CSE 5835 Final Report: Investigation of Dynamic Surrogate Models of Tumor Transport Phenomena

Due on May 6, 2022 at 11:59pm

Samuel Degnan-Morgenstern

Contents

1 Project Overview

Using a systems engineering perspective, drug delivery in tumors can be systematically modeled and optimized using rigorous methods in deterministic global optimization [1]. A first principles model of fluid and solute transport in tumors is used to analyzed the effect of the tumor microenviornment (TME) normalizing therapies that improve drug delivery [2, 3, 4, 5, 6]. A critical bottleneck in this process is the embedding of the partial differential equation (PDE) based transport model in the global optimization programs. Often times, global optimization with these PDE-based models take up to days to solve or in the worst case do not converge. In my senior thesis and a recent publication [1], I established artificial neural network (ANN) surrogate models that capture the input-output relationships between physiological parameters of interest and the corresponding accumulation profile that tracks drug delivery as depicted in Figure 1. This model utilized a multidimensional regression approach that learned the entire trajectory of the PDE using a large quantity of data. This model was successfully deployed in the global optimization studies to accelerate the solutions by several orders of magnitude. Despite this success, a more rigorous assessment of the machine learning pipeline is needed to develop the most efficient and accurate surrogate models.

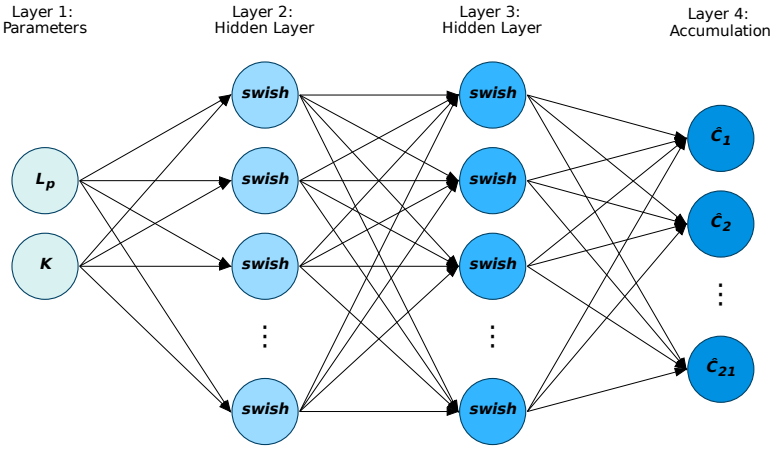

Figure 1: A multilayer perceptron (MLP) with 2 hidden layers, 24 nodes per layer, and the swish activation was previously used to construct a surrogate for a PDE-based tumor transport model. The model captures the mapping between two biological parameters of interest, vascular and interstitial hydraulic conductivity (L_p & K), and the dynamic accumulation profile.

In this work, I extend my previous modeling efforts to consider both the multidimensional regression approach and an alternative dynamic approach. The ultimate goal is to develop a model that effectively maps three biological parameters of interest to an accumulation profile while maximizing accuracy. The motivating framework for this is the explicit Euler formulation for numerically solving a differential equation. Specifically, for a given state variable y that varies over time we have the equation:

$$\frac{dy}{dt} = f\left(y, t\right) \tag{1}$$

which is discretized into n nodes and solved numerically by the following scheme:

$$y_{i+1} = y_i + \Delta t \cdot f\left(y_i, t_i\right), \ i \in \{1, 2, \ldots, n\}, \tag{2}$$

where y_i is the state variable at the ith node, t_i is the current time step, and Δt is the temporal discretization. This framework can be translated into a surrogate modeling approach using two strategies. The first is to

learn the full integration of the differential equation for steps $i = 1$ to $i = n$ in a single model using multidimensional regression. The other approach is to use a dynamic model to learn one time step of the explicit Euler integration such that the model maps the biological parameters, t_i, and y_i to y_{i+1}. This approach enables the use of advanced recurrent neural network models that are designed for modeling time series data with high accuracy. We will explore the development and performance of these models in this report.

2 Data

In this section, we will discuss the details of the data used for this project. Specifically, we will discuss the motivation for this specific dataset, the sampling technique, the sample complexity of the data, the distribution of data, the noise in the data, and the data pre-processing. An overview of this workflow can be visualized in the flowchart in Figure 2.

As previously mentioned, the motivation for developing these models is to create a surrogate model for the mechanistic transport model in (15) such that biological parameters of interest are mapped to a discrete accumulation profile. These biological parameters are vascular hydraulic conductivity L_p, interstitial hydraulic conductivity K, and nanocarrier radius R_s. The values of vascular and interstitial hydraulic conductivity quantify the efficacy of fluid and solute transport through the tumor vascular and tumor interstitum respectively; these parameters are highly dependent on the dosage of TME normalizing dexamethasone treatment. Furthermore, the nanocarrier radius is a design choice made by the clinical researchers that significantly influences the relative significance of diffusive and convective mass transport. In conjunction with experimental studies, global optimization is used to estimate these parameters from *in vivo* data to quantify the effect of different treatment strategies that are being studied. This procedure is challenging because the relationship between these physiological parameters and the resulting concentration profile in the tumor is highly nonlinear as a result of the governing physics discussed in Appendix A. Thus, the use of machine learning surrogate models are advantageous by creating a reduced complexity regression model that accelerates this procedure as shown in Wang et al. [1].

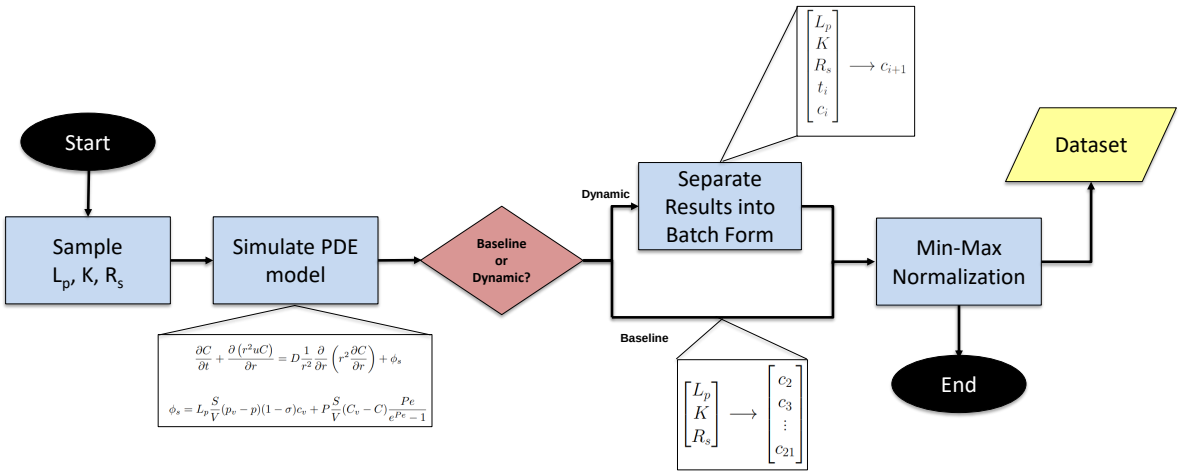

Figure 2: A flowchart representing the overview for data generation and pre-processing in this study. The biological parameters of interest are vascular hydraulic conductivity L_p, interstitial hydraulic conductivity K, and nanocarrier radius R_s. These parameters are sampled from uniform distributions. At each combination of the parameters, the mechanistic modeled described in (15) in Appendix A. For the baseline full integration models this data is then processed using min-max normalization and exported as the dataset. The dataset for the dynamic models is further processed by splitting each accumulation profile into 20 data points with the dynamic featurization as discussed in Section 3. The dynamic data is then scaled using min-max normalization and exported as the dataset.

To deliver an accurate surrogate model, it is necessary to create a dataset that is representative of the physiological conditions that could be possibly encountered while performing parameter estimation studies. As shown in Figure 2, the first step in doing so is sampling the physiological parameters. A Sobol sequence sampling procedure [7] was performed using the toolkit in Surrogates.jl v2.2.0 [8]. The Sobol sampling technique is a quasi random sampling method that efficiently fills the sample space compared to pure pseudo-random number generators [9]. This technique is used widely in the surrogate modeling literature, and it is well suited for accurately producing surrogate models. [10, 11]. For this procedure, lower and upper bounds were set on the physiological parameters as described in Table 1. The choice in these lower and upper limits represent physically viable and reasonable choices for the values of L_p, K, and R_s that would also be used in the search space for any optimization studies.

Table 1: The bounds of the input variables (L_p,K, R_s) used for the construction of the surrogate model are listed in this table.

Variable	Lower bound	Upper bound
L_p (cm/mm Hg-sec)	5.00×10^{-7}	5×10^{-6}
K (cm^2/mm Hg-sec)	5.00×10^{-7}	5×10^{-6}
$R_s(nm)$	5	30

Using this sampling protocol, 10^4 combinations of the biological parameters were selected such that we have a sample matrix $S \in \mathbb{R}^{3\times 10000}$. At each sample, the PDE model described in (15) was simulated over a time horizon of 5 minutes to generate a 20 node discretized accumulation profile $c_i \in \{2, 3, \ldots, 21\}$ where c_i denotes the accumulation value at the ith node. The reader should note that the initial concentration node c_1 is not recorded as it is trivially derived from the initial conditions in (19). This resulted in an accumulation matrix $C \in \mathbb{R}^{20\times 10000}$. Originally, the dataset was intended to be constructed using 10^6 samples, however, it was found that this dataset was too large to perform model assessment and train the models in a reasonable amount of time. Due to software bugs in the Julia programming language and the CUDA.jl framework, it was impossible to effectively train the models on the GPU. This limited model training exclusively to the CPU in a slower process. For this reason, the dataset was scaled down in this study. This sample complexity resulted in suitable validation and test performance for all models as discussed in Section 5.

Furthermore, for the dynamic models the data was further processed into batches of 20 such that each feature and target pair x_i, had the following form:

$$x_i = \begin{bmatrix} L_p \\ K \\ R_s \\ t_i \\ c_i \end{bmatrix} \tag{3}$$

$$y_i = c_{i+1}. \tag{4}$$

This resulted in a dataset consisting of 2×10^5 samples with a dynamic feature matrix $X \in \mathbb{R}^{5\times 200000}$ and target matrix $Y \in \mathbb{R}^{1\times 200000}$.

In this work, the data is sampled from well defined uniform distributions. Through the Sobol sequence sampling procedure, the resulting dataset will also be distributed uniformly as visualized in Figure 3 due to the underlying uniform distribution and the large number of samples. The uniform nature of this distribution leads to a very balanced dataset that is representative of the physically realizable physiological parameters. As this is a regression problem, the balance in the samples naturally leads to a representative and balanced target space. Furthermore, the target values are generated using a well defined mathematical model as outlined in Appendix A. The mechanistic tumor transport model is solved using the method of lines in DifferentialEquations.jl [12] using the state of the art stiff QNFD solver. As this data is model generated, there should be insignificant noise with some minor noise arising from the small approximation error associated with the numerical integration procedure.

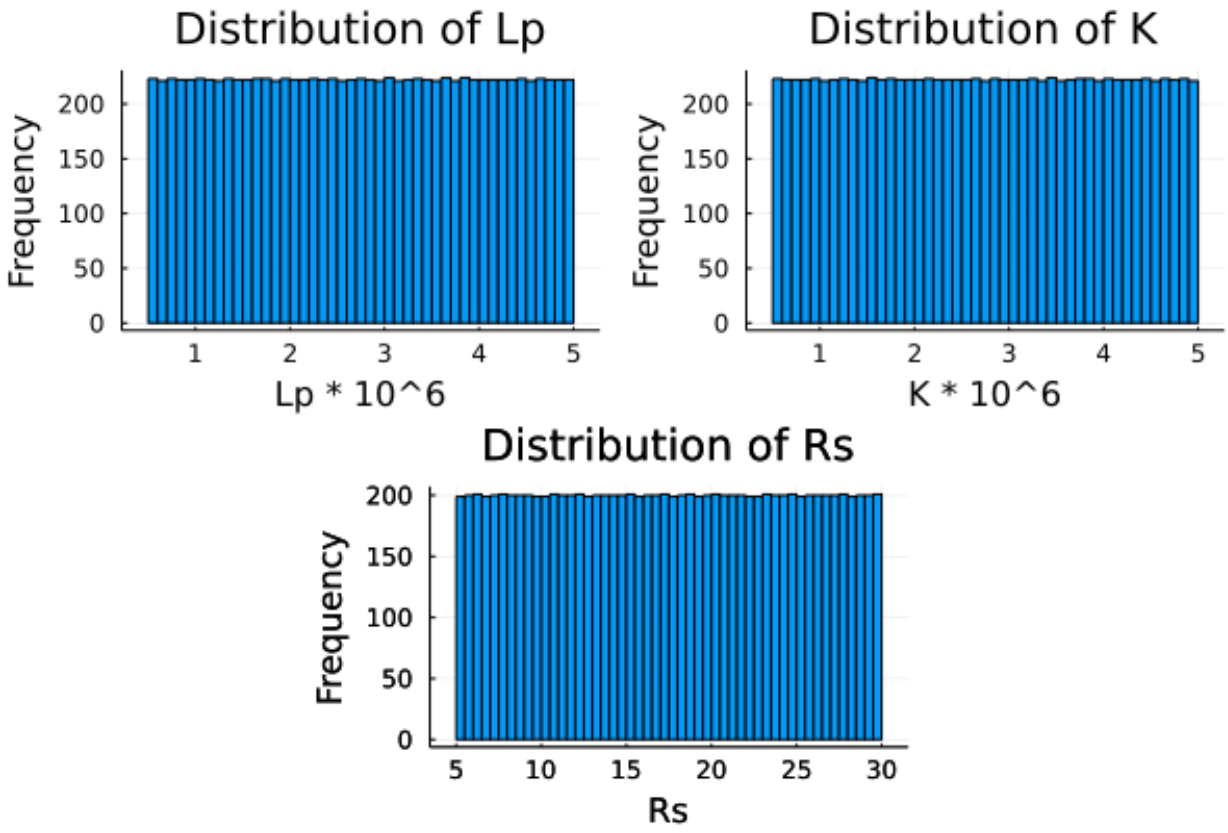

Figure 3: A visualization of the distribution of the biological parameters L_p, K,and R_s. The distribution of each parameter is approximately uniform.

Following data generation, the dataset was duplicated and processed into batch form with dynamic model featurization as described previously and in Section 3. As the datasets consist of features that differ by several orders of magnitude, each dataset was scaled using min-max normalization. As the bounds on the features are well defined, min-max normalization is a suitable choice to scale the data here. Lastly, each dataset was split into training, validation, and test sets using an 80/10/10 split. This step is critical for rigorously assessing candidate models and evaluating model performance.

In summary, in this section we have provided a thorough description of the dataset in this work. The data is generated by sampling physiological parameters of interest and simulating the mathematical model in (15) at each sample. The samples are uniformly distributed with minimal noise. The dataset is then scaled, split into training, validation, and test sets and exported for use in model development.

3 Feature Design

For the surrogate model construction, feature design is motivated by the ultimate goal of deploying the model in optimization studies as well as encoding relevant model parameters and variables. Specifically, the surrogate model is designed to capture the mapping relationship specified by the tumor transport model in Appendix A. As this is the case, we can choose model parameters of interest and understand their importance and effect in the physical model. The model parameters to be used as features are selected based on the goal of the optimization studies they will be used in. Lastly, the critical difference between the full integration and dynamic models is the use of features to encode the independent and dependent variables of the model.

3.1 Full Integration Model Features

The primary motivation for creating machine learning surrogate models is to deploy the models in optimization studies for parameter estimation such that the model is validated on *in vivo* data. This study is performed to quantify the physiological effect of different TME normalizing treatment with varying treatment dosages and nanocarrier sizes. From Martin et al. [6], we know the different treatment dosages have a strong effect on vascular hydraulic conductivity and interstitial hydraulic conductivity, while the nanocarrier size is a design choice of the clinical researchers. For this reason it is critical to encode these parameters directly in the surrogate model as features. The importance of each of these parameters will be discussed in this section.

Vascular hydraulic conductivity L_p quantifies the efficacy of fluid and solute transport through the porous vascular wall of the tumor vasculature. L_p is a function of the blood vessel properties as defined in (20). These properties are influenced by the TME normalizing treatments. Subsequently, L_p is a good feature choice as it encodes information about the TME that is affected by different treatments, and it has a well-defined correlation with the resulting concentration. Specifically, it is clear from (11),(12),(15), and (16) that L_p has a direct effect on the concentration. Intuitively this relationship makes sense because we know L_p is a function of the vasculature, and the size of blood vessels and the blood vessel pores will directly inhibit or enable better delivery of nanocarriers.

Interstitial hydraulic conductivity K quantifies the efficacy of fluid and solute transport through the tumor interstitum. This physiological transport is complex as visualized in Figure 4. K encodes information about the porosity of the tumor cells and the compression of the corresponding extracellular matrix. These properties are directly altered through TME normalizing therapy, as reflected in the dependence of K on the TME normalizing therapy dosage. As reflected in the model by (11),(12),(15), and (16), K is important in determining the resulting mass transport and nanocarrier accumulation. Due to the relevant physiological significant and direct correlation with the target, K is a reasonable feature choice for this model.

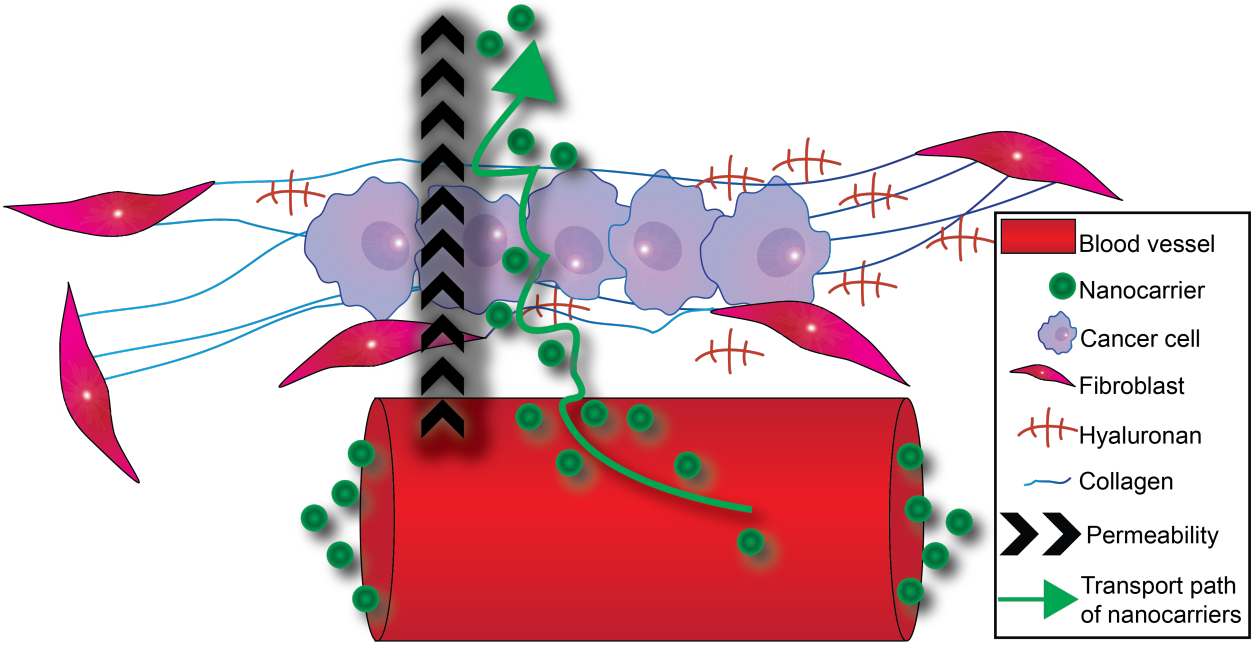

Figure 4: A schematic representing fluid and solute transport in the TME [13]. Nanocarriers flow through the porous blood vessels into the tumor interstitium through a combination of convective and diffusive transport.

Nanocarrier radius R_s encodes the size of the agent used to transport anticancer medicines. For model validation studies, treatments are analyzed at varying nanocarrier sizes. From the drug design study in Wang et al. [1], it is known the therapeutic effect of TME normalizing treatments is size dependent due to the altered transport properties. Thus, it is critical to encode the size in order to quantify the effect of different sized nanocarrier agents. From the underlying pore theory, the nanocarrier size is known to play a role in the permeability of nanocarriers. Specifically, from (23) the diffusion coefficient is dependent on the nanocarrier size which in turn has an effect on the permeability in (21) which in turn effects the source term of the solute transport model in (15) and (16). Furthermore, the inclusion of R_s also reflects the properties of several other model parameters. Specifically, the diffusion coefficient D, the solute reflection coefficient σ, and the nanocarrier circulation half life k_d are all dependent on the nanocarrier size. Thus, the use of just R_s eliminates the need for more physiological parameters that would otherwise be correlated. R_s is an important feature choice, as it encodes relevant information about the effect of different sized nanocarrier on the drug delivery that is valuable for the eventual deployment in parameter estimation optimization studies.

L_p, K, and R_s capture different information about the properties of the TME and the transport of fluid and solutes within the TME. For this reason, these features will not be directly correlated with each other. Specifically, L_p reflects information about the tumor vasculature while K reflects information about the tumor interstitium and the extracellular matrix, and R_s reflects the transport properties of the chosen solute. As we have shown, all these features correlate with the resulting concentration given by the mechanistic model. Due to the nature of the of the regression problem, there is a lack of invariance and symmetry of the model results with respect to different treatments and nanocarrier sizes. The model will vary smoothly with different TME normalizing treatments (and thus L_p and K values) and nanocarrier sizes.

3.2 Dynamic Model Features

In addition to the features encoding the physiological parameters of interest described in Subsection 3.1, the dynamic models also encode the discretized independent variable - time t_i and the discretized dependent variable - accumulation c_i, $i \in \{2, 3, \ldots, 21\}$. These features reflect the different formulation with respect to the numerical integration scheme in (2) as discussed in Section 1. This is important to this approach, as we are trying to encode additional structure and information about the explicit numerical integration scheme through this featurization.

Furthermore, t_i and c_i directly influence the governing solute transport model, as it depends directly on both these variables. Specifically, the vascular solute concentration explicitly depends on time as described in (17). Furthermore, the source term in (16) depends on the current concentration. For this reason, these two features encode additional information about the underlying mechanistic transport model.

The inclusion of t_i and c_i as features encodes important information about the transient nature of this model. This model formulation is simpler as the machine learning algorithm will be learning a single step of numerical integration such that it learns the RHS of (2). However, it is important to note that these two features will be correlated with eachother. The current concentration will depend on the current time step due to the transient nature of the model. Despite this correlation, it is important to still include both variables as features in this model because of the distinct and explicit dependence of the numerical integration scheme and the governing transport model on both of these variables.

4 Learning Algorithms

In this work, four different models using three different types of deep learning algorithms were used. Specifically, for the full integration approach solely a multilayer perceptron (MLP) model was used. For the

dynamic approach, a MLP, a vanilla recurrent neural network (RNN), and a Long-Short Term Memory (LSTM) model was used. In this section, we will discuss the strengths and weaknesses of all the chosen learning algorithms, the corresponding model assessment procedure, and the model training and optimization process. Model assessment was conducted using grid search for each learning algorithm. This procedure was done for simplicity in order to systematically perform model assessment in a reasonable amount of time given the multitude of learning algorithms tested. A record of the model assessment scores can be accessed in the project repository.

4.1 Full Integration Multilayer Perceptron

For the full integration approach, a MLP model was assessed. This algorithm was chosen as a baseline comparison in the investigation of the dynamic surrogate models. A MLP is one of the simplest forms of an artificial neural network or deep learning model. This type of model is composed of an input layer, a set of one or more hidden layers, and an output layer as visualized in Figure 5.

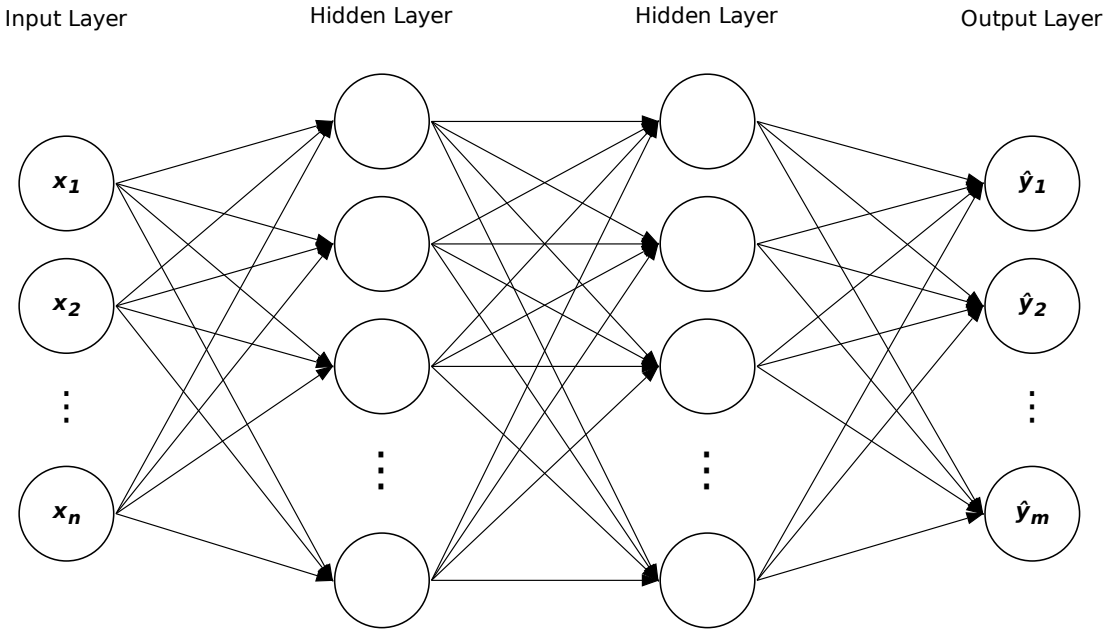

Figure 5: A schematic representing a fully connected feed-forward multilayer perceptron model

Each input layer and hidden layer is fully connected to the next layer with each edge connection having an associated weight. The value of each node is computed by first calculating the linear combination of the previous layer's nodes and associated weights, adding a bias term, then this value is taken as an input to an activation function which outputs the corresponding value of the node. The hyperparameters of MLP models are the number of hidden layers, number of nodes per hidden layer, and the activation function of a given layer.

MLPs are universal approximators that can represent any function to arbitrary degree of accuracy. This makes them well suited for a surrogate modeling regression task as we have in this work. Furthermore, MLPs have the ability to perform multidimensional regression with high accuracy [14]. MLPs can capture

nonlinear relationships as we have in this work, and they possess a relatively simple structure. Furthermore, the multidimensional regression approach allows this model to be compact and require only one evaluation. However, MLPs have a non-convex loss function that makes optimization challenging and non-deterministic. The development of MLPs also requires significant tuning to determine the optimal hyperparameters.

The problem of hyperparameter tuning and model assessment was addressed in this work using grid search. From the previous work on this surrogate modeling project in Wang et al.[1], it was known the model performed relatively well with 1-2 hidden layers and the *swish* and *tanh* activation functions. For this reason, we assessed these number of layers and activation functions with the addition of the *relu* activation function. The number of nodes per hidden layer was picked to be either 8,16, or 32 nodes. In summary, 18 candidate models were assessed with 1-2 hidden layers, the *swish*, *tanh*, or *relu* activation function, and 8, 16, or 32 nodes per hidden layers. The mean squared error (MSE) was recorded on the validation set for each model. It was found that a MLP with 2 hidden layer with 8 nodes per layer and the *relu* activation function as visualized in Figure 6 performed the best.

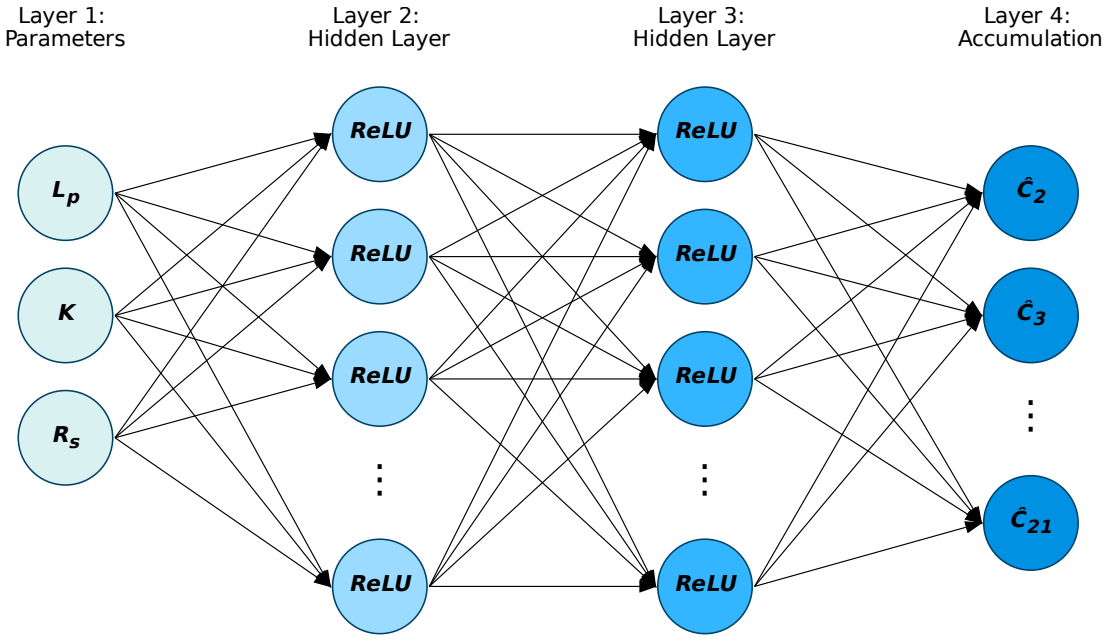

Figure 6: A visualization of the multilayer perceptron model used for the full integration surrogate modeling approach.

The final model was then constructed in the Julia programming language in Flux.jl [15]. The model was trained for 100 epochs using the MSE loss function, the ADAM optimizer [16], an exponentially decaying learning rate schedule, and early stopping criteria of an MSE value on the validation set of 2.5×10^{-6}. The performance metrics are reported in Section 5.

4.2 Dynamic Multilayer Perceptron

For the dynamic approach, three learning algorithms were investigated. The first of which is a MLP model. The learning algorithm is the same described in Subsection 4.1. Subsequently, the learning algorithm has

similar advantages and disadvantages with the most notable being its defining ability to accurately capture nonlinear mapping relationships in regression problems. The most notable difference in the strengths and weaknesses of this model is the simplicity afforded by the low dimensional target space of the dynamic MLP model. In theory, this should reduced the computational burden of the model during training. However, this model will required repeated evaluation to construct an accumulation profile.

Model assessment for the dynamic MLP was also conducted using grid search. The following hyperparameter configurations were tested: 1-2 hidden layers, the *relu* and *tanh* activation functions, and 8, 16, or 32 nodes per hidden layer. The MSE of each model on validation set was recorded. The model with the best performance consisted of 1 hidden layer,8 nodes per hidden layer, and the *relu* activation function as visualized in Figure 7.

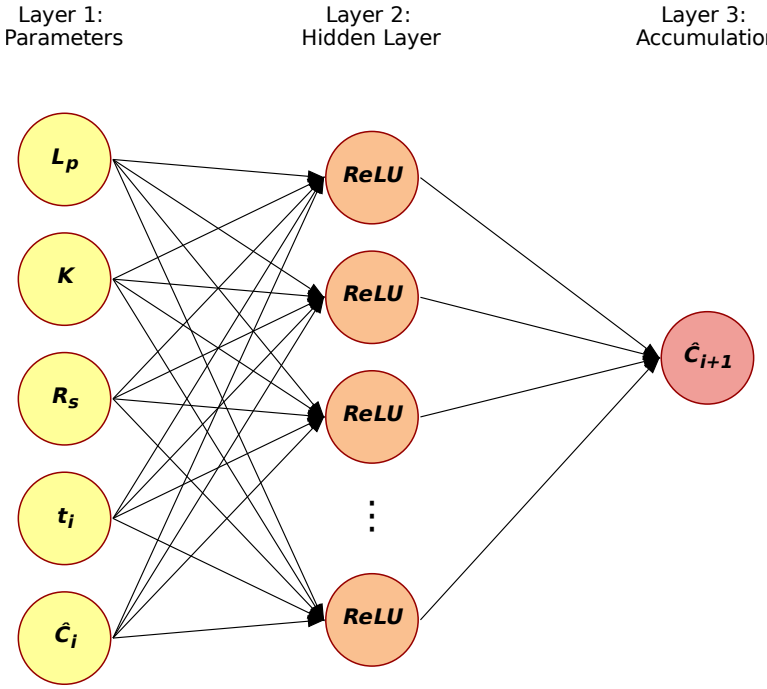

Figure 7: A visualization of the multilayer perceptron model used for the dynamic surrogate modeling approach.

The final model was then constructed and trained in Flux.jl [15]. Specifically, the model was trained for 100 epochs using the MSE loss function, the ADAM optimizer [16], an exponentially decaying learning rate schedule, and early stopping criteria of an MSE value on the validation set of 5×10^{-7}. The performance metrics are reported in Section 5.

4.3 Dynamic Vanilla Recurrent Neural Network

The second learning algorithm employed for the dynamic approach was a vanilla RNN model. RNNs are another class of deep learning models specifically designed for sequential data such as the time series data in this work. Specifically, RNNs are artificial neural networks with an additional hidden state for each node.

This hidden state is updated after each evaluation to reflect the current state of the model. This enables the model to learn arbitrarily long term dependencies within sequential data. Furthermore, RNNs can handle variable input sizes. In this work, we are interested in repeatedly evaluating the model to construct a sequential accumulation profile. The weaknesses in vanilla recurrent neural networks comes from the difficulty in training them. RNNs are trained through back-propagation on the unrolled computational graph, but due to the long term dependencies vanishing or exploding gradients can arise for complex RNN models.

Like the MLP models, model assessment was conducted using grid search. Only the *tanh* activation was tested as this is standard practice in the field. The hidden layers were varied between 1-2 recurrent hidden layers with 8, 16, or 32 nodes per hidden layer. The MSE of each model on the validation set was recorded. The model with the best performance had 1 recurrent hidden layer and 32 nodes in the hidden layer as visualized in Figure 8.

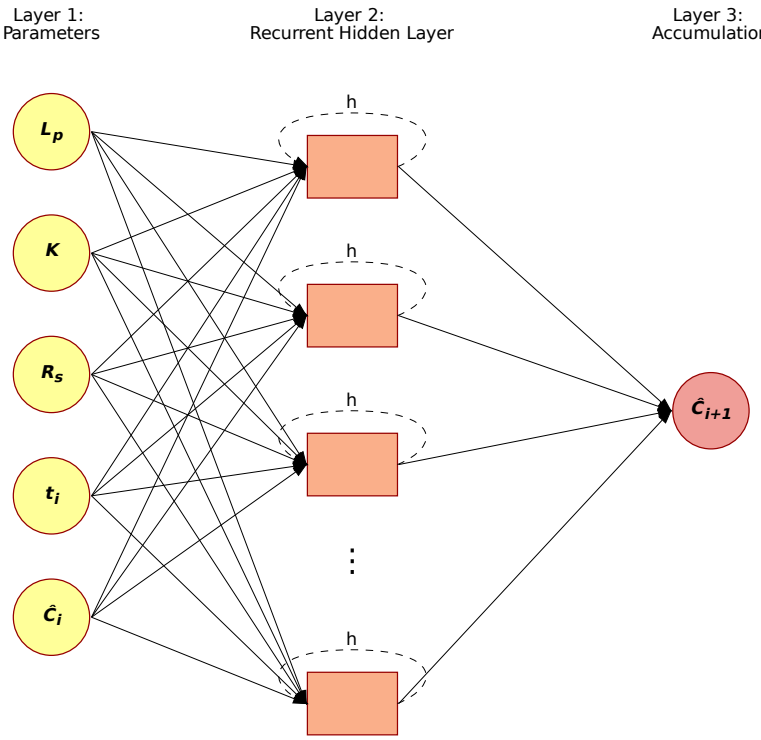

Figure 8: A visualization of the vanilla recurrent neural network model used for the dynamic surrogate modeling approach.

The model was constructed and trained in Flux.jl [15]. The model was trained for 100 epochs using the MSE loss function, the ADAM optimizer [16], a constant learning rate of 5×10^{-5}, and an early stopping criteria of an MSE value on the validation set of 5×10^{-7}. The performance metrics are reported in Section 5.

4.4 Dynamic Long-Short Term Memory Network

The final learning algorithm used in the dynamic approach is a LSTM model. A LSTM is a variant of the vanilla RNN described in 4.3. The key difference is that LSTM utilize three interacting layers consisting of different gates to manage the hidden state of the cell and the inputs and outputs. Forget gate layers decide

what information in the current cell state should be removed, input gate layers decide how and what values in the current cell state should be updated, and the output layer determines what information and values to output from the current cell state. This configuration hugely improves upon the strengths of RNNs, as the management of information prevents exploding or vanishing gradients during training. However, it is important to note that for a fixed sample complexity, like in this study, the LSTM model has a more complex hypothesis class. Specifically, there is a larger number of parameters that must be trained relative to MLPs and vanilla RNNS, and if this is done without enough data then there may be overfitting.

Like previously, model assessment was conducted using grid search. The same hyperparameter values as the vanilla RNN model were tested, specifically: 1-2 recurrent hidden layers and 8, 16,or 32 nodes per hidden layer. Following training, the validation MSE was recorded. The best performing model had 1 recurrent hidden layer with 8 nodes in the hidden layer as visualized in Figure 9.

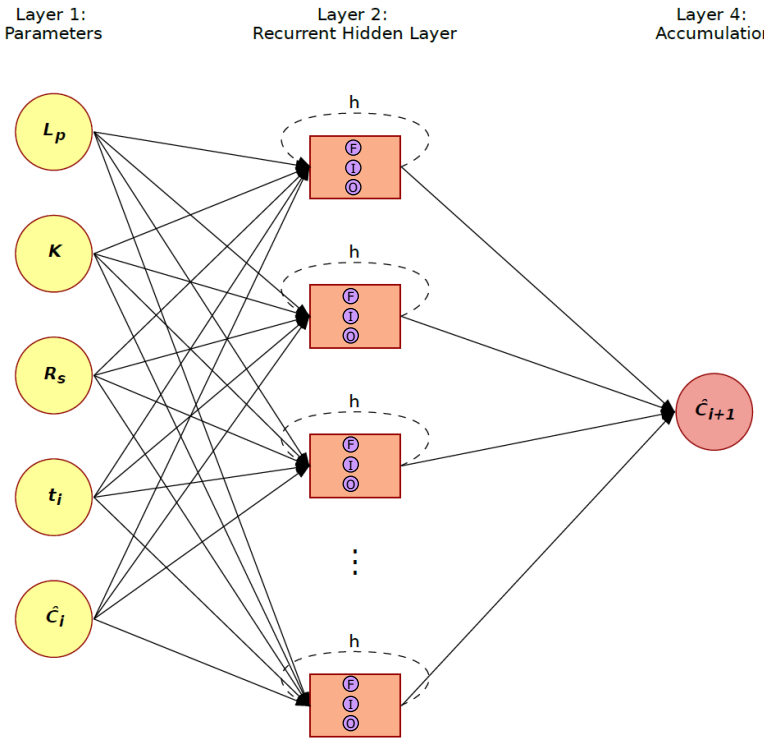

Figure 9: A visualization of the long-short term memory model used for the dynamic surrogate modeling approach. The reader should note that the F, I, and O symbols respectively denote the forget gate, input gate, and output gate in the LSTM model.

The model was constructed and trained in Flux.jl [15]. The model was trained for 100 epochs using the MSE loss function, the ADAM optimizer [16], an exponentially decaying learning rate schedule, and an early stopping criteria of an MSE value on the validation set of 1×10^{-7}. The performance metrics are reported in Section 5.

5 Model Performance

As this study is in a regression setting, a multitude of standard regression performance metrics are chosen to evaluate the surrogate models. In this section, we will present the chosen performance metrics, report the performance of the surrogate models, and provided further qualitative results and analysis.

Three primary performance metrics were used in this study including: mean squared errror (MSE), root mean squared error (RMSE), and mean percent error (MPE) also sometimes referred to as mean absolute percentage error. MSE is defined as follows

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5}$$

where N is the number of samples, y_i is the true output, and $\hat{y}_i$ is the predicted output. MSE is widely used in regression problems as it provides a simple measure of the error between the model predictions and the true target data. We have utilized MSE in this work for model assessment when comparing models of the same class. As MSE was used as the loss function, this gave a direct assessment of the performance. The second performance metric used in this study was the closely related RMSE, defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{6}$$

RMSE corresponds to the $\mathbf{L_2}$ norm which is a measure of the distance between two points. While similar to MSE, the RMSE metric confers several advantages as it is on the same order of the normalized data, and is thus more interpretable. Furthermore, the RMSE value of a naive model that just outputs the mean of the target data will be the standard deviation of the dataset. This provides a standard benchmark for assessing the performance of the model. The final quantitative performance metric used is the MPE value defined as:

$$MPE = \frac{100\%}{N}\sum_{i=1}^{N}\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right). \tag{7}$$

The MPE metric is commonly used in the machine learning regression literature [17], as it is similar to the mean absolute error performance metric. The MPE performance metric has the advantage of being a dimensionless metric that provides a relative estimate of error. Furthermore, MPE is given as a percentage so it is more easily interpretable than other regression metrics. The values of these performance metrics for all surrogate models are reported on the test set in Table 2.

Table 2: The performance metrics for the full integration and dynamic surrogate models are reported in this table. The MSE, RMSE, and MPE are recorded on the test set for each model.

Performance Metrics on Test Set			
Model	MSE	RMSE	MPE
Full Integration MLP	2.69×10^{-6}	1.64×10^{-3}	0.71%
Dynamic MLP	5.32×10^{-7}	7.29×10^{-4}	0.55%
Dynamic RNN	1.46×10^{-6}	1.21×10^{-3}	1.61%
Dynamic LSTM	4.10×10^{-7}	6.40×10^{-4}	0.85%

From Table 2, it appears that all the models perform quite well with most having a MPE of less than 2% error. However, it is important to note that these metrics were calculated using the data from the test set

where all sequence values are known *a priori*. However, for this application it is necessary to build up the full accumulation profile using only the physiological parameters and initial conditions. For the full integration model, this is done implicitly as it learns the full accumulation profile in one model. For the dynamic models, this requires repeated evaluation and feeding the output back into the model as an input. Following this procedure, the performance metrics are calculated using just the sample set and reported in Table 3.

Table 3: For the full integration and dynamic surrogate models, the MSE, RMSE, and MPE are recorded by building a predicted accumulation profile using the sample set and initial conditions for each model.

Performance Metrics using Sample Set			
Model	MSE	RMSE	MPE
Full Integration MLP	2.69×10^{-6}	1.64×10^{-3}	0.71%
Dynamic MLP	1.30×10^{-2}	1.14×10^{-1}	39.40%
Dynamic RNN	1.06×10^{-2}	1.03×10^{-1}	33.27%
Dynamic LSTM	1.34×10^{-2}	1.15×10^{-1}	38.79%

From the results in Table 3, it is clear that the full integration MLP significantly outperforms all of the dynamic models. This stems from the repeated evaluation of each dynamic surrogate model to construct a full accumulation profile. Each evaluation has a small error associated with it, and the slightly erroneous evaluation is fed back into the model both explicitly in the features and implicitly in the hidden state of the recurrent models. This procedure leads to the accumulation of error in the dynamic models. Specifically, early time nodes will have good predictions, but as the small errors begin to accumulate the models will become more and more inaccurate at later time nodes. This can be further visualized in the qualitative analysis in Figure 10.

It is important to note that a naive regression model would only predict the mean of the dataset. This naive model would have an RMSE score of the standard deviation. In this case, the normalized standard deviation of the dataset is 1.67×10^{-1}. Here all the models have performed better than this benchmark. However, the dynamic models have only slightly outperformed the naive model, whereas the full integration model has significantly outperformed this metric by two orders of magnitude. From this analysis, we can further conclude that the full integration model has truly learned a good predictive model.

Another unexpected result of this investigation is the relative performance of the dynamic models. On the test set, the dynamic MLP has a better MPE score than the recurrent models signifying potentially better relative accuracy. This phenomenon may be a result of the fixed sample complexity in this study. Specifically, the 2×10^5 data points may have been enough to effectively train the MLP without overfitting, but it may have not been enough for the recurrent models. Furthermore, the vanilla RNN has the worst performance of all models on the test set which is expected due to the difficult training these models as discussed previously.

Additionally, these relative performance metrics for the dynamic models become inverted when evaluating the performance metrics on the sample set. Specifically, the dynamic RNN performs the best followed by the LSTM model and the MLP model. The recurrent properties of the RNN and the LSTM likely gave these models a slight edge over the MLP models for accurately reconstructing the time series data. However, it is very unexpected that the RNN outperformed the LSTM model. This result may be caused by the different training processes and the additional complexity of the LSTM hypothesis class. The RNN was found to have good performance when trained using the ADAM optimizer with a constant learning rate of 5×10^{-5}. When training the LSTM, a constant learning rate was found to lead to significant overfitting. For this reason, an exponentially decaying learning rate was used which gave satisfactory performance with less overfitting.

Furthermore, the additional complexity of the LSTM hypothesis class makes these models more likely to overfit the data for a fixed sample complexity. Typically, the LSTM models will always work better, but as these are very simple recurrent surrogate models the observed result is likely an edge case of this rule of thumb. For future studies with more data and GPU training, the LSTM model is likely to outperform the vanilla RNN models significantly.

One deficit of all these performance metrics is they report error scores in aggregate. The practice of averaging the error leads to the loss of information on what types of samples the model performs well on and poorly on. To remedy this, we present a plot of the surrogate models versus the mechanistic model at several parameters of interest. Specifically, the samples of interest correspond to the results of the parameter estimation study in Wang et al. [1] where the surrogate models would be expected to perform especially well. These visualizations allow us to qualitatively assess the models performance at different regions of the sample set. The first treatment of interest is the control treatment with no administration of the TME normalizing agent dexamethasone (DEX) using a 70kDa tracer molecule. The results and discussion for the model performance at this sample are presented in Figure 10. The second treatment of interest is the 3 mg/kg DEX treatment with a 500 kDa tracer. The results and discussion for the model performance at this sample are presented in Figure 11. The visualizations of additional samples are plotted in Appendix B.

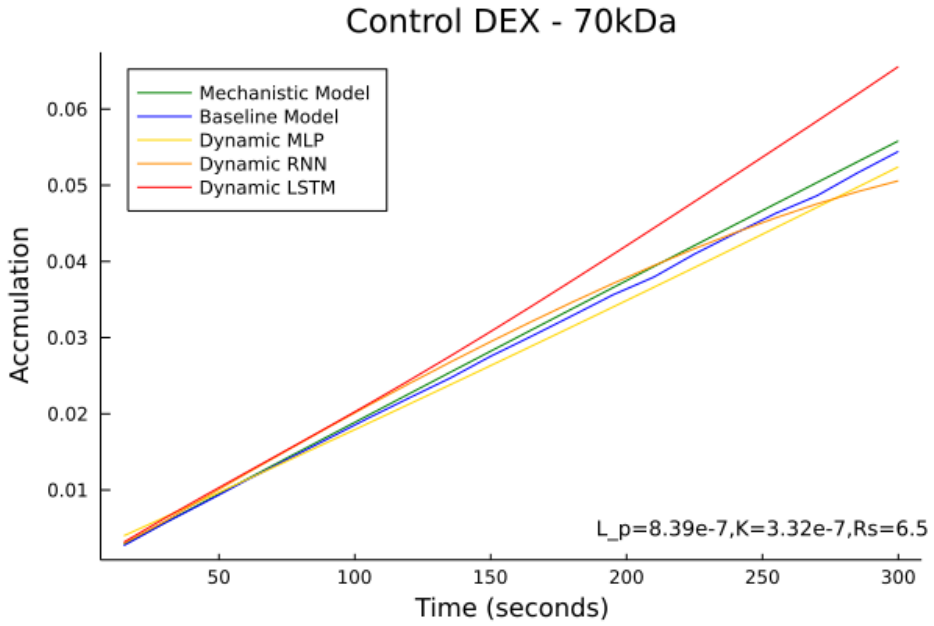

Figure 10: This figures presents the accumulation profile derived from the mechanistic PDE model in (15) and the accumulation predicted by the various surrogate models as indicated in the legend. The values of L_p, K, and R_s in this plot represent no TME normalizing DEX treatment and a 70kDa tracer molecule. From the results of Wang et al. [1], this therapy exhibits irregular and generally poor solute transport that is dominated by diffusion. This transport is notoriously challenging to model as it is governed by a mix of convective and diffusive transport. For this reasons, it is challenging to represent this complex transport phenomena using a surrogate model. It is evident that the closest model prediction is the baseline model, as it closely tracks the results of the mechanistic model over time. Interestingly, the dynamic models exhibit the accumulation of error in their predictions. Specifically, each model seems to deviate more and more from the mechanistic model over time. As discussed previously, this is a result from each evaluation leading to more and more error as the dynamic model reconstructs the accumulation profile.

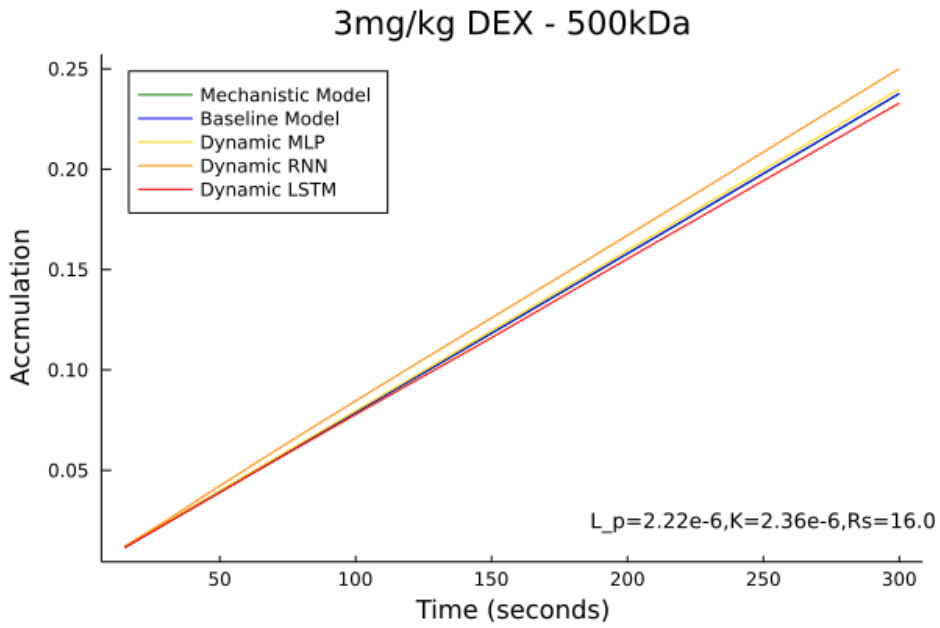

Figure 11: This figures presents the accumulation profile derived from the mechanistic PDE model in (15) and the accumulation predicted by the various surrogate models as indicated in the legend. The values of L_p, K, and R_s in this plot represent 3 mg/kg of the TME normalizing DEX treatment and a 500kDa tracer molecule. From the results of Wang et al. [1], this therapy exhibits effective transport that is dominated by convection. For this reason, the model is simpler and can be modeled more effectively by the surrogates. From the plot, it is clear that all the models track the mechanistic model very well. Notably, the dynamic RNN and LSTM begin to deviate towards the later time nodes. However, this deviation is significantly less than the that exhibited in Figure 10 due to the more regular transport properties that are easier to model accurately.

6 Answer to Instructor's Questions

6.1 Q1

My main question is about how the data is split and performance is measured. From your code and writeup it looks like the data is split into training, validation, and test simply randomly. However, for the same L_p, K, and R_s, the concentration states c_i are correlated in time. Did I understand correctly how the data is being split, and if so, what would be an alternative way to split the data so that there is no data leakage between training, validation, and test sets? (Hint: there is more than one way to think about this, one based on separation by time, and one based on grouping together data so that all datapoints in the group are only in train/val/or test.)

You are correct in your assessment that the data was split into training, validation, and test randomly. My original thought while performing this split was that for a given sample of the biological parameters the profile length would be 20 and thus evenly divisible by the number of samples in the training,validation, and test set (160,000 / 20,000 / 20,000 respectively). Therefore, in theory a given profile would end up in only one of the sets. I think a way to address the leakage problem could be to enforce this more formally. However, I believe a way that truly would improve performance would be to enforce temporal ordering.

An alternative way to split the data without leakage would be to group the data in time. Specifically, we could have 20 different training/validation/test splits for each time node t_i in a similar procedure to k-fold cross validation. Therefore, each time node t_i could be considered its own fold with its own training, validation, and test split. This way the model would learn only one time node at a time and not encounter future information during training from the mini-batch gradient descent. The training procedure could be to first train the model on the t_1 fold, assess the performance using the validation and test set and then repeat the procedure on the $t_2, t_3, \ldots, t_n$ folds.

6.2 Q2

The full integration model currently only makes predictions up to 20 timesteps, if I am understanding correctly. I assume the dynamic models' performance are also computed using 20 accumulated timesteps (is this correct?). Without doing any further experiments, how do you hypothesize the performance of the full integration model will change as the total number of predicted timesteps is increased? Similarly, how do you hypothesize the performance of the dynamic integration model will change?

Your assessment is correct. For the optimization study, we only consider a fixed time horizon of 300 seconds (5 minutes) with a $\Delta t = 25$s. These values were chosen to maximize the accuracy of observed experimental predictions and stay within the stability limits of the numerical integration (stability analysis conducted prior to work in Wang et al. [1]). Thus, the full integration model was constructed to predict 20 time steps and the dynamic model was evaluated 20 times with the Δt implicitly encoded in the model.

I would hypothesize that the performance of the full integration model would stay the same and then eventually decrease as more time steps are added. Specifically, the full integration model has been shown to have a relatively constant small error at each node. However, training the model itself would be more difficult due to the higher dimensional target space leading to a more inaccurate model.

I think the dynamic model performance would further decrease due to the accumulation of error discussed previously. Specifically, as more evaluations are required then the error would grow even more. One caveat is that if the data leakage problem was appropriately addressed the dynamic model might be more robust due to the fixed one dimensional target space. The dimensionality of the problem would not change, so in theory training the model would have a fixed difficulty.

Additionally, if the time horizon is expanded at the same time the time steps are increased this would also sharply decrease the performance of both models. The accumulation profiles in the range of 5 minutes are relatively linearly, however, they become significantly nonlinear as the time horizon increases (concentration saturates as the drug delivery approaches steady state).

6.3 Q3

While the full PDE considers C(r,t), the ML models only use the averaged C over space (do I understand this correctly?). Is it possible that different C(r,t) with the same average C at time t can lead to different behavior at time t+1? If so, how might this affect the performance of your dynamic models?

This is an interesting question, as it is something I actually encountered in my previous work in [1]. Indeed, the concentration is spatially averaged as the ultimate goal is to fit it with experimental data that can only be reported as a spatial average over time. It is possible that different combinations of the biological parameters can lead to different C(r,t) with the same spatial average at time t, but with different trajectories at time t

+ 1. From my perspective, the hope is that using the biological features at each time step would help encode information about the trajectory that would be informative for predicting the different concentrations at time t+1 even if the concentration at step t was the same.

However, even with this being the case, this loss of information in the averaging would detrimentally affect the performance of the dynamic models. For example, consider two models with different biological parameter combinations S_1 and S_2, with intersecting trajectories of the spatially averaged accumulation profiles. The dynamic model for S_1 may predict a trajectory that is similar to S_2 and vice versa because of the intersection.

In practice with the full integration models, this problem was resolved through an additional constraint on the optimization problem. From experimental measurements, we knew the peripheral pressure in the tumor which was encoded as a constraint and lead to a unique solution.

7 Conclusion

In this work, we present an investigation of two approaches to creating surrogate models for a PDE-based mechanistic model of fluid and solute transport in tumors. A data set is constructed by sampling three relevant biological parameters and solving (15) to create an accumulation profile that tracks drug delivery over time. The dataset is balanced and representative of the potential physiological parameters. Following data generation, two classes of models were built. A full integration model is used to map the physiological parameters L_p, K, and R_s to the full accumulation profile using multidimensional regression with an MLP model. Alternatively, a dynamic approach is used to map the biological parameters and the model variables t_i, and c_i at the ith time step to the concentration at the next time step using a MLP model, a vanilla RNN model, and an LSTM model. The performance is evaluated using MSE, RMSE, MPE, and qualitative plots. The study found that the full integration vastly outperformed the dynamic approaches due to the accumulation of error by the repeated evaluation of the dynamic model. From visual inspection, the surrogate models perform well on samples that reflect regular, convection dominated transport whereas the surrogate models perform worse on samples that reflect the non-normalized TME environment with irregular transport phenomena.

Appendices

A Tumor Transport Model

In this section, we describe the 1-dimensional tumor transport model introduced by Baxter and Jain [2, 3, 4, 5] to model fluid and solute transport in tumors. Tumors are spatially heterogeneous and consist of a necrotic core, a rapidly dividing outer region, and a large vasculature to support this continuous growth. These differences in characteristics cause a spatial dependence of physiological parameters that introduces a high degree of complexity. For the purpose of this study, we make the simplifying assumption of spatial homogeneity. Furthermore, we ignore the role of lymphatic and extravascular binding because of their insignificant presence in tumors. We do not explicitly consider the presence of microscopic structures in the tumor, such as blood vessels, cells, and the extracellular matrix (ECM), since we are focused primarily on capturing macroscopic phenomena. Spatial averaging is employed on the results of the solute transport model to provide a concise and direct description of the underlying dynamics. Finally, we assume that the sources of fluid and solute are continuously distributed throughout the tumor.

A.1 Fluid Transport

The interstitial fluid transport in a tumor follows Darcy's law for axisymmetric flow in spherical coordinates:

$$\mathbf{u} = -K\nabla\mathbf{p} \tag{8}$$

which simplifies to

$$u = -K\frac{dp}{dr}, \tag{9}$$

where u is the interstitial fluid velocity ($\frac{\text{cm}}{\text{s}}$), K is the hydraulic conductivity of the tumor interstitium ($\frac{\text{cm}^2}{\text{mm Hg - s}}$), p is the interstitial fluid pressure (mm Hg), and r is the radial position (cm). The fluid continuity equation for steady-state incompressible flow in a tumor in 1D spherical coordinates is given by:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2u\right) = L_p\frac{S}{V}(p_v - p), \tag{10}$$

where L_p is the hydraulic conductivity of the vascular wall ($\frac{\text{cm}^2}{\text{mm Hg - s}}$), $\frac{S}{V}$ is the ratio of the vascular surface area per unit volume (cm^{-1}), and p_v is the vascular pressure (mm Hg). Substituting (9) into (10), we obtain the steady-state fluid transport model as follows:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dp}{dr}\right) = \frac{\alpha^2}{R^2}(p - p_{ss}), \tag{11}$$

where

$$\alpha = R\sqrt{\frac{S}{V}\frac{L_p}{K}}. \tag{12}$$

Here, α is a dimensionless parameter, R is the radius of the tumor (cm), and p_{ss} is the steady-state interstitial pressure where the efflux and influx from the vasculature are equal. For this study, p_v and p_{ss} are equivalent.

This model has the following boundary conditions:

$$\left.\frac{dp}{dr}\right|_{r=0} = 0,$$
$$p|_{r=R} = p_\infty$$

which represents the no-flux symmetry condition at the center of the tumor and a fixed tissue pressure p_∞ (mm Hg) at the tumor edge R.

The analytical solution to the fluid transport model can be derived in dimensionless form as:

$$\hat{p}\left(\hat{r}\right) = \left(1 - \frac{\sinh\left(\hat{r}\alpha\right)}{\hat{r}\sinh\left(\alpha\right)}\right), \tag{13}$$

where

$$\hat{r} = \frac{r}{R},$$
$$\hat{p} = \frac{p - p_\infty}{p_{ss} - p_\infty}.$$

Here, $\hat{r}$ is the dimensionless form of the radial position in the tumor and $\hat{p}$ is the dimensionless form of the interstitial fluid pressure.

A.2 Solute Transport

Nanocarrier solute transport in tumors is characterized by the convection-diffusion equation:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = \nabla \cdot (D\nabla c) + \phi_s \tag{14}$$

which simplifies to the following in 1D spherical coordinates:

$$\frac{\partial c}{\partial t} + \frac{\partial\left(r^2 uc\right)}{\partial r} = D\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c}{\partial r}\right) + \phi_s. \tag{15}$$

Here, c is the solute concentration in the tumor interstitium ($\frac{\text{g}}{\text{mL}}$), D is the diffusion coefficient ($\frac{\text{cm}^2}{\text{s}}$), and ϕ_s is the distributed source term based on the vessel pore model for transcapillary exchange as given by:

$$\phi_s = L_p\frac{S}{V}(p_v - p)(1-\sigma)c_v + P\frac{S}{V}(c_v - c)\frac{Pe}{e^{Pe} - 1}. \tag{16}$$

Here, σ is the solute reflection coefficient, P is the vascular permeability of the solute ($\frac{\text{cm}}{\text{s}}$) through the vascular wall, and c_v is the vascular solute concentration that decays exponentially o

$$c_v = c_o e^{-t/k_d} \tag{17}$$

where c_o is the initial concentration of blood solutes ($\frac{\text{g}}{\text{mL}}$) and k_d is the half-life circulation of nanocarriers. Additionally, Pe is the Péclet number representing the ratio of convective flux to diffusive flux across the vascular wall as given by:

$$Pe = \frac{L_p(p_v - p)(1-\sigma)}{P}.$$

For large values of the Pe number (Pe>1000), we can simplify (16) by recognizing that the term $\frac{Pe}{e^{Pe}-1}$ approximately goes to 0. The source term for the high-Péclet (HP) regime is given by the following equation:

$$\phi_{s,HP} = L_p\frac{S}{V}(p_v - p)(1-\sigma)c_v. \tag{18}$$

It is assumed that there are initially no solutes in the tumor interstitium before injection; thus, the solute transport model has the following initial condition:

$$c(t = 0, r) = 0. \tag{19}$$

The solute transport model has the following boundary conditions:

$$-D \left.\frac{\partial c}{\partial r}\right|_{r=0} + uc(r=0) = 0,$$
$$c(r=R) = c_\infty,$$

which represents the no-flux condition in the center of the tumor and the fixed continuous concentration at the edge of the tumor, as denoted by c_∞ ($\frac{\text{g}}{\text{mL}}$) representing the concentration in the surrounding tissue.

A.3 Pore Theory

We model the pore theory based on the approach developed by Deen [18] and Bungay and Brenner [19]. Specifically, we assume the vascular pores to be cylindrical and we denote the transvascular hydraulic conductivity L_p, the vascular permeability P, and the solute reflection coefficient σ as follows:

$$L_p = \frac{\gamma r_o^2}{8\mu L} \tag{20}$$
$$P = \frac{\gamma H D_o}{L} \tag{21}$$
$$\sigma = 1 - W \tag{22}$$

where γ is the fraction of vascular surface area occupied by pores, r_o is the pore radius (nm), μ is the blood viscosity (mm Hg -s), and L is the thickness of the vascular wall (cm). D_0 is the diffusion coefficient of the nanocarrier diffusion coefficient in free solution at 37°C given by the Stokes-Einstein relationship as follows:

$$D_0 = \frac{k_B T}{6\pi\mu r_p} \tag{23}$$

where k_B is the Boltzmann constant, T is the absolute temperature, and r_p is the radius of the nanocarrier. H and W, respectively, denote the diffusive and convective hindrance factors and are given by:

$$H = \frac{6\pi\Phi}{K_t},$$
$$W = \frac{\Phi(2-\Phi)K_s}{2K_t},$$

where Φ is the partition coefficient denoting the equilibrium ratio of the average intrapore concentration to the average concentration in the bulk medium. For purely steric interactions between solutes and pore wall, the partition coefficient is given by $\Phi = (1-\lambda)^2$ and $\lambda = \frac{r_p}{r_0}$, where λ is the ratio between the particle size r_p (nm) and the pore size r_0 (nm). Furthermore, the K_t and K_s factors are given by:

$$K_t = \frac{9}{4}\pi^2\sqrt{2}(1-\lambda)^{-5/2}[1 + \sum_{k=1}^{2}\alpha_k(1-\lambda)^k] + \sum_{k=0}^{4}\alpha_{k+3}\lambda^k$$
$$K_s = \frac{9}{4}\pi^2\sqrt{2}(1-\lambda)^{-5/2}[1 + \sum_{k=1}^{2}\beta_k(1-\lambda)^k] + \sum_{k=0}^{4}\beta_{k+3}\lambda^k$$

where α_k and β_k are the coefficients defined in Table 4 below.

Table 4: The hydrodynamic coefficients for the cylindrical pore model are given in this table. [18].

k	1	2	3	4	5	6	7
α_k	-73/60	77293/50400	-22.5083	-5.6117	-0.3363	-1.216	1.647
β_k	7/60	-2227/50400	4.0180	-3.9788	-1.9215	4.392	5.006

B Additional Qualitative Plots

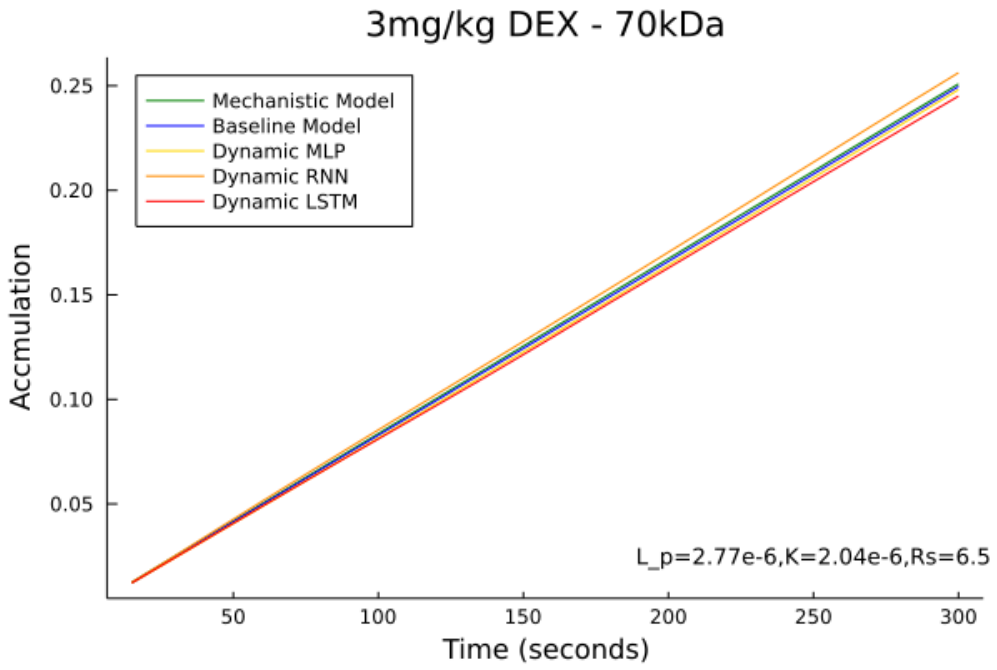

Figure 12: This figure presents the results of each model for the 3 mg/kg DEX treatment with a 70kDa tracer molecule. With this treatment, the TME is effectively normalized and exhibits regularized transport.

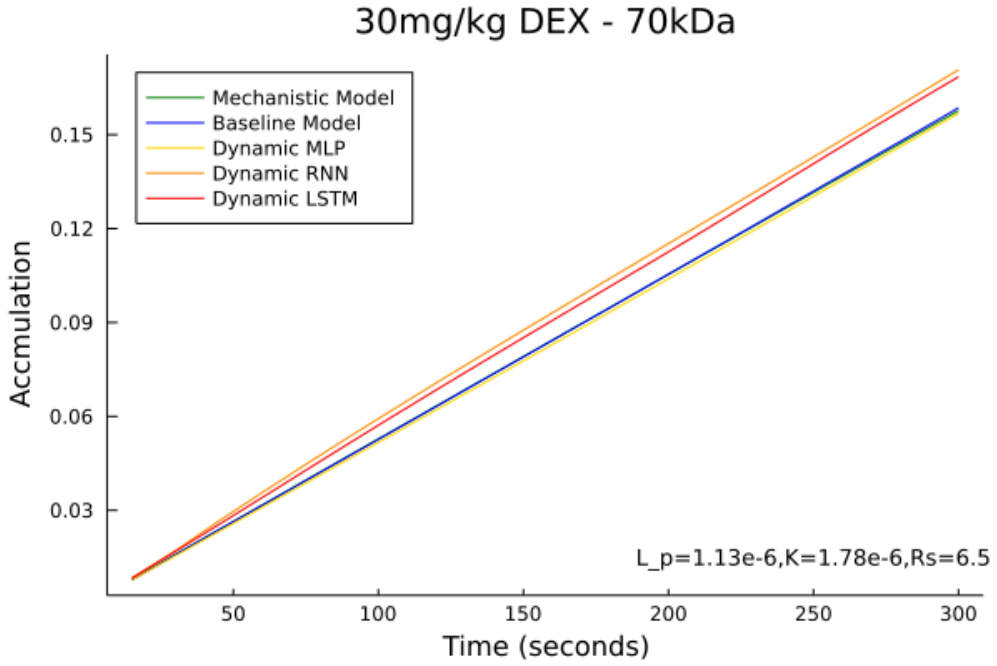

Figure 13: This figure presents the results of each model for the 30 mg/kg DEX treatment with a 70kDa tracer molecule. With this treatment, the treatment is not optimal, but exhibits better transport than the control case.

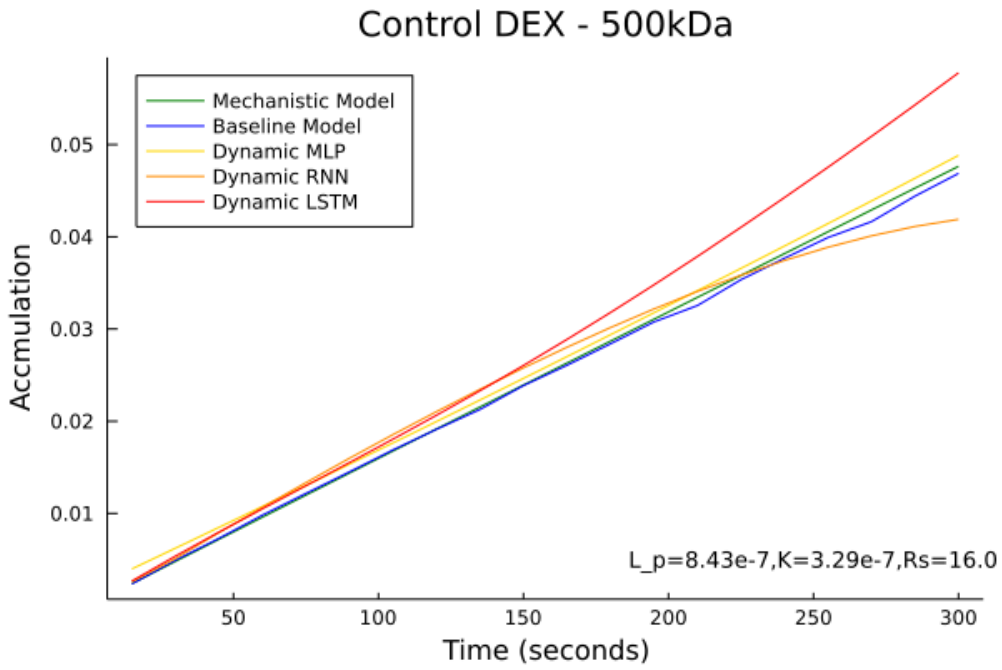

Figure 14: This figure presents the results of each model for the control DEX treatment with a 500kDa tracer molecule. Similarly to the 70kDa tracer control case, this treatment exhibits irregular transport characteristics.

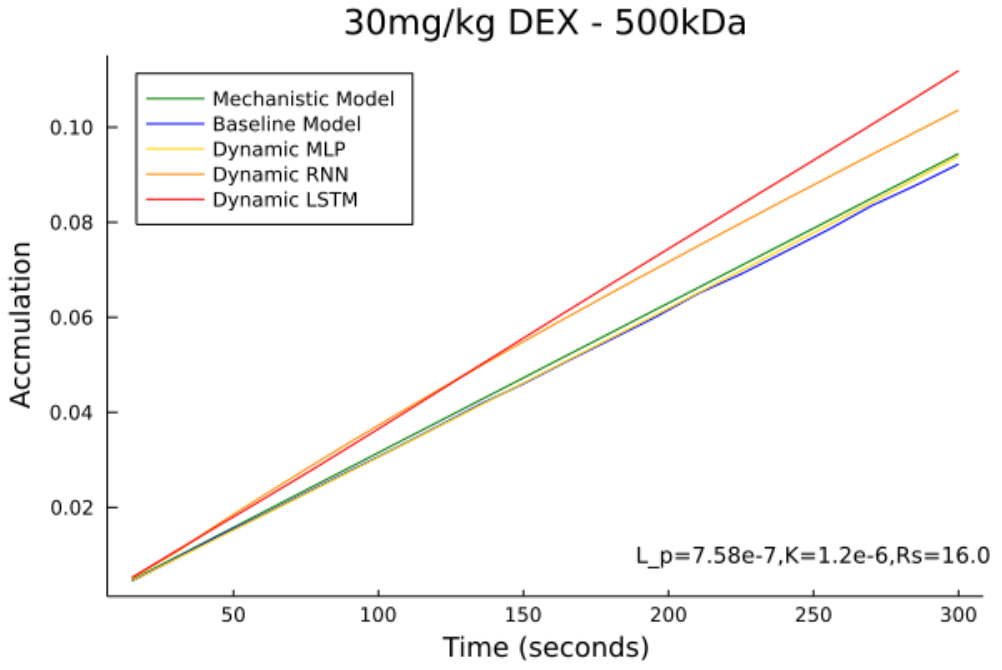

Figure 15: This figure presents the results of each model for the 30 mg/kg DEX treatment with a 500kDa tracer molecule.With this treatment, the treatment is not optimal, but exhibits better transport than the control case. Furthermore, it is worthy noting that recurrent models perform especially poor on this model.

C Link to Repository

The Github repository for this project can be accessed at the following link:
`https://github.com/stmorgenstern/DynamicTransportSurrogates.git`

References

[1] C. Wang et al. "Optimal Therapy Design With Tumor Microenvironment Normalization". In: *AIChE Journal* (2022).

[2] Laurence T Baxter and Rakesh K Jain. "Transport of fluid and macromolecules in tumors. I. Role of interstitial pressure and convection". In: *Microvascular research* 37.1 (1989), pp. 77–104.

[3] Laurence T Baxter and Rakesh K Jain. "Transport of fluid and macromolecules in tumors. II. Role of heterogeneous perfusion and lymphatics". In: *Microvascular research* 40.2 (1990), pp. 246–263.

[4] Laurence T Baxter and Rakesh K Jain. "Transport of fluid and macromolecules in tumors: III. Role of binding and metabolism". In: *Microvascular research* 41.1 (1991), pp. 5–23.

[5] Laurence T Baxter and Rakesh K Jain. "Transport of fluid and macromolecules in tumors. IV. A microscopic model of the perivascular distribution". In: *Microvascular research* 41.2 (1991), pp. 252–272.

[6] John D. Martin et al. "Dexamethasone Increases Cisplatin-Loaded Nanocarrier Delivery and Efficacy in Metastatic Breast Cancer by Normalizing the Tumor Microenvironment". In: *ACS Nano* 13.6 (June 2019), 6396–6408 10.1021/acsnano.8b07865.

[7] I.M Sobol'. "On the distribution of points in a cube and the approximate evaluation of integrals". In: *USSR Computational Mathematics and Mathematical Physics* 7.4 (1967), pp. 86–112. DOI: 10.1016/0041-5553(67)90144-9.

[8] Ludovico Bessi and Christopher Rackauckas. *Surrogates.jl*. 2019. URL: https://github.com/SciML/Surrogates.jl.

[9] Chen Wang et al. "An evaluation of adaptive surrogate modeling based optimization with two benchmark problems". In: *Environmental Modelling & Software* 60 (Oct. 2014), pp. 167–179. DOI: 10.1016/j.envsoft.2014.05.026. URL: https://doi.org/10.1016/j.envsoft.2014.05.026.

[10] Raul Yondo, Esther Andres, and Eusebio Valero. "A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses". In: *Progress in Aerospace Sciences* 96 (Jan. 2018), pp. 23–61. DOI: 10.1016/j.paerosci.2017.11.003. URL: https://doi.org/10.1016/j.paerosci.2017.11.003.

[11] Sarah E. Davis, Selen Cremaschi, and Mario R. Eden. "Efficient Surrogate Model Development: Impact of Sample Size and Underlying Model Dimensions". In: *13th International Symposium on Process Systems Engineering (PSE 2018)*. Elsevier, 2018, pp. 979–984. DOI: 10.1016/b978-0-444-64241-7.50158-0. URL: https://doi.org/10.1016/b978-0-444-64241-7.50158-0.

[12] Christopher Rackauckas and Qing Nie. "Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia". In: *Journal of Open Research Software* 5.1 (2017).

[13] John D. Martin et al. "Dexamethasone increases nanocarrier delivery by normalizing the tumor microenvironment". In: (Under review, 2018).

[14] Joseph Rynkiewicz. "Efficient estimation of multidimensional regression model using multilayer perceptrons". In: *Neurocomputing* 69.7-9 (Mar. 2006), pp. 671–678. DOI: 10.1016/j.neucom.2005.12.008. URL: https://doi.org/10.1016/j.neucom.2005.12.008.

[15] Mike Innes. "Flux: Elegant Machine Learning with Julia". In: *Journal of Open Source Software* (2018 10.21105/joss.00602).

[16] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[17] Alexei Botchkarev. "A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms". In: *Interdisciplinary Journal of Information, Knowledge, and Management* 14 (2019), pp. 045–076. DOI: 10.28945/4184. URL: https://doi.org/10.28945/4184.

[18] WM Deen. "Hindered transport of large molecules in liquid-filled pores". In: *AIChE Journal* 33.9 (1987), pp. 1409–1425.

[19] Peter M Bungay and Howard Brenner. "The motion of a closely-fitting sphere in a fluid-filled tube". In: *International Journal of Multiphase Flow* 1.1 (1973), pp. 25–56.