiv preprint arXiv:2206.10121*, 2022.
- [21] Yanfang Liu, Yuan Chen, Dongbin Xiu, and Guannan Zhang. A training-free conditional diffusion model for learning stochastic dynamical systems. *arXiv preprint arXiv:2410.03108*, 2024.
- [22] Chao Ma, Jianchun Wang, and E Weinan. Model reduction with memory and the machine learning of dynamical systems. *Communications in Computational Physics*, 25(4):947–962, 2019.
- [23] Brenden K Petersen, Mikel Landajuela, T Nathan Mundhenk, Claudio P Santiago, Soo K Kim, and Joanne T Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv preprint arXiv:1912.04871*, 2019.
- [24] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [25] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

- [26] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.
- [27] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.
- [28] Xing Shen, Xiaoliang Cheng, and Kewei Liang. Deep euler method: solving odes by approximating the local truncation error of the euler method. *arXiv preprint arXiv:2003.09573*, 2020.
- [29] Zezheng Song, Chunmei Wang, and Haizhao Yang. Finite expression method for learning dynamics on complex networks. *arXiv preprint arXiv:2401.03092*, 2024.
- [30] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [31] G Teobaldi, Al Ma’Mari, M Rogers, S Alghamdi, T Moorsom, S Lee, T Prokscha, H Luetkens, M Valvidares, M Flokstra, et al. Proceedings of the national academy of sciences. *Proceedings of the National Academy of Sciences of USA*, 2017.
- [32] Christopher Wilt, Jordan Thayer, and Wheeler Ruml. A comparison of greedy search algorithms. In *Proceedings of the International Symposium on Combinatorial Search*, volume 1 of *Lecture Notes in Computer Science*, pages 129–136, 2010.
- [33] John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020.

Department of Mathematics, University of Florida, Gainesville, FL 32611, USA.

Email address: jiananda.du@ufl.edu

Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Email address: Senwei.Liang@lbl.gov

Department of Mathematics, University of Florida, Gainesville, FL 32611, USA.

Email address: chunmei.wang@ufl.edu