Hybrid Modelling for Metabolic Pathway Optimization: The Impact of Data Availability and Prior Mechanistic Knowledge

Talitha Aarnoutse

Delft University of Technology



TU Delft

Hybrid Modelling for Metabolic Pathway Optimization: The Impact of Data Availability and Prior Mechanistic Knowledge

by

Talitha Aarnoutse

submitted in partial fulfillment of the requirements for the degree of Master of Science,
in computer science,
at the Delft University of Technology,
to be defended publicly on Wednesday July 2, 2026 at 15:00.

Thesis Advisor:	Dr. T.E.P.M.F. Abeel
Daily supervisor:	Dr. J. Schmitz
Committee member:	Dr. S. Dumančić
Project Duration:	November, 2025 - July, 2026
Faculty:	Faculty of Electrical Engineering, Mathematics & Computer Science, Delft

Cover: Cover image generated using ChatGPT (OpenAI) based on a custom prompt by the author.

TU Delft

Abstract

Metabolic engineering offers a promising route toward sustainable production of valuable chemicals by enabling the design of optimized microbial pathways. However, the vast combinatorial design space and the high cost of experimental validation make it challenging to efficiently identify optimal strain designs. Methods such as kinetic approaches that rely on Ordinary Differential Equations (ODEs) are interpretable, but need detailed knowledge of the underlying pathway. On the other hand, purely data-driven machine learning methods, require a large amount of training data and have more difficulty generalising to unseen conditions. Hybrid modelling approaches, such as Neural Ordinary Differential Equations (Neural ODEs) and Universal Differential Equations (UDEs), have emerged as a promising solution by combining mechanistic knowledge with data-driven learning. Despite their potential, it remains unclear how much data, and what type of data, is required to obtain reliable predictions in practical, data-limited settings. This study systematically investigates the relationship between data availability, model complexity, and predictive performance of hybrid kinetic models for a metabolic pathway producing p-coumaric acid. Using synthetically generated data, hybrid models with varying levels of mechanistic knowledge were trained and compared to fully data-driven approaches, including Neural ODEs and XGBoost. The influence of training set size (number of strains), temporal resolution (number of timepoints), prior mechanistic knowledge, reaction lumping, and observational noise was evaluated across multiple scenarios. Model performance was assessed in terms of prediction accuracy, training stability, and generalization to unseen strains. The results demonstrate that increasing the number of strains is generally more beneficial than increasing the number of timepoints, for predicting final product concentration. For predicting full metabolite concentrations it is more beneficial to increase the number of timepoints, but this requires a more careful trade-off. The hybrid model utilizes the underlying mechanistic knowledge to make moderate predictions even on a very limited number of timepoints. The hybrid models consistently outperform purely data-driven methods in low-data regimes, particularly when prior mechanistic knowledge is incorporated. While additional mechanistic knowledge improves predictive performance and reduces data requirements, it also increases sensitivity to noise and can negatively affect performance in highly constrained settings. Reaction lumping reduces model complexity but typically leads to lower performance compared to full network representations, especially when sufficient temporal data is available. Overall, this study provides practical insights into the trade-offs between data quantity, data type, and model complexity in hybrid metabolic modelling. The findings highlight that efficient experimental design should prioritize increasing the diversity of sampled strains over temporal resolution, and that incorporating mechanistic knowledge is most beneficial when data is scarce but noise levels remain limited.

Contents

Abstract i

1 Introduction 1

2 Research objective 3

3 Methods 4
- 3.0.1 Data generation . . . 4
- 3.0.2 Model architecture . . . 4
- 3.0.3 Model Evaluation and Performance Metrics . . . 7

4 Results 9
- 4.0.1 Synthetic data generation . . . 9
- 4.0.2 Hyper parameter tuning . . . 10
- 4.0.3 Influence of data quantity on hybrid model performance in comparison to step size 11
- 4.0.4 Effect of prior mechanistic network knowledge on the minimal data requirements for reliable model performance . . . 16
- 4.0.5 Effect of reaction lumping on the minimal data requirements for reliable model performance . . . 19
- 4.0.6 Robustness to noise . . . 23

5 Conclusion 25

6 AI Disclosure 26

7 DAIC Acknowledgement 26

References 27

A Hyper parameter tuning 29
- A.0.1 Performance individual hyper parameters . . . 29
- A.0.2 Batch size vs Neural width . . . 31

B Synthetic data generation 32

C Performance in comparison to data quantity 34

D Model performance vs mechanistic knowledge 42

E Model performance vs model complexity 44
- E.0.1 Stoichiometry for lumped networks . . . 44

F Robustness to noise 46

1 Introduction

Metabolic engineering enables the production of valuable compounds by modifying cellular pathways, offering a sustainable alternative to chemical processes that rely on non-renewable resources such as fossil fuels [14]. By genetically modifying microorganisms, it enables the production of food additives, pharmaceuticals, fuels, and many other products. These modifications redirect metabolic flux toward a desired product, by for example decreasing metabolic flux toward competing pathways that lead to unwanted byproducts [1]. However, optimizing these systems remains highly challenging due to the large combinatorial design space and the high cost and time associated with experimental validation.

Traditionally, pathway optimization has focused on identifying rate-determining steps and improving them sequentially [7]. While this approach can be effective, it often fails to capture the complex interactions between different pathway components. In contrast, combinatorial pathway optimization allows for the optimization of multiple pathway components simultaneously [9]. In combinatorial pathway optimization different variants of a metabolic pathway are created by assembling different strain designs. These strain designs are constructed by combining different genes with different promoters and other regulatory elements. However, due to the large size of DNA libraries, the number of possible designs often far exceed what can be experimentally evaluated [24].

A method that is often used to address this issue is the design-build-test-learn (DBTL) framework. In each DBTL cycle an initial design is created during the design phase, which is then built and tested for increased production. The resulting data is then analysed to identify influential components and guide the design of potentially improved strains for the next iteration. This iterative framework provides a structured approach to exploring the design space and has been successfully used to guide the identification of improved strain designs. However, this process can still be very time consuming and costly. As a result, there is a growing focus on strategies that improve process efficiency while reducing the need for extensive experimental data.

Over the last few years, machine learning has been increasingly integrated into the DBTL workflow to recommend new strain designs [19] [24] [17]. These models can learn complex relationships directly from experimental data without requiring detailed mechanistic knowledge of the underlying system. As a result, they are well suited to handling high-dimensional design spaces. These approaches have been shown to yield strong results [23], but they come at the cost of reduced interpretability. Machine learning models often function as black boxes, making it difficult to understand the biological mechanisms underlying their predictions. Furthermore, their ability to generalize beyond the training data can be limited, particularly in data-scarce settings.

In contrast, mechanistic kinetic models provide an interpretable framework for describing metabolic systems. These models use systems of Ordinary Differential Equations (ODEs) to describe how metabolite concentrations evolve over time, allowing for direct simulation of the effects of parameter perturbations. As such, they can provide valuable insights into which reactions or enzymes are rate-limiting and how changes in pathway components influence system behaviour. However, constructing accurate kinetic models requires detailed knowledge of reaction kinetics and parameters, which is often incomplete or unavailable in practice [8] [10].

The strengths and weaknesses of both these methods has motivated the development of hybrid modelling approaches, that combine mechanistic knowledge with data-driven learning. Methods such as; Universal Differential Equations (UDEs) and Neural Ordinary Differential Equations (NODEs), integrate neural networks into differential equation frameworks to model partially known systems. In these approaches, known parts of the system are described mechanistically, while unknown dynamics are learned using neural networks.

Recent studies have demonstrated the potential of NODEs and UDEs in systems biology [4] [21]. UDEs have been applied to chemical reaction networks and biological pathways, where they integrate mechanistic parameters with neural network components to capture unknown dynamics.

Alongside their promise, hybrid models also introduce new challenges. Training these models can be difficult due to stiff dynamics, limited and noisy data, and the interplay between mechanistic and learned components. Several strategies have been proposed to address these issues. Multi-start optimization, hyperparameter sampling, and regularisation have been shown to improve both predictive accuracy and interpretability [16].

Neural ODEs have demonstrated the ability to generalize to unseen knockout scenarios and irregularly sampled data. They can predict gene expression and metabolic fluxes beyond the training period and accommodate irregular sampling intervals [18] [6] [25]. However, hybrid models can sometimes suffer from unrealistic solutions, such as negative values for biochemical quantities. This problem can be alleviated by incorporating a state-dependent non-negativity factor [15]. Furthermore, it has been shown that incorporating physical constraints can guide the UDE towards more physiologically plausible solutions [3]. These constraint-based strategies are important for applications in metabolism, where concentrations, fluxes, and reaction rates are strictly limited by biological feasibility.

Research has also investigated the interpretability and identifiability of these hybrid methods. Many biologically realistic models may not be fully identifiable, meaning that multiple parameter combinations can produce the same output [2]. Furthermore, studies have distinguished between parametric and functional identifiability, demonstrating that functional identifiability is most critical for predictive power [11].

Despite these advances, a key question remains insufficiently explored: how much data, and what type of data, is required to train reliable hybrid models? In practical applications, experimental data are often scarce, expensive, and noisy. While previous work has addressed issues related to model architecture and training stability, there has been relatively little systematic investigation into how different data regimes influence model performance. In particular, it remains unclear how the number of strains, the temporal resolution of measurements, and the amount of prior mechanistic knowledge interact to determine predictive accuracy, stability, and generalization.

This study aims to address this gap by systematically analysing the influence of data quantity, data type, and model complexity on the performance of hybrid kinetic models for metabolic pathway optimization. By comparing different combinations of training set size, temporal resolution, and levels of mechanistic knowledge, this work seeks to identify the minimal data requirements for achieving reliable predictions. Ultimately, the goal is to provide practical insights into how experimental and modelling efforts can be optimally balanced, thereby improving the efficiency of the DBTL cycle and advancing the application of hybrid modelling in metabolic engineering.

2 Research objective

Despite recent progress in hybrid kinetic modelling, it remains unclear how much data, and what kind of data, is actually needed to obtain reliable predictions. Experimental data is costly to obtain and often sparse, making it crucial to understand which data regimes provide the greatest benefit. To address this gap, this study focuses on systematically evaluating the relationship between data availability and model performance. Therefore, the research question that this study aims to answer is:

How do data quantity and data type influence the predictive performance, stability, and generalization of hybrid kinetic models across different levels of model complexity?

In this context, data quantity refers to the number of strains and data type refers to the structure of the available data, including the temporal resolution and the number of measured metabolites.

More specifically, this study focuses on the following sub-questions:

- How does the number of strains relate to the amount of time steps?
- How does prior mechanistic knowledge of the network affect the minimal data requirements for reliable model performance?
- How does reaction lumping affect the minimal data requirements for reliable model performance?

This study aims to systematically investigate how different data regimes affect the predictive accuracy, stability, and generalization of hybrid kinetic models. To achieve this, hybrid models will be constructed with different levels of complexity. The underlying pathway of each of these models is the same, the difference lies in which part of the pathway is assumed to be prior knowledge. Model complexity is defined as the proportion of reactions described mechanistically versus those learned by the neural network. For the creation of each model, the Mass-conserving Neural ODE approach will be used, as previous work has demonstrated that this architecture outperforms other hybrid modelling approaches [12].

For each model, it will be investigated how predictive performance depends on the availability and type of data. More specifically, the influence of the number of strains and the number of time steps will be investigated.

All scenarios are evaluated based on three criteria:

- **Stability of model training**
  - Quantified as the proportion of multi-start initializations that successfully converge
- **Goodness of fit to observed data**
- **Performance on unseen data**
  - Extrapolation to new strains outside of the training data

By systematically varying both model complexity and data availability, this study aims to identify the minimal data requirements for obtaining stable and predictive hybrid kinetic models. Ultimately, the goal is to determine which types and quantities of data are optimal for reliable hybrid modelling at different levels of complexity. Understanding these relationships is essential for designing efficient experiments, as it informs whether resources should be allocated toward increasing the number of strains or improving temporal resolution. This information can help guide experimental design decisions in metabolic engineering and contribute to more efficient and targeted DBTL cycles.

3 Methods

3.0.1 Data generation

To systematically evaluate model performance across a wide range of data regimes, a synthetic dataset was generated using the JaxKineticModel package [22]. JaxKineticModel is a computational framework for modelling biochemical reaction networks, built on the JAX library. Within this framework, reaction systems can be defined using kinetic rate equations and subsequently simulated over time to obtain the metabolite trajectories. For this study, a kinetic model describing a batch process of the production of p-coumaric acid was used. The model contains eight different metabolites and ten reactions, that are all described by Michaelis-Menten kinetics [20] (Equation 3.1). Michaelis-Menten kinetics describes how the rate of a reaction depends on the concentration of the substrate. In biological systems, enzymes play an important role in catalysing reactions. With a high enough substrate concentration all enzymes become saturated, as a result the reaction rate approaches a maximum limit beyond which it can no longer increase. The Michaelis-Menten equation captures this behaviour:

$$v = \frac{V_{\max}[S]}{K_m + [S]} \tag{3.1}$$

Where v represents the reaction rate, $[S]$ the substrate concentration, V_{max} the maximum achievable reaction rate and Km the Michaelis constant. The model contains a total of 23 kinetic parameters that describe the reactions. Together the model describes a pathway from substrate to product containing lumped reactions of the glycolysis, the pentose-phosphate pathway, and the shikimate pathway [13]. To generate time-series data, the kinetic model is solved to obtain metabolite concentration trajectories over time. Given a set of timepoints, initial conditions and the parameters describing each reaction, the model simulates the trajectories over time and returns the concentration per time step for each of the metabolites.

To simulate the time-series data for multiple different strains, the V_{max} parameters of the reactions were systematically perturbed. For each parameter, a set of possible values was defined, representing different expression levels. New strain designs were then generated by sampling combinations of these perturbations. For each of the strain designs, the kinetic model was used to simulate the fermentation over time. The final dataset contains 40 time steps for each of the metabolites, which is subsampled to test the models on different amounts of training data.

3.0.2 Model architecture

Hybrid

The hybrid model combines mechanistic knowledge with data-driven learning using a mass-conserving Neural Ordinary Differential Equation (MCNO) framework [12]. In this approach, the known dynamics of the metabolic network are described using mechanistic reaction kinetics, while unknown components are represented by a neural network. The two components are integrated through an additive structure constrained by the stoichiometric matrix of the system (Figure 3.1, Equation 3.2).

$$\frac{d\mathbf{y}}{dt} = S\left(\mathbf{v}_{\text{mechanistic}}(\mathbf{y}, t) + \mathbf{v}_{\text{NN},\theta}(\mathbf{y}, t)\right) \tag{3.2}$$

Here, S denotes the stoichiometric matrix, $v_{mechanistic}$ represents fluxes calculated from known kinetic expressions, and $v_{NN,\theta}$ represents learned corrections to these fluxes predicted by a neural network.

A binary mask is used to define which reactions are treated mechanistically and which are learned by the neural component. For masked reactions, the neural network is responsible for predicting the full reaction flux, whereas for unmasked reactions, the mechanistic model provides the primary description.

The data is normalised to make sure the metabolites are weighed equally in the loss function. The loss function consists of three components; a mean squared error term, an L_2 regularization term on the neural network parameters and a penalty for invalid numbers (Equation 3.3).

$$\mathcal{L}_{\text{total}} = \frac{1}{N}\sum_{i=1}^{N} m_i(\hat{y}_i - y_i)^2 + 10^9 \frac{N_{\text{invalid}}}{N} + \lambda \sum_{p\in\theta} p^2 \tag{3.3}$$

Where m_i is a masking term that excludes non-finite values, N is the total number of predictions, $N_{invalid}$ is the number of excluded predictions, $\hat{y}_i$ and y_i are the true and predicted value, λ is the regularization coefficient and θ are the weights of the neural network. To ensure numerical values, the infinite values are masked out (Equation 3.4). A penalty term is applied to strongly guide the optimizer away from unstable regions.

$$m_i = \begin{cases} 1, & \text{if } \hat{y}_i \text{ and } y_i \text{ are finite} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

To prevent the model from producing negative values, a state-dependent non-negativity constraint is applied to the fluxes. Reaction rates are multiplied by a smooth sigmoid function when substrate concentrations approach zero. This prevents abrupt discontinuities while reducing the likelihood of negative concentrations. During inference, any remaining negative values are clamped to zero.

To increase training stability event-based stopping was added [12] (Equation 3.5).

$$\text{Stop integration if} \quad \exists i \,:\, y_i(t) < -10 \quad \vee \quad y_i(t) > 10^7 \tag{3.5}$$

This makes sure that integration is stopped early when the concentrations reach highly implausible values. To improve numerical stability during training, the neural network weights were initialized near zero by scaling them by 10^2 . This limits large initial perturbations to the mechanistic dynamics and prevents unstable trajectories during the early stages of optimization [12]. The model uses an initial small step size, which is automatically adjusted using PID-controlled error estimation based on the specified relative and absolute tolerances. Training is performed using the AdaBelief optimizer with global gradient norm clipping to prevent exploding gradients. A learning rate warm-up strategy was applied, where the learning rate increases from zero to its maximum value over an initial warm-up phase. This improves stability during early training by preventing large parameter updates that could destabilize the ODE solver.

The neural component consists of a fully connected multilayer perceptron (MLP) with tanh activation functions and dropout regularization. The network takes normalized metabolite concentrations and kinetic parameters as input and outputs a vector of reaction flux corrections, one for each reaction in the metabolic network. The models were trained for a maximum of 5000 steps and a maximum of 3 hours. The stopping criteria for training is based on validation loss convergence. Validation loss is calculated every 50 steps for 50 independent validation strains. To avoid models from stopping training early due to an early dip in loss, the patience was set to 500 steps. Checkpoints during training were saved every 200 steps. After training it was checked whether an earlier checkpoint obtained a lower validation loss, in case of overfitting towards the end of training due to the high patience. In that case the earlier checkpoint was used as the final model.

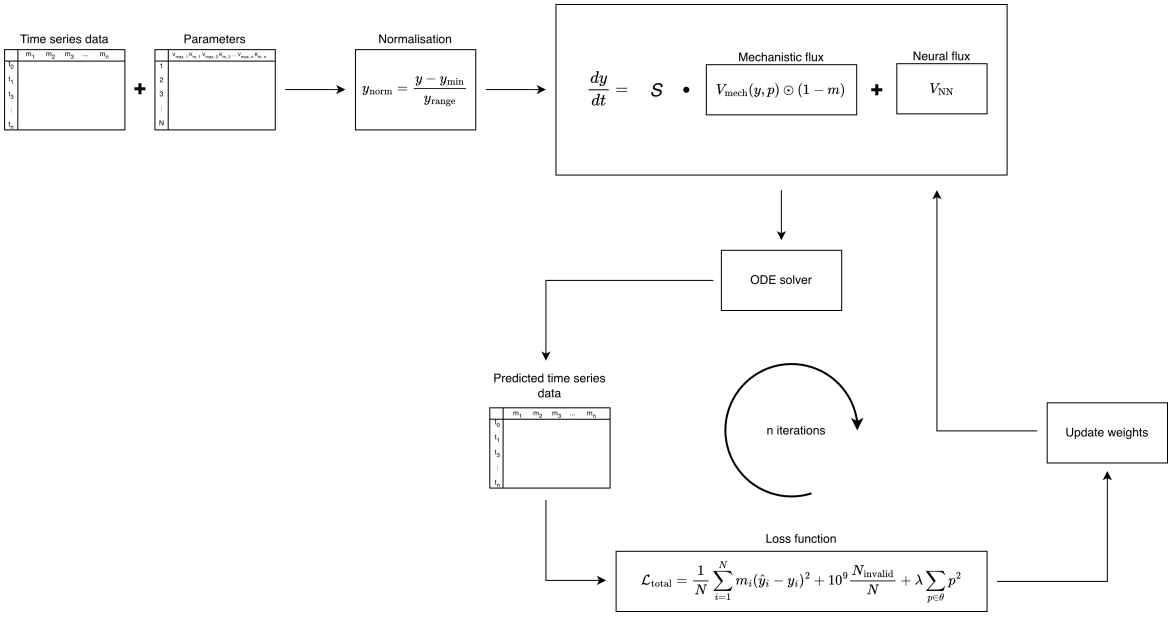


Figure 3.1: Schematic representation of the mass-conserving Neural Ordinary Differential Equation (MCNO) architecture. Mechanistic reaction fluxes are additively combined with neural network–predicted flux corrections and constrained by the stoichiometric matrix S. A reaction mask specifies which reactions are modelled mechanistically and which are learned by the neural component. The resulting flux vector is integrated over time to obtain metabolite concentration trajectories.

NODE

In addition to the hybrid model, a fully data-driven Neural ODE is used as a baseline for comparison. The architecture of the Neural ODE is similar to that of the hybrid model, with the key difference being the absence of mechanistic knowledge. (Figure B.2). Instead of modelling individual reaction fluxes, the Neural ODE directly learns the full system dynamics (Equation 3.6).

$$\frac{d\mathbf{y}(t)}{dt} = \mathrm{NN}_\theta(\mathbf{y}(t), t) \tag{3.6}$$

Whereas the hybrid model uses a neural network to predict a vector of reaction flux corrections, which are then combined with mechanistic fluxes and mapped to the state derivatives via the stoichiometric matrix, a Neural ODE directly learns the full state dynamics without explicit knowledge of individual reactions or stoichiometric structure.

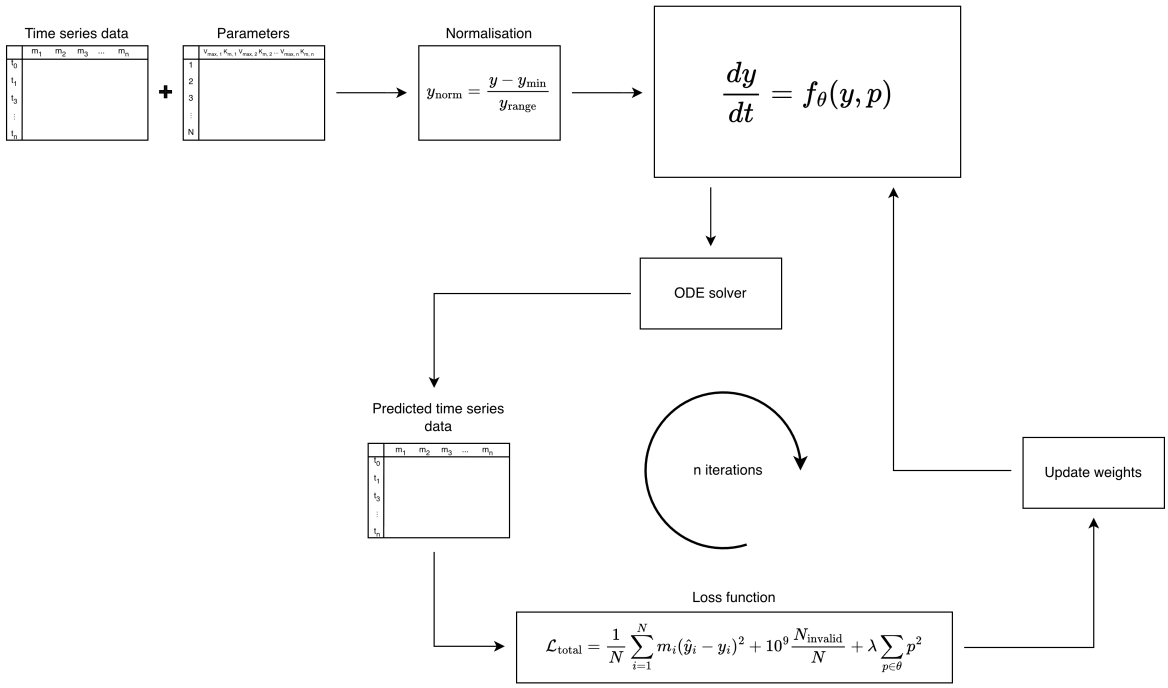


Figure 3.2: Schematic overview of the Neural Ordinary Differential Equation (NODE) architecture. The neural network directly predicts the time derivatives of metabolite concentrations from the system state, without incorporating explicit mechanistic or stoichiometric constraints. The predicted dynamics are integrated to produce concentration trajectories.

XGBoost

As a non-dynamical baseline, an XGBoost model is used to predict metabolite concentrations without explicitly modelling temporal dynamics. The model is implemented as a multi-output regressor by wrapping the XGBoost regressor in the MultiOutputRegressor class from sklearn.

Each metabolite is predicted using a separate XGBoost model. The input features consist of strain parameters and initial metabolite concentrations. The model predicts the metabolite concentration at a specific timepoint. By concatenating predictions across all timepoints, full trajectories are reconstructed.

Unlike the ODE-based models, the XGBoost approach treats each timepoint independently and does not exploit temporal continuity. This makes it less suitable for interpolation between observed data points, but provides a useful baseline for assessing the benefits of incorporating temporal and mechanistic structure.

3.0.3 Model Evaluation and Performance Metrics

To quantitatively evaluate and compare model performance across different training regimes and network architectures, standardized error metrics based on the Root Mean Squared Error (RMSE) and the Normalized Root Mean Squared Error (NRMSE) were computed. All calculations were performed on concentration trajectories, excluding the initial condition at $t = 0$. To prevent numerical instabilities or divergent trajectories from biasing aggregate performance statistics, a masking mechanism was implemented to exclude any non-finite predictions (NaN or $\pm\infty$) from the error calculations. The fraction of excluded points was tracked to ensure that an excessive number of predictions was not removed, which could otherwise bias the results.

The global trajectory error across all species and time points was evaluated by first computing the species-specific RMSE for each metabolite s, and then averaging over all metabolites:

$$\mathrm{RMSE}_s = \sqrt{\frac{1}{N_{s,\mathrm{finite}}} \sum_{i \in S_{\mathrm{eval}}} \sum_{t \in T_{\mathrm{eval}}} (y_{i,t,s} - \hat{y}_{i,t,s})^2}. \tag{3.7}$$

Here, $y_{i,t,s}$ and $\hat{y}_{i,t,s}$ denote the ground-truth and predicted concentrations of metabolite s for strain i at time point t, respectively. $N_{s,\mathrm{finite}}$ represents the total number of finite coordinates included in the calculation.

To facilitate comparison across metabolites with highly varying concentration scales, the species-specific NRMSE was calculated by normalizing RMSE_s by the global concentration range of that metabolite, R_s. To ensure fair comparison between different scenarios, the metabolite range is computed across all strains and time points in the dataset, including both test and train data:

$$\mathrm{NRMSE}_s = \left(\frac{\mathrm{RMSE}_s}{R_s} \right) \times 100, \tag{3.8}$$

where

$$R_s = \max_{i,t} (y_{i,t,s}) - \min_{i,t} (y_{i,t,s}) \,. \tag{3.9}$$

The aggregate trajectory metric, $\mathrm{NRMSE}_{\mathrm{all}}$, is then defined as the average of the species-specific NRMSE values:

$$\mathrm{NRMSE}_{\mathrm{all}} = \frac{1}{S} \sum_{s=1}^{S} \mathrm{NRMSE}_s, \tag{3.10}$$

where S is the total number of evaluated metabolites.

Finally, to evaluate endpoint yield prediction accuracy for the target product, p-coumaric acid (p-CA), the NRMSE is calculated at the final experimental time point ($t = t_{\mathrm{end}}$):

$$\mathrm{NRMSE}_{\mathrm{p-CA,\ final}} = \left(\frac{\sqrt{\frac{1}{N_{\mathrm{final,\ finite}}} \sum_{i \in S_{\mathrm{eval}}} (y_{i,t_{\mathrm{end}},\mathrm{p-CA}} - \hat{y}_{i,t_{\mathrm{end}},\mathrm{p-CA}})^2}}{R_{\mathrm{p-CA}}} \right) \times 100. \tag{3.11}$$

All metric calculations were implemented using JAX and NumPy to enable efficient, vectorized computation across large datasets.

4 Results

4.0.1 Synthetic data generation

To effectively analyse the performance of hybrid models on different amounts and types of data, a dataset of the fermentation of p-coumaric acid was synthetically generated. The advantage of synthetically generating the data is that this approach is not limited by the availability of experimental data. The models can be tested on a variety of different training sets, while having enough data left to construct an independent and unbiased test set. Using the JaxKineticModel package, a kinetic model was constructed of a pathway leading from substrate to p-coumaric acid, including biomass production and respiration reactions (Figure 4.1 A). The kinetic model is described by a set of ODEs that define how the concentration of each metabolite changes over time. All reactions in this network are described using Michaelis Menten kinetics, and each reaction is therefore described by the parameters V_{max} and K_m. To create the dataset of a large number of different strains, the V_{max} of the reactions were perturbed to create a number of distinct sets of parameters that each represent a different strain. Only the maximum reaction rate was perturbed, as changing the K_m would mean changing the enzyme that catalyses the reaction. This study focuses on simulating variations in promoter strength, as these represent more biologically relevant perturbations.

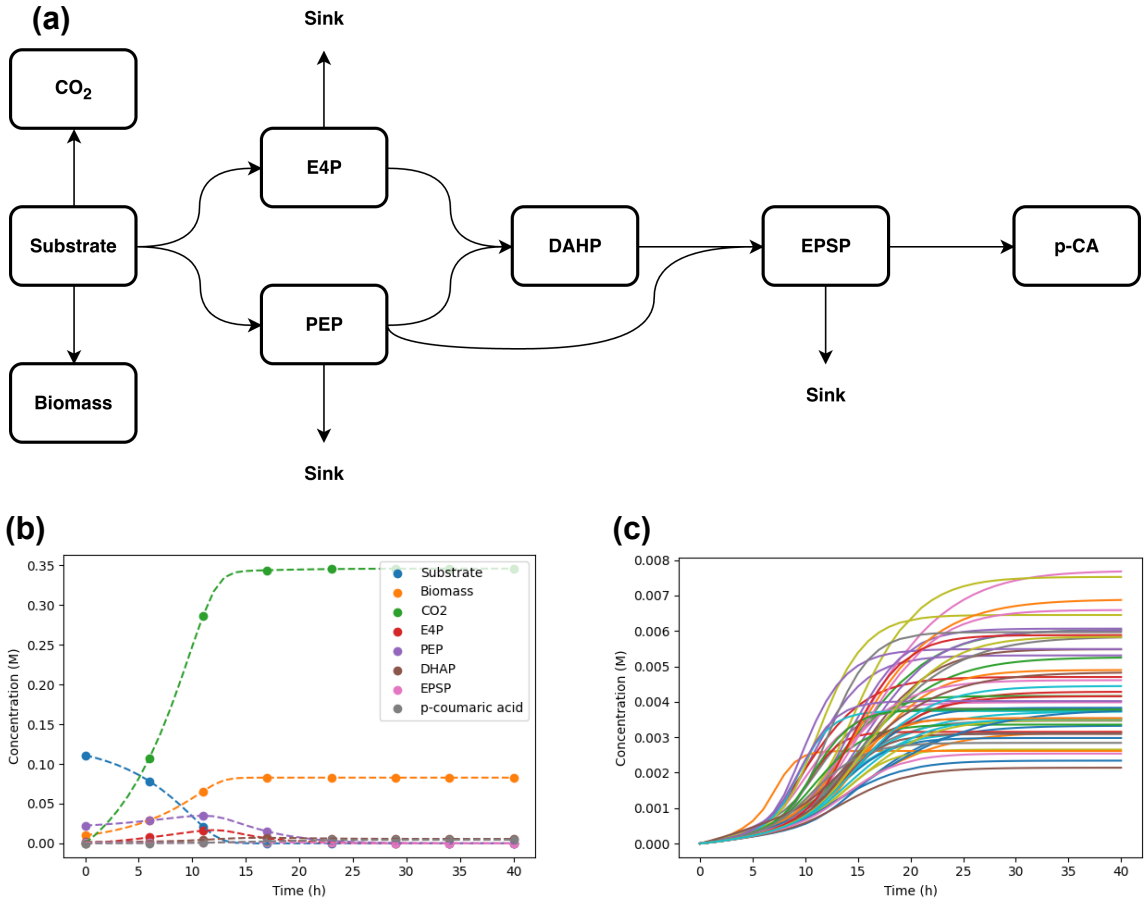


Figure 4.1: Overview of synthetic dataset generation. (a) Representation of the p-coumaric acid metabolic pathway. (b) Example metabolite concentration trajectories over time; dashed lines indicate full simulations (40 timepoints), while markers indicate subsampled observations. (c) Final product trajectories for a set of perturbed strains.

4.0.2 Hyper parameter tuning

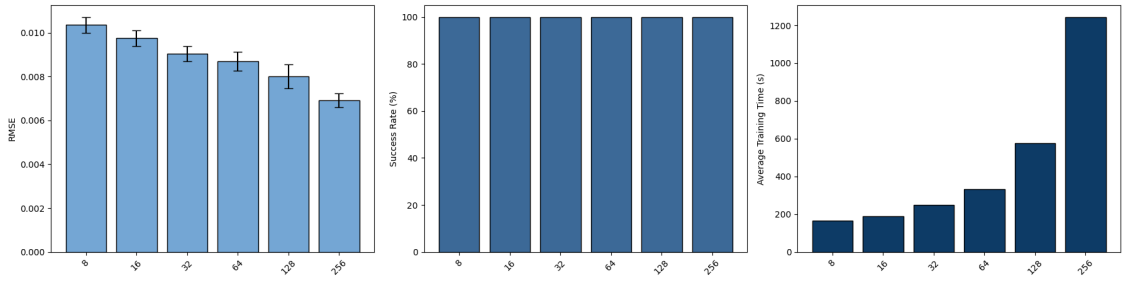
To identify an effective set of hyperparameters, a random search was performed over a predefined grid of configurations. A total of 700 hyperparameter combinations were evaluated, where each configuration is trained for three different random seeds to account for variability in model initialization. All models were trained on a dataset consisting of 250 strains and evaluated on an independent test set of 50 strains.

A random search strategy was chosen instead of an exhaustive grid search due to the high dimensionality of the hyperparameter space. The number of possible grid combinations grows exponentially with the number of parameters, making full coverage computationally infeasible. Moreover, grid search allocates equal resolution to all hyperparameters, even though in practice only a subset typically has a strong influence on model performance, while others have relatively minor effects. As a result, large portions of a grid can be spent evaluating uninformative or suboptimal combinations. Random search is more efficient in these scenarios because it explores a broader range of configurations and is more likely to sample effective regions of the space when only a few hyperparameters dominate performance.

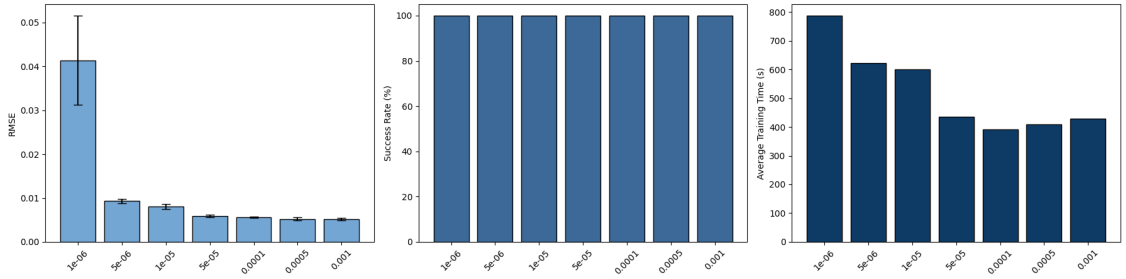
The best-performing configuration was selected based on mean prediction error, variability across runs, and total training time. To further refine this configuration, a local optimization procedure (hill climbing) was applied to the most influential numerical hyperparameters.

To assess the relative importance of individual hyperparameters, performance was evaluated across a range of values for each parameter (Figure 4.2, A.1). The results indicate that the learning rate, neural network width, and batch size have the strongest impact on model performance. Increasing the network width and batch size generally leads to improved predictive accuracy, likely due to increased model capacity and more stable gradient estimates during training. However, these improvements are accompanied by a substantial increase in training time, particularly when using large training datasets.

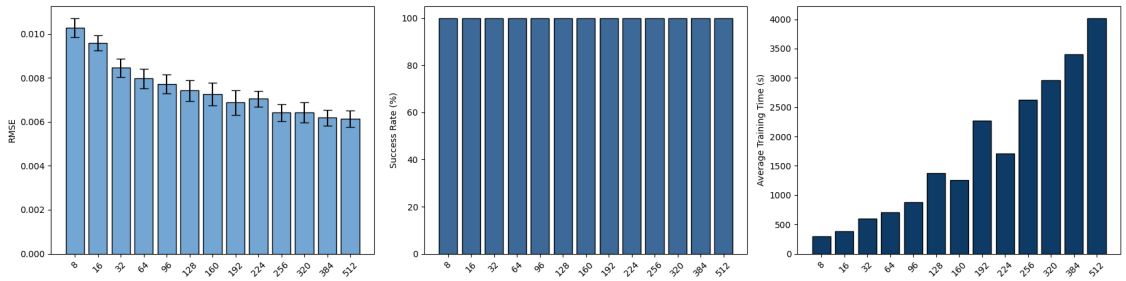
To balance predictive performance and computational efficiency, a grid of batch size and network width values was evaluated for larger training set sizes (Figure A.2). Based on these results, intermediate values were selected that significantly reduce training time while maintaining comparable predictive performance. This trade-off enables the evaluation of a larger number of experimental scenarios, which is essential for the systematic analysis of data regimes performed in this study. As the primary objective of this work is to identify general patterns rather than to optimize performance for a single scenario, slightly reduced accuracy was considered acceptable in exchange for broader experimental coverage.



(a) Batch size



(b) Learning rate



(c) Network width

Figure 4.2: Effect of key hyperparameters on model performance. From left to right: the average root mean squared error (RMSE) across all metabolites and timepoints, training success rate, and average training time. Panels show results for (a) batch size, (b) learning rate, and (c) neural network width.

4.0.3 Influence of data quantity on hybrid model performance in comparison to step size

Obtaining a complete mechanistic description of metabolic pathways is often challenging and resource-intensive. At the same time, purely data-driven approaches require large amounts of training data and may struggle to generalize beyond the observed domain. Hybrid modelling approaches, such as Universal Differential Equations (UDEs), aim to combine the strengths of both approaches by integrating mechanistic knowledge with data-driven components. In this work, the mass-conserving Neural ODE architecture is used, as it has been shown to achieve strong predictive performance in previous studies [12].

Despite the promising capabilities of hybrid models, it remains unclear how the quantity and structure of available data influence model performance. In particular, experimental data in metabolic engineering is often limited, and it is therefore important to understand whether it is more beneficial to increase the number of strains (data diversity) or the number of timepoints per strain (temporal resolution). Practically, this corresponds to a trade-off between high-throughput screening of many strains with limited measurements, and detailed time-series measurements for a smaller number of strains.

To address this question, hybrid models were trained on datasets with varying numbers of strains and timepoints. Training set sizes ranged from 8 to 500 strains, while the number of observed timepoints per strain ranged from 1 to 30. By systematically varying the size of the training data, the goal is to observe an optimum and determine which data type has the greatest impact. The hybrid model used in these experiments assumes that respiration and biomass formation are known and modelled mechanistically, while all other reactions are treated as unknown and learned by the neural network. This reflects a realistic scenario in which core metabolic processes are well-characterized, while product-specific pathways remain less understood.

Although all kinetic parameters are included in the dataset, this does not imply that the full mechanistic mapping is known. In practice, this corresponds to a situation where experimental inputs, such as promoter strengths, are available, but their exact mapping to model parameters is unknown. Since the data is synthetically generated based directly on the parameter values and not on the promoter strengths, these are not available in this case. Therefore, in practice this would require an extra mapping from the promoter strength, to the parameter perturbations. However, this is a different problem and lies outside the scope of this study.

The performance of the hybrid model was compared to a fully data-driven Neural ODE (NODE) and an XGBoost model. The NODE learns the full system dynamics without incorporating mechanistic knowledge, while the XGBoost model predicts metabolite concentrations independently at each timepoint without modelling temporal dynamics. For each configuration, models were trained on 10 different subsets and 5 random seeds per subset, resulting in a total of 50 trained models per scenario. Performance was evaluated using normalized root mean squared error (NRMSE) across all metabolites and timepoints.

The results reveal clear differences in how model performance depends on data quantity and temporal resolution (Figure 4.3). The XGBoost model requires a large number of timepoints to accurately reconstruct metabolite trajectories, as it does not exploit temporal continuity and is therefore unable to interpolate effectively between observations. The XGBoost is able to achieve a low error in the highest amount of data availability, outperforming the other two methods. In contrast, both the hybrid model and the Neural ODE are able to interpolate between sparse timepoints, outperforming the XGBoost in low data-regimes.

The Neural ODE struggles to reconstruct full trajectories when trained on very limited temporal data, particularly when only the final timepoint is observed. The hybrid model shows improved performance in this setting due to the incorporation of mechanistic knowledge. For example, the hybrid model is able to capture substrate dynamics even when trained on a single timepoint. As it encodes known processes such as respiration and biomass formation, which contribute to a large proportion of the substrate consumption. This provides a strong inductive bias that allows the model to infer plausible trajectories with minimal data (Figure C.3).

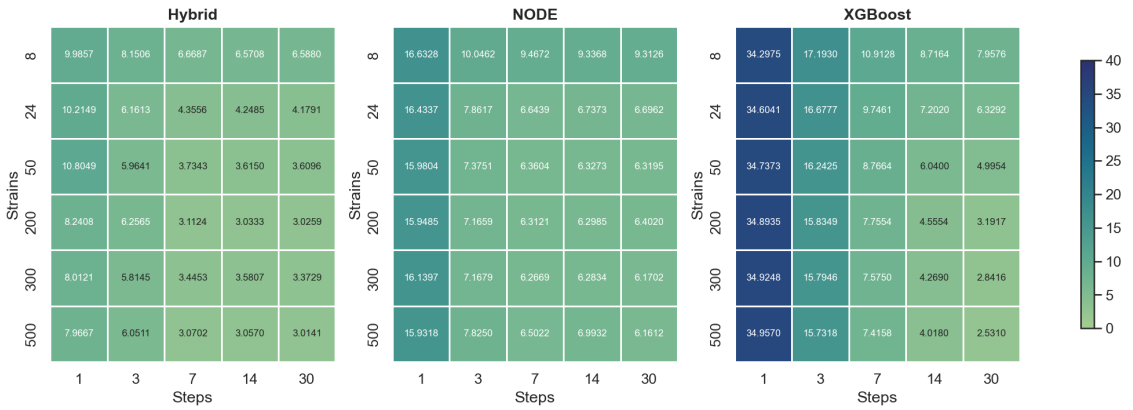


Figure 4.3: Heatmaps of mean normalized root mean squared error (NRMSE) across combinations of training set size (number of strains) and temporal resolution (number of timepoints). Results are shown for the hybrid model (left), Neural ODE (middle), and XGBoost (right), evaluated over all metabolites and timepoints (40 timepoints).

Table 4.1: Significant improvement when training on more than 7 timepoints.

Strains	Mean NRMSE (7 TP)	Mean NRMSE (> 7 TP)	% Improvement	P-value	Significant
8	6.669	6.579	1.34	0.437	No
24	4.356	4.214	3.26	0.123	No
50	3.734	3.612	3.27	0.184	No
200	3.112	3.030	2.66	0.088	No
300	3.445	3.477	-0.91	0.762	No
500	3.070	3.036	1.13	0.408	No

Beyond approximately 200 training strains, performance gains begin to saturate for all model types (Figure 4.4). For the NODE the training took longer to such an extent that it didn't converge within the set time limit of three hours. This is why the predictions get worse for the higher number of strains. From a certain amount of timepoints, adding more timepoints doesn't significantly improve the performance of the hybrid model. Increasing the number of timepoints above seven only does not lead to statistically significant improvements. (Table 4.1). Therefore, the optimal number of timepoints appears to be approximately seven. In settings with a low availability of data it is important to make a trade-off between the number of timepoints and the number of strains. Strictly prioritizing one of the other does not lead to the most optimal results.

Training stability was found to be comparable across models after hyperparameter tuning, with no major convergence failures observed (Figure C.1). However, when training the hybrid model on very limited data (1 timepoint and 8 strains) it starts to make invalid predictions (Figure C.2). This doesn't happen for any of the other configurations, suggesting that the hybrid model is slightly more unstable in the extremely low data-regimes.

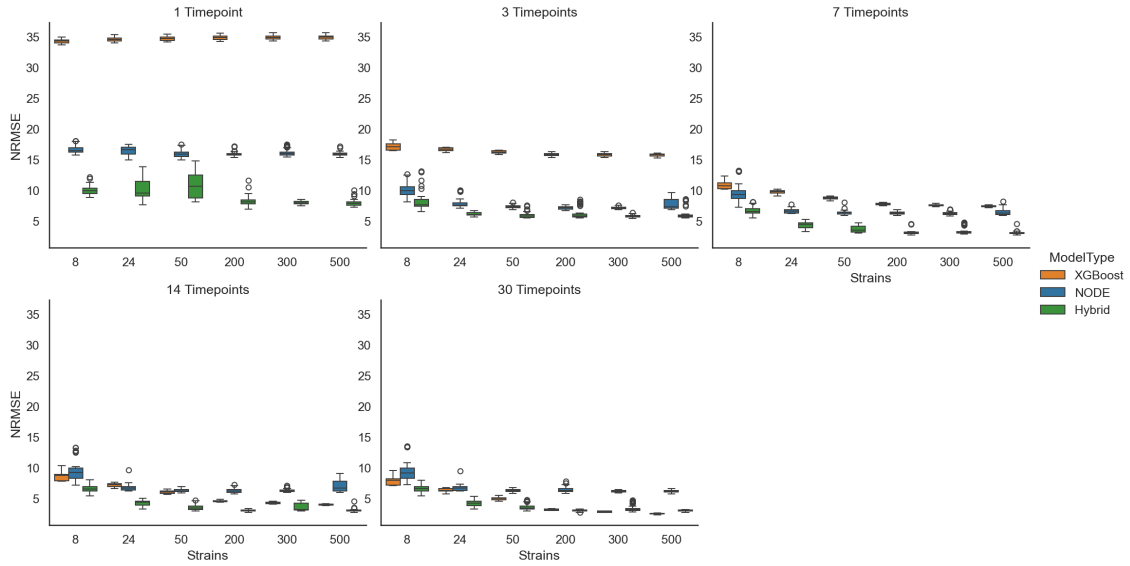


Figure 4.4: Distribution of model performance across scenarios. Boxplots show the median, interquartile range, and outliers of the NRMSE across all model types, evaluated over all metabolites and timepoints.

The XGBoost model achieves the lowest training error and exhibits minimal variance across runs, indicating strong fitting ability. However, it also shows the largest gap between training and test performance, indicating overfitting. The hybrid model performs slightly better than the NODE and slightly worse than the XGBoost. The variance over the runs and the error increases slightly when increasing the number of steps from 1 to 7, after that it stays consistent. The more timesteps added, the more XGBoost overfits to the training data (Figure 4.5). XGBoost does best at reproducing the training data,

but it overfits to the training data. All models overfit in low data regimes, but the NODE and hybrid model improve more than XGBoost when increasing the number of strains.

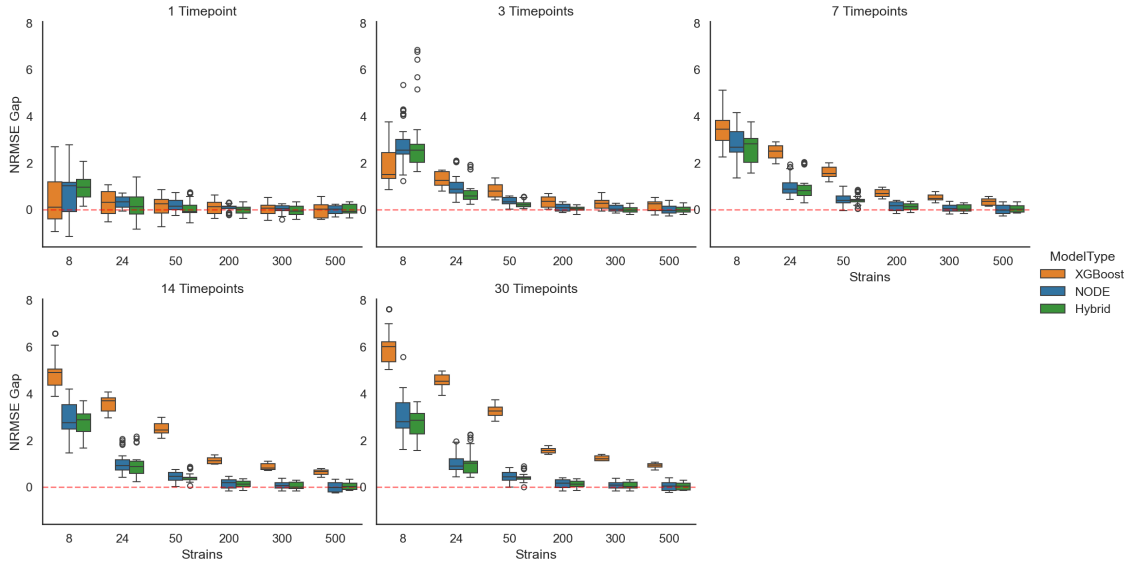


Figure 4.5: Distribution of train–test performance gap (NRMSE) across models and scenarios, evaluated over all metabolites and timepoints. Panels correspond to different numbers of training timepoints. The x-axis shows the number of training strains, and colors indicate model type.

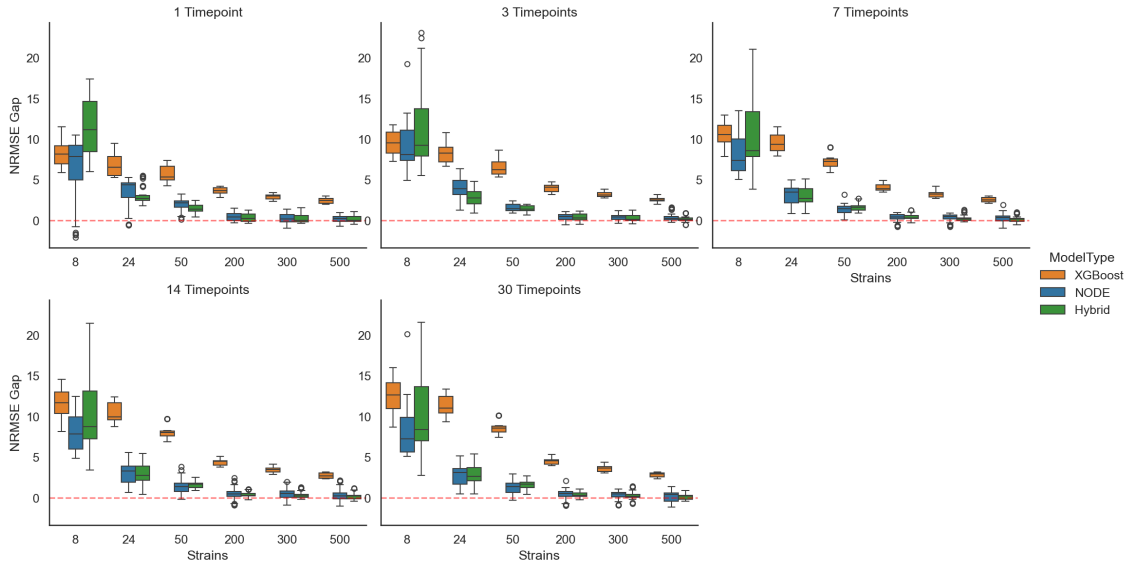


Figure 4.6: Distribution of train–test performance gap (NRMSE) for the final p-coumaric acid (p-CA) concentration across model types and scenarios. Panels correspond to different numbers of training timepoints, and the x-axis shows the number of training strains.

When evaluating performance specifically on the final p-coumaric acid (p-CA) concentration, the observed trends differ from those based on full trajectory evaluation (Figure 4.7). In this case, all models show similar dependence on training set size, with performance improving primarily as the number of strains increases. The number of timepoints has relatively little impact on final product prediction accuracy. For the hybrid model, training on a single timepoint even leads to slightly improved performance in some cases (Figure 4.8), suggesting that focusing the loss on the final endpoint can be beneficial when the endpoint is the primary objective. For XGBoost the extra timepoints do slightly improve the predictions, but only when trained on sufficient data. When training instead on a very small set of

strains, adding more timesteps decreases the performance. Most importantly, for the lower amounts of training data, the hybrid model outperforms the XGBoost. The XGBoost eventually catches up to the hybrid model and even starts to perform better, but only when increasing to a very large number of strains. The NODE overall performs the worst and is the most unstable (Figure 4.8).

For small training datasets (8 strains), XGBoost fails to capture the underlying distribution and instead predicts nearly constant values, appearing as horizontal lines rather than the expected diagonal pattern in the test set. In contrast, both the Neural ODE and hybrid models produce predictions that follow the true distribution, albeit with greater dispersion compared to scenarios with more training data (Figure C.5).

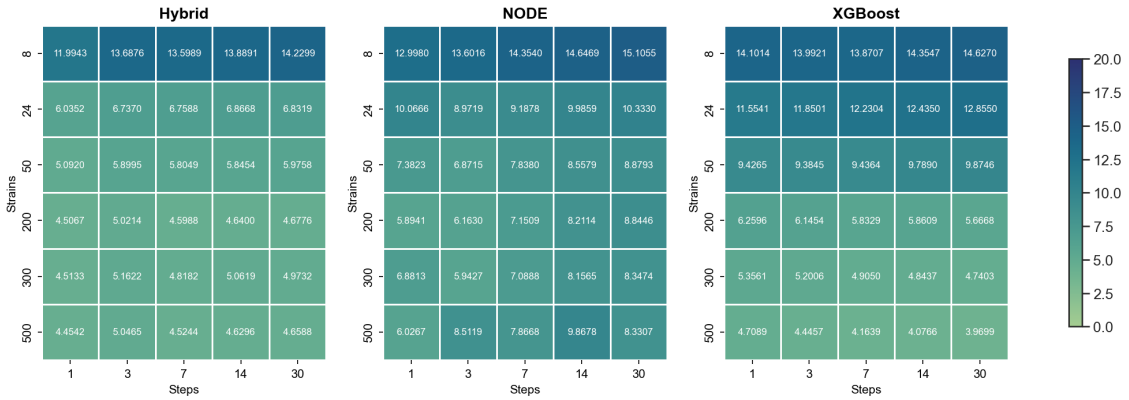


Figure 4.7: Heatmaps of mean NRMSE for final p-coumaric acid (p-CA) concentration predictions across varying training data configurations. Results are shown for hybrid model (left), Neural ODE (middle), and XGBoost (right).

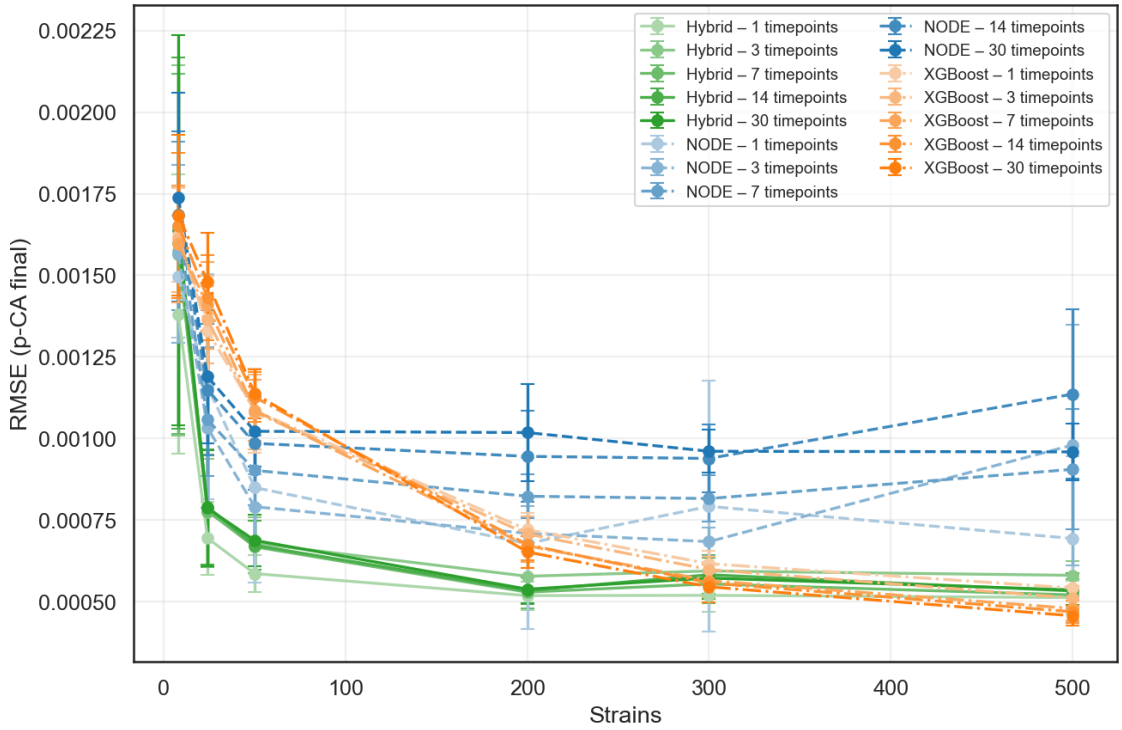


Figure 4.8: Root mean squared error (RMSE) of the final p-coumaric acid (p-CA) concentration as a function of training set size. Lines represent different model types, where color (hue) distinguishes between the hybrid model, Neural ODE, and XGBoost. Shades of the same color indicate the number of timepoints used during training. Error bars represent variability across runs.

When evaluating the stability of the final p-coumaric acid (p-CA) prediction across different subsets,

performance improves significantly for all models as the training set size increases, reaching a plateau around 200 strains (Figure 4.9). The variability between subsets is highest for the smallest training sets, particularly with only 8 strains, where performance is more sensitive to the specific subset composition. Within individual subsets, the Neural ODE shows the highest variability across runs, whereas the hybrid model and XGBoost remain more stable (Figure C.6). For small training sets, variability across different random seeds is also high, especially for the hybrid model, but this variability decreases substantially as the number of training strains increases.

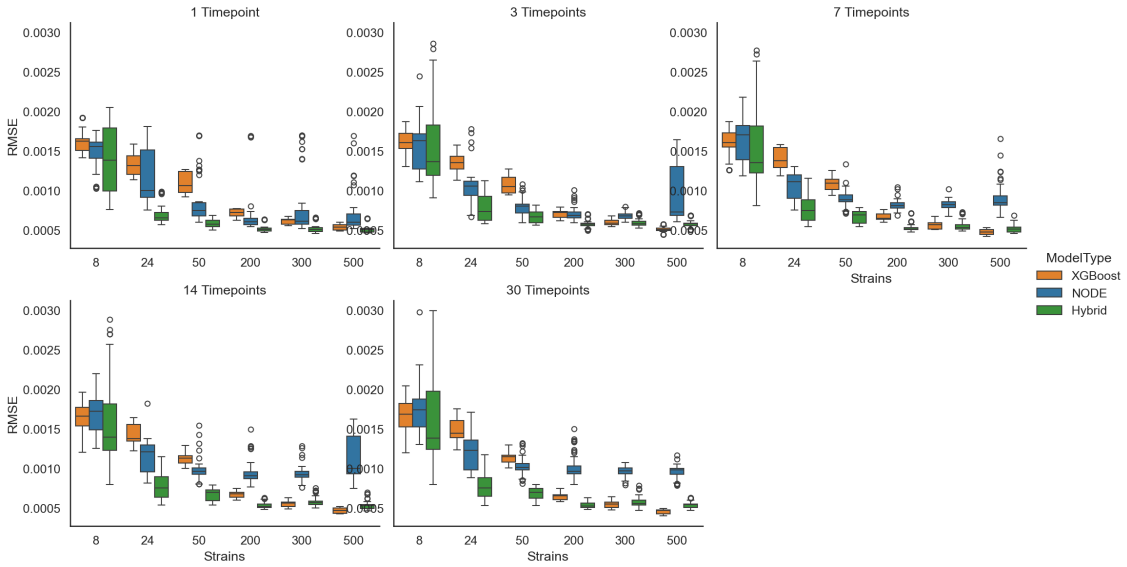


Figure 4.9: Distribution of model performance across scenarios. Boxplots show the median, interquartile range, and outliers of the RMSE across all model types, evaluated on only the final p-CA timepoint.

To assess how well the models capture extreme values, and under which conditions, the prediction error for individual strains in the test set was analysed as a function of the true final product concentration (Figure C.7). This is relevant in many practical settings, as accurately predicting the top producing strains is often the goal in metabolic pathway optimization. For XGBoost, the prediction error exhibits a clear increasing trend, particularly for smaller training set sizes, indicating that the model struggles to accurately predict high-producing strains. In contrast, the hybrid model shows a much weaker dependence of error on the true concentration, suggesting improved performance on outliers.

Overall, these results demonstrate the importance of making a trade-off between the amount of strains and timepoints, when the goal is to predict the entire metabolite trajectories. It is important to train on sufficient timepoints, and the overall error quickly saturates with the addition of more strains, suggesting that it is more beneficial to prioritize timepoints. However, the final metabolite concentration predictions do improve when adding more strains still (Figure C.4b). If the goal is solely to predict only the final product concentration, it is better to prioritize the number of strains. In very low-data regimes, training of the hybrid model becomes unstable, but this problem dissappears when slightly increasing the number of strains, or training on more timepoints. Despite this, the hybrid model still outperforms both the Neural ODE and XGBoost models in low-data settings. While hybrid models provide clear advantages in low-data settings due to their incorporation of mechanistic knowledge, sufficient data can allow simpler models such as XGBoost to achieve comparable performance.

4.0.4 Effect of prior mechanistic network knowledge on the minimal data requirements for reliable model performance

Thus far, the hybrid model was evaluated under the assumption that the entire product pathway is unknown and must be learned from data. However, in practice, parts of metabolic networks, such as central pathways (e.g., glycolysis), are often well characterized. This raises the question of how

incorporating prior mechanistic knowledge affects the amount and type of data required to train the model effectively.

To investigate this, five different model configurations were constructed in which varying subsets of reactions are masked. Masked reactions are treated as unknown and their fluxes are fully learned by the neural network, whereas unmasked reactions are modelled mechanistically using Michaelis–Menten kinetics. This way, the degree of prior knowledge incorporated into the model is systematically varied.

For each masking configuration, the number of training strains and timepoints was varied in the same manner as in the previous section. For each combination of masking scenario and data regime, 10 subsets were sampled and each subset was trained with 5 random seeds, resulting in 50 trained models per configuration. Model performance was evaluated using both the NRMSE across all metabolites and timepoints, and the RMSE of the final p-coumaric acid (p-CA) concentration (Figure 4.11).

When evaluating only the final p-CA prediction, it is observed that especially in scenarios with higher levels of mechanistic knowledge (e.g., when only the product pathway is masked), increasing the number of timepoints can lead to a worse performance. This effect is observed much less noticeably in the fully masked model. A likely explanation for this effect is related to how the loss function distributes attention across timepoints. When only a single timepoint is available, the model focuses entirely on predicting the final concentration. However, when multiple timepoints are included, the loss is distributed over all timepoints, reducing the relative contribution of the final point. As a result, the model may optimize overall trajectory accuracy at the expense of endpoint precision. This issue could potentially be mitigated by assigning a higher weight to the final timepoint in the loss function when endpoint prediction is the primary objective. This effect becoming more pronounced as the model obtains more mechanistic knowledge, indicates that the more knowledge the more constrained the model becomes and the more trouble it has to fit the trajectory data.

The results show that incorporating mechanistic knowledge improves model performance, with the lowest errors obtained for the model with the least unknowns. However, the amount of data required to reach maximum performance remains largely unchanged. For all masking configurations, performance saturates at approximately 200 training strains. Therefore, the addition of knowledge does not decrease the data requirements, but it lowers the threshold. Because of this, models with more mechanistic knowledge can achieve comparable performance to less knowledgeable models with fewer training samples. For example, in the configuration where only the product reaction is unknown, performance at 50 strains is better than that of the fully masked model at 200 strains (Figure 4.10). This demonstrates that prior knowledge effectively reduces the amount of data required to achieve a given level of accuracy. However, these benefits are most pronounced in extreme configurations, where either nearly all reactions are masked or nearly all are known. For intermediate cases, where only a small number of reactions are added as prior knowledge, the performance differences are minimal. The lines for the intermediate cases all fall in the same spot (Figure D.1).

The variance for the scenario with only the reaction to product masked, when trained on 7 timepoints and 500 strains, is very large. This is because some seeds within this configurations are very slow, as a result they time out very early before the models have learned anything. The increased training time is not caused by the training data, as large difference occur for seeds with the same subset. Therefore, it is likely caused by the NN weight initialization. It is unclear why it happens specifically for this scenario, but not other scenarios, that for example have more or less timepoints or a different level of masking.

The addition of mechanistic knowledge has a large impact on performance when evaluated on full metabolite trajectories. The model that masks only the final reaction consistently achieves significantly lower trajectory error than the other configurations, even when trained on very limited temporal data. For example, this model achieves better performance when trained on a single timepoint than the fully masked model trained on 30 timepoints, highlighting the strong benefit provided by prior knowledge. The largest improvement in performance is observed when unmasking the reactions directly downstream of the substrate (Figure D.2). Beyond this point, unmasking additional intermediate reactions yields only marginal gains. A further improvement is observed when all reactions except the final step are known. When trained on only 8 strains, the model that includes all reactions except sink reactions shows a large variance in performance. This likely occurs because the model is highly constrained

but lacks sufficient data to accurately learn the remaining unknown components, particularly the sink reactions. As a result, the model struggles to generalize across different training subsets.

It should be noted that in the final predictions some non-finite values occurred in the final predictions for some of the configurations (Figure D.3). As these values are excluded from the performance metrics, this may introduce a slight optimistic bias in certain results. However, the proportion of invalid predictions is very low across all scenarios, and is primarily observed in low-data regimes (8 strains). Therefore, the impact on the reported results is limited. Notably, these values occur most frequently in the configuration where only the sink and product reactions are masked, suggesting that this scenario is comparatively less stable.

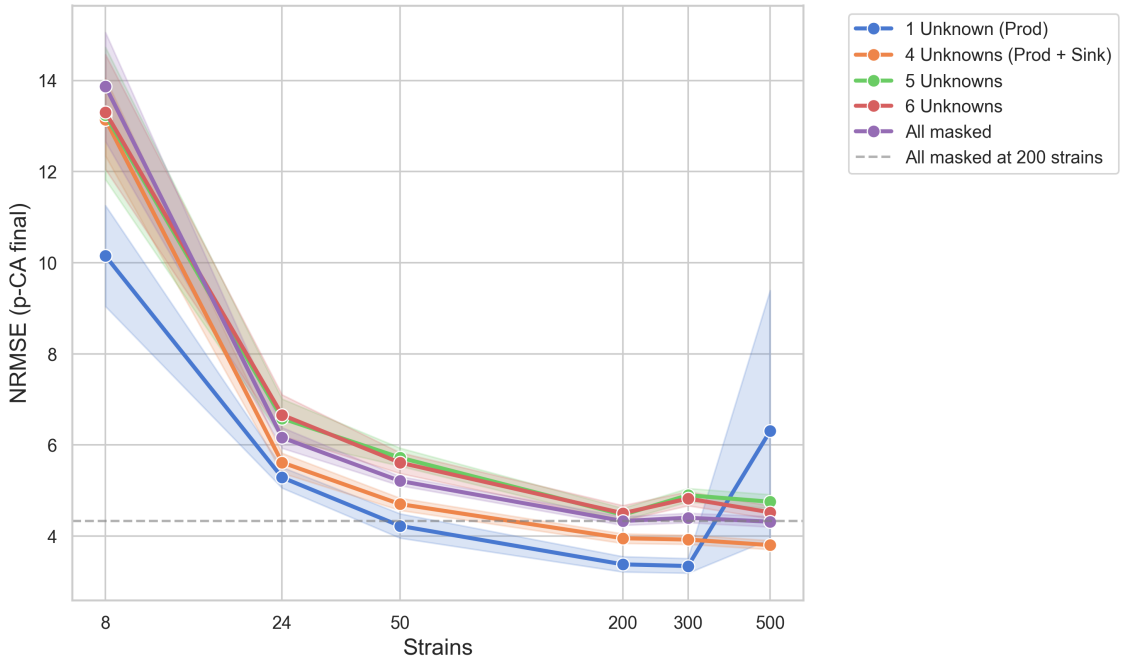


Figure 4.10: Effect of prior mechanistic knowledge on model performance. Line plot of final p-CA prediction error (NRMSE) as a function of training set size for different masking scenarios. All models were trained on 7 timepoints. The dashed line indicates performance of the fully masked model at 200 strains. Shaded regions indicate variability across runs (standard deviation).

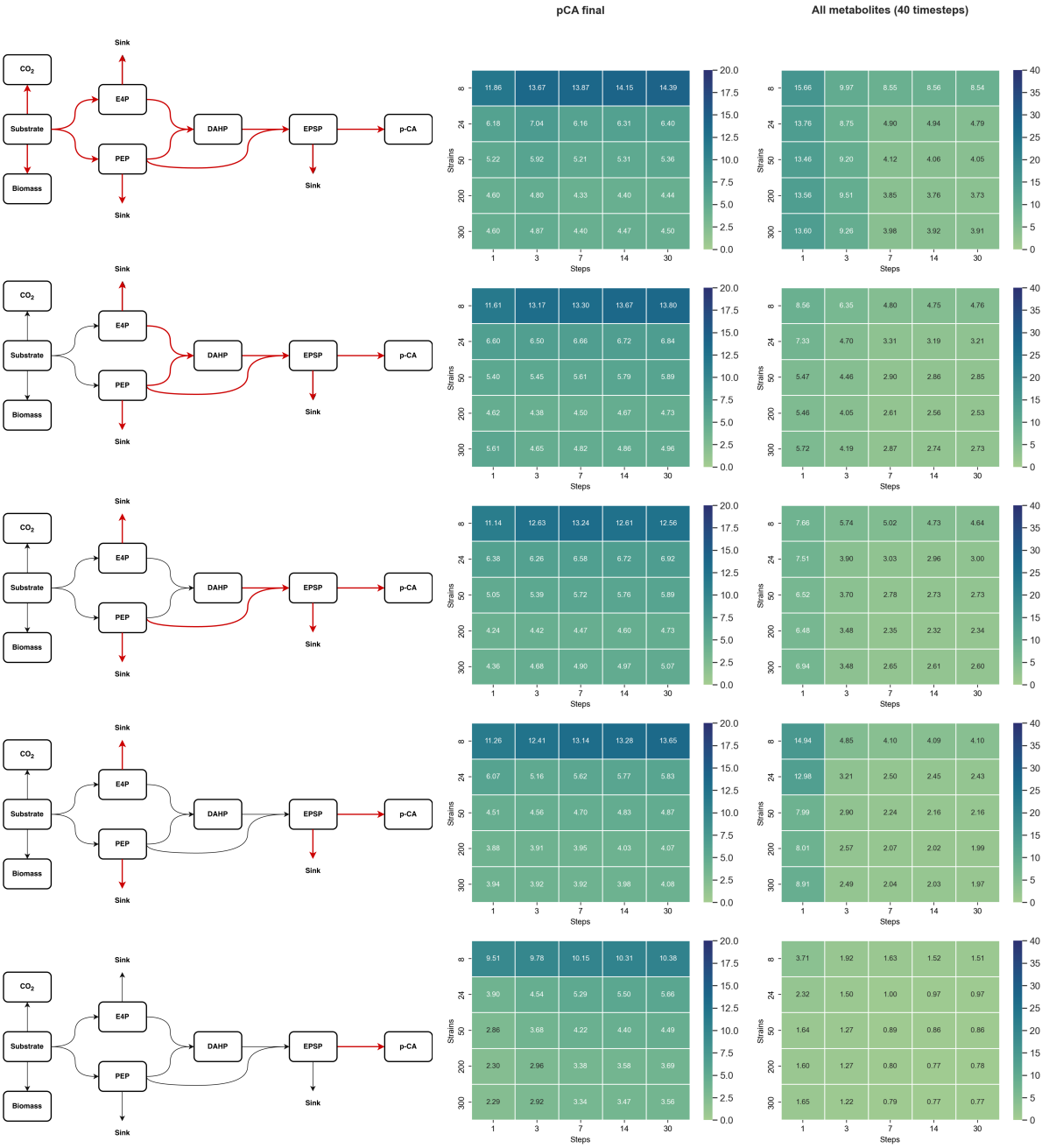


Figure 4.11: Impact of reaction masking on model performance. Showing the metabolic pathways indicating masked reactions in red (Left), heatmaps of mean RMSE for final p-CA prediction (Middle) and heatmaps of mean NRMSE across all metabolites and timepoints (Right).

4.0.5 Effect of reaction lumping on the minimal data requirements for reliable model performance

We have thus far assumed that all steps in the metabolic network are known. However, in practice it is often unclear which specific reactions take place and which intermediate metabolites are involved. In this case multiple reactions can be lumped together to form one reaction which comprises multiple steps in the actual pathway. Therefore, this study investigates whether lumping reactions together significantly affects the performance and if it requires more training data to achieve similar performance. On the other hand, sometimes having too many restrictions can negatively influence the performance. This is because it causes the model greater difficulty in fitting the data. In previous research it was

found that the mass-conserving hybrid model architecture works best when only a select amount of the reactions is masked [12]. Therefore, instead of masking reactions that are unknown, reactions can be lumped together, leaving the model more room for optimization. This study investigates if in the scenario where none of the reactions are known it is better to lump the reactions. Meaning that instead of using a Neural Network to individually predict the flux for a number of unknown reactions, the network now predicts the flux of all these reactions together at once. To answer both these questions, there are two different approaches. In the first approach three different pathways are created with different degrees of lumping (Figure 4.15). In each of these pathways all the reactions are masked, to simulate the scenario where none of the parameters are known for calculating the flux. In the second approach we lump all the reactions together that were marked as unknown in the masking scenarios of the previous question (Figure 4.16). When lumping reactions, it is important to create an additional reaction to account for missing sink reactions, otherwise the mass cannot leave the system and the model has no way to account for that, which results in bad predictions.

A small number of non-finite prediction values were also observed in the lumped model configurations (Figure E.1). Similar to the masking scenarios, these values occurred at very low frequencies and were primarily confided to low-data regimes. As such, their impact on the reported performance metrics is limited.

The NRMSE is calculated for all timesteps of the metabolites that are in the training data, so for the lumped counterpart of the fully masked model of the previous section, this is just substrate and product. For these models the original stoichiometry was kept but condensed into a matrix containing only the metabolites that are present in the lumped pathway. This means that the stoichiometry now isn't balanced, as some of the intermediate metabolites with their sink reactions are missing. The other way of doing it would be to calculate stoichiometry directly from the known reactions, but this means that the stoichiometry is not exactly true to the ground truth. Both options were tried and compared with each other, using the second approach still achieves similar performance to the first approach, even with the simplified stoichiometry (E.0.1).

The final p-CA concentration prediction is not affected significantly by the lumping of the models (Figure 4.13). However, this is only the case when trained on cross-sectional data. For the less complex models, the p-CA predictive performance degrades much quicker when adding more timepoints than for the model with the most complete underlying network. But the best prediction of the most lumped model, which is when trained using only 1 timepoint, does not improve when increasing the complexity (the 3 best lines fall in the same spot). When assessing the models based on complete metabolite trajectories, each model improves over the less complex models. This gap is largest when trained on less timesteps (Figure 4.12).

Whereas it previously was observed that masking less reactions does not significantly change the data requirements to reach the performance cap. For different levels of lumping, this is not the case. It was noticed that previously, with the full underlying network, the model reached saturation around 50 strains, when evaluated on full metabolite trajectories. For the most lumped model, this is still the case, as performance does not improve when including more training strains. For the other two models they do increase in performance beyond 50 strains (Figure 4.12). Suggesting that reaction lumping increases the data requirements. Conversely, having more knowledge of the individual network reactions, decreases the amount of training data needed.

When comparing the models from the previous section to models in which all unknown reactions are lumped, it is noticed that in most cases the full network model outperforms the lumped variant. This is especially the case for the first model, where all reactions are masked/lumped. In this case the lumped model has too little information and performs far worse than the fully masked model. However, that is only the case when trained on more than one timestep. When the models are trained on cross-sectional data, both models perform very similarly for all model complexities. In some cases, the lumped variant even achieves a lower error when trained on cross-sectional data. Each seed from both model versions were tested on the same subset, so a paired t-test could be used to compute the statistical significance (Figure 4.14). The lumped variant does achieve a significant improvement over the full model in a few cases, but not in enough cases to state that one way is always better than the other. This does indicate that when in practice not all individual reactions are known, performance does not decrease drastically.

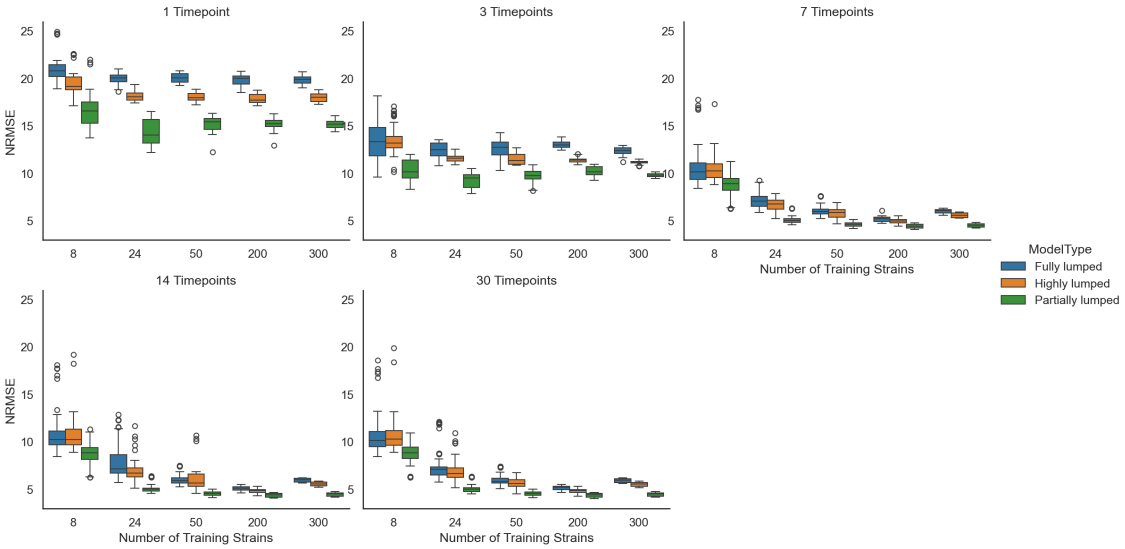


Figure 4.12: Distribution of NRMSE across lumped pathway models. Boxplots summarize performance across all scenarios for different levels of reaction lumping.

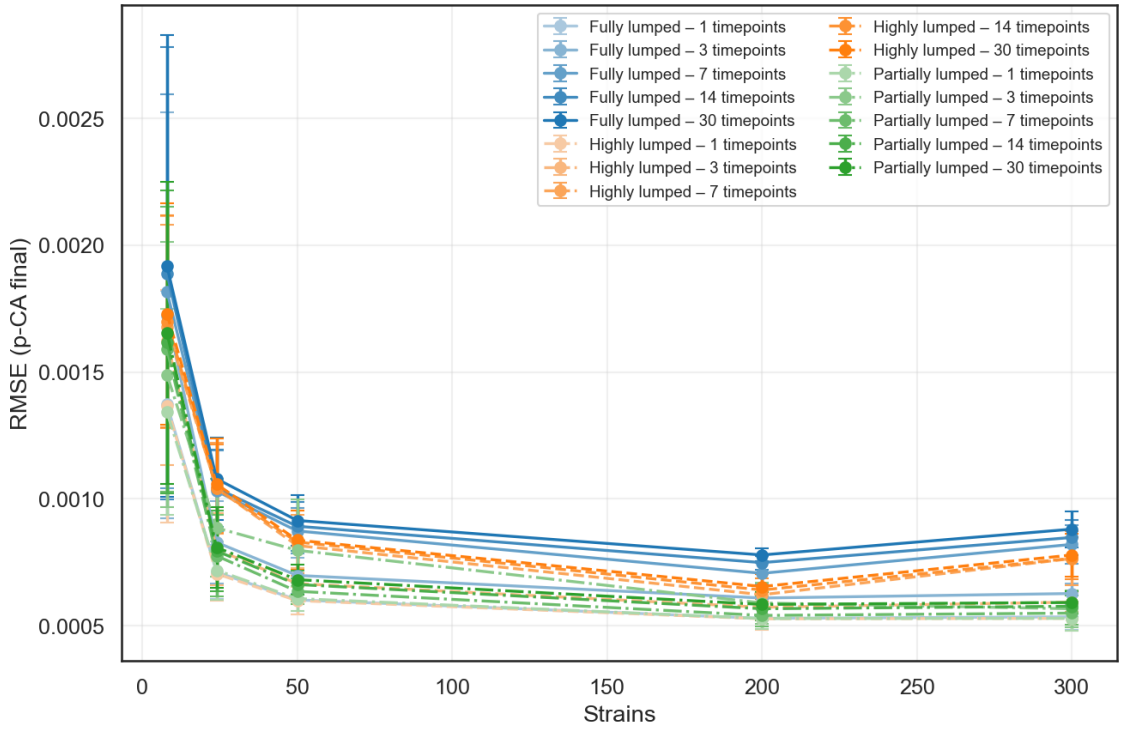


Figure 4.13: Final p-CA prediction error (RMSE) as a function of training set size and degree of reaction lumping. Lines indicate different timepoint configurations.

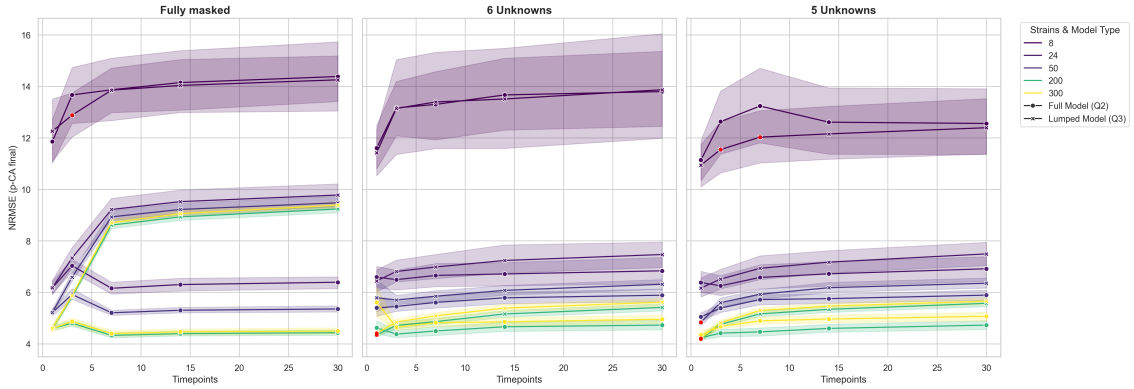


Figure 4.14: Comparison between fully specified (4.0.4) and lumped (4.0.5) variants across different pathway configurations. Panels correspond to three model structures with increasing levels of mechanistic detail. The x-axis shows the number of training timepoints, and the y-axis shows the NRMSE of the final p-coumaric acid (p-CA) concentration. Colors indicate the number of training strains, while markers distinguish model types (circles: full model, crosses: lumped model). Red markers indicate scenarios in which the lumped model significantly outperforms the full model based on a paired statistical test.

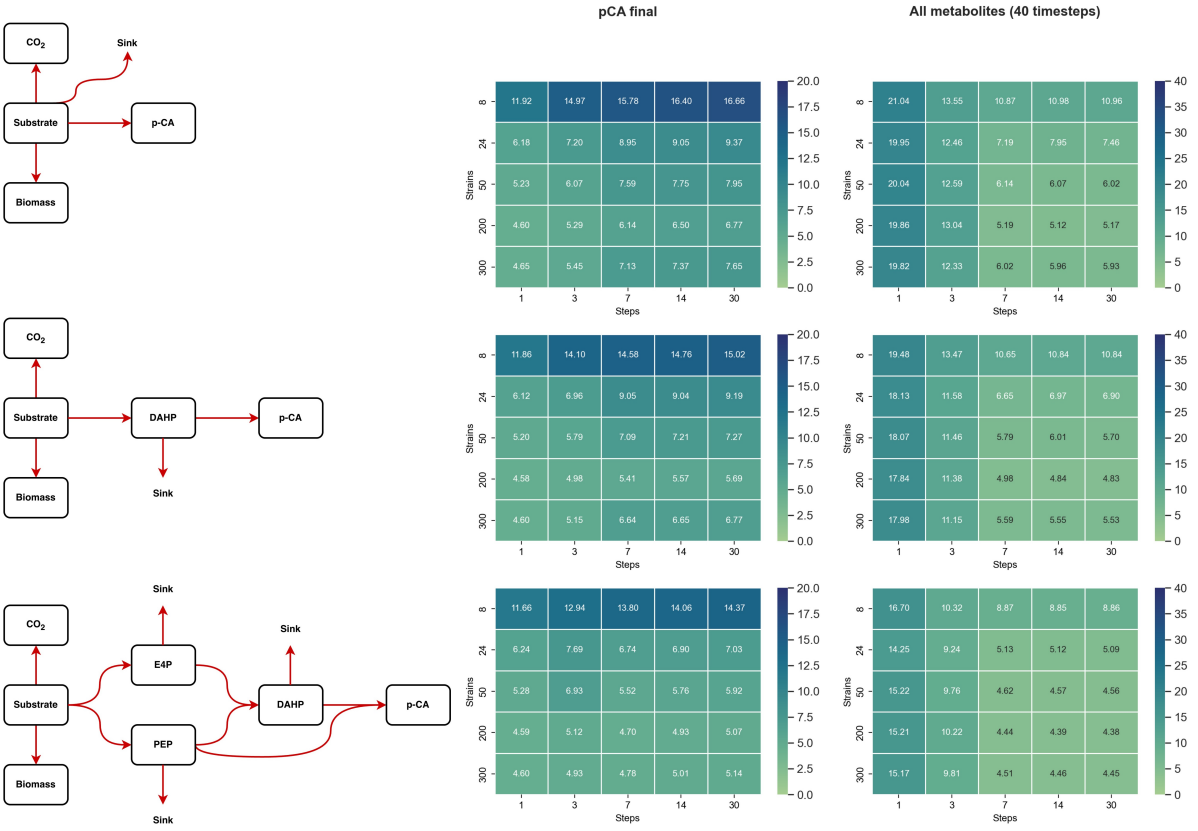


Figure 4.15: Impact of reaction lumping on model performance. Showing the metabolic pathway representations of the lumped reactions (Left), heatmaps of mean RMSE for final p-CA prediction (Middle) and heatmaps of mean NRMSE across all metabolites and timepoints (Right).

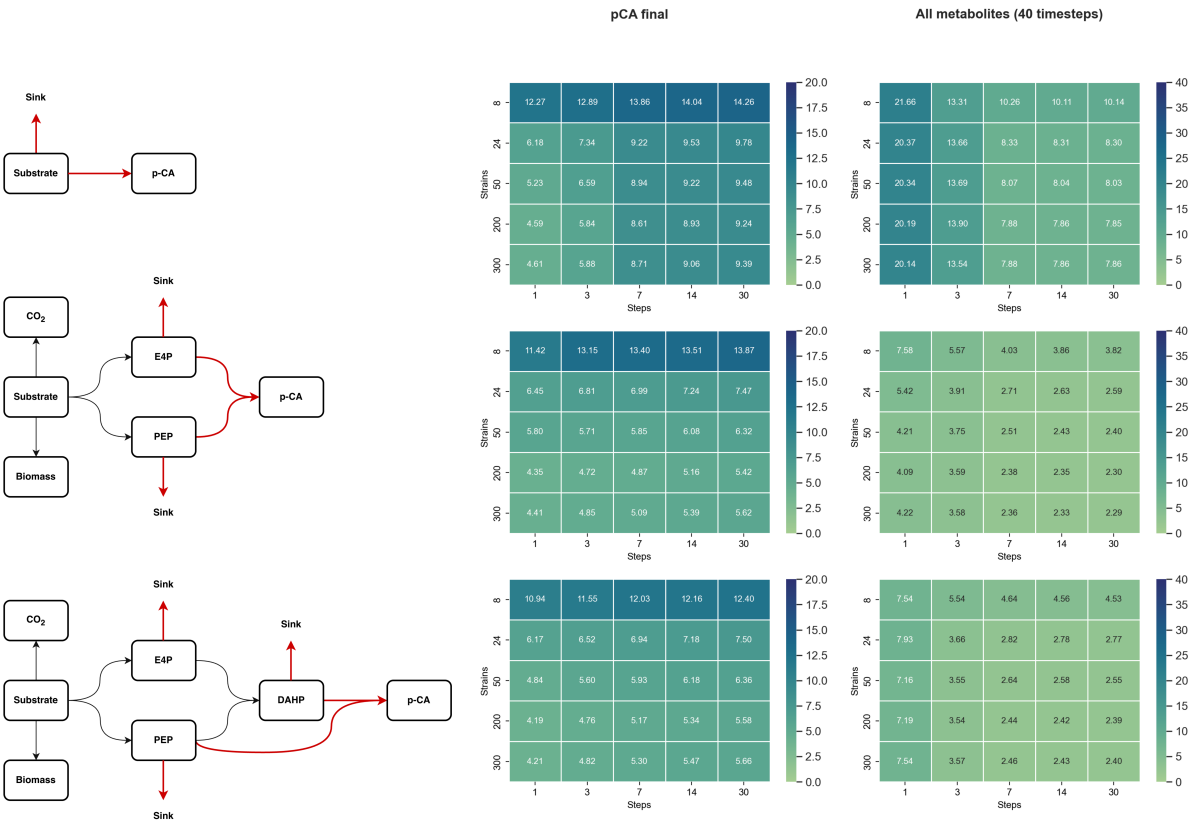


Figure 4.16: Combined masking and lumping scenarios. Showing the metabolic pathway representations of the lumped reactions indicating masked reactions in red (Left), heatmaps of mean RMSE for final p-CA prediction (Middle) and heatmaps of mean NRMSE across all metabolites and timepoints (Right).

4.0.6 Robustness to noise

In the previous experiments all the data that was used to train the models on was the ground truth data. However, in practical settings it rarely occurs that measured data has no noise. To assess how robust the hybrid models of different complexities are to different levels of noise in decreasing dataset sizes, three new datasets were constructed. These datasets were created by adding 5%, 10% or 20% Gaussian noise to the original dataset.

The experiments have shown that the hybrid model is quite robust to noise. The performance doesn't degrade drastically when increasing the noise levels (Figure 4.17). This effect is particularly pronounced for models with less underlying mechanistic knowledge, the most constrained model is the least robust to noise. This model was doing significantly better than the other two in a noise-free setting. However, when increasing the noise levels, the most constrained model takes a big negative jump in performance, towards the other two models. This effect is even worse when trained on a very small training set. Without noise when trained on 8 strains the model with most prior knowledge outperforms the other two. However, regardless of the amount of timepoints, when adding noise, performance deteriorates quicker compared to the other models. This effect is also visible when assessing only on the error of the final p-CA concentration (Figure F.1).

It seems that some specific scenarios are more prone to getting stuck in a bad region, which causes optimization to be very slow, and as a result these models have not learned enough in the set time limit to make useful predictions. This is for example seen in the most constrained scenario from the different levels of masking at 7 timepoints and 500 strains (Section 4.0.4). It also occurs at 20% noise for specific combinations of training data. Interestingly, for the model of the lowest complexity, this happens at 7 timepoints as well, both for 200 and 300 strains. For the model of medium complexity however, this only occurs at 14 timepoints, also at both 200 and 300 strains.

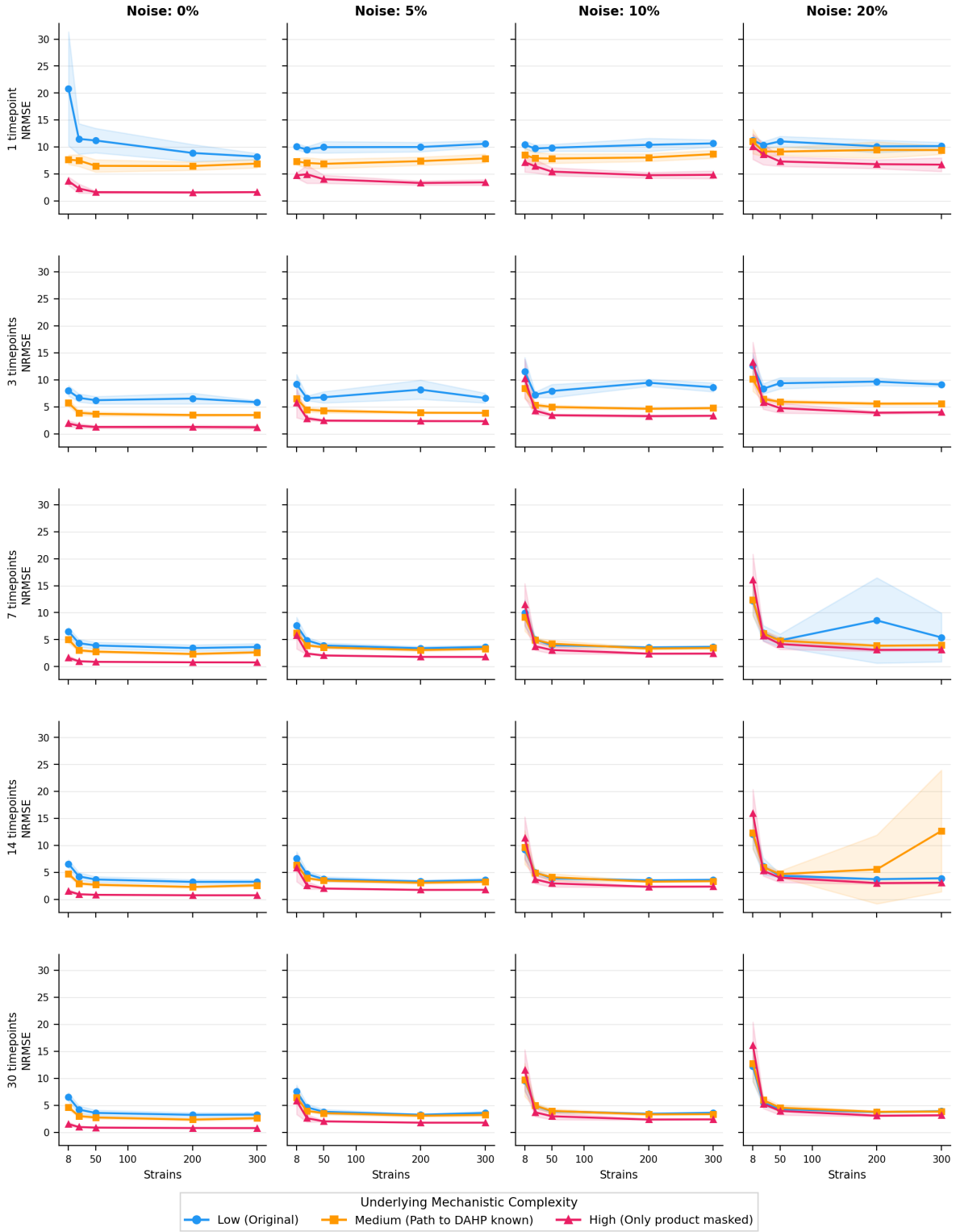


Figure 4.17: Effect of observational noise on model performance across different data regimes and levels of mechanistic complexity. Panels are organized by noise level (columns: 0%, 5%, 10%, 20%) and number of timepoints used for training (rows: 1, 3, 7, 14, and 30 timepoints). The x-axis shows the number of training strains, and the y-axis shows the mean normalized root mean squared error (NRMSE) evaluated across all metabolites and timepoints. Colored lines represent different levels of mechanistic knowledge, and shaded regions indicate variability across runs (standard deviation).

5 Conclusion

This study investigated the role of data quantity, data type, and mechanistic model structure in determining the performance of hybrid kinetic models for metabolic pathway optimisation. By systematically varying training set size, temporal resolution, prior mechanistic knowledge, and model complexity, several key insights into the behaviour of hybrid modelling approaches were obtained.

The results demonstrate that increasing the number of strains in the training data is generally more beneficial than increasing the number of timepoints per strain. This is especially true for predicting final product concentrations. When reconstructing full metabolite trajectories, a more careful trade-off is needed to achieve the most optimal performance, depending on the goal. Having enough timepoints is necessary for achieving a tolerable overall error, but the final metabolite concentration predictions improve further when adding more strains. Consequently, for practical applications within the design-build-test-learn (DBTL) cycle, prioritizing high-throughput screening across many strains is more effective than obtaining densely sampled time-series data from a limited number of strains.

Hybrid models show clear advantages over purely data-driven methods, particularly in low-data regimes. By incorporating prior mechanistic knowledge, hybrid models achieve improved generalization and require fewer data points to reach comparable performance levels. However, this benefit depends on the quality and completeness of the prior knowledge. While increasing mechanistic constraints generally improves predictive accuracy, overly constrained models can become harder to train and more sensitive to both data scarcity and noise.

The inclusion of additional mechanistic information can significantly decrease the amount of timepoints needed when evaluated on full metabolite trajectories, and the amount of strains needed when evaluated on final product concentration. However, this is mostly noticed for extreme differences in masking level, so when everything is masked and when almost nothing is masked. As when masking only one or two reactions barely any differences are noticed. Having more additional mechanistic information does not necessarily reduce the maximum amount of data required to reach a saturation in performance. All configurations show a saturation effect, with limited improvement beyond approximately 200 training strains. Therefore, more knowledge does not decrease the amount of data needed to get the maximum performance, but it lowers the threshold. It is still true that with less data and more knowledge a comparable prediction level can be achieved with a less knowledgeable model. But again, this difference is mostly noticed in the extreme cases.

Reaction lumping presents a trade-off between model simplicity and predictive accuracy. While lumped models reduce complexity and may perform comparably under cross-sectional data conditions, they generally underperform compared to full network models when temporal information is available. Furthermore, insufficient representation of sink reactions can lead to poor performance and unphysical behaviour, highlighting the importance of preserving key structural properties of the system. However, the prior knowledge does not have to be completely perfect to make accurate predictions. Which was shown when using different methods for constructing the stoichiometric matrix.

Finally, it was shown that models with more mechanistic constraints are more sensitive to noise, particularly in low-data regimes. While these models perform best under ideal, noise-free conditions, their performance deteriorates more rapidly when noise is introduced, especially in low data conditions. Indicating a trade-off between interpretability, constraint strength, and robustness.

It is important to note that all experiments in this study were conducted using synthetically generated data for a single metabolic pathway. While the use of synthetic data enables controlled experimentation and access to ground-truth dynamics, it may not fully capture the complexity, variability, and measurement noise present in real biological systems. Furthermore, the findings are derived from a specific pathway structure and may not directly generalize to all metabolic networks. Despite these limitations, the main trends identified in this study; the importance of strain diversity over temporal resolution and the benefits of mechanistic information under low-data conditions, are likely not specific to the pathway

considered here and are consistent with broader observations in complex biological systems. Therefore, it is reasonable to expect that these insights extend to other pathways and practical applications in metabolic engineering. However, further validation on experimental datasets and across a broader range of metabolic systems is required to confirm the generality of these results.

In a broader context, these results highlight the importance of inductive bias and data structure in learning complex dynamical systems. The findings demonstrate that incorporating domain knowledge can substantially improve sample efficiency, particularly in low-data regimes, while also introducing trade-offs in robustness and model flexibility. This aligns with broader trends in scientific machine learning, where hybrid approaches aim to combine data-driven models with structured prior knowledge to improve generalization and reliability. More generally, this work contributes to understanding how different types of data, cross-sectional versus temporal, affect learning performance, which is relevant beyond metabolic modelling to other applications involving time-series prediction and system identification.

In conclusion, this work demonstrates that effective hybrid modelling requires a careful balance between data availability and model complexity. For practical applications, it is recommended to prioritize larger and more diverse datasets over high temporal resolution, to incorporate mechanistic knowledge where available but avoid over-constraining the model, and to account for noise when designing experiments. These insights contribute to a better understanding of how to efficiently design experiments and build reliable hybrid models for metabolic engineering.

6 AI Disclosure

This thesis made use of generative AI tools to assist with writing and code development. ChatGPT (OpenAI) was used as a writing assistant to help improve clarity, rephrase sentences, and enhance the overall flow of the text. In addition, Gemini Flash (Google) was used as a supportive tool for coding assistance and debugging. All final content, analysis, and conclusions remain the responsibility of the author.

7 DAIC Acknowledgement

Research reported in this work was partially or completely facilitated by computational resources and support of the Delft AI Cluster (DAIC) at TU Delft (RRID: SCR_025091), but remains the sole responsibility of the authors, not the DAIC team. [5]

References

[1] Selçuk Aslan, Elad Noor, and Arren Bar□Even. Holistic bioengineering: rewiring central metabolism for enhanced bioproduction. *Biochemical Journal*, 474(23):3935–3950, 2017.

[2] Mitchell Daneker, Zhen Zhang, E. Karniadakis George and Lu Lu □Systems biology: Identifiability analysis and parameter identification via systems-biology informed neural networks. *arXiv preprint*, 2022.

[3] Max de Rooij, Balázs Erdős, Natal A. W. van Riel, and Shauna D. O'Donovan. Physiology-informed regularisation enables training of universal differential equation systems for biological applications. *PLoS Computational Biology*, 21(1):e1012198, 2025.

[4] Max de Rooij, Natal A. W. van Riel, and Shauna D. O'Donovan. Conditional universal differential equations capture population dynamics and interindividual variation in c□peptide production. *npj Systems Biology and Applications*, 11(1):84, 2025.

[5] Delft AI Cluster (DAIC). The delft ai cluster (daic), rrid:scr_025091, 2024.

[6] R. Erbe et al. Transcriptomic forecasting with neural ordinary differential equations. *Computational Biology and Chemistry*, 2023.

[7] Herbert M. Huttanus, Jiayuan Sheng, and Xueyang Feng. Metabolic engineering for production of small molecule drugs: Challenges and solutions. *Fermentation*, 2(1):4, 2016.

[8] Mohammad Mazharul Islam and Rajib Saha. Kinetic modeling of metabolism: Present and future. *Current Opinion in Systems Biology*, 26:10–17, 2021.

[9] Markus Jeschek, Daniel Gerngross, and Sven Panke. Combinatorial pathway optimization for streamlined metabolic engineering. *Current Opinion in Biotechnology*, 47:142–151, 2017.

[10] Gengjie Jia, Gregory Stephanopoulos, and Rudiyanto Gunawan. Incremental parameter estimation of kinetic metabolic network models. *BMC Systems Biology*, 6:142, 2012.

[11] Torkel E. Loman and Ruth E. Baker. Functional and parametric identifiability for universal differential equations applied to chemical reaction networks. *arXiv preprint*, 2025.

[12] Bálint Magyar. Hybrid machine learning models for metabolic engineering, July 2025. Supervisors: Thomas Abeel and Paul van Lent.

[13] Sara Moreno-Paz, Rianne van der Hoek, Elif Eliana, Priscilla Zwartjens, Silvia Gosiewska, Vitor A. P. Martins dos Santos, Joep Schmitz, and Maria Suarez-Diez. Machine learning-guided optimization of p-coumaric acid production in yeast. *ACS Synthetic Biology*, 13(4):1312–1322, 2024.

[14] Annabel C. Murphy. Metabolic engineering is key to a sustainable chemical industry. *Natural Product Reports*, 28(8):1406–1425, 2011.

[15] Maren Philipps, Antonia Körner, Jakob Vanhoefer, Dilan Pathirana, and Jan Hasenauer. Non-negative universal differential equations with applications in systems biology. *arXiv preprint*, 2024.

[16] Maren Philipps, Nina Schmid, and Jan Hasenauer. Current state and open problems in universal differential equations for systems biology. *npj Systems Biology and Applications*, 11:101, 2025.

[17] Kristin V. Presnell and Hal S. Alper. Systems metabolic engineering meets machine learning: A new era for data□driven metabolic engineering. *Biotechnology Journal*, 14(9):e1800416, 2019.

[18] Santanu Rathod, Pietro Liò, and Xiao Zhang. Predicting time□varying flux and balance in metabolic systems using structured neural□ode processes. *CoRR*, abs/2410.14426, 2024.

[19] Blake A. Simmons, Christopher J. Petzold, Steven W. Singer, Aindrila Mukhopadhyay, Deepti Tanjore, Joshua G. Dunn, and Hector Garcia□Martin. Machine learning for metabolic engineering: A review. *Metabolic Engineering*, 63:34–60, 2021.

[20] Bharath Srinivasan. A guide to the michaelis–menten equation: steady state and beyond. *The FEBS Journal*, 289(20):6086–6098, 2022.

[21] Anna C. M. Thöni, William E. Robinson, Yoram Bachrach, Wilhelm T. S. Huck, and Tal Kachman. Modeling chemical reaction networks using neural ordinary differential equations. *Journal of Chemical Information and Modeling*, 65(9):4346–4352, 2025.

[22] P. van Lent, O. Bunkova, B. Magyar, L. Planken, J. Schmitz, and T. Abeel. Jaxkineticmodel: Neural ordinary differential equations inspired parameterization of kinetic models. *PLOS Computational Biology*, 21(7):e1012733, 2025.

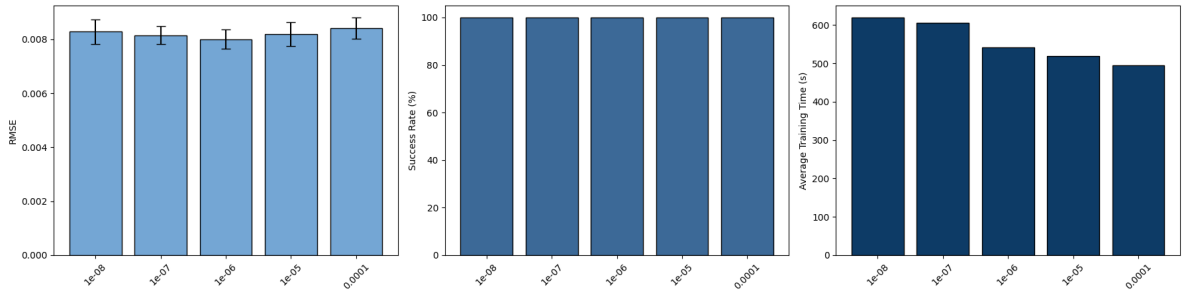
[23] Paul van Lent, Thomas Abeel, Rianne van der Hoek, Joep Schmitz, Sara Moreno Paz, Irsan Kooi, and Moniek Jonkers. Machine learning□assisted pathway optimization in large combinatorial design spaces: a p□coumaric acid case study. *bioRxiv*, 2025. preprint, posted 18 June 2025.

[24] Paul van Lent, Joep P. J. Schmitz, and Thomas Abeel. Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering. *ACS Synthetic Biology*, 12(9):2588–2599, 2023.

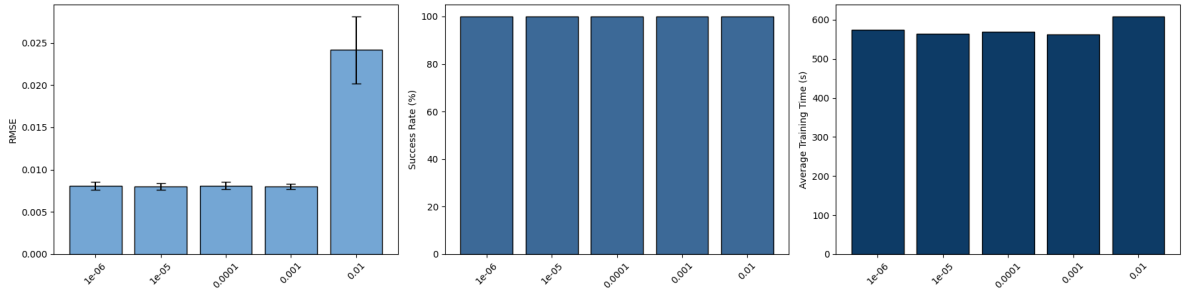
[25] Jiaqi Zhang, Erica Larschan, Jeremy Bigness, and Ritambhara Singh. scnode: generative model for temporal single□cell transcriptomic data prediction. *Bioinformatics*, 40(Supplement_2):ii146–ii154, 2024.

A Hyper parameter tuning

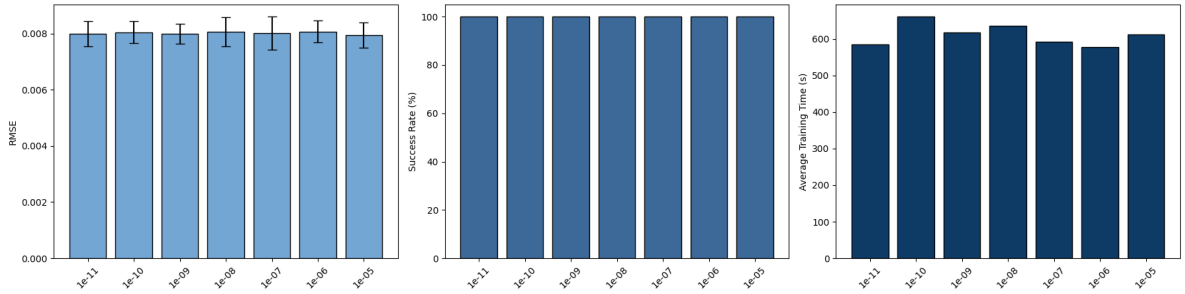
A.0.1 Performance individual hyper parameters



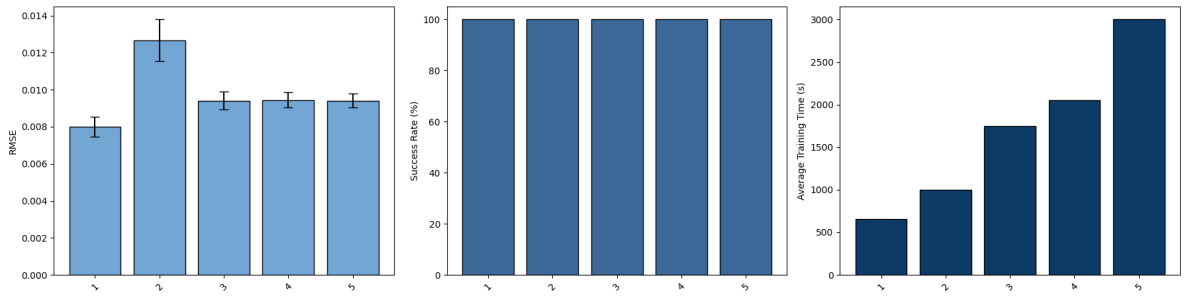
(a) Absolute tolerance



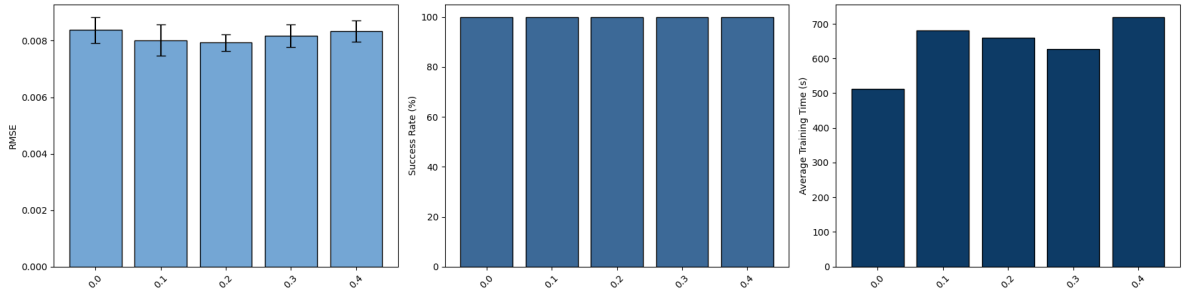
(b) Relative tolerance



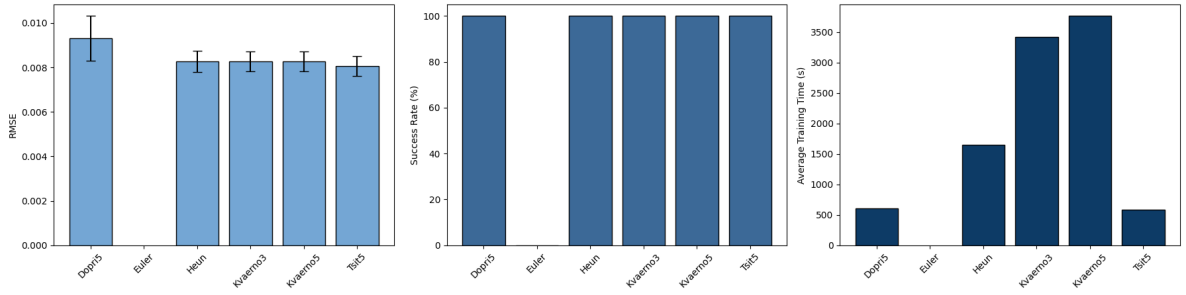
(c) Initial stepsize



(d) Neural depth



(e) Dropout



(f) Solver

Figure A.1: Hyperparameter sensitivity analysis. Panels show the effect of (a) absolute tolerance, (b) relative tolerance, (c) initial step size, (d) network depth, (e) dropout rate, and (f) ODE solver on performance metrics (RMSE, success rate, and training time).

A.0.2 Batch size vs Neural width

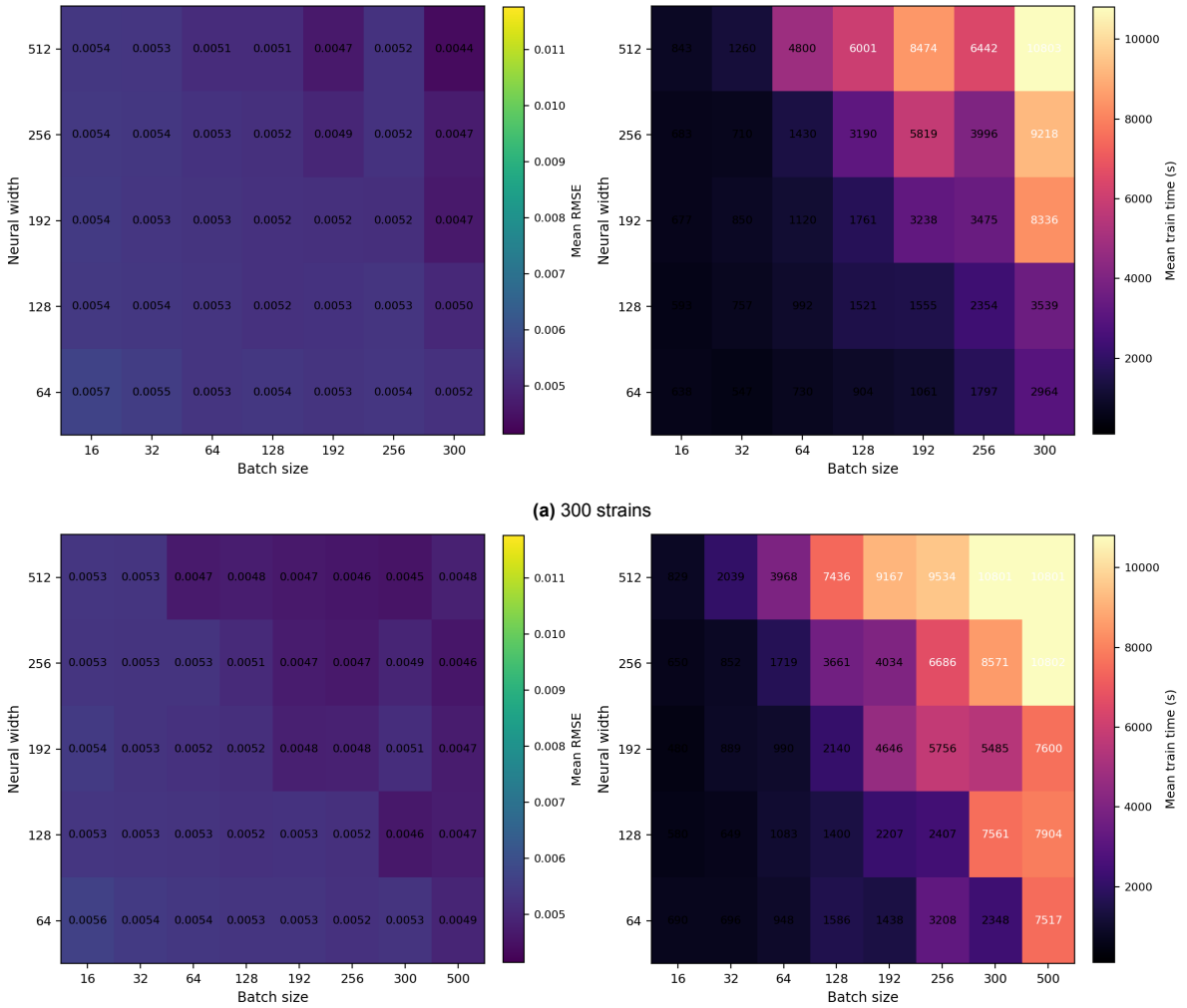


Figure A.2: Interaction between batch size and neural network width. Heatmaps show mean RMSE (left) and average training time (right) for (a) 300 strains and (b) 500 strains.

B Synthetic data generation

Mass balance check

To make sure that the mass balance closes, the data was simulated using the model with the sink reactions removed, such that all mass that goes into the system should remain in the system.

From the initial conditions we have:

$$3.3\,\mathrm{C_6H_{12}O_6} + 0.01\,\mathrm{CH_{1.8}O_{0.5}N_{0.2}} + 0.022\,\mathrm{C_3H_5O_6P} + 0.002\,\mathrm{C_7H_{13}O_{10}P} + 0.001\,\mathrm{C_{10}H_{13}O_{10}P}.$$

The simulated output of the fermentation at the final time point is:

$$1.788\,\mathrm{CH_{1.8}O_{0.5}N_{0.2}} + 11.996\,\mathrm{CO_2} + 0.224\,\mathrm{C_4H_9OP} + 0.1149\,\mathrm{C_7H_{13}O_{10}P} + 0.511\,\mathrm{C_9H_8O_3} (\text{Table } B.1).$$

To check whether the carbon mass balance holds, everything is multiplied by the number of carbon atoms in each compound:

$$3.3 \cdot 6 + 0.01 + 0.022 \cdot 3 + 0.002 + 0.001 \cdot 10 \;\rightarrow\; 1.788 + 11.996 + 0.224 \cdot 4 + 0.1149 \cdot 7 + 0.511 \cdot 9.$$

This results in 19.9 on the left-hand side and 20.1 on the right-hand side.

Table B.1: Metabolite concentrations over time

step	substrate	c_biomass	co2	e4p	pep	dhap	epsp	pcoumaric_acid
0	3.3300e+00	0.0100	0.0000	0.0000	2.2406e-02	0.0020	1.0000e-03	0.0000
1	3.2872e+00	0.0325	0.1492	0.0099	3.5398e-02	0.0024	1.0560e-03	3.46e-04
2	3.1840e+00	0.0866	0.5092	0.0296	6.1543e-02	0.0063	2.0212e-03	1.497e-03
3	2.8135e+00	0.2812	1.8111	0.0848	1.2593e-01	0.0227	9.7808e-03	1.4938e-02
4	1.6153e+00	0.9115	6.0757	0.2251	2.4237e-01	0.0604	3.0947e-02	1.1585e-01
5	3.4651e-41	1.7884	11.9847	0.2244	4.1647e-05	0.1149	1.0866e-02	4.9968e-01
6	3.3418e-40	1.7884	11.9956	0.2244	-7.2064e-26	0.1149	1.2756e-14	5.1056e-01
7	-6.7110e-45	1.7884	11.9956	0.2244	7.9410e-29	0.1149	1.1722e-18	5.1056e-01

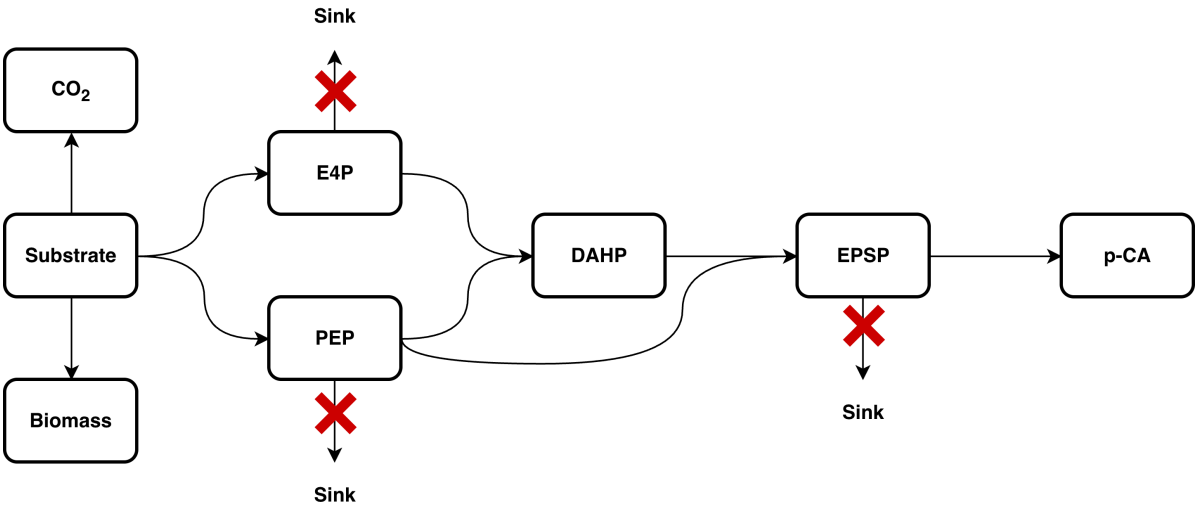


Figure B.1: Metabolic pathway used for mass balance validation, with sink reactions removed to enforce conservation of mass.

Metabolite Trajectories

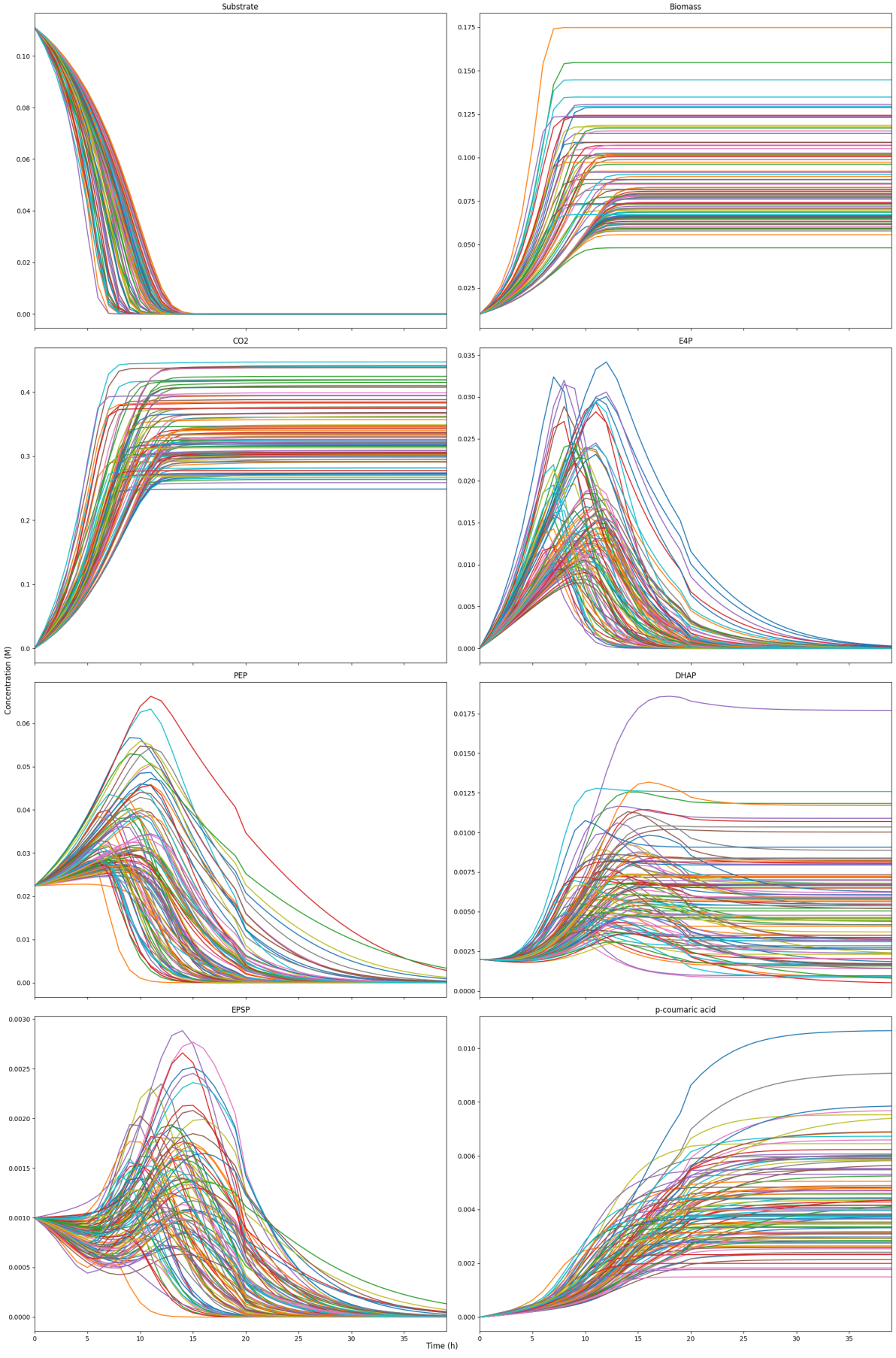


Figure B.2: Simulated metabolite concentration trajectories over time for a representative subset of perturbed strains. Each line represents a different strain.

C Performance in comparison to data quantity

Success rate

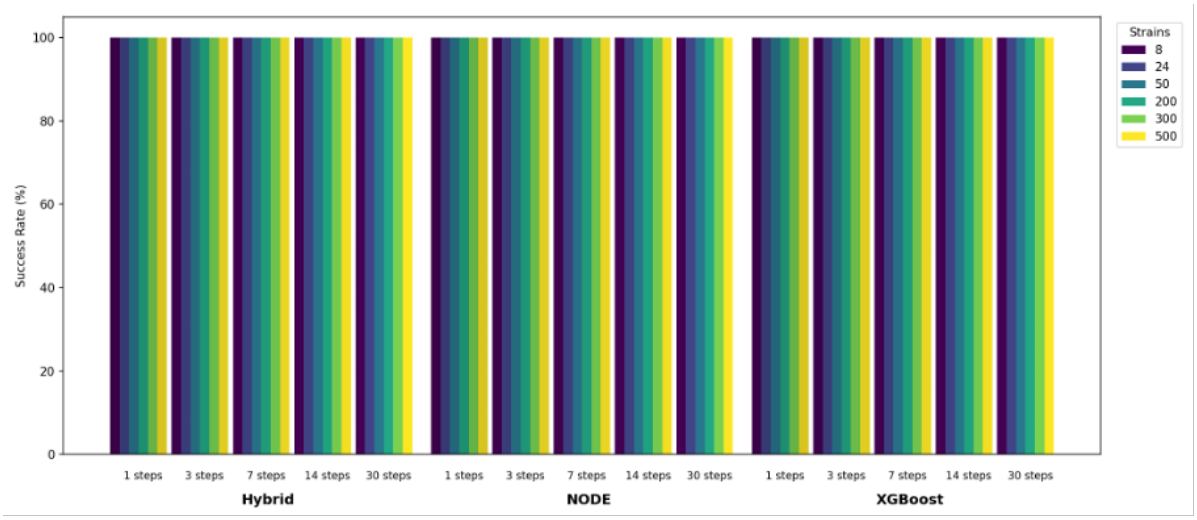


Figure C.1: Training success rate across all model types and data configurations. Bars represent the percentage of successful training runs for combinations of model type (Hybrid, Neural ODE, XGBoost), number of training timepoints, and training set size.

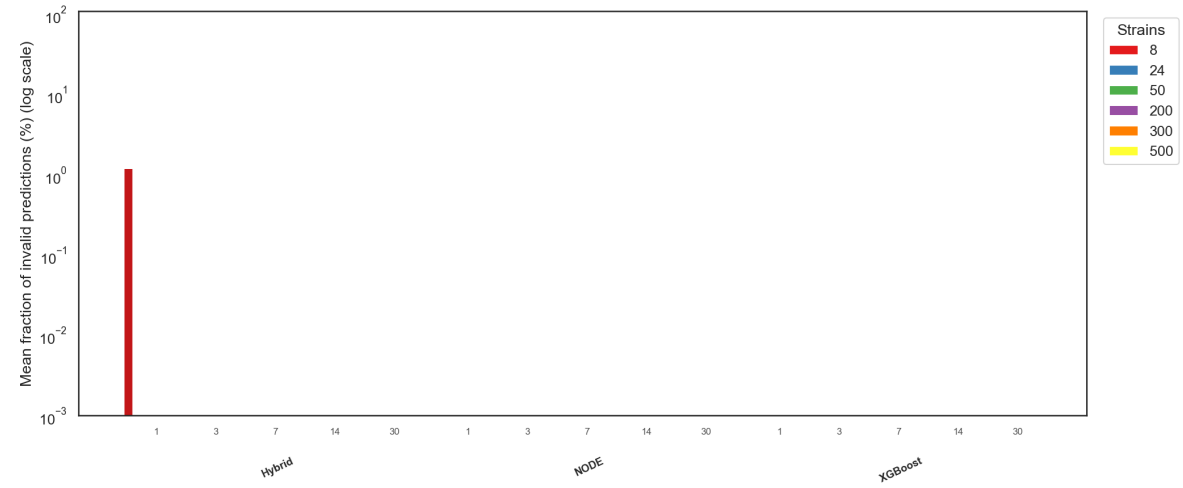
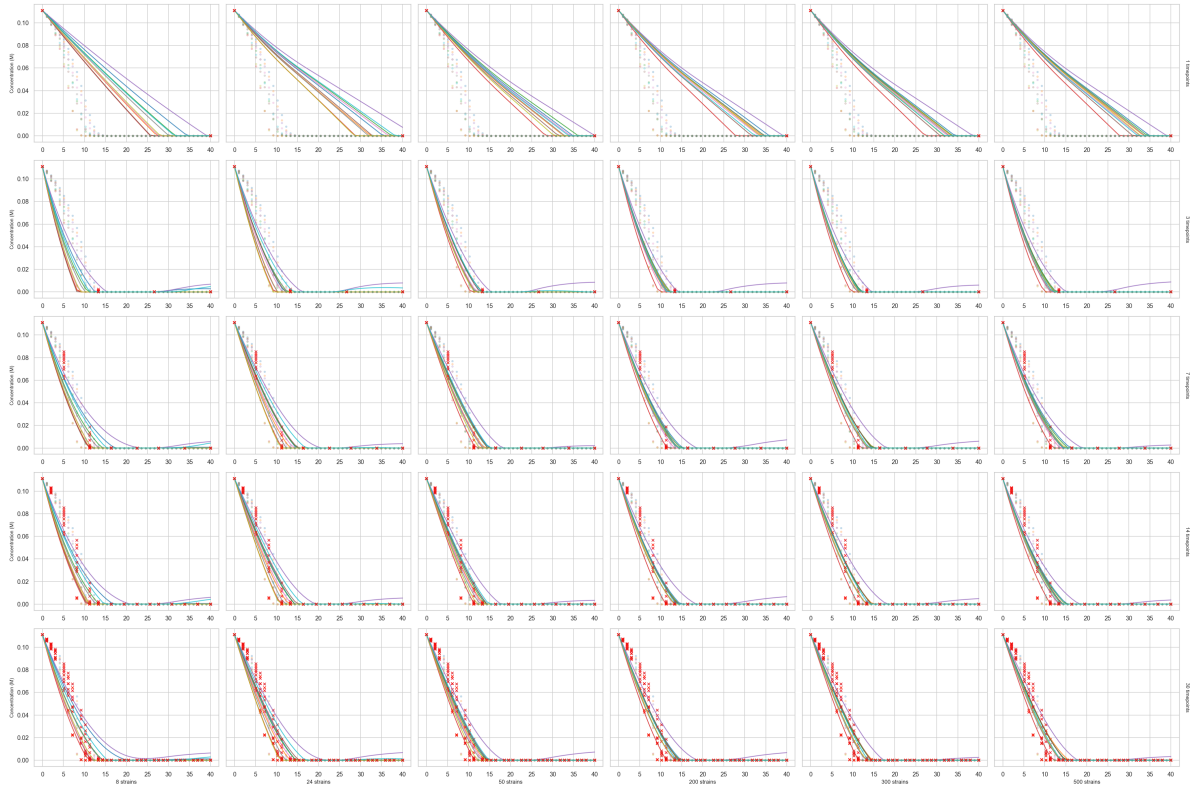


Figure C.2: Mean fraction of invalid evaluation points (NaN/Inf) (%) across model types. Values are shown on a logarithmic scale and represent the percentage of excluded data points for different training configurations.

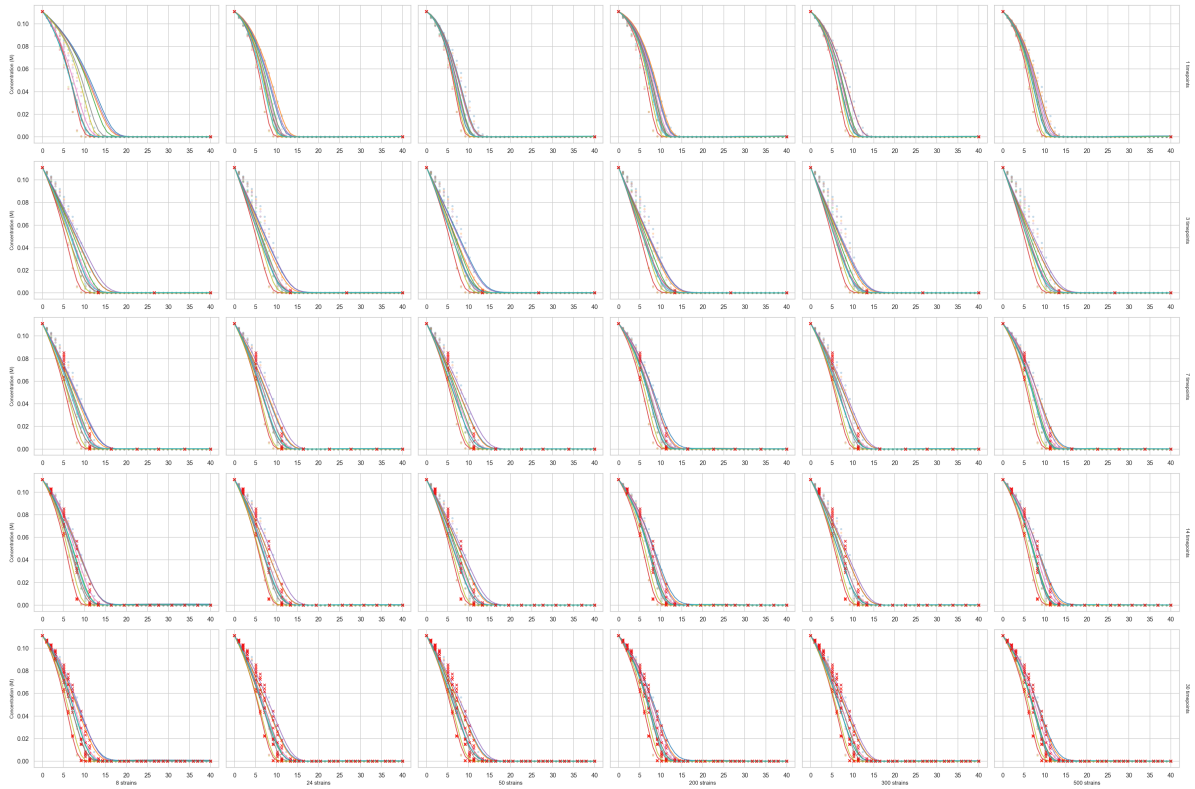
Predicted metabolite trajectories

(a)



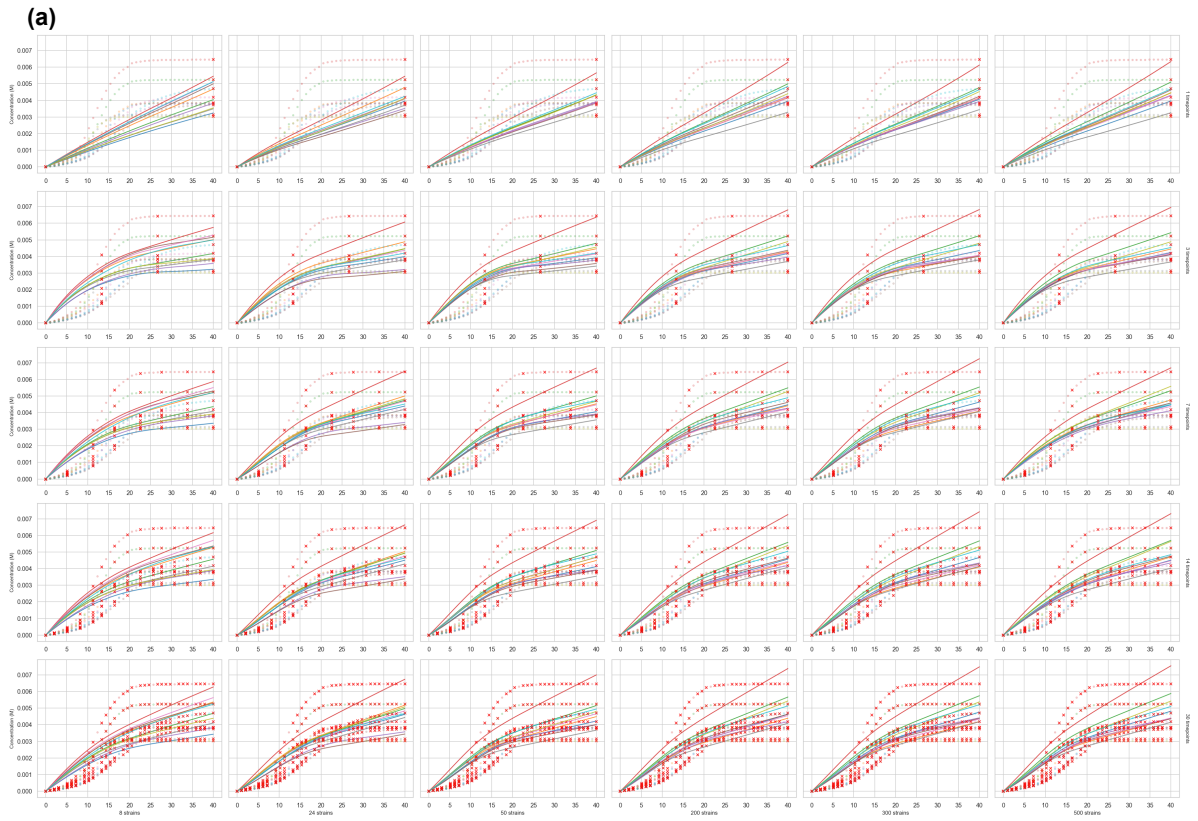
(a) NODE

(b)

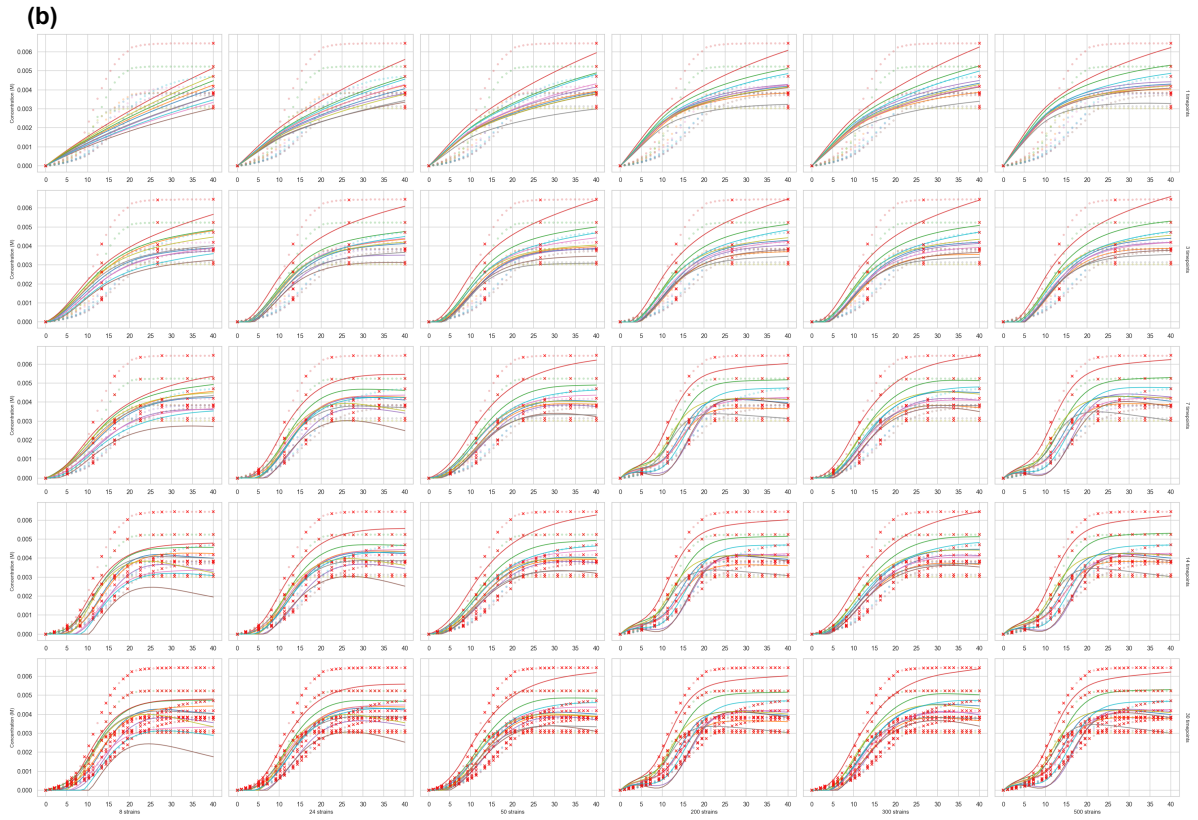


(b) Hybrid

Figure C.3: Predicted substrate concentration trajectories for unseen test strains. Panels compare (a) Neural ODE and (b) hybrid model predictions. Solid lines represent model predictions, while markers indicate ground truth observations. Red markers denote timepoints included in the training data. The figure represents a typical example, showing the predictions for one of the fifty trained models.



(a) NODE

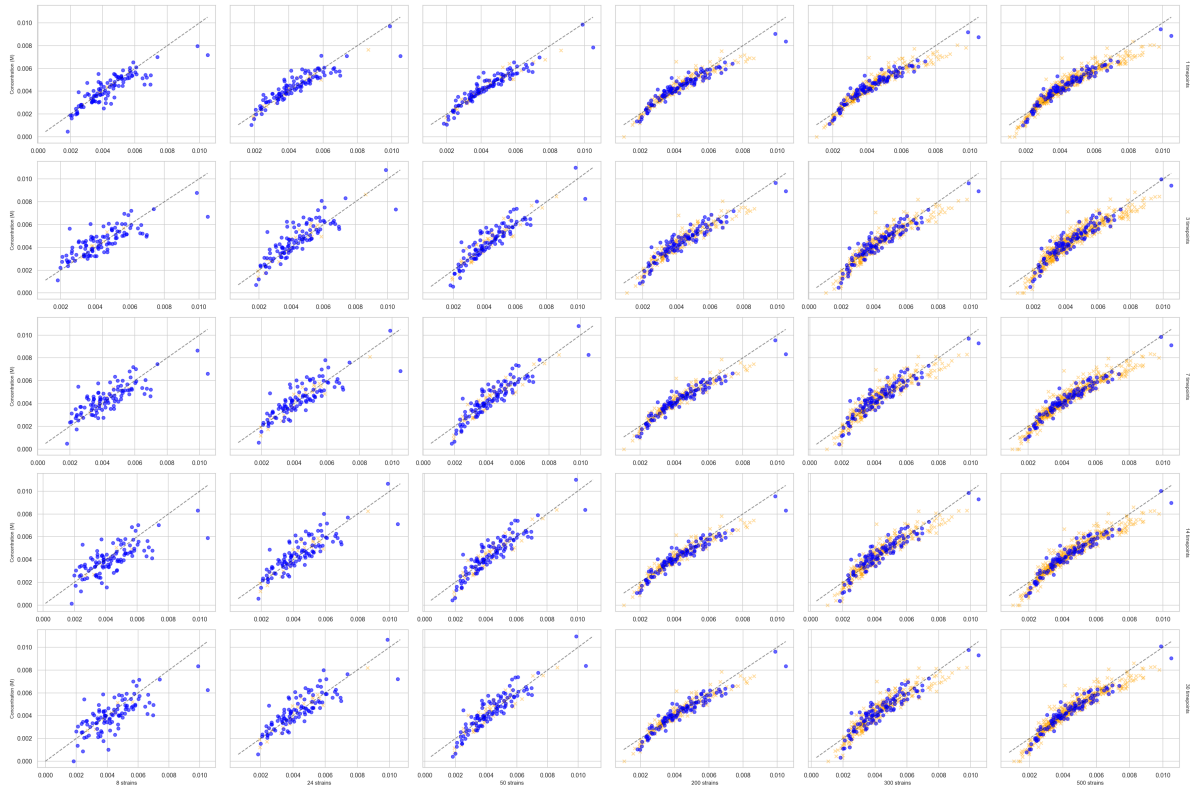


(b) Hybrid

Figure C.4: Predicted product concentration trajectories for unseen test strains. Panels compare (a) Neural ODE and (b) hybrid model predictions. Solid lines represent model predictions, while markers indicate ground truth observations. Red markers denote timepoints included in the training data. The figure represents a typical example, showing the predictions for one of the fifty trained models.

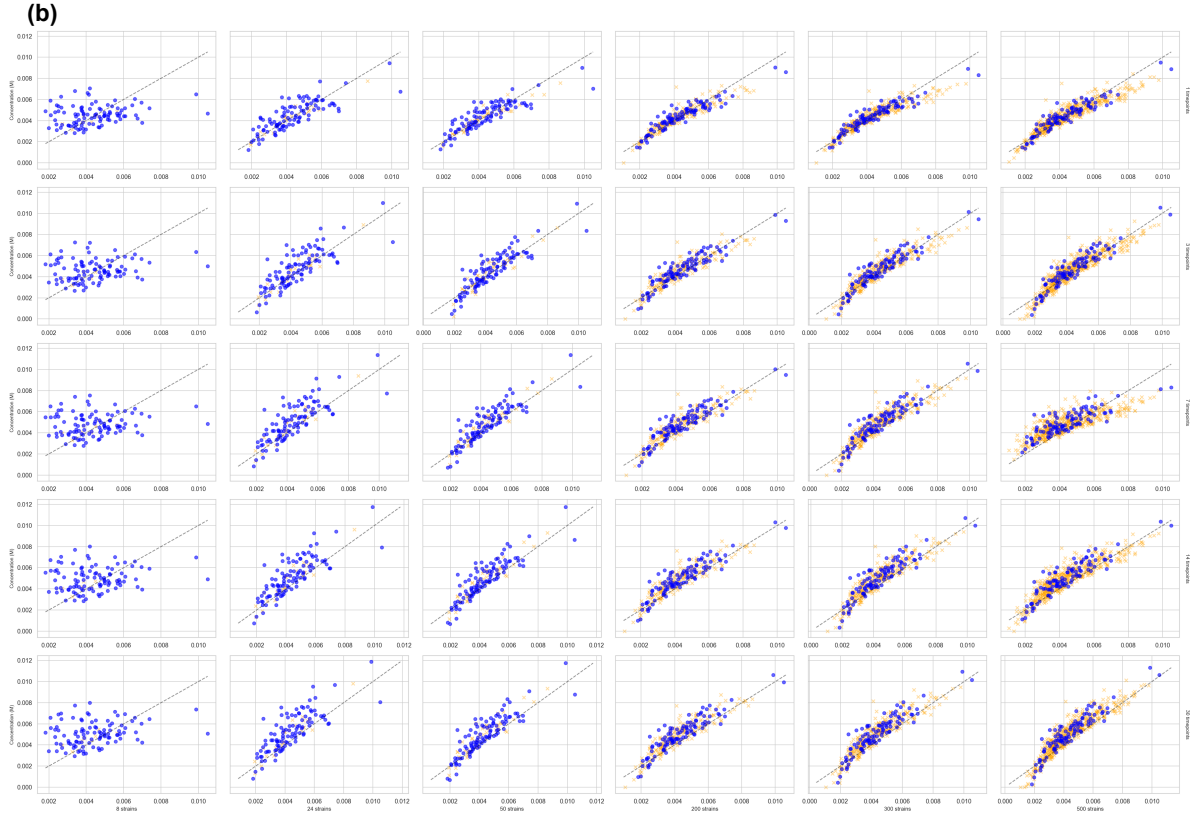
Final p-CA prediction

(a)

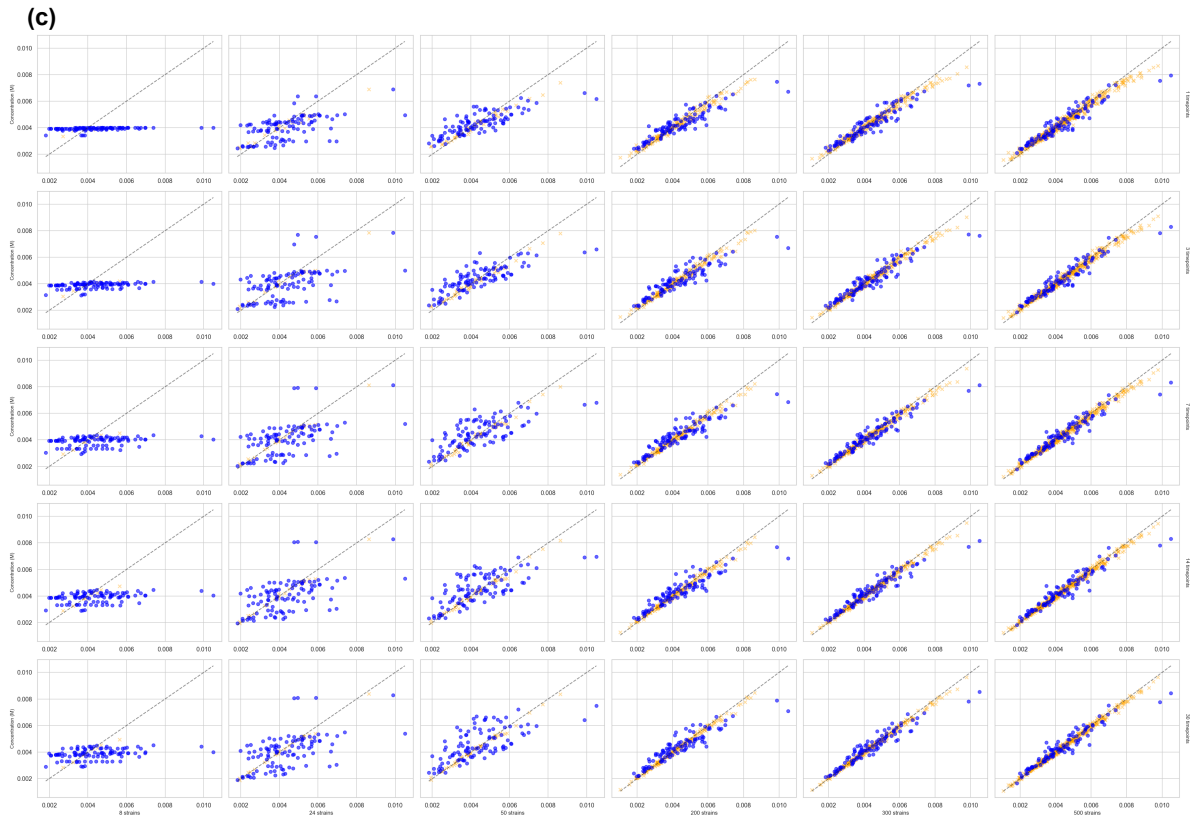


(a) Hybrid model

(b)



(b) NODE



(c) XGBoost

Figure C.5: Scatter plots of predicted versus true final p-CA concentrations for all model types across scenarios. The figures represent a typical example, showing the predictions for one of the fifty trained models.

Variance within subsets

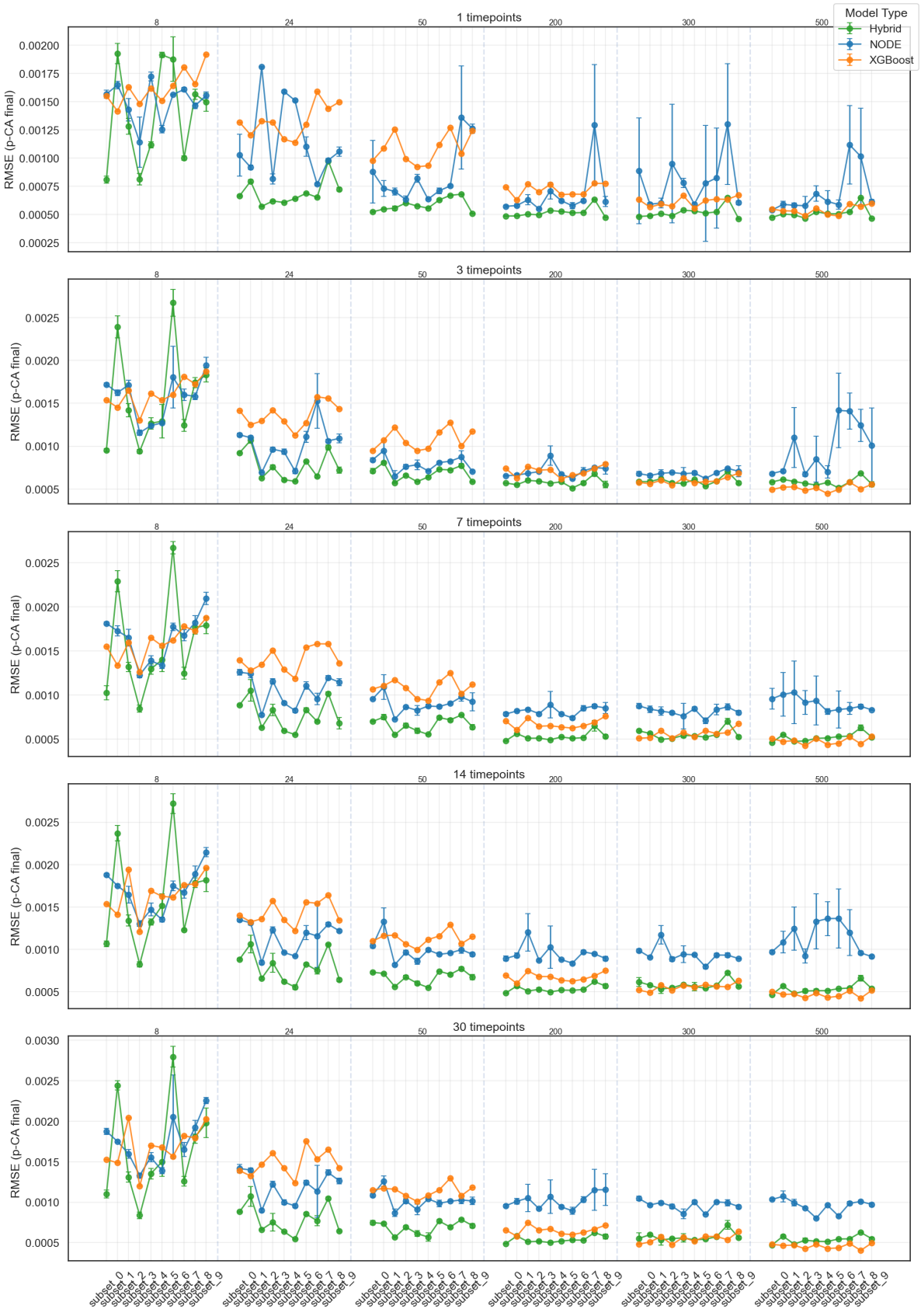
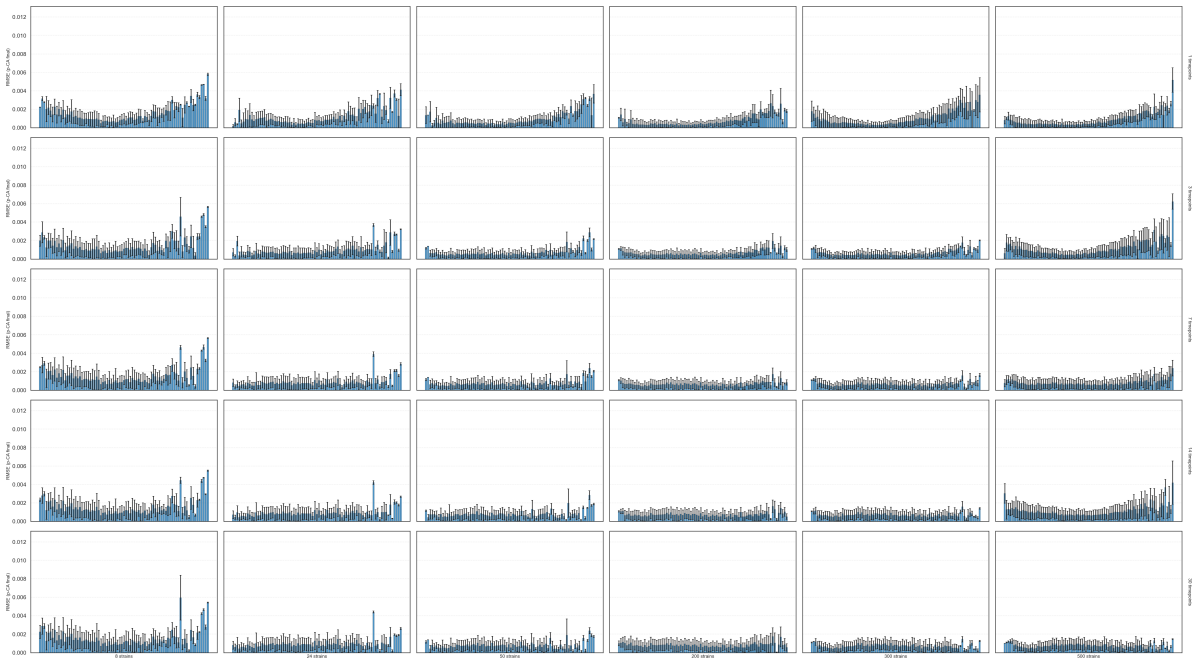


Figure C.6: Variation in model performance across subsets. Lines represent the average error per subset, with error bars indicating variability across different random seeds. Subsets are grouped by training set size and model configuration.

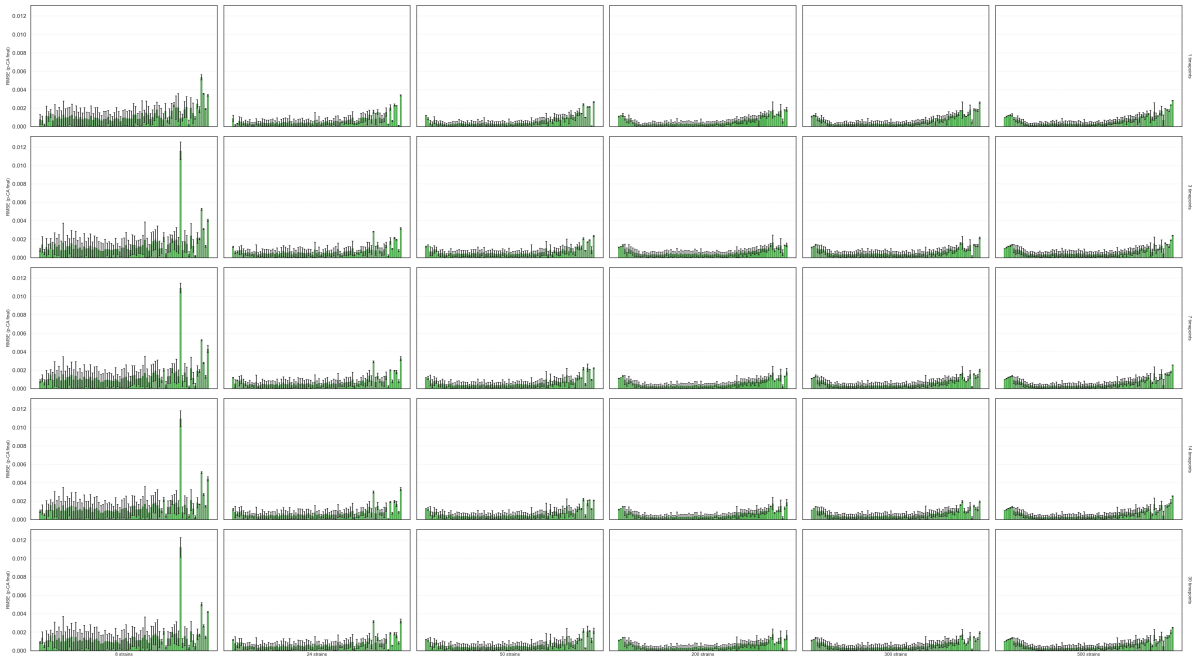
Error per strain

(a)



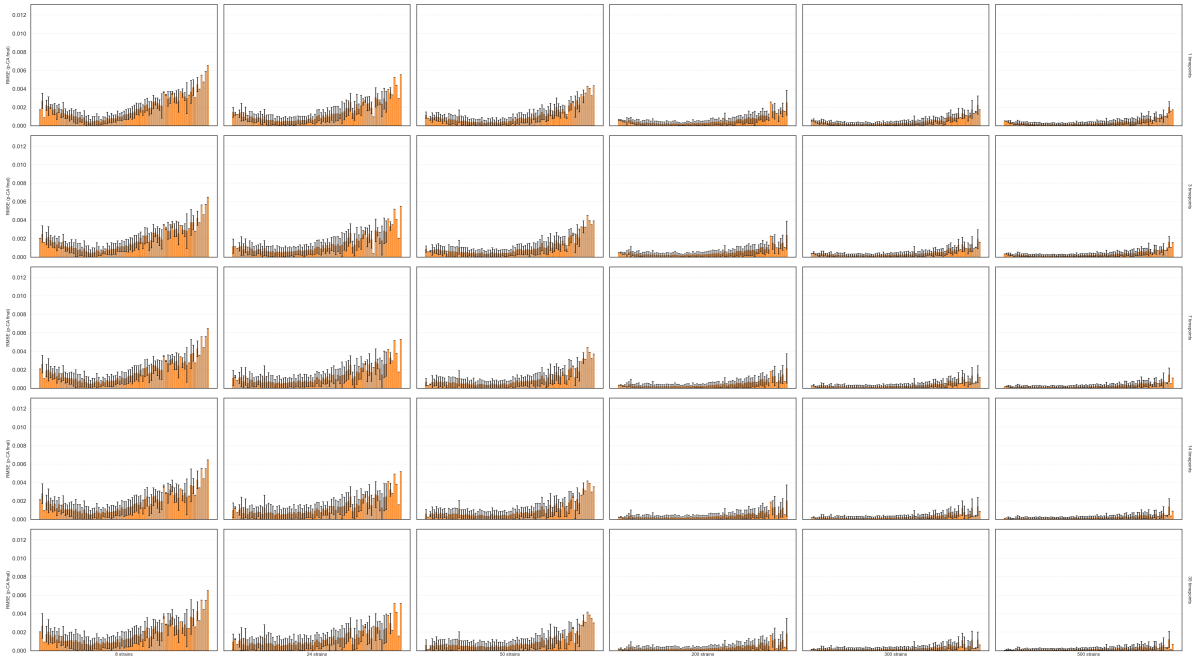
(a) NODE

(b)



(b) Hybrid

(c)



(c) XGBoost

Figure C.7: Mean RMSE per test strain for final p-CA prediction, sorted by true final concentration. Results are shown for different model types.

D Model performance vs mechanistic knowledge

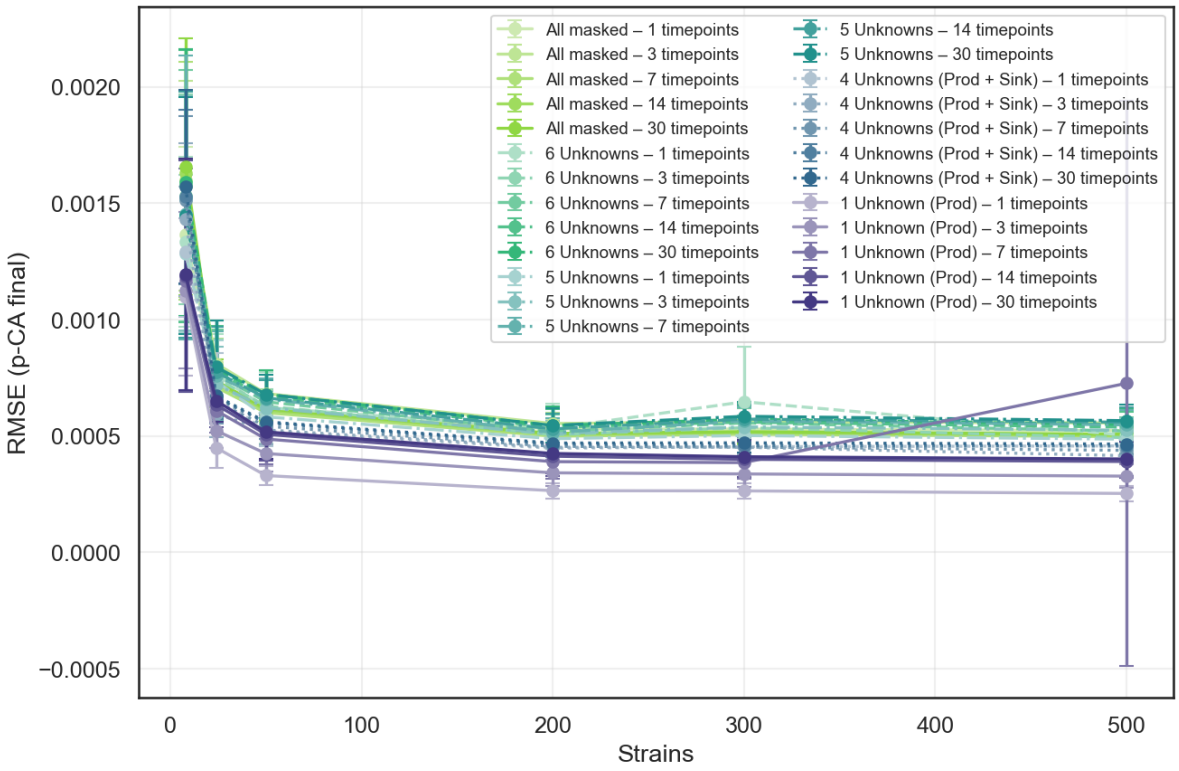


Figure D.1: Effect of mechanistic knowledge on model performance. Mean RMSE with error bars is shown across masking levels, timepoints, and training set sizes.

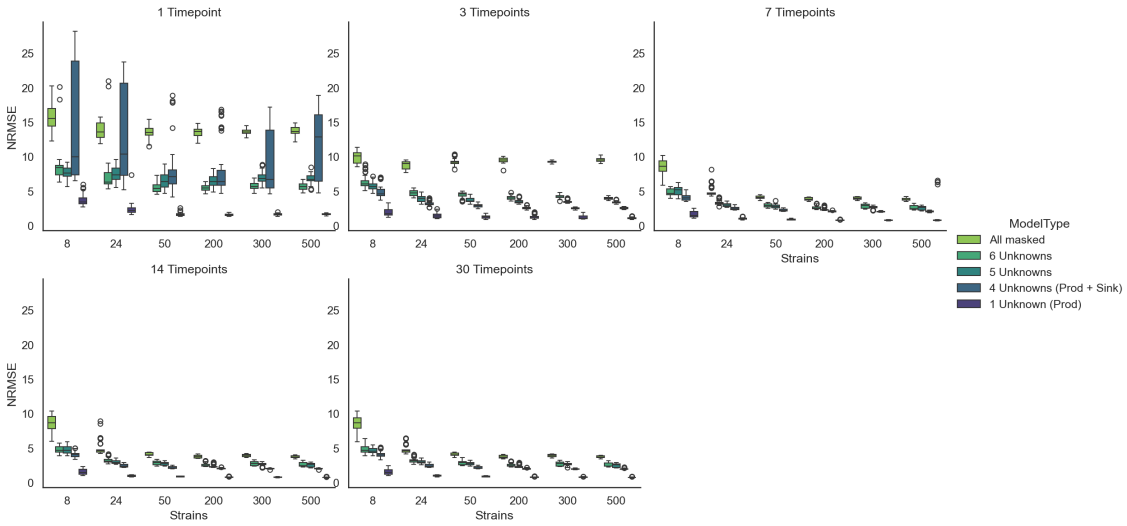


Figure D.2: Distribution of NRMSE across different levels of underlying mechanistic knowledge. Boxplots summarize performance across all scenarios for different levels of reaction masking. Evaluated on all timepoints and metabolites.

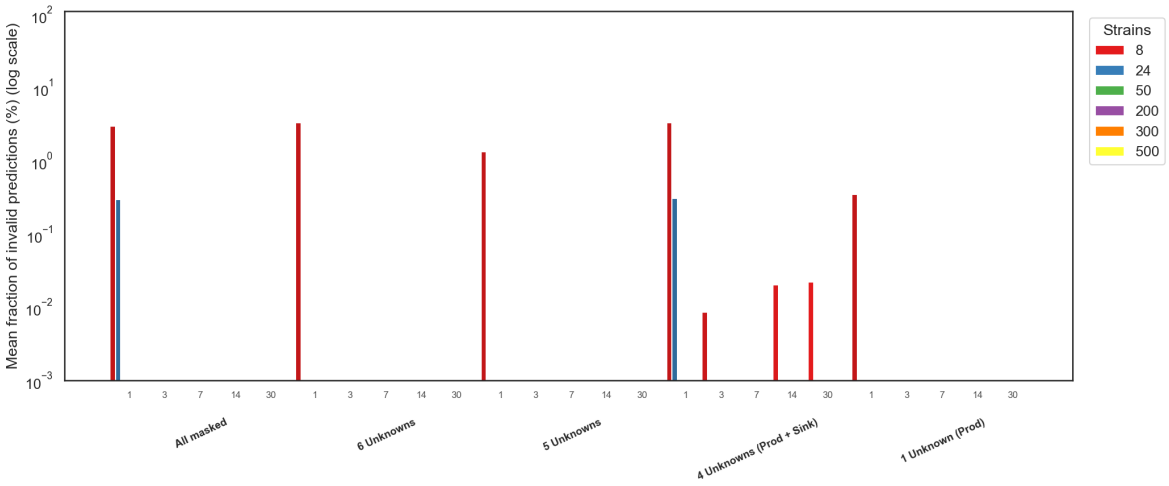


Figure D.3: Mean fraction of invalid evaluation points (NaN/Inf) (%) across masking scenarios. Values are shown on a logarithmic scale and represent the percentage of excluded data points for different training configurations.

E Model performance vs model complexity

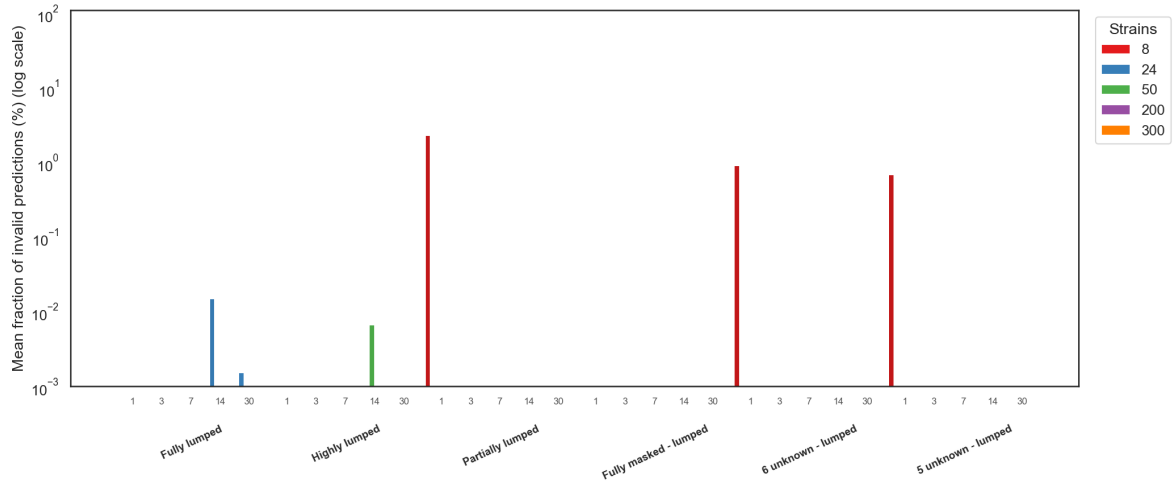


Figure E.1: Mean fraction of invalid evaluation points (NaN/Inf) (%) across lumping scenarios. Values are shown on a logarithmic scale and represent the percentage of excluded data points for different training configurations.

E.0.1 Stoichiometry for lumped networks

To train a model using a lumped version of the metabolic pathway a new stoichiometric matrix is required to match the new total number of reactions and metabolites. This study considered two ways of creating the new matrix. One way is to keep the original stoichiometry, but condense it into a smaller matrix based on the new number of reactions. For example, in the ground truth pathway, a total of 1 mole of substrate is converted, through different metabolites, into one mole of product. In the lumped version, that doesn't have these intermediate metabolites, we can immediately turn 3 mole substrate into one mole of product. However, this means that the mass balance now doesn't exactly close, because of the missing metabolites that disappear through sink reactions in the original pathway, that the new pathway doesn't account for. An issue could be that this information therefore wouldn't be known in a real setting, as you need detailed information of the complete pathway to know the exact stoichiometry, that is why a different approach was also tried. In this approach the matrix was constructed by calculating the stoichiometry based on the amount of carbon atoms in the metabolites. For example, glucose has 6 carbon atoms and it is converted into p-CA, which has 9 carbon atoms. So, from that information a matrix is constructed in which 3 mole of substrate is converted into 2 mole of product. This means that for this approach the balance does close, but the information is not exactly accurate with the underlying ground truth as one mole more is produced per mole of substrate compared to the ground truth. Both approaches were tried to see how much of a difference this makes. The problem only applies for the models that no longer contain the E4P metabolite, as this is where the extra mass from substrate disappears. Otherwise there is only one way to construct the stoichiometric matrix. Therefore, this applies to three of the lumped scenarios (Figure E.2). When comparing the errors for both approaches it can be concluded that the first approach does give a slightly lower error, but the difference is very minimal. The difference mostly noticeable when assessing only the final product concentration and training on a very limited number of strains (8). Therefore, when the true underlying stoichiometry is not known and it is based only on the known reactions, it does not significantly degrade the performance. So, not having this information does not have a prohibitive negative effect on the performance. The experiments in the main text were all performed using the first approach.

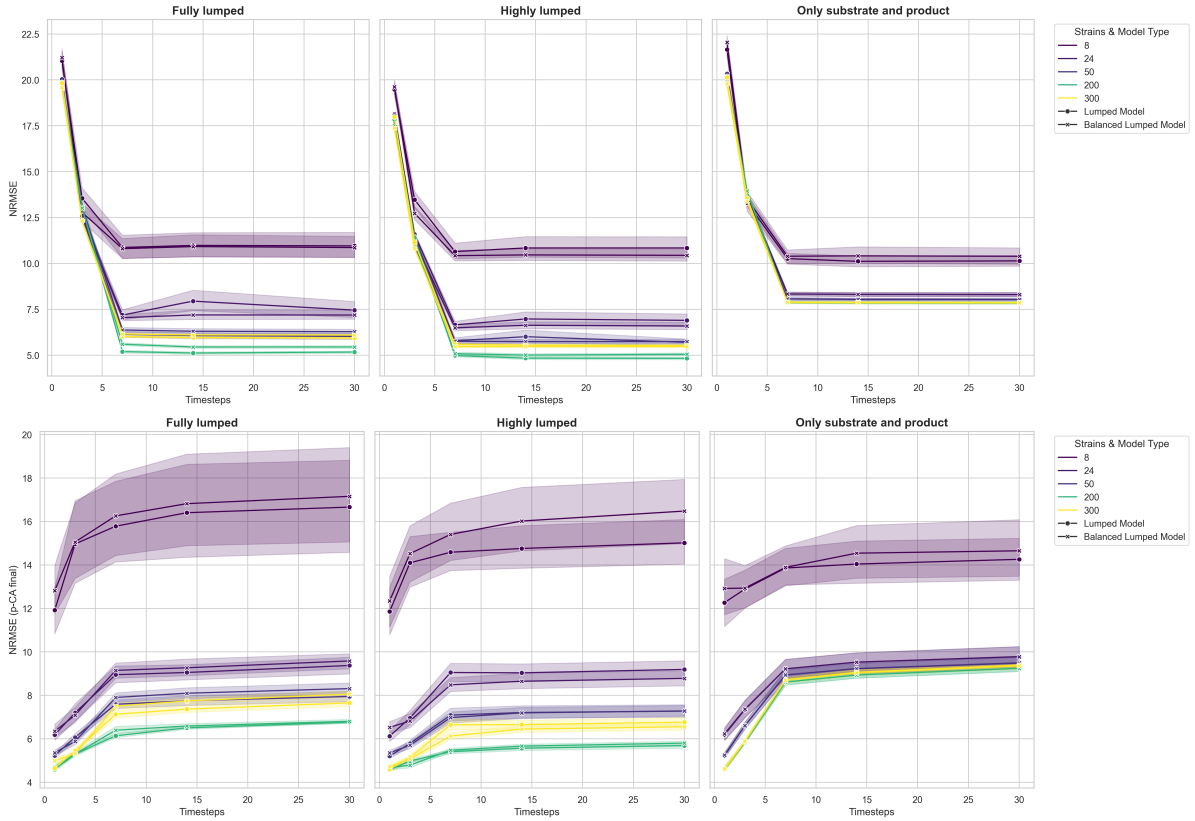


Figure E.2: Comparison of two approaches for constructing stoichiometric matrices in lumped models. Top row: NRMSE across timepoints; bottom row: final product error. Colours denote training set size, symbols distinguish construction methods.

F Robustness to noise

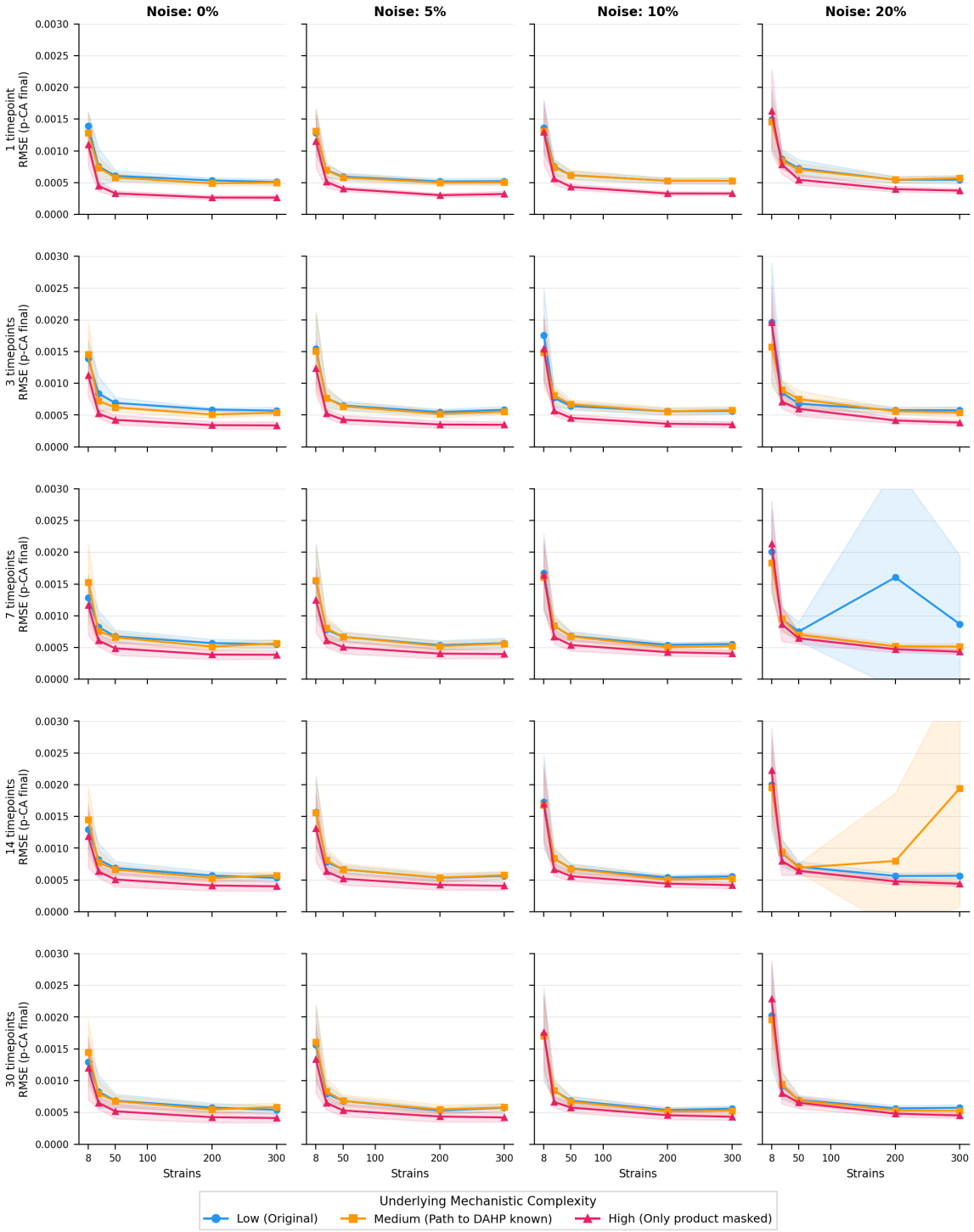


Figure F.1: Effect of observational noise on final product prediction accuracy. Panels are organized by noise level (columns: 0%, 5%, 10%, 20%) and number of training timepoints (rows: 1, 3, 7, 14, and 30). The x-axis shows the number of training strains, and the y-axis shows the root mean squared error (RMSE) of the final p-coumaric acid (p-CA) concentration. Colored lines represent different levels of mechanistic complexity, and shaded regions indicate variability across runs.