Data-Driven Discovery of Mechanistic Ecosystem Models with LLMs

Scott Spillias1,2[[1]](#footnote-20)

Jacob Rogers3

Fabio Boschetti5

Magda Guglielmo6

Rowan Trebilco1,2

SukYee Yong6

Beth Fulton1,2

2025-06-10

Abstract

Ecosystem models are essential for ecosystem management, but their development traditionally requires significant time and expertise, creating bottlenecks in addressing urgent environmental challenges. We present “AI for Models of Ecosystems” (AIME), a novel framework that integrates large language models (LLMs) with evolutionary optimization to automate the discovery of interpretable ecological models from time-series data. AIME addresses the inverse problem of inferring ecologically meaningful mechanistic models that explain observed data while maintaining biological plausibility. We validate AIME through two complementary marine case studies: (1) a nutrient-phytoplankton-zooplankton model to test recovery of known ecological relationships, and (2) a Crown-of-Thorns starfish model to assess practical utility for reef management. In the first case, our best models displayed almost perfect recovery of known ecological dynamics while maintaining strong predictive performance across multivariate time-series. In the second case, AIME generated models comparable to human expert models, with the best model successfully capturing COTS outbreak dynamics and demonstrating strong out-of-sample predictive power. AIME produces interpretable models with meaningful parameters that capture real biological processes, facilitating scientific insight and management applications. By dramatically accelerating model development while maintaining scientific rigor, AIME offers a powerful new tool for researchers and managers facing urgent ecological challenges in rapidly changing environments.

1CSIRO Environment, Hobart, Australia  
2Centre for Marine Socio-Ecology, University of Tasmania, Hobart, Australia  
3Department of Agriculture, Fisheries and Forestry, Canberra, Australia  
4CSIRO Environment, St. Lucia, Australia  
5CSIRO Environment, IOMRC Crawley, Australia  
6CSIRO IM&T, Eveleigh, Australia

**Keywords:** Artificial Intelligence, Ecological modelling, Evolutionary Algorithms, Large Language Models, Marine Ecosystems, Crown-of-Thorns Starfish, Great Barrier Reef, time-series forecasting

# Introduction

Ecosystem models provide essential tools for managing complex interactions between nature and people , but their development traditionally requires significant time and expertise, creating a bottleneck in addressing urgent environmental challenges — particularly as climate change demands rapid, adaptable approaches for ecosystem management .

Artificial Intelligence offers promising solutions to these modelling challenges, with potential to accelerate model development and enhance adaptability . While initial efforts to apply AI in ecological modelling focused on machine learning approaches that rely on black-box methods , emerging techniques in equation discovery and automated scientific discovery show particular promise . These methods can derive interpretable mathematical relationships directly from data, offering advantages over statistical emulators when modelling novel environmental conditions . Similarly, attempts to leverage large language models (LLMs) for direct time-series prediction , though successful in other fields, are unsuited for producing reliable ecological insights or testing management interventions. For instance, work on multimodal LLMs for environmental prediction achieves impressive accuracy in forecasting physical variables like streamflow and water temperature, but does not address the mechanistic relationships needed for ecosystem management.

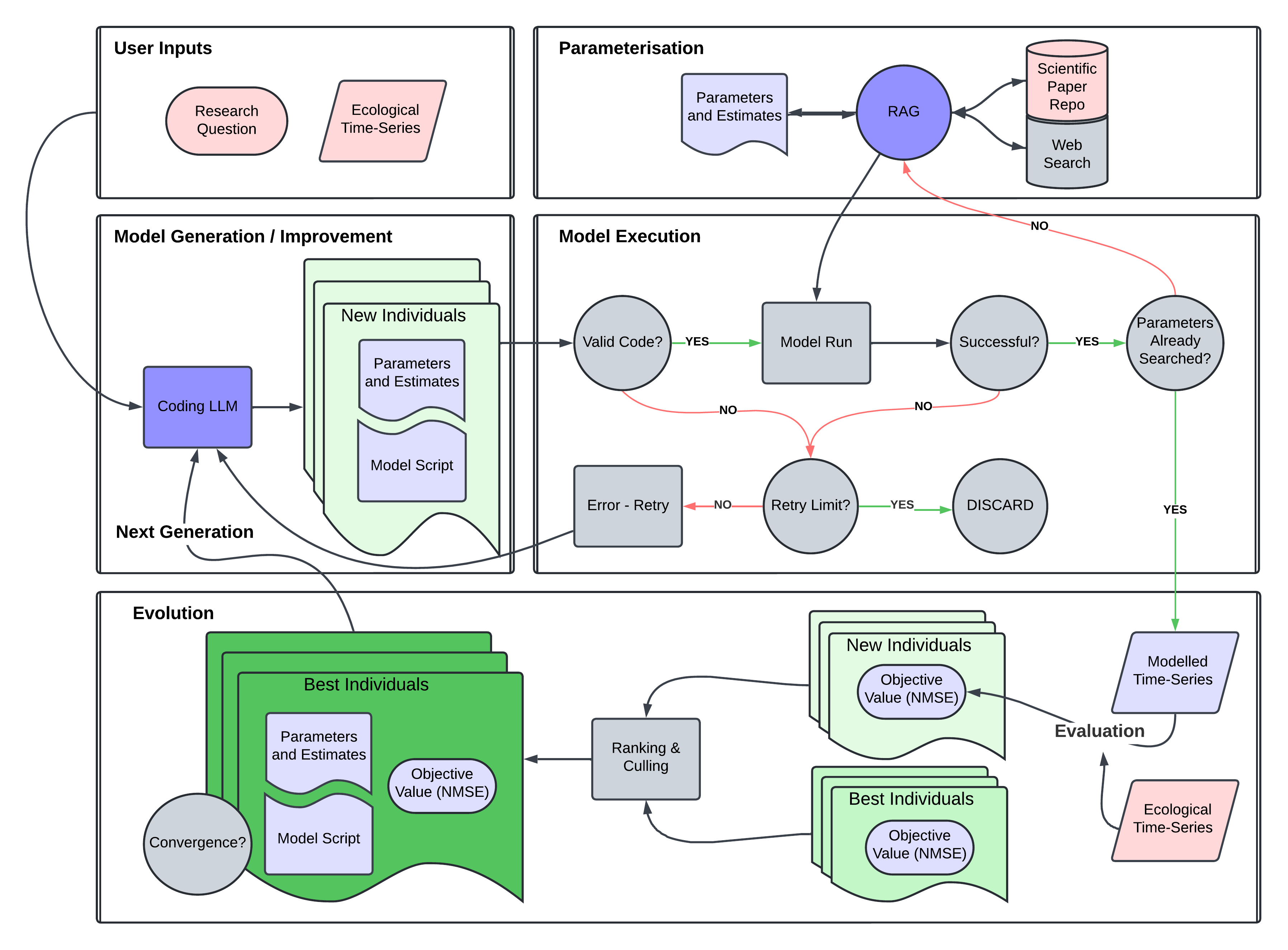
Rather than using AI to replace traditional modelling approaches, recent advances in AI coding capabilities suggest a more promising direction . Language models like o3-mini and Claude can assist in constructing mechanistic models , maintaining interpretability while accelerating the development process . Recent demonstrations of LLMs automating scientific processes, from autonomous chemical experimentation to biomedical research , and even fully automated scientific discovery , highlight their potential for systematic scientific work. The key challenge lies in developing frameworks that can systematically harness these capabilities while ensuring scientific rigor and maintaining human oversight in the discovery process . To the best of our knowledge, such an approach has not been attempted yet in ecological modelling.

To address this challenge, we present “AI for Models of Ecosystems” (AIME), a novel framework that requires minimal inputs - only time-series data and research questions and aims to produce ecologically sound mathematical models that explain the time-series data. AIME addresses the inverse problem of inferring ecologically meaningful mechanistic models and parameters that causally explain observed data. The inferred models can then be used to test management interventions and scenarios. When solving inverse problems through numerical optimization, practitioners usually predetermine the model and its allowable parameter ranges, which defines both the parameter space and its mapping to the observation space. Less commonly, the model itself is inferred directly from observations, as in Scientific Machine Learning (SciML), System Identification, and Automated Algorithm Discovery. Successful model reconstruction in such scenarios requires extensive data to ensure statistical relations are represented in the observations. In addition, a common challenge across all inverse problem approaches is the need to impose constraints that prevent numerically accurate yet empirically unrealistic solutions. AIME differs from traditional and SciML approaches by leveraging information available to current LLMs to impose ecologically meaningful constraints on both forward models and parameter ranges. This addresses the significant limitation of sparse ecological observations. Within this framework, AIME employs Template Model Builder (TMB, a R/C++ library for efficient parameter estimation in complex nonlinear models) as its foundation, providing a rigorous statistical framework for ecological modelling. The system operates through an iterative process where an LLM generates candidate model structures as TMB-compatible equations, which are then evaluated against time-series data using normalized objective functions (where lower values indicate better model fit). These models undergo evolutionary optimization across multiple generations, where successful structures (referred to as “individuals”) are selected and improved upon, while underperforming models are culled. This evolutionary approach systematically produces increasingly accurate representations of ecosystem dynamics.

We validate AIME through two complementary case studies that test different aspects of ecological modelling, each involving both dependent variables (state variables predicted by the model) and forcing variables (external drivers affecting the system). First, we evaluate the framework’s ability to recover fundamental ecological theory using synthetic data generated from a well-established nutrient-phytoplankton-zooplankton (NPZ) model . This controlled experiment tests AIME’s equation-learning capabilities by comparing discovered equations against known mathematical relationships that represent core ecological processes. For our NPZ case study, we additionally evaluate models using ecological accuracy scores (on a scale of 0-8), which measure how well the generated models recover known ecological mechanisms from the reference model. These scores assess specific components like nutrient uptake, phytoplankton growth, and zooplankton dynamics, with higher scores indicating closer alignment with established ecological theory. Second, we assess AIME’s ability to address management-relevant predictions using synthetic data based on Crown-of-Thorns starfish (COTS) populations on the Great Barrier Reef, derived from existing MICE models . The COTS case study, with three dependent variables (COTS abundance, fast-growing coral cover, and slow-growing coral cover) and two forcing variables (temperature and chlorophyll-a), tests the framework’s robustness while focusing on a specific management challenge: predicting outbreaks in a complex predator-prey system. Through systematic comparison of different LLMs (o3-mini from OpenAI, Claude Sonnet models from Anthropic, and Gemini models from Google), we evaluate how different AI capabilities affect model performance. This comparative approach demonstrates how our evolutionary framework can both recover theoretical ecological relationships and generate practical models for ecosystem management.

# Methods

At its core, AIME integrates Large Language Models (LLMs) for generating and modifying model structures, Template Model Builder (TMB) for statistical parameter estimation, and evolutionary algorithms for systematic model improvement.



Conceptual diagram of the automated ecological modelling framework, AIME. The workflow consists of four main components: (1) User Inputs, where research questions and ecological time-series data are provided; (2) Parameterisation, utilizing RAG-enhanced literature search to estimate parameter values; (3) Model Generation/Improvement, where the Coding LLM creates new individuals with model scripts and parameters; (4) Model Execution, where the LLM’s model code is implemented and TMB is used to optimise paramter values; and (5) Evolution, which evaluates model performance through individual assessment, error handling, and ranking-based selection.

## AIME Framework

### Model Generation and Improvement

AIME uses LLMs to write and modify computer code through Aider , which is a coding assistant that can create, modify, and interpret local files. Aider can be used in the command-line or called within python scripts, as we have done here, and can receive text and/or images as input, depending on whether the underlying LLM is ’multi-modal’ (i.e., can interpret text and images). Each model instance, referred to as an individual in our evolutionary framework, consists of three components: (1) a TMB-compatible dynamic model written in C++ that implements a system of equations, (2) a parameters file containing initial values and bounds, and (3) a documentation file explaining the ecological meaning of the model equations (see Section [7.1](#subsec:initial_model_prompt) for the complete prompt).

The LLM generates initial parameter estimates for pre-testing model structure before optimization begins. For each parameter, it assigns a priority number that determines optimization order, following established practices in ecosystem modelling .

When improving existing models, if multi-modal (i.e. can receive images as input) the LLM analyzes performance plots comparing predictions to historical data, otherwise the LLM receives a structured file showing the model fit residuals. After interpreting the model fit, AIME makes targeted, ecologically meaningful changes to model equations, implementing one modification at a time to maintain transparency and traceability of successful modelling strategies (see Section [7.3](#subsec:model_improvement_prompt)).

### Parameterisation

Upon initialization, the LLM estimates parameter values for each parameter supplied in the model. This initial estimation allows for the subsequent execution of the model, and the discovery of structural or syntactical errors in the LLM-generated code. If a model is successful in compiling and running, AIME goes on to find evidence to support better values for parameters. Building on the success of LLM-based extraction from ecological literature , the system implements a Retrieval-Augmented Generation (RAG) architecture to search scientific literature (see Section [6](#subsec:rag_architecture) for detailed RAG implementation).

The RAG process works as follows: First, the system prompts an LLM to create detailed semantic descriptions of each parameter, expanding beyond the basic descriptions provided by the coding LLM. For example, if the coding LLM defines a parameter as "growth rate of phytoplankton," the RAG system might expand this to "maximum specific growth rate of marine phytoplankton in nutrient-rich conditions, measured per day." These enhanced descriptions aim to improve the relevance of search results when querying literature databases.

To find appropriate parameter values, the RAG system employs a structured, multi-source search strategy. First, it searches a local collection of scientific papers (see Section [5](#subsec:curated_literature) for the complete collection used for the CoTS case study) using ChromaDB as a persistent vector store, with documents processed into semantic chunks to enable precise retrieval. Second, it queries the Semantic Scholar database , focusing on highly-cited papers in biology, mathematics, and environmental science. Third, it performs general web searches through the Serper API to capture additional relevant sources. The system combines results from all three sources to build a comprehensive understanding of each parameter’s possible values and ecological meaning.

The RAG system then uses LLMs to extract numerical values from the search results, determining not only parameter values but also their valid ranges. The prompt instructs the LLM to identify minimum, maximum, and typical values for each parameter, along with their units and citation information (see Section [7.2](#subsec:parameter_enhancement_prompt)). All parameter information is stored in a structured JSON database that includes minimum and maximum bounds, units, and citations to source literature. For this proof of concept effort, all parameters, including those with values found from literature, are treated as estimable parameters in the optimization process, with literature-derived values, mins and maxes serving as initial estimates and bounds rather than fixed constraints. This approach allows the optimization process to refine parameter values while still benefiting from literature-informed starting points and biologically plausible ranges.

### Model Execution and Error Handling

Because LLMs often make trivial mistakes in their outputs, we developed an error handling system to address common issues. On occasion, the LLM coder will attempt to create a system of equations with circular logic (which we refer to as ’data leakage’). Data leakage occurs when the model directly uses observed values from the current time step to predict those same values, instead of properly predicting values using only information from previous time steps. To prevent this, we implement a set of code validation checks to ensure that the submitted model is properly formatted and free from logical inconsistencies.

The framework executes models through TMB , an approach which underpins several marine ecosystem modelling frameworks . TMB provides automatic differentiation techniques for efficient parameter estimation, with optimization priorities assigned following established practices . TMB’s prioritization system operates through recording which parameters and intermediate calculations are actually needed for the objective function and its derivatives. This computational efficiency is particularly valuable for complex ecological models, as TMB automatically identifies which parameters influence specific likelihood components and focuses derivative calculations only on relevant parameters, significantly accelerating the optimization process.

For models that pass initial validation, AIME addresses compilation errors through automated analysis of error messages and implementation of appropriate fixes. For numerical instabilities, the system employs progressive simplification of model structure while maintaining ecological relevance. Each model variant receives up to five iterations of fixes, with later iterations favoring simpler model structures that can be iteratively improved. The specific prompts used for error handling are provided in Section [7.4](#subsec:error_handling_prompt).

### Model Evaluation

For each response variable , we calculate a normalized mean squared error:

where represents observed values for variable at time , represents corresponding model predictions, is the standard deviation of the observed values for variable , and is the number of observations. The final objective function value is the mean across all response variables:

where is the number of response variables. This approach ensures that each time series contributes equally to the objective function regardless of its scale or units. For simplisticity in this proof-of-concept, we did not weight the time-series in the objective function, however this might prove useful in future work to prioritize uncovering key dynamics.

### Evolutionary Algorithm Implementation

The system maintains a population of model instances, which we refer to as ‘individuals’, where each individual represents a complete model implementation including its equations, parameters, and performance metrics. Within each generation, individuals undergo parameter optimization using Template Model Builder to find optimal parameter values for their current model structure.

After parameter optimization, individuals are evaluated based on their prediction accuracy. Those achieving the lowest prediction errors (objective values) are selected to become parents for the next generation, while less well-performing individuals are culled and non-functioning ones (those that fail to compile or execute) are discarded.

At the beginning of each new generation, the system creates new individuals in two ways: by making targeted structural modifications to the best-performing parent individuals from the previous generation, and by creating entirely new individuals from scratch when there are not enough functioning individuals.

## Validation Experiments

We conducted two complementary validation case studies of AIME. The first validation experiment aimed to see if AIME could recover known model equations from synthetic time-series data, whilst the second validation experiment examined real-world applicability through modelling a set of time-series where noise was added to the synthetic data \*\*\*need specific understadning of how these data were collected from Jacob\*\*\*

### Retrieving Model Equations – NPZ Case Study

We conducted a controlled experiment using synthetic time-series data generated by a well-established nutrient-phytoplankton-zooplankton (NPZ) model from , whose dynamics are well-studied . The complete system of equations is presented in Section [7.5](#subsec:npz_evaluation_prompt) of the Supplementary Information. This validation tested our framework’s ability to rediscover established ecological relationships from synthetic data where the underlying equations of a system are known, providing a rigorous assessment of the system’s equation-learning capabilities.

In addition to monitoring the convergence of AIME’s modelled time-series towards the provided time-series data, we evaluated the framework’s ability to recover six key ecological characteristics from the original model, each based on a discrete term in the system of three equations: nutrient uptake by phytoplankton with Michaelis-Menten kinetics and self-shading, nutrient recycling through zooplankton predation and excretion, environmental mixing of nutrients, phytoplankton growth through nutrient uptake, phytoplankton losses through mortality and predation, and zooplankton population dynamics.

During evolution, for each ‘best performer’ in a generation, we used Claude Sonnet-3.5 to evaluate each model and provide a score between 0 and 1 for each ecological characteristic. The scoring system was designed to be interpretable and verifiable by ecological experts. For each characteristic, the LLM was provided with:

* The original equation term from the reference NPZ model
* The corresponding term from the generated model
* Specific criteria for scoring the similarity between the two terms

For example, when evaluating nutrient uptake by phytoplankton, a score of 1.0 would be assigned if the generated model correctly implemented Michaelis-Menten kinetics with self-shading (matching the form ), while a score of 0.5 might be given if only the basic Michaelis-Menten term was present without self-shading. A score of 0 would indicate no representation of nutrient uptake. The complete evaluation prompt with detailed scoring criteria for each ecological characteristic is provided in Section [7.5](#subsec:npz_evaluation_prompt). This additional evaluation allowed us to better understand whether objective value improvements were indeed related to improved ecological understanding, or whether they were instead related to spurious mathematical relationships with limited ecological basis. We ran this evolutionary process using Sonnet-3.5, in four individuals for 60 generations and a convergence threshold of 0.05.

We focused on the framework’s ability to recover known ecological relationships from synthetic data. We analyzed the relationship between ecological accuracy scores and objective values using two-sided Pearson’s product-moment correlation tests (cor.test in R). For each ecological characteristic, we calculated the correlation coefficient (r) and tested the null hypothesis that the true correlation is 0, with the alternative hypothesis that it is not 0. The resulting p-value indicates the probability of observing such a correlation by chance if no true relationship exists, with values below 0.05 considered statistically significant. This approach allowed us to evaluate whether improvements in predictive accuracy were achieved through mathematically sound ecological mechanisms rather than through overfitting.

### Fitting Real World Data – CoTS Case Study

The Crown-of-Thorns starfish (CoTS) case study examined real-world applicability through modelling populations of CoTS and their prey, coral, on the Great Barrier Reef. This case study also made two external forcing time-series available to AIME, sea-surface temperature and CoTS larval immigration quanitites. We tested the leading LLMs (o3-mini from OpenAI, Claude Sonnet 3.6 and 3.7 from Anthropic, and Gemini-2.0 Flash and Gemini-2.5-pro from Google) within our framework and evaluated AIME’s ability to match the model created by a human expert in the same context. \*\*\*this needs more elaboration, including citations to published models, etc.\*\*\*

Due to time and cost constraints we only performed limited tests using each LLM, where we initialized populations of four individuals for ten generations each. We calibrated these population parameters by balancing the cost of running an individual population against the rate of convergence that we found in our initial tests of the system.

We tracked several key performance metrics for each population:

* Runtime performance: Total runtime and per-generation computation time
* Error resolution: Number of iterations required to achieve successful model implementation in each generation
* Model stability: Proportion of successful, culled (underperforming), and numerically unstable models per generation Z

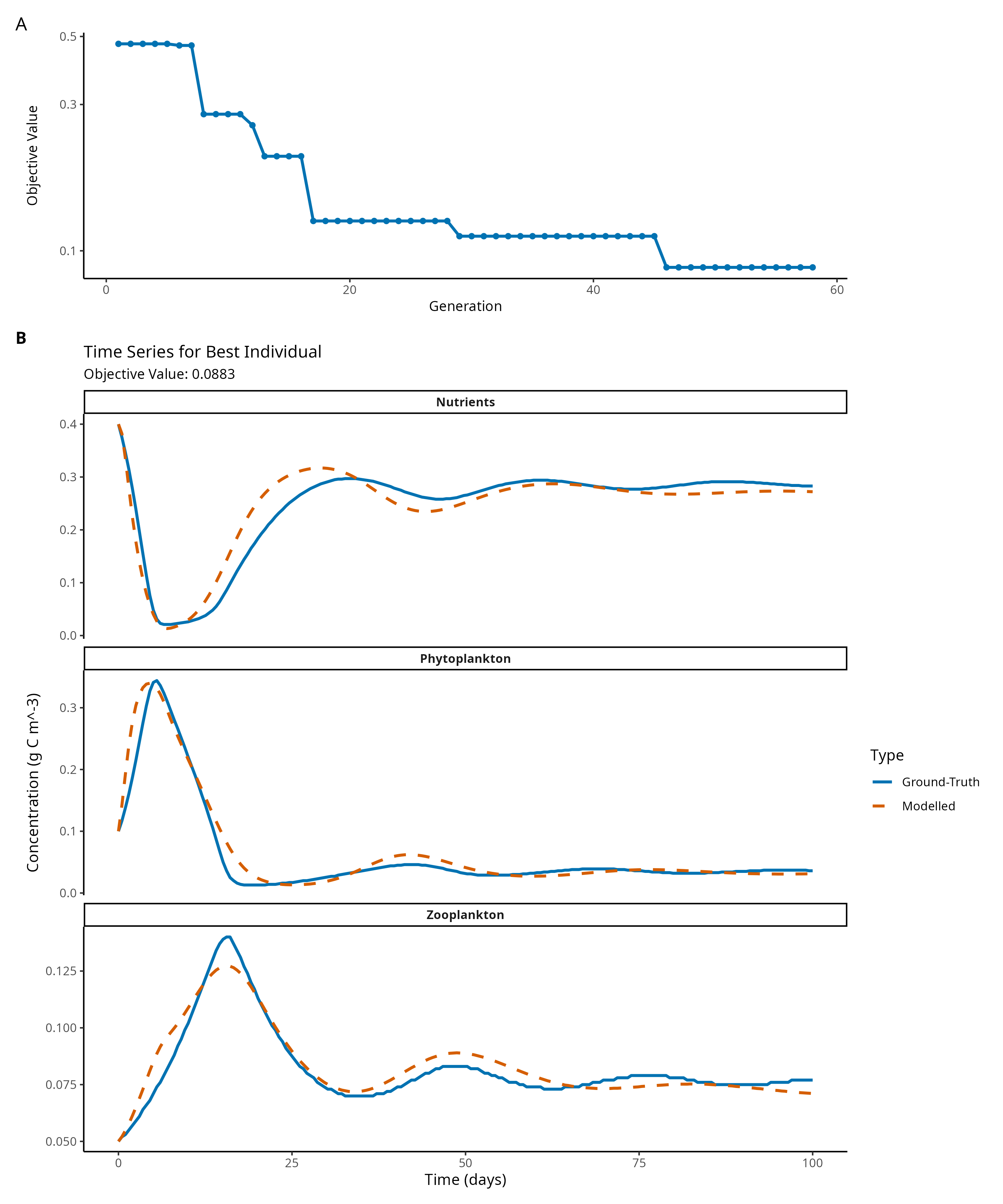
We also analyzed the evolutionary trajectories of successful models by tracking their lineage from initial to final states, documenting the frequency and magnitude of improvements across generations. This included measuring the number of generations required to reach best performance and the proportion of attempts that resulted in improved models.

We also implemented a single evolution where we constructed a temporal cross-validation approach by partitioning the time series data into training (pre-2000, approximately 70%) and testing (2000-2005, approximately 30%) sets. This allowed us to evaluate both in-sample fit and out-of-sample prediction accuracy. For each ecosystem component (COTS abundance, fast-growing coral cover, and slow-growing coral cover), we calculated root mean square error (RMSE), mean absolute error (MAE), and R² values to quantify prediction accuracy. By comparing these metrics against those of the human-developed reference model, we could assess whether our automated approach could match expert-level performance in a real-world ecological application.

# Results

## Retrieving Model Equations – NPZ Case Study

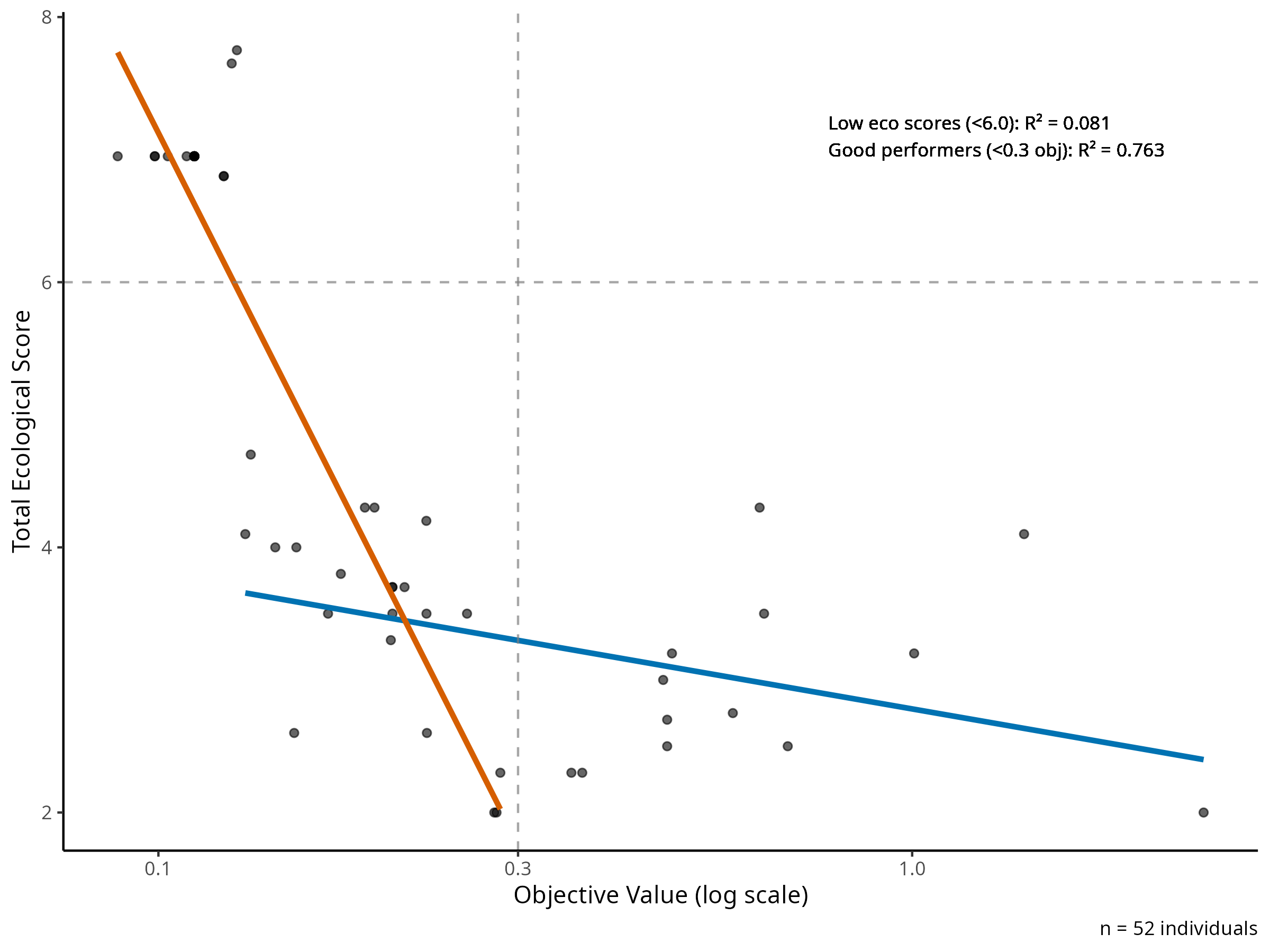
Analysis of the NPZ validation study revealed that while AIME did not perfectly recover the original model equations after 60 generations, it achieved substantial success in reproducing key ecological mechanisms in its best-performing models. The top models achieved objective values as low as 0.0883 (lower is better) while maintaining high ecological accuracy scores (maximum total score of 7.75 out of 8; higher is better), demonstrating remarkable accuracy in reproducing NPZ dynamics. These best performers showed particularly strong recovery of phytoplankton growth dynamics and zooplankton equations (scores up to 1.0 for both; where 1.0 reflects perfect recovery of underlying dynamics), demonstrating the framework’s ability to rediscover fundamental ecological relationships. However, even the best models struggled to recover nutrient mixing terms (maximum score of 0 - indicating this mechanism was never suggested in any capacity), suggesting some ecological mechanisms were more challenging to identify from time-series data alone.



Evolution and performance of the best NPZ model. (A) Training progress showing the objective value across generations on a log-scale. (B) Time series comparison between ground-truth and modelled NPZ dynamics for the best-performing model (objective value = 0.0883). The plots show the temporal evolution of nutrient, phytoplankton, and zooplankton concentrations (g C m-3). Blue solid lines represent ground-truth data, while orange dashed lines show model predictions, demonstrating the model’s ability to capture key ecological patterns and phase relationships between trophic levels.

Importantly, we found negative correlations between objective values and ecological accuracy scores, indicating that improvements in model fit were generally achieved through ecologically sound mechanisms rather than overfitting. The strongest correlation was observed for phytoplankton growth equations (r = -0.461, p = 0.002), followed by nutrient uptake (r = -0.399, p = 0.008). The total ecological score also showed a significant negative correlation with objective values (r = -0.380, p = 0.012), suggesting that models achieving better fits tended to incorporate more correct ecological mechanisms.

The best-performing models achieved objective values as low as 0.112, demonstrating strong predictive accuracy while maintaining meaningful ecological structure. A detailed analysis of individual ecological characteristics (see Figure [6](#fig:ecological_characteristics) in Supplementary Materials) revealed that some mechanisms were more readily recovered than others.



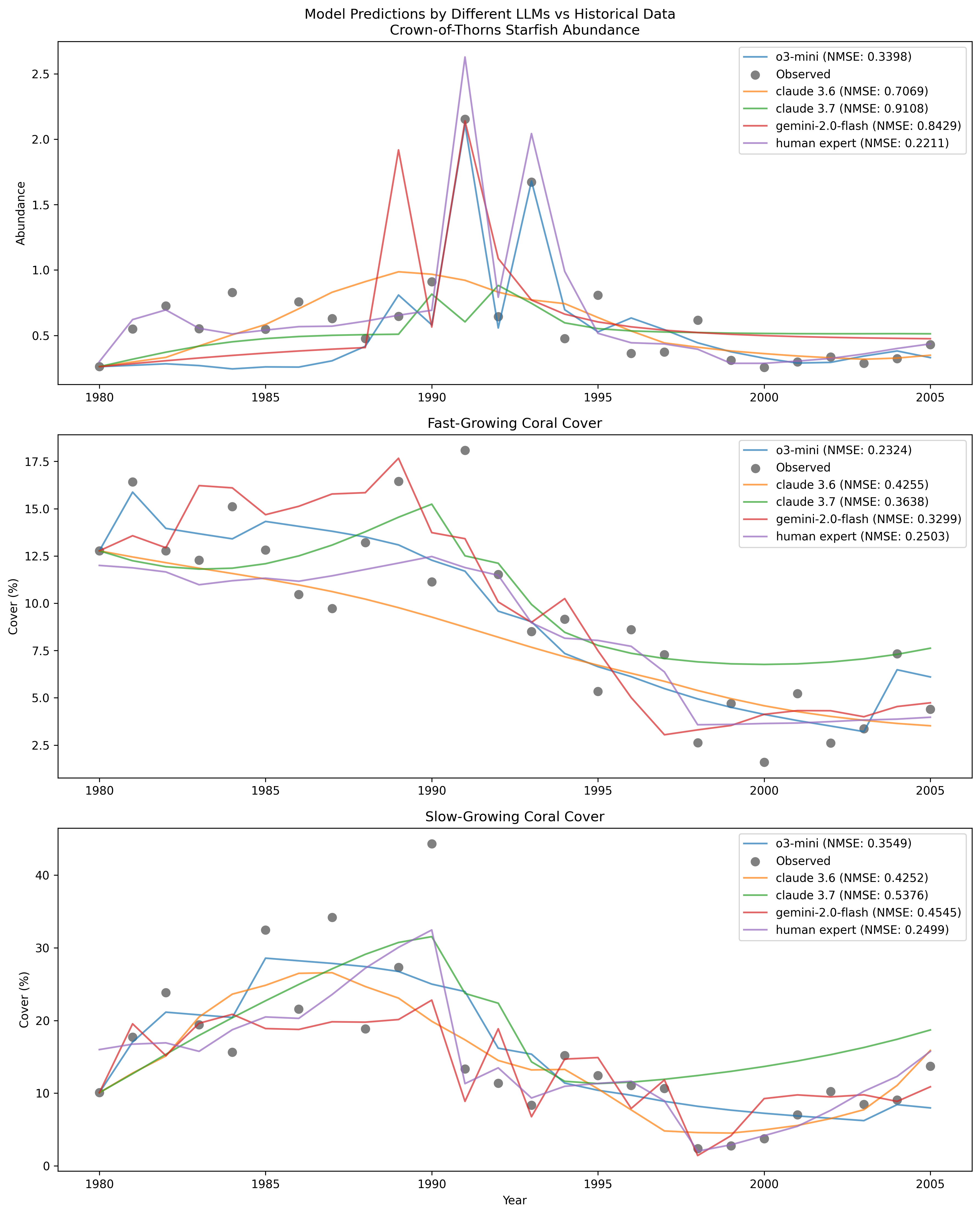
Log-log Relationship between total ecological accuracy score and model performance (objective value). Lower objective values indicate better model fit, while higher ecological scores indicate closer alignment with known NPZ model mechanisms. The data distribution appears to cluster into two groups—models with total ecological scores above 6 and those below 6—suggesting a potential threshold effect where certain key ecological mechanisms significantly improve model performance. This pattern may be driven by specific equation terms that, once correctly implemented, substantially enhance model accuracy.

## Fitting Messy ‘Real-World’ Data – CoTS Case Study

For all but one LLM, AIME was able to generate ecosystem models with prediction accuracy that was comparable to expert-developed models. After ten generations, analysis across LLM configurations revealed systematic performance patterns, with o3-mini achieving the best objective value (0.4213), followed by Claude Sonnet-3.6 (0.4992), Gemini 2.0 Flash (0.5216), and Claude Sonnet-3.7 (0.5808). Surprisingly, despite high scores on common benchmarks, Google’s Gemini-2.5-pro was not able to produce a single numerically stable model after five generations, and thus we terminated its process. Component-specific analysis revealed varying levels of prediction accuracy, with models showing strongest performance in predicting fast-growing coral cover and slow-growing coral cover, while maintaining reasonable accuracy for the more volatile COTS abundance patterns (Figure [4](#fig:llm_comparison)).

Detailed analysis of the best-performing models revealed substantial structural diversity in their ecological mechanisms (see Supplementary Materials for full comparison). The models employed markedly different approaches to temperature effects, with implementations ranging from asymmetric Gaussian responses (o3-mini) to linear thresholds (Claude Sonnet-3.7) and standard Gaussian functions (Claude Sonnet-3.6 ). Similarly, the models varied in their representation of COTS-coral interactions, with approaches including coral-dependent reproduction with density-dependent mortality (o3-mini), resource-dependent reproduction via coral consumption (Claude Sonnet-3.6 ), and logistic growth with food limitation (Claude Sonnet-3.7). These structural differences highlight the framework’s ability to explore diverse ecological mechanisms while maintaining comparable predictive performance.

There were notable differences in the behaviour and speed of convergence for all of the LLMs we tested Figure [9](#sec:convergence). The o3-mini model completed ten generations in as little as 41 minutes. The two Claude LLMs were able to generate well-functioning models in a single generation, but often struggled to improve upon previous generations, whereas o3-mini (the only ‘reasoning’ model we included in this study) was able to consistently improve upon previous generations. Due to this behaviour, we decided to run o3-mini for an additional 60 generations (70 generations total), to see if it would converge at or near the human model. Ultimately, it did reach an objective value of 0.297, and importantly, it was able to capture the outbreak dynamics of the CoTS as requested in the research topic (Figure [4](#fig:llm_comparison)).



Comparison of model predictions across ecosystem components. The plots show observed versus predicted values for COTS abundance, fast-growing coral cover, and slow-growing coral cover, demonstrating the models’ ability to capture key ecological patterns and relationships. Objective values (obj) shown in the legend represent the normalised mean squared error for all three variables, where lower values indicate better model performance. NB: Gemini-2.5-pro is not shown as it failed to produce a numerically stable model.

### Time-Series Prediction Performance

Using Sonnet 3.5 as the base LLM, we tested whether models could be created that are capable of predicting out of sample datapoints. Quantitative evaluation of out-of-sample predictions (2000-2005) revealed varying performance across components, with particularly strong predictive power for slow-growing coral cover (R² = 0.895, RMSE = 2.60, MAE = 1.86).

For fast-growing coral cover, the model achieved moderate predictive accuracy (RMSE = 2.33, MAE = 1.92, R² = 0.182), effectively capturing the general declining trend while showing some deviation in precise values. COTS population predictions demonstrated reasonable accuracy in absolute terms (RMSE = 0.19, MAE = 0.17) despite a lower R² value (0.019).

Figure [5](#fig:validation_combined) illustrates these prediction capabilities, showing both training period performance (pre-2000) and out-of-sample predictions (2000-2005). The validation results are particularly noteworthy given the challenge of simultaneously predicting multiple interacting ecological components across different temporal scales. The model’s ability to maintain consistent error metrics (RMSE and MAE) while capturing both rapid population dynamics and slower coral cover changes suggests it has successfully identified fundamental ecological relationships governing reef ecosystem dynamics.

![](data:application/pdf;base64,)

Time-series validation of the best-performing AI model showing predictions against observed data. The model was trained on pre-2000 data (white background) and validated on unseen 2000-2005 data (pink shaded region). Top: COTS abundance predictions showing capture of population variability. Middle: Fast-growing coral cover predictions demonstrating tracking of decline patterns. Bottom: Slow-growing coral cover predictions illustrating strong capture of recovery dynamics. Orange lines represent model predictions, blue dots show observed data.

# Discussion

Our complementary validation studies demonstrate the viability of AI-driven automation in ecological model development and calibration. The NPZ validation revealed AIME’s ability to recover known ecological relationships from synthetic data, with the best models achieving high ecological accuracy scores (up to 7.75 out of 8) while maintaining strong predictive performance (objective values as low as 0.112). While the framework did not perfectly reconstruct the original equations after 50 generations, it successfully identified key mechanisms like Michaelis-Menten kinetics and predator-prey interactions. The negative correlation between ecological accuracy and objective values suggests that improvements in model fit were achieved through discovery of correct ecological relationships rather than overfitting. This ability to balance model fit with ecological realism represents a significant advance in automated ecological modelling.

The CoTS case study further demonstrated AIME’s practical utility, with the framework successfully generating models that were comparable to human expert models, with the caveat that AI performance was less consistent across all of the variables. The consistent performance across multiple populations suggests that evolutionary algorithms can effectively navigate the challenging balance between model complexity and practical utility in real-world applications.

Notably, the AI-generated models achieved comparable predictive performance to the human expert model despite substantial structural differences (see Supplementary Materials for detailed comparison). While the human model implemented an age-structured COTS population with explicit age classes and a Beverton-Holt stock-recruitment relationship, the AI models generally employed simpler, unstructured population approaches. Similarly, the human model featured an explicit prey-switching function for COTS predation preference between coral types, whereas AI models used various functional responses ranging from coral-dependent reproduction to logistic growth with food limitation. These structural differences highlight an important trade-off: the human model exhibited greater mechanistic detail reflecting domain expertise, while AI models achieved similar performance with more parsimonious formulations. This is consistent with other work showing that multiple numerically valid representations of the same ecological system can coexist , with each offering different insights into underlying mechanisms. This method provides a valuable opportunity for researchers to explore diverse ecological mechanisms while maintaining comparable predictive performance. The ability to generate multiple, distinct models for a given system opens up exciting avenues for future research, allowing for the exploration of different ecological hypotheses and the potential for ensemble modelling approaches.

## Contrasting Approaches to AI in Ecological Modelling

Recent advances in AI have demonstrated remarkable capabilities in ecological time-series prediction. Studies using transformer architectures and diffusion models, including multimodal approaches like LITE , have shown high accuracy in direct forecasting of environmental variables . While these methods effectively handle challenges like missing data and distribution shifts, they treat the system as a black box, learning patterns directly from time-series data without explicitly modelling underlying mechanisms. While our study did not directly compare our approach with black-box methods, our NPZ validation study suggests a key advantage of our approach: the ability to provide insights into fundamental ecological processes like nutrient cycling or predator-prey dynamics through explicit model discovery. This interpretability is a theoretical advantage over black-box approaches, though comparative studies would be needed to fully evaluate the relative strengths of each approach in specific ecological contexts.

Our evolutionary approach fundamentally differs by using AI to generate actual ecological models rather than make direct predictions. Instead of training neural networks to forecast future values, AIME evolves interpretable models with meaningful parameters that capture real biological and physical processes. This distinction is crucial for several reasons. First, our generated models provide scientific insight into system behavior, revealing mechanisms and relationships that direct prediction approaches typically cannot without careful oversight . Second, the models maintain biological plausibility through explicit parameter constraints and mechanistic formulations, ensuring their utility for management applications. Third, because they capture fundamental processes rather than just patterns, these models can potentially be transferred to new scenarios and used to explore management interventions. The relationship between our approach and direct prediction methods is nuanced. Recent time-series prediction approaches using transformer architectures have achieved impressive accuracy, with mean squared errors as low as 0.001-0.04 for normalized predictions and root mean squared errors reduced by up to 52% compared to traditional methods . While our evolved models may not always match these pure prediction accuracies, they offer advantages in interpretability and scientific insight.

Importantly, these approaches need not be viewed as mutually exclusive. Comparing mechanistic models with black-box predictions can be particularly insightful, especially when the two approaches diverge. For instance, under novel conditions like future climate scenarios, differences in predictions could highlight processes that are not well-captured by mechanistic models or reveal patterns that black-box approaches detect but cannot explain. When predictive approaches outperform mechanistic models, this divergence can guide researchers toward missing processes or relationships that should be incorporated into mechanistic understanding. Our framework demonstrates that it’s possible to achieve both reasonable predictive accuracy and meaningful ecological interpretability, with each approach offering complementary strengths.

This focus on model generation rather than direct prediction aligns with the needs of ecosystem-based management, where understanding system dynamics is as important as predictive accuracy. The interpretability of our evolved models enables managers to assess the credibility of predictions and understand the mechanisms driving system behavior thereby allowing for informed management interventions, advantages not readily available with black-box prediction approaches. Work examining automated scientific discovery emphasizes the importance of maintaining human oversight while leveraging AI’s computational capabilities . Our approach directly addresses this need by producing interpretable models that facilitate meaningful human oversight while leveraging AI’s capabilities for systematic exploration of model space.

## Implications for Ecosystem-Based Fisheries Management

The successful application of AIME to Crown-of-Thorns starfish populations on the Great Barrier Reef demonstrates its potential for informing ecosystem-based management decisions. The framework’s capacity to capture both short-term outbreak dynamics and longer-term ecosystem changes provides managers with valuable insights for intervention planning. The comparable performance between AI-generated models and human expert approaches suggests that automated modelling could complement traditional methods, accelerating the development and evaluation of management strategies.

The success of our modelling framework in capturing complex ecosystem dynamics while maintaining interpretability suggests it could be valuable for understanding CoTS outbreaks in other regions. Recent studies have shown that ecosystem shifts involving CoTS can have long-lasting effects that are difficult to reverse , making it crucial to detect and respond to outbreaks early. Notably, the speed of implementation achieved through our framework significantly outpaces what a human modeller can manage working independently, with model generation and refinement occurring in hours rather than weeks or months. This enhanced efficiency creates capacity for more timely intervention in response to emerging ecological threats, a critical advantage when management actions depend on rapid model development and deployment. Our model’s ability to integrate multiple data sources, both locally and from web search, and account for both biological and environmental factors provides a robust foundation for developing early warning systems and evaluating potential management interventions.

## Limitations and Future Directions

Despite promising results, several limitations warrant consideration. The observed variation in convergence rates across populations suggests that initial conditions significantly influence model evolution trajectories. While the best performing population achieved rapid convergence within five generations, other populations required more than twice as many generations to approach similar performance levels.

An important limitation in our current implementation is the treatment of all model parameters as estimable quantities in the optimization process, even when well-established values exist in the literature. While our RAG system successfully retrieves some literature-based values and ranges for parameters, these are only used as initial estimates and bounds rather than as fixed quantities. This approach may lead to unnecessary parameter estimation and potential deviation from biologically meaningful values. Future versions of the framework should distinguish between parameters that truly need estimation and those that could be fixed based on reliable literature values. This would not only reduce the parameter space for optimization but also better incorporate established ecological knowledge into the modelling process and make the process less resource-intensive when doing calculations.

There are numerous future avenues for validating, improving, and extending this framework. First, there are several hyper-parameters that likely control the success and speed of convergence of the framework (LLM-choice, temperature, number of individuals per generation, prompt construction, etc.). Systematic testing across these choices may reveal optimal configurations for convergence. In particular, the comparative analysis of different AI configurations (as detailed in Section [3.2](#sec:cots_data) and Figure [7](#fig:status_distribution)) reveals trade-offs between model choice and rate of improvement. While the o3-mini configuration was able to consistently iteratively improve, the Sonnet models were able to perform well in a single generation but then did not consistently improve. Future work could explore hybrid approaches that leverage the strengths of different AI configurations at various stages of model development, or that employ different LLMs consecutively over multiple generations. Further, ongoing testing of new LLMs as they are released may yield considerable gains in efficiency and cost-saving. Second, we have tested a relatively simple ecosystem model with three dependent variable time-series and two forcing variable time-series. Simple systems like these will be limited in real-world utility, and therefore testing on more complex systems with tens or hundreds of time-series will be needed. Incorporating spatial components may also be possible and will greatly improve the utility of this framework. Third, accessing relevant scientific information for the parameter RAG search is limited by the user’s ability to either curate a local database of relevant materials, or access scientific papers online. Fourth, we have demonstrated that it is possible for this LLM-based system to generate multiple, distinct models for a given system. Choosing between similarly performing, but ecologically distinct models may be necessary for experts with ecological knowledge, or perhaps employing approaches that ensemble multiple plausible models may allow for the reduction in uncertainty

In conclusion, AIME represents a significant advancement in ecological modelling that bridges the gap between computational efficiency and ecological insight. By dramatically accelerating model development while maintaining scientific rigour, this framework offers a powerful new tool for researchers and managers facing urgent ecological challenges. As environmental pressures intensify globally, the capacity to rapidly develop, test, and deploy ecologically sound models will become increasingly valuable for effective conservation and management of marine ecosystems.

# Supplementary Information: An AI-Driven Framework for Automated Generation of Marine Ecosystem Models

# Curated Literature Collection

The local document collection used in this case study was carefully curated to provide comprehensive coverage of marine ecosystem modeling approaches, with particular focus on COTS-coral dynamics and management interventions. The collection encompasses several key research areas:

* Ecosystem Modeling Frameworks: established foundational principles for ecosystem approaches to fisheries, while introduced Models of Intermediate Complexity for Ecosystem assessments (MICE). explored optimal model complexity levels.
* COTS Management and Ecology: provided a comprehensive thirty-year review of COTS research. developed models for COTS outbreak management, while analyzed corallivore culling impacts under bleaching scenarios.
* Ecological Regime Shifts: investigated predator-driven regime shifts in marine ecosystems. provided insights into ecological tipping points through ecosystem modeling.
* Management Interventions: examined large-scale interventions on the Great Barrier Reef. explored harvest control implications using MICE models.
* Model Application Guidelines: provided critical guidelines for adapting ecosystem models to new applications. demonstrated multispecies production model applications for analyzing ecological and fishing effects.
* Integrated Systems: and explored integrated multi-trophic aquaculture modeling, providing insights into coupled biological systems. analyzed trade-offs in seaweed farming between food production, livelihoods, marine biodiversity, and carbon sequestration benefits.

These papers were selected based on their direct relevance to COTS population dynamics, coral reef ecology, and ecosystem modeling approaches. The collection provided both specific parameter values and broader ecological context for model development.

# RAG Architecture Implementation

The Retrieval-Augmented Generation (RAG) system facilitates parameter search and extraction from scientific literature. The system employs two primary search strategies: a local search of user-curated documents and a comprehensive web search. For local search, the system uses ChromaDB as a persistent vector store to maintain an indexed collection of scientific papers and technical documents specifically curated by research teams for their ecological systems. These documents are processed into semantic chunks of approximately 512 tokens with small overlaps to preserve context while enabling precise retrieval of relevant information.

The parameter search process begins with the generation of enhanced semantic descriptions for each parameter. These descriptions are crafted to improve search relevance by capturing the ecological and mathematical context in which the parameters are used. The system first searches the user-curated local documents using embeddings generated through Azure OpenAI’s embedding service. When necessary, it extends to web-based sources through two channels: querying the Semantic Scholar database for highly-cited papers in biology, mathematics, and environmental science, and conducting broader literature searches through the Serper API to capture additional relevant sources.

The search results from both local and web sources are processed through an LLM to extract numerical values. The system applies consistent validation across both search pathways, identifying minimum and maximum bounds, ensuring unit consistency, and validating source reliability. When direct parameter values are not found in either the local collection or web sources, the system defaults to the initial estimates from the coding LLM. All extracted information, including parameter values, valid ranges, and complete citation details, is stored in a structured JSON database for reproducibility and future reference.

The RAG system implements automatic retry mechanisms when initial searches fail to yield usable results. Each retry attempt follows a structured progression: first accessing the curated local collection through ChromaDB queries, then expanding to Semantic Scholar for peer-reviewed literature, and finally utilizing Serper API for broader scientific content. This progressive broadening of scope, while maintaining focus on ecologically relevant sources, ensures robust parameter estimation even in cases where direct measurements are sparse in the literature.

# AI Prompts Used in Model Development

The development of the model relied on several carefully crafted prompts to guide the artificial intelligence system. These prompts were designed to ensure numerical stability, proper likelihood calculation, and clear model structure. The following sections detail the exact prompts used at each stage of model development.

## Initial Model Creation

The initial model creation utilized a comprehensive prompt that emphasized three key aspects of model development. The prompt used for model initialization was:

Please create a Template Model Builder model for the following topic:[PROJECT\_TOPIC]. Start by writing intention.txt, in which you provide a concise summary of the ecological functioning of the model. In model.cpp, write your TMB model with the following important considerations:  
  
1. NUMERICAL STABILITY:  
- Always use small constants (e.g., Type(1e-8)) to prevent division by zero  
- Use smooth transitions instead of hard cutoffs in equations  
- Bound parameters within biologically meaningful ranges using smooth penalties rather than hard constraints  
  
2. LIKELIHOOD CALCULATION:  
- Always include observations in the likelihood calculation, don't skip any based on conditions  
- Use fixed minimum standard deviations to prevent numerical issues when data values are small  
- Consider log-transforming data if it spans multiple orders of magnitude  
- Use appropriate error distributions (e.g., lognormal for strictly positive data)  
  
3. MODEL STRUCTURE:  
- Include comments after each line explaining the parameters (including their units and how to determine their values)  
- Provide a numbered list of descriptions for the equations  
- Ensure all important variables are included in the reporting section  
- Use `\_pred' suffix for model predictions corresponding to `\_dat' observations

## Parameter Enhancement

To enhance parameter descriptions for improved semantic search capabilities, the following prompt was employed:

Given a mathematical model about [PROJECT\_TOPIC], enhance the semantic descriptions of these parameters to be more detailed and searchable. The model code shows these parameters are used in the following way:  
  
[MODEL\_CONTENT]  
  
For each parameter below, create an enhanced semantic search, no longer than 10 words, that can be used for RAG search or semantic scholar search.

## Model Improvement

For iterative model improvements, the system utilized this prompt:

Improve the fit of the following ecological model by modifying the equations in this TMB script. Only make ONE discrete change most likely to improve the fit. Do not add stochasticity, but you may add other ecological relevant factors that may not be present here already.  
  
You may add additional parameters if necessary, and if so, add them to parameters.json. Please concisely describe your ecological improvement in intention.txt and then provide the improved model.cpp and parameters.json content.

## Error Handling Prompts

For compilation errors, the system used this prompt:

model.cpp failed to compile. Here's the error information:  
  
[ERROR\_INFO]  
  
Do not suggest how to compile the script

For data leakage issues, the system employed this detailed prompt:

Data leakage detected in model equations. The following response variables cannot be used to predict themselves:  
  
To fix this:  
1. Response variables ([RESPONSE\_VARS]) must be predicted using only:  
 - External forcing variables ([FORCING\_VARS])  
 - Other response variables' predictions (\_pred variables)  
 - Parameters and constants  
2. Each response variable must have a corresponding prediction equation  
3. Use ecological relationships to determine how variables affect each other  
  
For example, instead of:  
 slow\_pred(i) = slow \* growth\_rate;  
Use:  
 slow\_pred(i) = slow\_pred(i-1) \* growth\_rate \* (1 - impact\_rate \* cots\_pred(i-1));  
  
Please revise the model equations to avoid using response variables to predict themselves.

For numerical instabilities, the system used an adaptive prompt that became progressively more focused on simplification after multiple attempts:

The model compiled but numerical instabilities occurred. Here's the error information:  
  
[ERROR\_INFO]  
  
[After 2+ attempts: Consider making a much simpler model that we can iteratively improve later.]  
Do not suggest how to compile the script

## NPZ Case Study - Recovering Equations

The model implementation can be compared to the original NPZ equations from :

Our generated model captures several key ecological processes from the original system:

1. Nutrient uptake by phytoplankton following Michaelis-Menten kinetics
2. Quadratic zooplankton mortality
3. Nutrient recycling through zooplankton excretion
4. Environmental mixing effects

For evaluating the ecological characteristics of generated models against the NPZ reference model, the system used this prompt. The prompt used for all evaluations was:

Compare this C++ model against the following criteria that should be present in the NPZ model equation by equation.  
The mathematical structure should be identical, even if variable names differ.  
  
For each equation (dN/dt, dP/dt, dZ/dt), check these components:  
- nutrient\_equation\_uptake: In dN/dt: Nutrient uptake by phytoplankton with Michaelis-Menten kinetics (N/(e+N)) and self-shading (a/(b+c\*P))  
- nutrient\_equation\_recycling: In dN/dt: Nutrient recycling from zooplankton via predation (beta\*lambda\*P^2/(mu^2+P^2)\*Z) and excretion (gamma\*q\*Z)  
- nutrient\_equation\_mixing: In dN/dt: Environmental mixing term (k\*(N0-N))  
- phytoplankton\_equation\_growth: In dP/dt: Phytoplankton growth through nutrient uptake (N/(e+N))\*(a/(b+c\*P))\*P  
- phytoplankton\_equation\_loss: In dP/dt: Phytoplankton losses through mortality (r\*P), predation (lambda\*P^2/(mu^2+P^2)\*Z), and mixing ((s+k)\*P)  
- zooplankton\_equation: In dZ/dt: Zooplankton growth through predation (alpha\*lambda\*P^2/(mu^2+P^2)\*Z) and mortality (q\*Z)  
  
For each characteristic:  
1. Score 1.0 if the mathematical structure is equivalent, regardless of:  
 - Variable names (e.g., 'N' vs 'nutrients' vs 'N[0]')  
 - Programming syntax (e.g., 'pow(x,2)' vs 'x^2')  
 - Code organization  
2. Verify the mathematical operations are in the same order  
3. Confirm all coefficients are present (even if named differently)  
  
Provide your response in JSON format:  
{  
 "qualitative\_description": "Overall description of how well the model matches the criteria",  
 "characteristic\_scores": {  
 "characteristic\_name": {  
 "score": 0.0 to 1.0,  
 "explanation": "How it is implemented"  
 }  
 }  
}

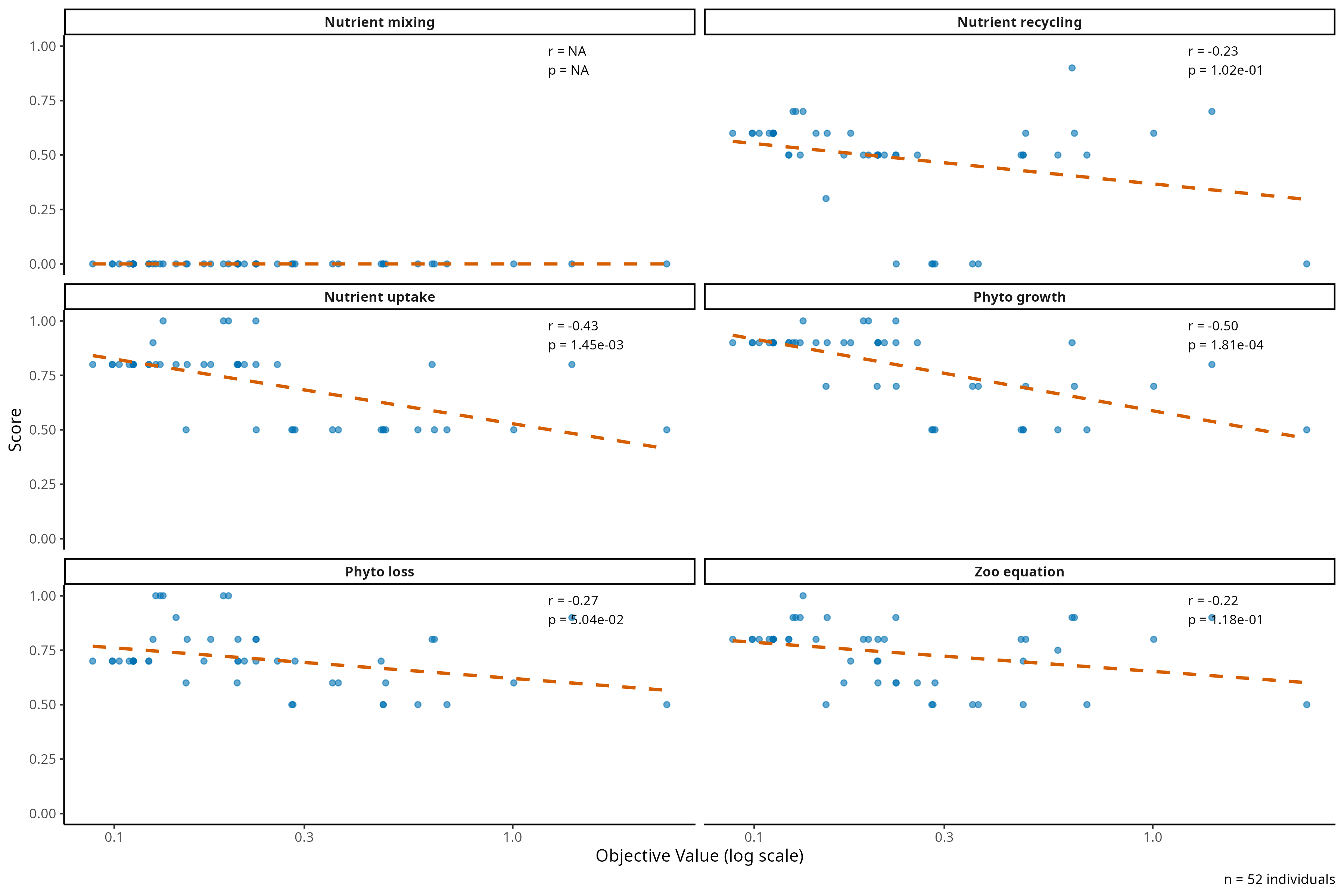
To validate the scoring approach, we first tested the evaluation system on a model containing the exact equations from the original NPZ model. The system correctly identified and scored all characteristics with perfect accuracy (score of 1.0 for each component), as shown in this test evaluation output:

{  
 "qualitative\_description": "A nutrient-phytoplankton-zooplankton (NPZ) model with detailed interactions representing nutrient uptake, predation, mortality, and environmental mixing",  
 "characteristic\_scores": {  
 "nutrient\_equation\_uptake": {  
 "score": 1.0,  
 "explanation": "Exact Michaelis-Menten kinetics with N/(e+N) and self-shading term a/(b+c\*P), implemented identically"  
 },  
 "nutrient\_equation\_recycling": {  
 "score": 1.0,  
 "explanation": "Includes both predation-based recycling (beta\*lambda\*P^2/(mu^2+P^2)\*Z) and zooplankton excretion (gamma\*q\*Z)"  
 },  
 "nutrient\_equation\_mixing": {  
 "score": 1.0,  
 "explanation": "Environmental mixing term k\*(N0-N) present with correct implementation"  
 },  
 "phytoplankton\_equation\_growth": {  
 "score": 1.0,  
 "explanation": "Growth term matches exactly: (N/(e+N))\*(a/(b+c\*P))\*P"  
 },  
 "phytoplankton\_equation\_loss": {  
 "score": 1.0,  
 "explanation": "Includes mortality (r\*P), predation (lambda\*P^2/(mu^2+P^2)\*Z), and mixing ((s+k)\*P)"  
 },  
 "zooplankton\_equation": {  
 "score": 1.0,  
 "explanation": "Zooplankton growth through predation (alpha\*lambda\*P^2/(mu^2+P^2)\*Z) and mortality (q\*Z)"  
 }  
 }  
}

This validation test confirmed that the evaluation system could correctly identify and score ecological characteristics when present.

# NPZ Validation

The NPZ validation study evaluated AIME’s ability to recover known ecological relationships from synthetic data. Figure [6](#fig:ecological_characteristics) shows the relationship between model performance (objective value) and ecological accuracy scores for each characteristic of the NPZ model. The negative correlations across multiple characteristics suggest that improvements in model fit were achieved through discovery of correct ecological mechanisms rather than overfitting.



Relationship between ecological accuracy scores and model performance for each NPZ model characteristic. Each panel shows how well models recovered a specific ecological mechanism (score from 0-1) versus their predictive accuracy (objective value). Lower objective values indicate better model fit. Two-sided Pearson’s product-moment correlation coefficients (r) and their associated p-values are shown for each characteristic.

## Best Performing NPZ Model

This model achieved an objective value of 0.0883.

### Model Description

The following model represents our framework’s attempt to recover the NPZ dynamics from . The model aims to capture three key components:

* Nutrient uptake and recycling
* Phytoplankton growth and mortality
* Zooplankton predation and dynamics

### Model Intention

\section{Ecological Intention}  
  
A key modification was made to incorporate direct nutrient recycling from zooplankton grazing activity. In marine systems, zooplankton feeding is often inefficient, with a significant portion of consumed phytoplankton being released as dissolved nutrients rather than being assimilated into biomass or entering the detritus pool. This "sloppy feeding" process creates an important feedback loop where grazing can stimulate new primary production through rapid nutrient recycling.  
  
The recycling efficiency is temperature-dependent, reflecting how metabolic rates and feeding mechanics vary with temperature. This creates an adaptive feedback where warmer conditions lead to both increased grazing pressure and faster nutrient recycling, better capturing the coupled nature of predator-prey interactions in planktonic systems.  
  
The modification introduces a direct pathway from grazing to dissolved nutrients, complementing the slower recycling through the detritus pool. This better represents the multiple timescales of nutrient cycling in marine food webs and helps explain how high productivity can be maintained even under intense grazing pressure.

### Model Implementation

#include <TMB.hpp>  
template<class Type>  
Type objective\_function<Type>::operator() ()  
{  
 // Data  
 DATA\_VECTOR(Time); // Time points (days)  
 DATA\_VECTOR(N\_dat); // Nutrient observations (g C m^-3)  
 DATA\_VECTOR(P\_dat); // Phytoplankton observations (g C m^-3)  
 DATA\_VECTOR(Z\_dat); // Zooplankton observations (g C m^-3)  
   
 // Create default temperature vector if not provided  
 vector<Type> Temp(Time.size());  
 Temp.fill(Type(20.0)); // Default temperature of 20°C  
   
 // Parameters  
 PARAMETER(r\_max); // Maximum phytoplankton growth rate (day^-1)  
 PARAMETER(K\_N); // Half-saturation constant for nutrient uptake (g C m^-3)  
 PARAMETER(g\_max); // Maximum zooplankton grazing rate (day^-1)  
 PARAMETER(K\_P); // Half-saturation constant for grazing (g C m^-3)  
 PARAMETER(alpha\_base); // Baseline zooplankton assimilation efficiency  
 PARAMETER(alpha\_max); // Maximum additional assimilation efficiency  
 PARAMETER(K\_alpha); // Half-saturation for nutrient-dependent efficiency  
 PARAMETER(m\_P); // Base phytoplankton mortality rate (day^-1)  
 PARAMETER(m\_P\_N); // Nutrient-dependent phytoplankton mortality (day^-1)  
 PARAMETER(s\_P); // Base phytoplankton sinking rate (day^-1)  
 PARAMETER(s\_P\_max); // Maximum additional nutrient-stress sinking rate (day^-1)  
 PARAMETER(m\_Z); // Base zooplankton mortality rate (day^-1)  
 PARAMETER(m\_Z\_N); // Nutrient-dependent zooplankton mortality (day^-1)  
 PARAMETER(r\_D); // Detritus remineralization rate (day^-1)  
 PARAMETER(sigma\_N); // SD for nutrient observations  
 PARAMETER(sigma\_P); // SD for phytoplankton observations  
 PARAMETER(sigma\_Z); // SD for zooplankton observations  
 PARAMETER(I\_opt); // Optimal light intensity  
 PARAMETER(beta); // Light attenuation coefficient  
 PARAMETER(k\_w); // Light attenuation coefficient due to phytoplankton self-shading  
 PARAMETER(E\_p); // Activation energy for photosynthetic efficiency (eV)  
 PARAMETER(theta\_P); // Temperature sensitivity of grazing selectivity  
 PARAMETER(eta\_max); // Maximum nutrient uptake efficiency multiplier  
 PARAMETER(k\_eta); // Steepness of uptake efficiency response  
 PARAMETER(N\_crit); // Critical nutrient concentration for efficiency switch  
 PARAMETER(eta\_base); // Baseline nutrient uptake efficiency  
   
  
 // Constants for numerical stability  
 const Type eps = Type(1e-8);  
 const Type min\_conc = Type(1e-10); // Minimum concentration  
 const Type max\_dt = Type(0.1); // Maximum time step  
   
 // Initialize negative log-likelihood  
 Type nll = 0.0;  
   
 // Smooth penalties to keep parameters in biological ranges  
 nll -= dnorm(log(r\_max), Type(0.0), Type(1.0), true); // Keep r\_max positive  
 nll -= dnorm(log(K\_N), Type(-3.0), Type(1.0), true); // Keep K\_N positive  
 nll -= dnorm(log(g\_max), Type(-1.0), Type(1.0), true); // Keep g\_max positive  
 nll -= dnorm(log(K\_P), Type(-3.0), Type(1.0), true); // Keep K\_P positive  
 nll -= dnorm(logit(alpha\_base), Type(0.0), Type(2.0), true); // Keep alpha\_base between 0 and 1  
 nll -= dnorm(logit(alpha\_max), Type(0.0), Type(2.0), true); // Keep alpha\_max between 0 and 1  
 nll -= dnorm(log(K\_alpha), Type(-3.0), Type(1.0), true); // Keep K\_alpha positive  
 nll -= dnorm(log(m\_P), Type(-3.0), Type(1.0), true); // Keep m\_P positive  
 nll -= dnorm(log(m\_Z), Type(-3.0), Type(1.0), true); // Keep m\_Z positive  
 nll -= dnorm(log(r\_D), Type(-3.0), Type(1.0), true); // Keep r\_D positive  
   
 // Vectors to store predictions  
 vector<Type> N\_pred(Time.size());  
 vector<Type> P\_pred(Time.size());  
 vector<Type> Z\_pred(Time.size());  
 vector<Type> D\_pred(Time.size());  
   
 // Initial conditions (ensure positive)  
 N\_pred(0) = exp(log(N\_dat(0) + eps));  
 D\_pred(0) = Type(0.1); // Initial detritus concentration  
 P\_pred(0) = exp(log(P\_dat(0) + eps));  
 Z\_pred(0) = exp(log(Z\_dat(0) + eps));  
   
 // Numerical integration using 4th order Runge-Kutta  
 for(int t = 1; t < Time.size(); t++) {  
 Type dt = Time(t) - Time(t-1);  
   
 // Use fixed small time steps for stability  
 Type h = Type(0.1); // Fixed step size  
 int n\_steps = 10; // Fixed number of steps  
   
 Type N = N\_pred(t-1);  
 Type P = P\_pred(t-1);  
 Type Z = Z\_pred(t-1);  
 Type D = D\_pred(t-1);  
   
 for(int step = 0; step < n\_steps; step++) {  
 // Temperature scaling (Arrhenius equation)  
 Type T\_K = Temp(t) + Type(273.15); // Convert to Kelvin  
 Type T\_ref = Type(293.15); // Reference temp (20°C)  
 Type E\_a = Type(0.63); // Activation energy (eV)  
 Type k\_B = Type(8.617e-5); // Boltzmann constant (eV/K)  
   
 // Temperature scaling factor (simplified)  
 // General metabolic temperature scaling  
 Type temp\_scale = exp(E\_a \* (Type(1.0)/T\_ref - Type(1.0)/T\_K) / k\_B);  
 // Photosynthesis-specific temperature scaling  
 Type photo\_eff = exp(E\_p \* (Type(1.0)/T\_ref - Type(1.0)/T\_K) / k\_B);  
 // Bound scaling factors to prevent numerical issues  
 temp\_scale = Type(0.5) + Type(0.5) \* temp\_scale;  
 photo\_eff = Type(0.5) + Type(0.5) \* photo\_eff;  
   
 // Calculate seasonal light intensity   
 Type season = Type(0.6) \* sin(Type(2.0) \* M\_PI \* Time(t) / Type(365.0));  
 Type I = I\_opt \* (Type(1.0) + season);  
   
 // Light limitation factor with self-shading  
 Type I\_effective = I \* exp(-k\_w \* P); // Reduce light based on phytoplankton density  
 Type light\_limitation = (I\_effective/I\_opt) \* exp(Type(1.0) - I\_effective/I\_opt);  
   
 // Temperature-dependent grazing selectivity  
 Type K\_P\_T = K\_P \* (Type(1.0) + theta\_P \* (temp\_scale - Type(1.0)));  
   
 // Calculate nutrient-dependent uptake efficiency with baseline  
 Type eta\_N = eta\_base + (eta\_max - eta\_base) / (Type(1.0) + exp(-k\_eta \* (N - N\_crit)));  
   
 // Calculate temperature and light dependent rates  
 Type uptake = r\_max \* temp\_scale \* photo\_eff \* light\_limitation \* eta\_N \* N \* P / (K\_N + N + eps);  
   
 Type grazing = g\_max \* temp\_scale \* P \* Z / (K\_P\_T + P + eps);  
   
 // Detritus remineralization (temperature dependent)  
 Type remin = r\_D \* temp\_scale \* D\_pred(t-1);  
   
 // System of differential equations  
 // Calculate nutrient-dependent assimilation efficiency first  
 Type alpha\_N = alpha\_base + alpha\_max \* (N / (N + K\_alpha + eps));  
   
 // Calculate temperature-dependent nutrient recycling efficiency from grazing  
 Type recycling\_eff = Type(0.3) \* temp\_scale; // Base 30% recycling, modified by temperature  
 Type grazing\_recycle = recycling\_eff \* (1 - alpha\_N) \* grazing;  
   
 Type dN = -uptake + remin + grazing\_recycle;  
   
 // Enhanced mortality and sinking under nutrient stress  
 Type nutrient\_stress = m\_P\_N \* K\_N / (N + K\_N + eps);  
 Type sinking = (s\_P + s\_P\_max \* K\_N / (N + K\_N + eps)) \* P;  
 Type dP = uptake - grazing - (m\_P + nutrient\_stress) \* P - sinking;  
 // Enhanced zooplankton mortality under nutrient stress  
 Type Z\_nutrient\_stress = m\_Z\_N \* K\_N / (N + K\_N + eps);  
 Type dZ = alpha\_N \* grazing - (m\_Z \* Z + Z\_nutrient\_stress) \* Z;  
 Type dD = m\_P \* P + m\_Z \* Z \* Z + (1 - alpha\_N) \* grazing - remin;  
   
 // Euler integration step  
 N += h \* dN;  
 P += h \* dP;  
 Z += h \* dZ;  
   
 // Ensure concentrations stay positive  
 N = exp(log(N + eps));  
 P = exp(log(P + eps));  
 Z = exp(log(Z + eps));  
 D += h \* dD;  
 D = exp(log(D + eps));  
 }  
   
 // Store final values  
 N\_pred(t) = N;  
 P\_pred(t) = P;  
 Z\_pred(t) = Z;  
 D\_pred(t) = D;  
 }  
   
 // Likelihood calculations using lognormal distribution  
 Type min\_sigma = Type(0.01); // Minimum standard deviation  
 for(int t = 0; t < Time.size(); t++) {  
 nll -= dnorm(log(N\_dat(t) + eps), log(N\_pred(t) + eps),   
 exp(log(sigma\_N + min\_sigma)), true);  
 nll -= dnorm(log(P\_dat(t) + eps), log(P\_pred(t) + eps),   
 exp(log(sigma\_P + min\_sigma)), true);  
 nll -= dnorm(log(Z\_dat(t) + eps), log(Z\_pred(t) + eps),   
 exp(log(sigma\_Z + min\_sigma)), true);  
 }  
   
 // Report predictions  
 REPORT(N\_pred);  
 REPORT(P\_pred);  
 REPORT(Z\_pred);  
 REPORT(D\_pred);  
  
   
 return nll;  
}

### Model Parameters

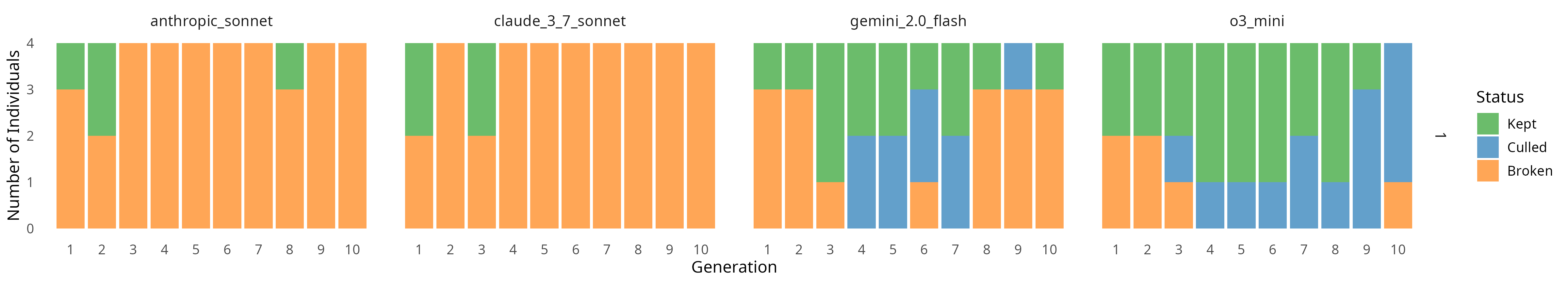
{  
 "parameters": [  
 {  
 "parameter": "r\_max",  
 "value": 1.0,  
 "description": "Maximum phytoplankton growth rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Maximum photosynthetic carbon fixation rate in marine ecosystems",  
 "citations": [  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC1913777/",  
 "https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020JG005719",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0967064506001263"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "K\_N",  
 "value": 0.1,  
 "description": "Half-saturation constant for nutrient uptake (g C m^-3)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Nutrient limitation threshold for phytoplankton growth dynamics",  
 "citations": [  
 "https://www.nature.com/articles/s41467-023-40774-0",  
 "https://www.sciencedirect.com/science/article/pii/S0043135420309428",  
 "https://www.sciencedirect.com/science/article/pii/S1568988325000125"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "g\_max",  
 "value": 0.4,  
 "description": "Maximum zooplankton grazing rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Peak predation rate of zooplankton on phytoplankton populations",  
 "citations": [  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC3031578/",  
 "https://academic.oup.com/plankt/article/22/6/1085/1587539",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0025556413001466"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "K\_P",  
 "value": 0.2,  
 "description": "Half-saturation constant for grazing (g C m^-3)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Prey density threshold controlling zooplankton consumption rates",  
 "citations": [  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC9124482/",  
 "https://academic.oup.com/icesjms/article/71/2/254/781831",  
 "https://aslopubs.onlinelibrary.wiley.com/doi/10.1002/lno.10632"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "alpha\_base",  
 "value": 0.2,  
 "description": "Baseline zooplankton assimilation efficiency (dimensionless)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Minimum efficiency of energy transfer from prey to zooplankton consumers under nutrient-poor conditions",  
 "citations": [  
 "https://link.springer.com/article/10.1007/s10750-017-3298-9",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0022098198000736"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "alpha\_max",  
 "value": 0.3,  
 "description": "Maximum additional zooplankton assimilation efficiency (dimensionless)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum additional assimilation efficiency possible under optimal nutrient conditions",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/abs/pii/S0022098198000736",  
 "https://www.int-res.com/articles/meps/139/m139p267.pdf"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "K\_alpha",  
 "value": 0.1,  
 "description": "Half-saturation constant for nutrient-dependent efficiency (g C m^-3)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Nutrient concentration at which additional assimilation efficiency reaches half maximum",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/abs/pii/S0022098198000736"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "m\_P",  
 "value": 0.1,  
 "description": "Phytoplankton mortality rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Natural death and senescence rate of phytoplankton communities",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/abs/pii/S0079661123001854",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0146638002001067",  
 "https://www.tandfonline.com/doi/full/10.1080/09670262.2018.1563216"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "m\_Z",  
 "value": 0.05,  
 "description": "Base zooplankton mortality rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Baseline natural mortality rate of zooplankton populations",  
 "citations": [  
 "https://academic.oup.com/icesjms/article/81/6/1164/7697287",  
 "https://www.nature.com/articles/s41558-023-01630-7",  
 "https://www.sciencedirect.com/science/article/pii/S0048969723041281"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "m\_Z\_N",  
 "value": 0.1,  
 "description": "Nutrient-dependent zooplankton mortality rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Additional zooplankton mortality rate under nutrient-poor conditions",  
 "citations": [  
 "https://doi.org/10.4319/lo.2009.54.4.1025",  
 "https://doi.org/10.1016/j.seares.2016.07.002"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "r\_D",  
 "value": 0.1,  
 "description": "Detritus remineralization rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Rate at which detrital organic matter is converted back to bioavailable nutrients through bacterial decomposition",  
 "citations": [  
 "https://doi.org/10.4319/lo.1992.37.6.1307",  
 "https://doi.org/10.1016/j.marchem.2007.01.006",  
 "https://doi.org/10.1016/j.dsr2.2008.04.029"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "sigma\_N",  
 "value": 0.2,  
 "description": "Standard deviation for nutrient observations (log scale)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 4,  
 "enhanced\_semantic\_description": "Measurement uncertainty in marine nutrient concentration observations",  
 "processed": true  
 },  
 {  
 "parameter": "sigma\_P",  
 "value": 0.2,  
 "description": "Standard deviation for phytoplankton observations (log scale)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 4,  
 "enhanced\_semantic\_description": "Observational variability in phytoplankton biomass measurements",  
 "processed": true  
 },  
 {  
 "parameter": "sigma\_Z",  
 "value": 0.2,  
 "description": "Standard deviation for zooplankton observations (log scale)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 4,  
 "enhanced\_semantic\_description": "Statistical dispersion of zooplankton population density estimates",  
 "processed": true  
 },  
 {  
 "parameter": "E\_a",  
 "value": 0.63,  
 "description": "Activation energy for metabolic scaling (eV)",  
 "source": "literature",  
 "import\_type": "CONSTANT",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Activation energy controlling temperature dependence of biological rates based on metabolic theory",  
 "citations": [  
 "https://doi.org/10.1111/ele.12308",  
 "https://doi.org/10.1126/science.1114383",  
 "https://doi.org/10.1038/nature04095"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "I\_opt",  
 "value": 150.0,  
 "description": "Optimal light intensity for photosynthesis (W/m^2)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Optimal irradiance level for phytoplankton photosynthetic efficiency",  
 "citations": [  
 "https://doi.org/10.4319/lo.1997.42.7.1552",  
 "https://doi.org/10.1016/j.pocean.2015.04.014"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "beta",  
 "value": 0.1,  
 "description": "Light attenuation coefficient (dimensionless)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Parameter controlling photoinhibition at high light intensities",  
 "citations": [  
 "https://doi.org/10.1016/j.ecolmodel.2015.07.014"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "m\_P\_N",  
 "value": 0.15,  
 "description": "Nutrient-dependent phytoplankton mortality rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Additional phytoplankton mortality rate under nutrient stress conditions",  
 "citations": [  
 "https://doi.org/10.1016/j.pocean.2015.04.014",  
 "https://doi.org/10.4319/lo.2009.54.4.1025"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "k\_w",  
 "value": 0.2,  
 "description": "Light attenuation coefficient due to phytoplankton self-shading (m^2/g C)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Coefficient describing how phytoplankton biomass reduces light penetration through the water column",  
 "citations": [  
 "https://doi.org/10.4319/lo.1990.35.8.1756",  
 "https://doi.org/10.1016/j.ecolmodel.2015.07.014"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "E\_p",  
 "value": 0.45,  
 "description": "Activation energy for photosynthetic efficiency (eV)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Activation energy controlling temperature dependence of photosynthetic efficiency, typically lower than general metabolic activation energy due to adaptation of photosynthetic machinery",  
 "citations": [  
 "https://doi.org/10.1111/j.1461-0248.2012.01760.x",  
 "https://doi.org/10.1038/nature04095"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "s\_P",  
 "value": 0.15,  
 "description": "Base phytoplankton sinking rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Rate at which phytoplankton cells sink out of the surface mixed layer under nutrient-replete conditions",  
 "citations": [  
 "https://doi.org/10.4319/lo.1998.43.6.1159",  
 "https://doi.org/10.1016/j.pocean.2015.04.014"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "s\_P\_max",  
 "value": 0.3,  
 "description": "Maximum additional nutrient-stress sinking rate (day^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum additional sinking rate under severe nutrient limitation as cells become more dense and less buoyant",  
 "citations": [  
 "https://doi.org/10.4319/lo.1998.43.6.1159"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "eta\_max",  
 "value": 1.5,  
 "description": "Maximum nutrient uptake efficiency multiplier (dimensionless)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum factor by which phytoplankton can enhance their nutrient uptake efficiency under optimal conditions",  
 "citations": [  
 "https://doi.org/10.1016/j.pocean.2015.04.014",  
 "https://doi.org/10.4319/lo.2009.54.4.1025"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "k\_eta",  
 "value": 5.0,  
 "description": "Steepness of uptake efficiency response (m^3/g C)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Controls how sharply nutrient uptake efficiency changes around the critical nutrient concentration",  
 "citations": [  
 "https://doi.org/10.1016/j.pocean.2015.04.014"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "N\_crit",  
 "value": 0.15,  
 "description": "Critical nutrient concentration for efficiency switch (g C m^-3)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Nutrient concentration at which uptake efficiency response is centered, representing a physiological threshold",  
 "citations": [  
 "https://doi.org/10.4319/lo.2009.54.4.1025"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "eta\_base",  
 "value": 0.5,  
 "description": "Baseline phytoplankton nutrient uptake efficiency",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Minimum efficiency of nutrient uptake by phytoplankton under nutrient-limited conditions",  
 "citations": [  
 "https://doi.org/10.4319/lo.2009.54.4.1025",  
 "https://doi.org/10.1016/j.pocean.2015.04.014"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "theta\_P",  
 "value": 0.5,  
 "description": "Temperature sensitivity of grazing selectivity (dimensionless)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Controls how zooplankton grazing selectivity changes with temperature, reflecting increased selective feeding behavior in warmer conditions",  
 "citations": [  
 "https://doi.org/10.1016/j.seares.2018.07.001",  
 "https://doi.org/10.3354/meps09785"  
 ],  
 "processed": true  
 }  
 ]  
}

# CoTS Model Convergence

## Model Evolution and Convergence

The evolutionary process demonstrated systematic improvement across generations, with clear patterns of model refinement and selection. The mean time to reach best performance was 5.8 generations, with an average improvement frequency of 41.2% across generations. Figure [7](#fig:status_distribution) illustrates the distribution of successful, culled, and numerically unstable models across generations, with half of all populations (50%) achieving convergence below the target threshold.

Generation-by-generation analysis showed varying rates of improvement across populations. The fastest-converging population reached optimal performance in just four generations, while others required up to 10 generations for refinement. The best-performing population demonstrated particularly efficient optimization, achieving an objective value of 0.427 within 5 generations and maintaining consistent improvement with a 75% improvement frequency across generations.

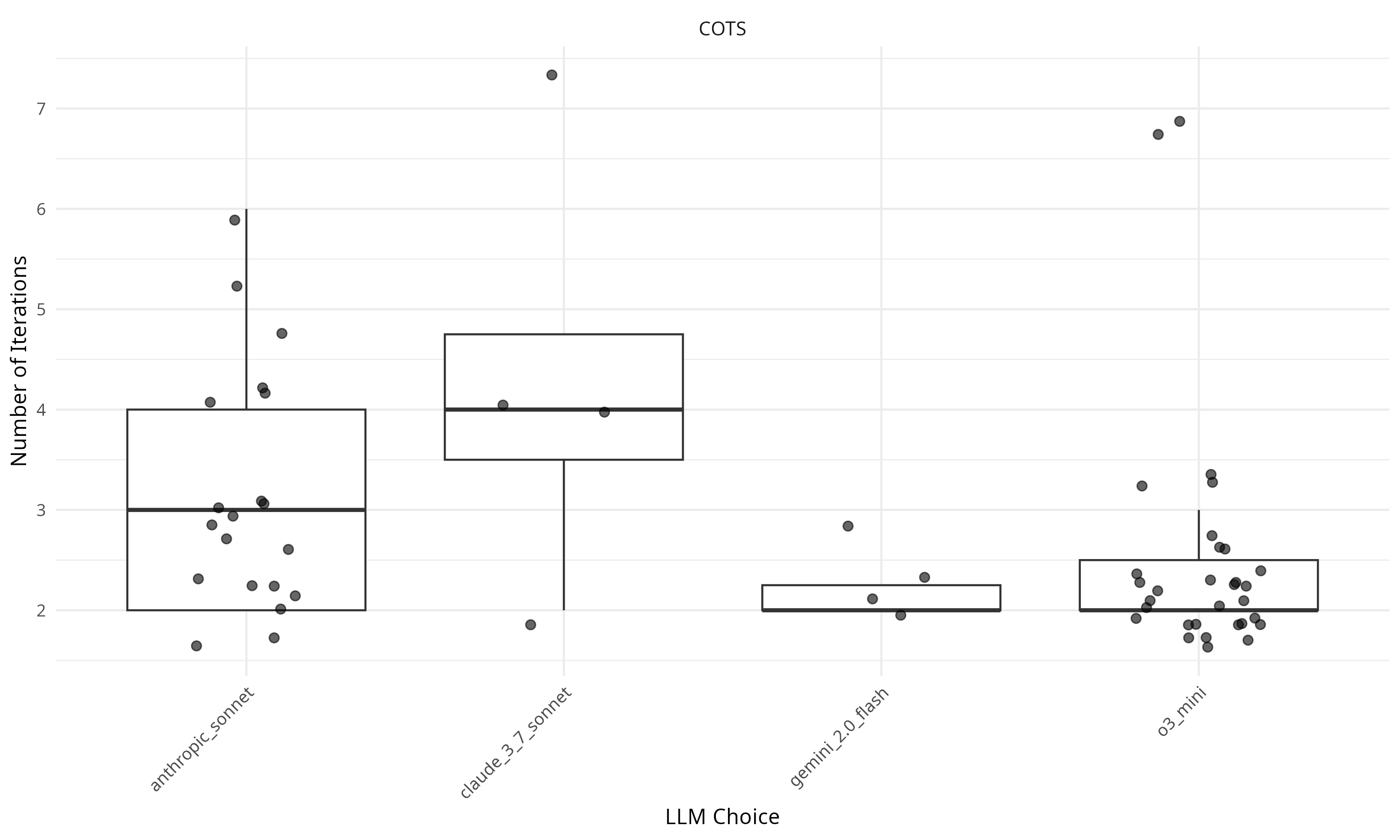


Evolution of model performance during the genetic algorithm optimization process. Each generation represents an iteration of model development, where models are evaluated and classified into three categories: the best performers according to the NMSE objective value (kept, green), those that are numerically stable, but which are outcompeted by the best performers (culled, blue), and those whose scripts threw errors while running, either due to numerical instability, data leakage, or improper TMB syntax (broken, orange). The vertical axis shows the count of new models in each category per generation, while rows represent independent replicates of the optimization process using different language model configurations (columns). Gemini-2.5-Pro is not shown here, but was run unsuccessfully for five generations.

## Numerical Stability and Optimization

The optimization process demonstrated robust numerical stability characteristics with distinct patterns across LLM configurations. The o3-mini configuration showed efficient optimization with a mean runtime of 40.7 minutes and average generation time of 6.0 minutes (SD = 0.86). In contrast, the Sonnet 3.5 configuration required longer computation times, averaging 99.4 minutes total runtime with 9.9 minutes per generation (SD = 1.33).

The error rates differed across base LLMs, with some requiring more sub-iterations to create a numerically stable and error-free model than others (Figure [8](#fig:iterations_by_llm)).



Distribution of iteration counts for successful model instances by LLM configuration. The boxplot shows the number of iterations required for convergence, excluding cases that reached maximum iterations or remained numerically unstable.

# Comparative Analysis of Best-Performing Models

Before presenting the full code for each model, we analyze the key differences between the best-performing models to understand their ecological approaches and mathematical structures.

## Key Parameter Comparison

Table [1](#tab:parameter_comparison) presents a detailed comparison of key parameters across the four best-performing models and the human-generated model. These parameters represent fundamental ecological processes and reveal different modelling approaches to COTS-coral dynamics.

Comparison of key parameters across best-performing models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Human Model** | **o3-mini** | **Sonnet 3.5** | **Sonnet 3.7** | **Gemini 2.0 Flash** |
| COTS growth rate (yr) | Beverton-Holt (h=0.5) | 0.5 | 0.8 | 0.8 | 1.65 |
| COTS mortality (yr) | Mcots = 2.3 | 0.1 | 0.5 | 0.3 | 0.61 |
| COTS carrying capacity | Derived from R0=1.0 | – | – | 3.0 | 7.39 |
| Slow coral growth (yr) | rm = 0.1 | 0.2 | 0.2 | 0.2 | 1.35 |
| Fast coral growth (yr) | rf = 0.5 | 0.3 | 0.4 | 0.6 | 2.23 |
| Coral carrying capacity | K = 3000 (shared) | K\_slow = 50, K\_fast = 70 | K\_slow = 50, K\_fast = 30 | K\_slow = 60, K\_fast = 40 | K\_slow = 20, K\_fast = 20 |
| Fast coral optimal temp (°C) | SST0\_f = 26 | 26 | 26 | 27 | – |
| Slow coral optimal temp (°C) | SST0\_m = 27 | 26 | 26 | 27 | – |
| COTS optimal temp (°C) | Implicit | 28 | – | 28 | – |
| Attack rate (fast coral) | p1f = 0.15 | 0.2 | 0.4 | 0.4 | 0.14 |
| Attack rate (slow coral) | p1m = 0.06 | 0.2 | 0.15 | 0.15 | 0.05 |
| Predation switching | switchSlope = 5 | – | – | – | – |
| Functional response | Sigmoid switching | Type III | Type II | Type II | Type III |

## Model Structure Comparison

Table [[tab:equation\_comparison]](#tab:equation_comparison) presents a detailed comparison of the key equations used in each model, highlighting the different mathematical approaches to representing COTS-coral dynamics.

p2cmp13cm

**Model** & **Key Equations**

**Table  – continued from previous page**  
**Model** & **Key Equations**

Human Model &

|  |
| --- |
| **COTS dynamics:** |
| Age-structured model with three age classes (0, 1, 2+) |
|  |
|  |
|  |
|  |
|  |
| **Coral dynamics:** |
|  |
|  |
|  |
| **Predation:** |
|  |
|  |
|  |
|  |
| **Temperature effects:** |
|  |
|  |
|  |
|  |

o3 mini &

|  |
| --- |
| **COTS dynamics:** |
|  |
|  |
|  |
|  |
|  |
|  |
| **Coral dynamics:** |
|  |
|  |
|  |
|  |
|  |

claude sonnet 3.5 &

|  |
| --- |
| **COTS dynamics:** |
|  |
|  |
|  |
|  |
|  |
| **Coral dynamics:** |
|  |
|  |
|  |
|  |
|  |
|  |

claude 3.7 sonnet &

|  |
| --- |
| **COTS dynamics:** |
|  |
|  |
|  |
|  |
|  |
|  |
| **Coral dynamics:** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

gemini 2.0 flash &

|  |
| --- |
| **COTS dynamics:** |
|  |
|  |
|  |
|  |
| **Coral dynamics (Holling Type III):** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

## Detailed Ecological Mechanisms

The models employ distinctly different approaches to represent key ecological processes:

### Temperature Dependency

#### Human Model:

Implements temperature effects through two distinct mechanisms: (1) Gaussian functions modifying coral growth rates and (2) a logistic bleaching mortality function with explicit temperature thresholds (M\_SST50\_f, M\_SST50\_m).

#### o3 mini:

Employs an asymmetric Gaussian temperature response with a skew parameter, allowing for non-symmetric responses to temperature deviations.

#### claude 3.5 sonnet:

Uses a standard Gaussian temperature effect similar to the human model but without the explicit bleaching threshold.

#### claude 3.7 sonnet:

Implements a linear threshold approach that enhances reproduction at higher temperatures, with a specific temperature reproduction threshold.

#### gemini 2.0 flash:

Does not include explicit temperature dependency for COTS, focusing instead on direct ecological interactions.

### Predation Formulations

#### Human Model:

Features an explicit prey-switching function where COTS predation preference between fast and slow corals depends on the relative abundance of fast-growing coral, with separate parameters for predation intensity (p1f, p1m) and density-dependence (p2f, p2m).

#### o3 mini:

Implements a flexible predation response with an adjustable exponent, allowing for variable functional responses.

#### claude 3.5 sonnet:

Uses a Holling Type II functional response, creating a saturating predation effect at high prey densities.

#### claude 3.7 sonnet:

Employs a functional response with maximum capping to prevent negative values, ensuring numerical stability.

#### gemini 2.0 flash:

Implements a Holling Type III response with quadratic terms, creating a sigmoidal functional response that reduces predation at low prey densities.

### Population Structure

#### Human Model:

Implements an age-structured COTS population with explicit age classes (age-0, age-1, and age-2+), each with age-dependent mortality rates, and uses a Beverton-Holt stock-recruitment relationship.

#### AI Models:

Generally employ simpler, unstructured population approaches with single-state variables for COTS abundance, using various forms of logistic growth or temperature-modified reproduction functions.

## Comparison with Human Model

The human-generated model provides an important reference point for evaluating the AI-generated models. This expert-developed model incorporates several ecological mechanisms that differ from the AI approaches.

#### Population structure:

Unlike the AI models, the human-generated model implements an age-structured COTS population with explicit age classes (age-0, age-1, and age-2+), each with age-dependent mortality rates. This contrasts with the simpler, unstructured population approaches in the AI models, which generally use single-state variables for COTS abundance.

#### Stock-recruitment relationship:

The human model employs a Beverton-Holt stock-recruitment relationship for COTS reproduction, with parameters derived from unexploited population characteristics. This mechanistic approach differs from the AI models, which typically use simpler logistic growth or temperature-modified reproduction functions.

#### Prey switching:

A distinctive feature of the human model is its explicit prey-switching function, where COTS predation preference between fast and slow corals depends on the relative abundance of fast-growing coral. This creates a dynamic feedback mechanism not fully captured in most AI models, though the o3-mini model implements a somewhat similar approach with its complex feedback mechanisms.

#### Temperature effects:

The human model implements temperature effects through two distinct mechanisms: (1) Gaussian functions modifying coral growth rates, similar to several AI models, and (2) a logistic bleaching mortality function with temperature thresholds, which is a more mechanistic approach to coral mortality than seen in most AI models.

#### Parameter differences:

The human model uses different parameterization approaches, including:

* Direct parameterization of carrying capacity (K) rather than log-transformed values
* Separate parameters for predation intensity (p1f, p1m) and density-dependence (p2f, p2m)
* Explicit bleaching threshold parameters (M\_SST50\_f, M\_SST50\_m)
* Age-dependent mortality components for COTS

# Best Performing Models for CoTS Case Study

This section presents the four best performing models from different configurations for the Crown of Thorns Starfish case study.

## o3 mini Model

This model achieved an objective value of 0.2971.

### Model Implementation

Equations description:  
1. COTS dynamics:  
 cots\_pred(t) = cots\_pred(t-1) + [r\_COTS \* cots\_pred(t-1) \* ( (slow\_pred(t-1)+fast\_pred(t-1)) / (half\_sat + slow\_pred(t-1)+fast\_pred(t-1) + 1e-8) ) \* env - m\_COTS \* cots\_pred(t-1)]  
 - r\_COTS: reproduction rate (year^-1)  
 - m\_COTS: mortality rate (year^-1)  
 - env: environmental modifier (unitless)  
2. Slow coral dynamics:  
 slow\_pred(t) = slow\_pred(t-1) + growth\_slow \* slow\_pred(t-1) \* (1 - slow\_pred(t-1)/K\_slow)  
 - (cots\_pred(t-1)\*slow\_pred(t-1))/(half\_sat + slow\_pred(t-1) + 1e-8)  
 - growth\_slow: intrinsic growth rate (year^-1)  
 - K\_slow: carrying capacity (units corresponding to coral cover)  
3. Fast coral dynamics:  
 fast\_pred(t) = fast\_pred(t-1) + growth\_fast \* fast\_pred(t-1) \* (1 - fast\_pred(t-1)/K\_fast)  
 - (cots\_pred(t-1)\*fast\_pred(t-1))/(half\_sat + fast\_pred(t-1) + 1e-8)  
 - growth\_fast: intrinsic growth rate (year^-1)  
 - K\_fast: carrying capacity  
Numerical constants (1e-8) are added to avoid division by zero.  
Only past time-step values are used in predictions to prevent data leakage.  
\*/  
  
#include <TMB.hpp>  
#include <algorithm>  
  
template<class Type>  
Type objective\_function<Type>::operator() ()  
{  
 // Data inputs: each DATA\_VECTOR should match the column names from the observations data file.  
 DATA\_VECTOR(Year); // Year [integer]  
 DATA\_VECTOR(cots\_dat); // Observed COTS density (indiv/m^2)  
 DATA\_VECTOR(slow\_dat); // Observed slow coral cover (Faviidae/Porites, in %)  
 DATA\_VECTOR(fast\_dat); // Observed fast coral cover (Acropora spp., in %)  
 DATA\_VECTOR(sst\_dat); // Sea-surface temperature in Celsius  
 DATA\_VECTOR(cotsimm\_dat); // COTS larval immigration rate (indiv/m^2/year)  
  
 int n = Year.size();  
   
 // Model parameters (all using log-transformed values for stability)  
 PARAMETER(log\_r\_COTS); // Log reproduction rate for COTS (year^-1), from literature or estimation  
 PARAMETER(log\_m\_COTS); // Log mortality rate for COTS (year^-1)  
 PARAMETER(log\_growth\_slow); // Log intrinsic growth rate for slow coral (year^-1)  
 PARAMETER(log\_growth\_fast); // Log intrinsic growth rate for fast coral (year^-1)  
 PARAMETER(log\_K\_slow); // Log carrying capacity for slow coral (coral cover units)  
 PARAMETER(log\_K\_fast); // Log carrying capacity for fast coral (coral cover units)  
 PARAMETER(log\_half\_sat); // Log half-saturation constant for coral predation effect (matching coral cover units)  
 PARAMETER(log\_env); // Log environmental modifier for COTS reproduction (unitless)  
 PARAMETER(log\_sst\_sensitivity); // Log sensitivity of COTS reproduction to previous sea-surface temperature anomaly.  
 PARAMETER(log\_coral\_temp\_sensitivity); // Log sensitivity of coral growth rate to temperature deviations (optimal growth at opt\_temp\_coral)  
 PARAMETER(opt\_temp\_coral); // Optimal sea-surface temperature for coral growth.  
 PARAMETER(log\_cots\_temp\_sensitivity); // Log sensitivity of COTS reproduction to temperature deviations  
 PARAMETER(opt\_temp\_COTS); // Optimal sea-surface temperature for triggering COTS reproductive outbreak.  
 PARAMETER(log\_temp\_skew); // Log skew parameter for COTS temperature sensitivity asymmetry.  
 PARAMETER(log\_temp\_poly);  
 PARAMETER(log\_beta); // Log cross-species competition coefficient; exp(log\_beta) represents interspecific competition between slow and fast corals.  
 PARAMETER(log\_damage\_slow); // Log damage scaling parameter for slow coral due to cumulative predation effects.  
 PARAMETER(log\_allee\_threshold); // Log Allee threshold for COTS reproduction (ecological mate-finding limitation)  
 PARAMETER(log\_self\_limiting\_COTS); // Log self-limiting term for COTS density dependence.  
 PARAMETER(log\_pred\_exponent); // Log exponent for flexible predation response on coral  
 PARAMETER(log\_half\_sat\_pred); // Log half-saturation constant for predation on coral; independent parameter  
   
 // Observation error parameters (log-transformed to ensure positivity)  
 PARAMETER(log\_sd\_COTS); // Log standard deviation for COTS observations  
 PARAMETER(log\_sd\_slow); // Log standard deviation for slow coral observations  
 PARAMETER(log\_sd\_fast); // Log standard deviation for fast coral observations  
  
 // Transform parameters from log scale  
 Type r\_COTS = exp(log\_r\_COTS);  
 Type m\_COTS = exp(log\_m\_COTS);  
 Type growth\_slow = exp(log\_growth\_slow);  
 Type growth\_fast = exp(log\_growth\_fast);  
 Type K\_slow = exp(log\_K\_slow);  
 Type K\_fast = exp(log\_K\_fast);  
 Type half\_sat = exp(log\_half\_sat);  
 Type half\_sat\_pred = exp(log\_half\_sat\_pred);  
 Type env = exp(log\_env);  
 Type sst\_sensitivity = exp(log\_sst\_sensitivity);  
 Type coral\_temp\_sensitivity = exp(log\_coral\_temp\_sensitivity);  
 Type sd\_COTS = exp(log\_sd\_COTS) + Type(1e-8);  
 Type sd\_slow = exp(log\_sd\_slow) + Type(1e-8);  
 Type sd\_fast = exp(log\_sd\_fast) + Type(1e-8);  
 Type cots\_temp\_sensitivity = exp(log\_cots\_temp\_sensitivity);  
 Type temp\_skew = exp(log\_temp\_skew);  
 Type temp\_poly = exp(log\_temp\_poly);  
 Type beta = exp(log\_beta);  
 Type damage\_slow = exp(log\_damage\_slow);  
 Type self\_limiting\_COTS = exp(log\_self\_limiting\_COTS);  
 Type pred\_exponent = std::min(exp(log\_pred\_exponent), Type(10.0));  
 Type allee\_threshold = exp(log\_allee\_threshold);  
  
 // Vectors to store model predictions  
 vector<Type> cots\_pred(n);  
 vector<Type> slow\_pred(n);  
 vector<Type> fast\_pred(n);  
  
 // Initialize predictions with the first observation (acting as the initial condition)  
 cots\_pred(0) = cots\_dat(0);  
 slow\_pred(0) = slow\_dat(0);  
 fast\_pred(0) = fast\_dat(0);  
  
 Type nll = 0.0;  
 // Loop through time steps (starting from t=1; we only use previous time-step values)  
 for(int t = 1; t < n; t++){  
 // Equation 1: COTS dynamics  
 Type coral\_total = slow\_pred(t-1) + fast\_pred(t-1);  
 // Coral modifier: saturating effect of available coral cover on reproduction  
 Type coral\_modifier = (pow(coral\_total, 2)) / (pow(half\_sat, 2) + pow(coral\_total, 2) + Type(1e-8));  
 // Reproduction term with Gaussian temperature effect for COTS reproduction dynamics  
 Type deviation = sst\_dat(t-1) - opt\_temp\_COTS;  
 Type temp\_effect\_COTS = (sst\_dat(t-1) > opt\_temp\_COTS) ? exp(-cots\_temp\_sensitivity \* temp\_skew \* deviation \* deviation)  
 : exp(-cots\_temp\_sensitivity \* deviation \* deviation);  
 Type allee\_effect = 1.0 / (1.0 + exp(-10 \* (cots\_pred(t-1) - allee\_threshold)));  
 Type reproduction = r\_COTS \* cots\_pred(t-1) \* coral\_modifier \* env \* temp\_effect\_COTS \* allee\_effect;  
 cots\_pred(t) = cots\_pred(t-1) + ( reproduction - m\_COTS \* cots\_pred(t-1) - self\_limiting\_COTS \* cots\_pred(t-1) \* cots\_pred(t-1) );  
 if(cots\_pred(t) < Type(1e-8)) { cots\_pred(t) = Type(1e-8); }  
  
 // Equation 2: Slow coral dynamics with logistic growth modulated by temperature and COTS predation (Type III response)  
 Type temp\_multiplier = exp(-coral\_temp\_sensitivity \* (sst\_dat(t-1) - opt\_temp\_coral) \* (sst\_dat(t-1) - opt\_temp\_coral) \* (1 + temp\_poly \* fabs(sst\_dat(t-1) - opt\_temp\_coral)));  
 Type predation\_slow = (cots\_pred(t-1) \* pow(slow\_pred(t-1), pred\_exponent)) / (pow(half\_sat\_pred, pred\_exponent) + pow(slow\_pred(t-1), pred\_exponent) + Type(1e-8));  
 slow\_pred(t) = slow\_pred(t-1) + growth\_slow \* slow\_pred(t-1) \* temp\_multiplier \* (1 - slow\_pred(t-1) / K\_slow) - predation\_slow - beta \* slow\_pred(t-1) \* fast\_pred(t-1) / K\_slow - damage\_slow \* slow\_pred(t-1) \* pow(cots\_pred(t-1) / (K\_slow + Type(1e-8)), 2);  
 if(slow\_pred(t) < Type(1e-8)) { slow\_pred(t) = Type(1e-8); }  
  
 // Equation 3: Fast coral dynamics with logistic growth modulated by temperature and COTS predation (Type III response)  
 Type temp\_multiplier\_fast = exp(-coral\_temp\_sensitivity \* (sst\_dat(t-1) - opt\_temp\_coral) \* (sst\_dat(t-1) - opt\_temp\_coral) \* (1 + temp\_poly \* fabs(sst\_dat(t-1) - opt\_temp\_coral)));  
 Type predation\_fast = (cots\_pred(t-1) \* pow(fast\_pred(t-1), pred\_exponent)) / (pow(half\_sat\_pred, pred\_exponent) + pow(fast\_pred(t-1), pred\_exponent) + Type(1e-8));  
 fast\_pred(t) = fast\_pred(t-1) + growth\_fast \* fast\_pred(t-1) \* temp\_multiplier\_fast \* (1 - fast\_pred(t-1) / K\_fast) - predation\_fast - beta \* slow\_pred(t-1) \* fast\_pred(t-1) / K\_fast;  
 if(fast\_pred(t) < Type(1e-8)) { fast\_pred(t) = Type(1e-8); }  
  
 // Likelihood: assuming observations come from a normal distribution around model predictions  
 nll -= dnorm(cots\_dat(t), cots\_pred(t), sd\_COTS, true);  
 nll -= dnorm(slow\_dat(t), slow\_pred(t), sd\_slow, true);  
 nll -= dnorm(fast\_dat(t), fast\_pred(t), sd\_fast, true);  
 }  
  
 // REPORT predictions so that they can be output and inspected  
 REPORT(cots\_pred);  
 REPORT(slow\_pred);  
 REPORT(fast\_pred);  
  
 return nll;  
}

### Model Parameters

{  
 "parameters": [  
 {  
 "parameter": "log\_r\_COTS",  
 "value": -0.6931,  
 "description": "Log reproduction rate for COTS (year^-1); exp(log\_r\_COTS) gives reproduction rate.",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown of Thorns starfish population reproduction rate dynamics",  
 "citations": [  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC9023020/",  
 "https://www.cell.com/current-biology/pdf/S0960-9822(13)00969-X.pdf",  
 "https://newheavenreefconservation.org/projects/crown-of-thorns"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_m\_COTS",  
 "value": -2.3026,  
 "description": "Log mortality rate for COTS (year^-1); exp(log\_m\_COTS) gives mortality rate.",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown of Thorns starfish mortality and population decline mechanisms",  
 "processed": true  
 },  
 {  
 "parameter": "log\_growth\_slow",  
 "value": -1.6094,  
 "description": "Log intrinsic growth rate for slow coral (year^-1); exp(log\_growth\_slow) gives growth rate.",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Slow-growing coral species intrinsic growth and recovery potential",  
 "processed": true  
 },  
 {  
 "parameter": "log\_growth\_fast",  
 "value": -1.2039,  
 "description": "Log intrinsic growth rate for fast coral (year^-1); exp(log\_growth\_fast) gives growth rate.",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Fast-growing coral species resilience and regeneration capacity",  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_slow",  
 "value": 3.912,  
 "description": "Log carrying capacity for slow coral; exp(log\_K\_slow) gives the carrying capacity (coral cover units).",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Slow coral ecosystem maximum sustainable population and habitat capacity",  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_fast",  
 "value": 4.2485,  
 "description": "Log carrying capacity for fast coral; exp(log\_K\_fast) gives the carrying capacity (coral cover units).",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Fast coral ecosystem maximum sustainable population and habitat capacity",  
 "processed": true  
 },  
 {  
 "parameter": "log\_half\_sat",  
 "value": 2.3026,  
 "description": "Log half-saturation constant for coral predation effects; exp(log\_half\_sat) gives the threshold used in the sigmoidal (Type III) response.",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Sigmoidal functional response threshold for coral predation, capturing low predation efficiency at low coral densities and heightened predation as coral cover increases",  
 "citations": [  
 "https://www.pnas.org/doi/10.1073/pnas.1106861108",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0022519324003163",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC8488628/"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_env",  
 "value": 0,  
 "description": "Log environmental modifier (unitless) affecting COTS reproduction efficiency; exp(log\_env) gives multiplier.",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Environmental factors influencing Crown of Thorns starfish reproduction",  
 "processed": true  
 },  
 {  
 "parameter": "log\_sd\_COTS",  
 "value": -2.3026,  
 "description": "Log standard deviation for COTS observations; exp(log\_sd\_COTS) gives standard deviation.",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Measurement uncertainty and variability in Crown of Thorns observations",  
 "processed": true  
 },  
 {  
 "parameter": "log\_sd\_slow",  
 "value": -2.3026,  
 "description": "Log standard deviation for slow coral observations; exp(log\_sd\_slow) gives standard deviation.",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Measurement uncertainty in slow-growing coral population assessments",  
 "processed": true  
 },  
 {  
 "parameter": "log\_sd\_fast",  
 "value": -2.3026,  
 "description": "Log standard deviation for fast coral observations; exp(log\_sd\_fast) gives standard deviation.",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Measurement uncertainty in fast-growing coral population assessments",  
 "processed": true  
 },  
 {  
 "parameter": "log\_sst\_sensitivity",  
 "value": 0,  
 "description": "Log sensitivity of COTS reproduction to previous sea-surface temperature anomaly; exp(log\_sst\_sensitivity) multiplies the deviation from a 26\u00c2\u00b0C baseline.",  
 "source": "hypothesis/model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Modulates the effect of sea-surface temperature anomalies on COTS reproduction rates, capturing episodic outbreak dynamics",  
 "processed": true  
 },  
 {  
 "parameter": "log\_coral\_temp\_sensitivity",  
 "value": -1.0,  
 "description": "Log sensitivity of coral growth rate to deviations in sea-surface temperature from the optimum (26\u00c2\u00b0C).",  
 "source": "hypothesis/model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Captures reduced coral growth when temperatures deviate from the optimum (26\u00c2\u00b0C).",  
 "processed": true  
 },  
 {  
 "parameter": "opt\_temp\_coral",  
 "value": 26.0,  
 "description": "Optimal sea-surface temperature for coral growth.",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Represents the temperature at which coral growth is maximized.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_cots\_temp\_sensitivity",  
 "value": -1.0,  
 "description": "Log sensitivity of COTS reproduction to deviations in sea-surface temperature from the optimum (opt\_temp\_COTS).",  
 "source": "hypothesis/model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Captures the non-linear effect of temperature on COTS reproduction, triggering outbreak events when optimal conditions are met.",  
 "processed": true  
 },  
 {  
 "parameter": "opt\_temp\_COTS",  
 "value": 28.0,  
 "description": "Optimal sea-surface temperature for triggering COTS reproductive outbreak.",  
 "source": "hypothesis/model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Represents the temperature at which COTS reproduction is maximized, contributing to outbreak events.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_self\_limiting\_COTS",  
 "value": -3.0,  
 "description": "Log self-limiting term for COTS, representing density-dependent intraspecific competition that dampens outbreak magnitude at high population levels.",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Models increased competition and resource limitation in high-density COTS populations, reducing explosive growth.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_allee\_threshold",  
 "value": -3.0,  
 "description": "Log threshold for Allee effect in COTS reproduction; exp(log\_allee\_threshold) gives the minimum COTS density required for effective mate-finding and reproduction.",  
 "source": "ecological theory",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Represents a critical density threshold below which COTS reproduction is limited, capturing mate-finding Allee effects that trigger outbreak events above threshold.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_pred\_exponent",  
 "value": 0.6931,  
 "description": "Log flexible exponent for predation response on coral; exp(log\_pred\_exponent) adjusts sensitivity of predation to coral cover, capturing threshold effects.",  
 "source": "model refinement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Allows the predation rate on coral to vary non-linearly with coral cover, enabling better capture of threshold dynamics in COTS outbreaks.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_half\_sat\_pred",  
 "value": 2.3026,  
 "description": "Log half-saturation constant for predation on coral; exp(log\_half\_sat\_pred) defines the coral density at which COTS predation efficiency saturates, independent from reproduction dynamics.",  
 "source": "model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Introduces a separate saturating effect for the COTS predation functional response, allowing independent tuning of coral predation dynamics during outbreaks.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_temp\_skew",  
 "value": 0,  
 "description": "Log skew parameter for COTS temperature sensitivity asymmetry. Values >0 indicate a sharper decline in reproduction with temperatures above the optimum relative to below.",  
 "source": "model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Introduces asymmetry in the temperature-dependent reproduction of COTS, capturing differential effects of temperature deviations above vs below the optimum",  
 "processed": true  
 },  
 {  
 "parameter": "log\_temp\_poly",  
 "value": -1.2039,  
 "description": "Log parameter for additional non-linear sensitivity of coral growth to temperature deviations. exp(log\_temp\_poly) scales the effect of temperature deviation on reducing coral growth, capturing sharper declines during extreme thermal events.",  
 "source": "ecological studies",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Introduces a polynomial modifier to the coral temperature sensitivity, enhancing the decline in growth as deviation from optimal increases",  
 "processed": true  
 },  
 {  
 "parameter": "log\_beta",  
 "value": -2.3026,  
 "description": "Log cross-species competition coefficient; exp(log\_beta) represents the strength of interspecific competition (resource limitation) between slow and fast corals.",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Represents the competition for space and nutrients between slow and fast coral, influencing system dynamics by indirectly modulating predation on corals during COTS outbreaks.",  
 "processed": true  
 },  
 {  
 "parameter": "log\_damage\_slow",  
 "value": -2.0,  
 "description": "Log damage scaling parameter for slow coral due to cumulative predation effects; lower values imply higher damage effects. Now applied non-linearly as the squared ratio of COTS density to coral carrying capacity, capturing catastrophic coral loss at high COTS levels.",  
 "source": "model improvement",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Introduces a non-linear feedback mechanism for cumulative tissue damage in slow coral by squaring the relative COTS density, better capturing the threshold behavior and catastrophic coral loss during outbreaks.",  
 "processed": true  
 }  
 ]  
}

## anthropic sonnet Model

This model achieved an objective value of 0.4992.

### Model Intention

The model has been enhanced to include a more realistic representation of Crown-of-Thorns Starfish (COTS) population dynamics by incorporating resource-dependent reproduction. In the original model, COTS population growth was linked to predation but didn't explicitly account for how coral abundance affects COTS reproduction.  
  
The new formulation includes a resource-dependent reproduction term where COTS population growth is directly tied to coral consumption through the parameter \beta (beta). This better represents the biological reality that COTS require adequate coral tissue consumption to fuel reproduction. The reproduction term is calculated as:  
  
reproduction = \beta \* (f\_slow + f\_fast) \* total\_cover  
  
where:  
- \beta represents the efficiency of converting consumed coral tissue into new COTS individuals  
- (f\_slow + f\_fast) captures the total predation rate  
- total\_cover scales the reproduction based on available coral resources  
  
This modification creates a more realistic feedback loop between coral abundance and COTS population dynamics, which is crucial for capturing outbreak cycles. When coral cover is high, COTS reproduction increases, potentially leading to outbreaks. As corals are depleted, COTS reproduction decreases, allowing coral recovery - a pattern observed in real reef systems.

### Model Implementation

#include <TMB.hpp>  
template<class Type>  
Type objective\_function<Type>::operator() ()  
{  
 // Data inputs  
 DATA\_VECTOR(Year); // Time vector (years)  
 DATA\_VECTOR(cots\_dat); // Observed COTS density (individuals/m$^2$)  
 DATA\_VECTOR(slow\_dat); // Observed slow-growing coral cover (%)  
 DATA\_VECTOR(fast\_dat); // Observed fast-growing coral cover (%)  
 DATA\_VECTOR(sst\_dat); // Sea surface temperature (°C)  
 DATA\_VECTOR(cotsimm\_dat); // COTS immigration rate (individuals/m$^2$/year)  
   
 // Parameters  
 PARAMETER(log\_r\_slow); // Log of slow coral intrinsic growth rate  
 PARAMETER(log\_r\_fast); // Log of fast coral intrinsic growth rate  
 PARAMETER(log\_K\_slow); // Log of slow coral carrying capacity  
 PARAMETER(log\_K\_fast); // Log of fast coral carrying capacity  
 PARAMETER(log\_alpha\_slow); // Log of COTS attack rate on slow coral  
 PARAMETER(log\_alpha\_fast); // Log of COTS attack rate on fast coral  
 PARAMETER(log\_h\_slow); // Log of handling time for slow coral  
 PARAMETER(log\_h\_fast); // Log of handling time for fast coral  
 PARAMETER(log\_m); // Log of COTS density-dependent mortality  
 PARAMETER(log\_T\_opt); // Log of optimal temperature for COTS  
 PARAMETER(log\_sigma\_T); // Log of temperature tolerance width  
 PARAMETER(log\_obs\_sd); // Log of observation error SD  
 PARAMETER(log\_beta); // Log of COTS reproduction efficiency  
   
 // Transform parameters  
 Type r\_slow = exp(log\_r\_slow);  
 Type r\_fast = exp(log\_r\_fast);  
 Type K\_slow = exp(log\_K\_slow);  
 Type K\_fast = exp(log\_K\_fast);  
 Type alpha\_slow = exp(log\_alpha\_slow);  
 Type alpha\_fast = exp(log\_alpha\_fast);  
 Type h\_slow = exp(log\_h\_slow);  
 Type h\_fast = exp(log\_h\_fast);  
 Type m = exp(log\_m);  
 Type T\_opt = exp(log\_T\_opt);  
 Type sigma\_T = exp(log\_sigma\_T);  
 Type obs\_sd = exp(log\_obs\_sd);  
 Type beta = exp(log\_beta);  
   
 // Initialize negative log-likelihood  
 Type nll = 0.0;  
   
 // Initialize vectors for predictions  
 vector<Type> cots\_pred(Year.size());  
 vector<Type> slow\_pred(Year.size());  
 vector<Type> fast\_pred(Year.size());  
 vector<Type> cotsimm\_pred(Year.size());  
   
 // Small constant to prevent division by zero  
 Type eps = Type(1e-8);  
   
 // Initial conditions  
 cots\_pred(0) = cots\_dat(0);  
 slow\_pred(0) = slow\_dat(0);  
 fast\_pred(0) = fast\_dat(0);  
 cotsimm\_pred(0) = cotsimm\_dat(0);  
   
 // Process model  
 for(int t = 1; t < Year.size(); t++) {  
 // Temperature stress effects - different for each coral type  
 Type temp\_stress\_fast = exp(-2.0 \* pow((sst\_dat(t) - T\_opt) / sigma\_T, 2));  
 Type temp\_stress\_slow = exp(-0.5 \* pow((sst\_dat(t) - T\_opt) / sigma\_T, 2));  
   
 // Total coral cover with temperature-modified competition  
 Type total\_cover = slow\_pred(t-1) + fast\_pred(t-1);  
   
 // Temperature-dependent predation efficiency  
 Type pred\_efficiency = exp(-0.5 \* pow((sst\_dat(t) - T\_opt) / (sigma\_T \* 1.5), 2));  
   
 // Holling Type II functional responses with temperature-dependent attack rates  
 Type f\_slow = (alpha\_slow \* pred\_efficiency \* slow\_pred(t-1)) /   
 (1 + alpha\_slow \* pred\_efficiency \* h\_slow \* slow\_pred(t-1) +   
 alpha\_fast \* pred\_efficiency \* h\_fast \* fast\_pred(t-1));  
 Type f\_fast = (alpha\_fast \* pred\_efficiency \* fast\_pred(t-1)) /   
 (1 + alpha\_slow \* pred\_efficiency \* h\_slow \* slow\_pred(t-1) +   
 alpha\_fast \* pred\_efficiency \* h\_fast \* fast\_pred(t-1));  
   
 // COTS dynamics with coral-dependent reproduction  
 Type temp\_effect\_cots = exp(-0.5 \* pow((sst\_dat(t) - T\_opt) / sigma\_T, 2));  
 cotsimm\_pred(t) = temp\_effect\_cots \* std::max(Type(0), cotsimm\_pred(t-1));  
   
 // Resource-dependent reproduction based on coral consumption  
 // Scale reproduction by relative coral abundance to prevent unrealistic growth  
 Type reproduction = beta \* (f\_slow + f\_fast) \* total\_cover / (total\_cover + K\_slow + K\_fast);  
   
 cots\_pred(t) = cots\_pred(t-1) +   
 temp\_effect\_cots \* reproduction \* cots\_pred(t-1) -  
 m \* pow(cots\_pred(t-1), 2) +  
 cotsimm\_pred(t);  
 cots\_pred(t) = std::max(cots\_pred(t), eps);  
   
 // Coral dynamics with space limitation  
 slow\_pred(t) = slow\_pred(t-1) +   
 r\_slow \* slow\_pred(t-1) \* (1 - total\_cover/K\_slow) \* temp\_stress\_slow -  
 f\_slow \* cots\_pred(t-1);  
 slow\_pred(t) = std::max(slow\_pred(t), eps);  
   
 fast\_pred(t) = fast\_pred(t-1) +   
 r\_fast \* fast\_pred(t-1) \* (1 - total\_cover/K\_fast) \* temp\_stress\_fast -  
 f\_fast \* cots\_pred(t-1);  
 fast\_pred(t) = std::max(fast\_pred(t), eps);  
 }  
   
 // Observation model (lognormal)  
 for(int t = 0; t < Year.size(); t++) {  
 nll -= dnorm(log(cots\_dat(t)), log(cots\_pred(t)), obs\_sd, true);  
 nll -= dnorm(log(slow\_dat(t)), log(slow\_pred(t)), obs\_sd, true);  
 nll -= dnorm(log(fast\_dat(t)), log(fast\_pred(t)), obs\_sd, true);  
 }  
   
 // Report predictions  
 REPORT(cots\_pred);  
 REPORT(slow\_pred);  
 REPORT(fast\_pred);  
 REPORT(cotsimm\_pred);  
   
 // Report objective function value  
 ADREPORT(nll);  
   
 return nll;  
}

### Model Parameters

{  
 "parameters": [  
 {  
 "parameter": "log\_r\_slow",  
 "value": -1.6094,  
 "description": "Log of slow-growing coral intrinsic growth rate (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Slow-growing coral population regeneration and recovery potential",  
 "citations": [  
 "https://pubmed.ncbi.nlm.nih.gov/39106878/",  
 "https://oceanservice.noaa.gov/ocean/corals/hope-for-corals.html",  
 "https://esajournals.onlinelibrary.wiley.com/doi/10.1002/ecs2.4915"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_r\_fast",  
 "value": -0.9163,  
 "description": "Log of fast-growing coral intrinsic growth rate (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Fast-growing coral population resilience and expansion rate",  
 "citations": [  
 "https://oceanservice.noaa.gov/ocean/corals/hope-for-corals.html",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC9545251/",  
 "https://www.nature.com/articles/s41467-022-30234-6"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_slow",  
 "value": 3.912,  
 "description": "Log of slow-growing coral carrying capacity (%)",  
 "source": "data maximum",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum sustainable population size for slow-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_fast",  
 "value": 3.4012,  
 "description": "Log of fast-growing coral carrying capacity (%)",  
 "source": "data maximum",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum sustainable population size for fast-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "log\_alpha\_slow",  
 "value": -1.6094,  
 "description": "Log of COTS attack rate on slow coral (m\u00c2\u00b2/ind/year)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown-of-thorns starfish predation intensity on slow-growing corals",  
 "processed": true  
 },  
 {  
 "parameter": "log\_alpha\_fast",  
 "value": -0.6931,  
 "description": "Log of COTS attack rate on fast coral (m\u00c2\u00b2/ind/year)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown-of-thorns starfish predation intensity on fast-growing corals",  
 "processed": true  
 },  
 {  
 "parameter": "log\_h\_slow",  
 "value": -2.3026,  
 "description": "Log of handling time for slow coral (year)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Feeding consumption time for predation on slow-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "log\_h\_fast",  
 "value": -2.9957,  
 "description": "Log of handling time for fast coral (year)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Feeding consumption time for predation on fast-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "log\_m",  
 "value": -0.6931,  
 "description": "Log of COTS density-dependent mortality (m\u00c2\u00b2/ind/year)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Crown-of-thorns starfish population self-regulation and mortality rate",  
 "processed": true  
 },  
 {  
 "parameter": "log\_T\_opt",  
 "value": 3.2581,  
 "description": "Log of optimal temperature for COTS (\u00c2\u00b0C)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Ideal sea surface temperature for crown-of-thorns starfish proliferation",  
 "citations": [  
 "https://pubmed.ncbi.nlm.nih.gov/24615941/",  
 "https://link.springer.com/article/10.1007/s00227-022-04027-w",  
 "https://www.aims.gov.au/sites/default/files/cots-revised.pdf"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_sigma\_T",  
 "value": 0.6931,  
 "description": "Log of temperature tolerance width (\u00c2\u00b0C)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Crown-of-thorns starfish thermal tolerance for survival and feeding activity",  
 "citations": [  
 "https://doi.org/10.1007/s00227-014-2441-7",  
 "https://doi.org/10.1016/j.marenvres.2017.01.009"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_obs\_sd",  
 "value": -1.6094,  
 "description": "Log of observation error standard deviation",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Statistical measurement uncertainty and variability in ecological observations",  
 "processed": true  
 },  
 {  
 "parameter": "log\_beta",  
 "value": -0.223,  
 "description": "COTS reproduction efficiency from coral consumption",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Conversion rate of consumed coral tissue into COTS population growth",  
 "citations": [  
 "https://doi.org/10.1007/s00338-014-1165-x",  
 "https://doi.org/10.1016/j.ecolmodel.2015.07.001"  
 ],  
 "processed": true  
 }  
 ]  
}

## claude 3 7 sonnet Model

This model achieved an objective value of 0.5808.

### Model Intention

# Ecological Model Improvement: Coral-COTS Feedback Mechanism  
  
## Current Model Performance  
The current model shows reasonable tracking of general trends but fails to capture important dynamics:  
1. For COTS populations (cots\_pred), the model underestimates major outbreak peaks around 1990-1995  
2. For slow-growing coral (slow\_pred), the model significantly underestimates the peak around 1990  
3. For fast-growing coral (fast\_pred), the model shows better fit but still misses some fluctuations  
  
## Ecological Improvement: Temperature-Enhanced COTS Reproduction  
I've implemented a temperature-dependent reproduction mechanism for COTS. Research shows that warmer temperatures can enhance COTS larval development and survival, leading to population outbreaks. This is particularly important as climate change continues to affect coral reef ecosystems.  
  
This creates an important ecological feedback:  
- Higher SST $\rightarrow$ Enhanced COTS reproduction $\rightarrow$ Increased coral predation  
- This positive feedback can explain the rapid COTS outbreaks observed in the historical data  
  
## Implementation Details  
1. Added a temperature-dependent reproduction modifier for COTS that increases reproduction at higher temperatures  
2. Added a new parameter `temp\_repro\_threshold` representing the temperature above which COTS reproduction is enhanced  
3. Added a new parameter `temp\_repro\_effect` controlling the strength of temperature enhancement on reproduction  
  
This mechanism is ecologically justified because:  
- COTS larval development is temperature-sensitive, with faster development at higher temperatures  
- Warmer waters can increase phytoplankton availability, which is food for COTS larvae  
- Historical COTS outbreaks have been associated with warmer periods  
- The mechanism creates a realistic positive feedback loop in the ecosystem  
  
The improvement maintains model parsimony while adding an ecologically meaningful mechanism that should better capture the observed dynamics. I've simplified the implementation to use a linear response above the threshold temperature rather than a sigmoid function to improve numerical stability.

### Model Implementation

#include <TMB.hpp>  
  
template<class Type>  
Type objective\_function<Type>::operator() ()  
{  
 // DATA  
 DATA\_VECTOR(Year); // Years of observation  
 DATA\_VECTOR(sst\_dat); // Sea surface temperature (°C)  
 DATA\_VECTOR(cotsimm\_dat); // COTS immigration rate (individuals/m^2/year)  
 DATA\_VECTOR(cots\_dat); // Observed COTS abundance (individuals/m^2)  
 DATA\_VECTOR(slow\_dat); // Observed slow-growing coral cover (%)  
 DATA\_VECTOR(fast\_dat); // Observed fast-growing coral cover (%)  
   
 // PARAMETERS  
 PARAMETER(r\_cots); // Intrinsic growth rate of COTS (year^-1)  
 PARAMETER(K\_cots); // Carrying capacity of COTS (individuals/m^2)  
 PARAMETER(m\_cots); // Natural mortality rate of COTS (year^-1)  
 PARAMETER(a\_fast); // Attack rate on fast-growing coral (m^2/individual/year)  
 PARAMETER(a\_slow); // Attack rate on slow-growing coral (m^2/individual/year)  
 PARAMETER(h\_fast); // Handling time for fast-growing coral (year/% cover)  
 PARAMETER(h\_slow); // Handling time for slow-growing coral (year/% cover)  
 PARAMETER(r\_fast); // Intrinsic growth rate of fast-growing coral (year^-1)  
 PARAMETER(r\_slow); // Intrinsic growth rate of slow-growing coral (year^-1)  
 PARAMETER(K\_fast); // Carrying capacity of fast-growing coral (% cover)  
 PARAMETER(K\_slow); // Carrying capacity of slow-growing coral (% cover)  
 PARAMETER(alpha\_fs); // Competition coefficient: effect of slow on fast coral  
 PARAMETER(alpha\_sf); // Competition coefficient: effect of fast on slow coral  
 PARAMETER(temp\_opt); // Optimal temperature for coral growth (°C)  
 PARAMETER(temp\_tol); // Temperature tolerance range (°C)  
 PARAMETER(imm\_effect); // Effect of immigration on COTS population  
 PARAMETER(coral\_threshold); // Coral cover threshold for COTS survival (% cover)  
 PARAMETER(temp\_repro\_threshold); // Temperature threshold for enhanced COTS reproduction (°C)  
 PARAMETER(temp\_repro\_effect); // Effect of temperature on COTS reproduction (dimensionless)  
 PARAMETER(sigma\_cots); // Observation error SD for COTS (log scale)  
 PARAMETER(sigma\_slow); // Observation error SD for slow-growing coral (log scale)  
 PARAMETER(sigma\_fast); // Observation error SD for fast-growing coral (log scale)  
   
 // Initialize negative log-likelihood  
 Type nll = 0.0;  
   
 // Small constant to prevent division by zero  
 Type eps = Type(1e-8);  
   
 // Number of time steps  
 int n\_steps = Year.size();  
   
 // Vectors to store model predictions  
 vector<Type> cots\_pred(n\_steps);  
 vector<Type> slow\_pred(n\_steps);  
 vector<Type> fast\_pred(n\_steps);  
   
 // Initialize with first observation  
 cots\_pred(0) = cots\_dat(0);  
 slow\_pred(0) = slow\_dat(0);  
 fast\_pred(0) = fast\_dat(0);  
   
 // Time series simulation  
 for (int t = 1; t < n\_steps; t++) {  
 // Temperature effect on coral growth (Gaussian response curve)  
 Type temp\_effect = exp(-0.5 \* pow((sst\_dat(t-1) - temp\_opt) / temp\_tol, 2));  
   
 // Total coral cover (food availability for COTS)  
 Type total\_coral = slow\_pred(t-1) + fast\_pred(t-1);  
   
 // Functional responses for COTS feeding on corals (Type II)  
 Type denom = 1.0 + a\_fast \* h\_fast \* fast\_pred(t-1) + a\_slow \* h\_slow \* slow\_pred(t-1);  
 Type F\_fast = (a\_fast \* fast\_pred(t-1)) / denom;  
 Type F\_slow = (a\_slow \* slow\_pred(t-1)) / denom;  
   
 // Food limitation effect on COTS (smooth transition at threshold)  
 Type food\_limitation = 0.1 + 0.9 / (1.0 + exp(-5.0 \* (total\_coral - coral\_threshold)));  
   
 // Temperature effect on COTS reproduction  
 Type temp\_effect\_cots = 1.0;  
 if (sst\_dat(t-1) > temp\_repro\_threshold) {  
 temp\_effect\_cots = 1.0 + temp\_repro\_effect \* (sst\_dat(t-1) - temp\_repro\_threshold) / 2.0;  
 }  
   
 // COTS population dynamics  
 Type density\_factor = std::max(Type(0.0), Type(1.0 - cots\_pred(t-1) / K\_cots));  
 Type cots\_growth = r\_cots \* cots\_pred(t-1) \* density\_factor \* food\_limitation \* temp\_effect\_cots;  
 Type cots\_mortality = m\_cots \* cots\_pred(t-1);  
 Type cots\_immigration = imm\_effect \* cotsimm\_dat(t-1);  
   
 cots\_pred(t) = std::max(eps, cots\_pred(t-1) + cots\_growth - cots\_mortality + cots\_immigration);  
   
 // Fast-growing coral dynamics  
 Type competition\_fast = (fast\_pred(t-1) + alpha\_fs \* slow\_pred(t-1)) / K\_fast;  
 competition\_fast = std::min(Type(1.0), competition\_fast);  
   
 Type fast\_growth = r\_fast \* fast\_pred(t-1) \* (1.0 - competition\_fast) \* temp\_effect;  
 Type fast\_predation = std::min(fast\_pred(t-1), F\_fast \* cots\_pred(t-1));  
   
 fast\_pred(t) = std::max(eps, fast\_pred(t-1) + fast\_growth - fast\_predation);  
   
 // Slow-growing coral dynamics  
 Type competition\_slow = (slow\_pred(t-1) + alpha\_sf \* fast\_pred(t-1)) / K\_slow;  
 competition\_slow = std::min(Type(1.0), competition\_slow);  
   
 Type slow\_growth = r\_slow \* slow\_pred(t-1) \* (1.0 - competition\_slow) \* temp\_effect;  
 Type slow\_predation = std::min(slow\_pred(t-1), F\_slow \* cots\_pred(t-1));  
   
 slow\_pred(t) = std::max(eps, slow\_pred(t-1) + slow\_growth - slow\_predation);  
 }  
   
 // Calculate negative log-likelihood  
 Type min\_sigma = Type(0.01);  
   
 for (int t = 0; t < n\_steps; t++) {  
 // COTS abundance likelihood  
 Type sigma\_cots\_t = std::max(min\_sigma, sigma\_cots);  
 nll -= dnorm(log(cots\_dat(t) + eps), log(cots\_pred(t) + eps), sigma\_cots\_t, true);  
   
 // Slow-growing coral cover likelihood  
 Type sigma\_slow\_t = std::max(min\_sigma, sigma\_slow);  
 nll -= dnorm(log(slow\_dat(t) + eps), log(slow\_pred(t) + eps), sigma\_slow\_t, true);  
   
 // Fast-growing coral cover likelihood  
 Type sigma\_fast\_t = std::max(min\_sigma, sigma\_fast);  
 nll -= dnorm(log(fast\_dat(t) + eps), log(fast\_pred(t) + eps), sigma\_fast\_t, true);  
 }  
   
 // Report predictions  
 REPORT(cots\_pred);  
 REPORT(slow\_pred);  
 REPORT(fast\_pred);  
   
 return nll;  
}

### Model Parameters

{  
 "parameters": [  
 {  
 "parameter": "r\_cots",  
 "value": 0.8,  
 "description": "Intrinsic growth rate of COTS population (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown-of-thorns starfish population intrinsic reproduction potential",  
 "citations": [  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC9023020/",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0025326X18304776",  
 "https://www.mdpi.com/journal/diversity/special\_issues/crown\_thorns\_starfish"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "K\_cots",  
 "value": 3.0,  
 "description": "Carrying capacity of COTS population (individuals/m^2)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum sustainable Crown-of-thorns starfish population density",  
 "citations": [  
 "https://www.nature.com/articles/s41598-023-46749-x",  
 "https://coralreefs.org/wp-content/uploads/2019/01/Session-72-5-13th-ICRS-Thummasan-et-al-session-72NS.pdf",  
 "https://esajournals.onlinelibrary.wiley.com/doi/full/10.1002/ecs2.4580"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "m\_cots",  
 "value": 0.3,  
 "description": "Natural mortality rate of COTS (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Natural mortality rate of Crown-of-thorns starfish population",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/pii/S0048969724054329",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC9023020/",  
 "https://www.researchgate.net/publication/324765841\_Mortality\_rates\_of\_small\_juvenile\_crown-of-thorns\_starfish\_Acanthaster\_planci\_on\_the\_Great\_Barrier\_Reef\_Implications\_for\_population\_size\_and\_larval\_settlement\_thresholds\_for\_outbreaks"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "a\_fast",  
 "value": 0.4,  
 "description": "Attack rate of COTS on fast-growing coral (m^2/individual/year)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Predation rate on fast-growing coral by Crown-of-thorns",  
 "citations": [  
 "https://www2.gbrmpa.gov.au/our-work/programs-and-projects/crown-thorns-starfish/coral-predator-outbreaks",  
 "https://www.sciencedirect.com/science/article/pii/S0048969724028389",  
 "https://www.aims.gov.au/research-topics/marine-life/crown-thorns-starfish"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "a\_slow",  
 "value": 0.15,  
 "description": "Attack rate of COTS on slow-growing coral (m^2/individual/year)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Predation rate on slow-growing coral by Crown-of-thorns",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/pii/S0048969724028389",  
 "https://www.reefresilience.org/pdf/COTS\_Nov2003.pdf",  
 "https://www.sciencedirect.com/science/article/pii/S0048969724054329"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "h\_fast",  
 "value": 0.1,  
 "description": "Handling time for COTS feeding on fast-growing coral (year/% cover)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Feeding handling time for fast-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "h\_slow",  
 "value": 0.2,  
 "description": "Handling time for COTS feeding on slow-growing coral (year/% cover)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Feeding handling time for slow-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "r\_fast",  
 "value": 0.6,  
 "description": "Intrinsic growth rate of fast-growing coral (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Growth rate of fast-recovering branching coral species",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/pii/S0960982224001519",  
 "https://www.nature.com/articles/s41598-017-03085-1",  
 "https://www.eurekalert.org/news-releases/1036591"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "r\_slow",  
 "value": 0.2,  
 "description": "Intrinsic growth rate of slow-growing coral (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Growth rate of slow-recovering massive coral species",  
 "citations": [  
 "https://esajournals.onlinelibrary.wiley.com/doi/10.1002/ecy.4510",  
 "https://www.soest.hawaii.edu/soestwp/announce/news/surprising-growth-rates-discovered-in-worlds-deepest-photosynthetic-corals/",  
 "https://www.sciencedirect.com/science/article/pii/S0925857418303094"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "K\_fast",  
 "value": 40.0,  
 "description": "Carrying capacity of fast-growing coral (% cover)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum cover potential for fast-growing coral species",  
 "citations": [  
 "https://www.aims.gov.au/information-centre/news-and-stories/highest-coral-cover-central-northern-reef-36-years",  
 "https://coralreef.noaa.gov/education/coralfacts.html",  
 "https://www.nature.com/articles/s41467-023-37858-2"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "K\_slow",  
 "value": 60.0,  
 "description": "Carrying capacity of slow-growing coral (% cover)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Maximum cover potential for slow-growing coral species",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/abs/pii/S0925857422002312",  
 "https://coralreef.noaa.gov/education/coralfacts.html",  
 "https://www.sciencedirect.com/science/article/pii/S0925857418303094"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "alpha\_fs",  
 "value": 0.5,  
 "description": "Competition coefficient: effect of slow-growing on fast-growing coral (dimensionless)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Interspecific competition impact of slow on fast corals",  
 "processed": true  
 },  
 {  
 "parameter": "alpha\_sf",  
 "value": 0.3,  
 "description": "Competition coefficient: effect of fast-growing on slow-growing coral (dimensionless)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Interspecific competition impact of fast on slow corals",  
 "processed": true  
 },  
 {  
 "parameter": "temp\_opt",  
 "value": 27.0,  
 "description": "Optimal temperature for coral growth (\u00c2\u00b0C)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Optimal sea temperature for coral ecosystem health",  
 "citations": [  
 "https://coral.org/en/coral-reefs-101/what-do-corals-reefs-need-to-survive/",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC8917797/",  
 "https://marsh-reef.org/index.php?threads/best-temperature-for-coral-growth.51752/"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "temp\_tol",  
 "value": 2.0,  
 "description": "Temperature tolerance range for coral (\u00c2\u00b0C)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Coral thermal tolerance range for ecosystem resilience",  
 "citations": [  
 "https://www.nature.com/articles/s41467-023-40601-6",  
 "https://www.nature.com/articles/srep17639",  
 "https://esajournals.onlinelibrary.wiley.com/doi/10.1002/fee.2483"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "imm\_effect",  
 "value": 0.8,  
 "description": "Effect of immigration on COTS population (dimensionless)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Immigration influence on Crown-of-thorns starfish population dynamics",  
 "processed": true  
 },  
 {  
 "parameter": "coral\_threshold",  
 "value": 5.0,  
 "description": "Coral cover threshold for COTS survival (% cover)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 3,  
 "enhanced\_semantic\_description": "Minimum coral cover required for starfish population survival",  
 "processed": true  
 },  
 {  
 "parameter": "temp\_repro\_threshold",  
 "value": 28.0,  
 "description": "Temperature threshold for enhanced COTS reproduction (\u00c2\u00b0C)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Temperature above which COTS larval development and survival is enhanced",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/abs/pii/S0141113620305328",  
 "https://www.sciencedirect.com/science/article/abs/pii/S0022098121001751",  
 "https://portlandpress.com/emergtoplifesci/article/6/1/67/230852/Crown-of-thorns-starfish-life-history-traits"  
 ],  
 "processed": true,  
 "found\_value": 30.0,  
 "found\_min": 28.0,  
 "found\_max": 32.0  
 },  
 {  
 "parameter": "temp\_repro\_effect",  
 "value": 1.5,  
 "description": "Effect of temperature on COTS reproduction (dimensionless)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 2,  
 "enhanced\_semantic\_description": "Magnitude of temperature enhancement on COTS reproduction rate",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/abs/pii/S0141113620305328",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC4325318/",  
 "https://archimer.ifremer.fr/doc/00656/76808/93560.pdf"  
 ],  
 "processed": true,  
 "found\_value": 3.02,  
 "found\_min": 1.2,  
 "found\_max": 4.9  
 },  
 {  
 "parameter": "sigma\_cots",  
 "value": 0.2,  
 "description": "Observation error standard deviation for COTS (log scale)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 4,  
 "enhanced\_semantic\_description": "Measurement uncertainty for Crown-of-thorns starfish population estimates",  
 "processed": true  
 },  
 {  
 "parameter": "sigma\_slow",  
 "value": 0.3,  
 "description": "Observation error standard deviation for slow-growing coral (log scale)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 4,  
 "enhanced\_semantic\_description": "Measurement uncertainty for slow-growing coral cover estimates",  
 "processed": true  
 },  
 {  
 "parameter": "sigma\_fast",  
 "value": 0.3,  
 "description": "Observation error standard deviation for fast-growing coral (log scale)",  
 "source": "initial estimate",  
 "import\_type": "PARAMETER",  
 "priority": 4,  
 "enhanced\_semantic\_description": "Measurement uncertainty for fast-growing coral cover estimates",  
 "processed": true  
 }  
 ]  
}

## gemini 2.0 flash Model

This model achieved an objective value of 0.5216.

### Model Intention

Model Assessment:  
The current model captures general trends but struggles with the magnitude and timing of COTS outbreaks. Coral predictions also deviate from observed data. The Holling type II functional response oversimplifies COTS predation.  
  
Proposed Improvement:  
Incorporate a Holling type III functional response for COTS predation on coral. This includes a quadratic term in the denominator, allowing for a slower initial increase in predation rate at low COTS densities, potentially reflecting COTS aggregation behavior and a saturation constant to limit the magnitude of outbreaks.

### Model Implementation

#include <TMB.hpp>  
  
template<class Type>  
Type objective\_function<Type>::operator() ()  
{  
 // ------------------------------------------------------------------------  
 // 1. Data and Parameters:  
 // ------------------------------------------------------------------------  
  
 // --- Data: ---  
 DATA\_VECTOR(Year); // Time variable (year)  
 DATA\_VECTOR(cots\_dat); // COTS abundance data (individuals/m2)  
 DATA\_VECTOR(slow\_dat); // Slow-growing coral cover data (%)  
 DATA\_VECTOR(fast\_dat); // Fast-growing coral cover data (%)  
 DATA\_VECTOR(sst\_dat); // Sea surface temperature data (Celsius)  
 DATA\_VECTOR(cotsimm\_dat); // COTS larval immigration rate (individuals/m2/year)  
  
 // --- Parameters: ---  
 PARAMETER(log\_r\_cots); // Log of intrinsic growth rate of COTS (year^-1)  
 PARAMETER(log\_K\_cots); // Log of carrying capacity of COTS (individuals/m2)  
 PARAMETER(log\_m\_cots); // Log of natural mortality rate of COTS (year^-1)  
 PARAMETER(log\_p\_cots); // Log of predation rate on COTS (year^-1)  
 PARAMETER(log\_K1\_cots); // Log of half-saturation constant for COTS predation (individuals/m2)  
 PARAMETER(log\_a\_fast); // Log of attack rate of COTS on fast-growing coral (m2/individual/year)  
 PARAMETER(log\_a\_slow); // Log of attack rate of COTS on slow-growing coral (m2/individual/year)  
 PARAMETER(log\_K\_fast); // Log of carrying capacity of fast-growing coral (%)  
 PARAMETER(log\_K\_slow); // Log of carrying capacity of slow-growing coral (%)  
 PARAMETER(log\_r\_fast); // Log of growth rate of fast-growing coral (year^-1)  
 PARAMETER(log\_r\_slow); // Log of growth rate of slow-growing coral (year^-1)  
 PARAMETER(log\_m\_fast); // Log of mortality rate of fast-growing coral (year^-1)  
 PARAMETER(log\_m\_slow); // Log of mortality rate of slow-growing coral (year^-1)  
 PARAMETER(log\_temp\_sensitivity\_fast); // Log of temperature sensitivity of fast-growing coral (Celsius^-1)  
 PARAMETER(log\_temp\_sensitivity\_slow); // Log of temperature sensitivity of slow-growing coral (Celsius^-1)  
 PARAMETER(log\_sigma\_cots); // Log of standard deviation of COTS observation error  
 PARAMETER(log\_sigma\_slow); // Log of standard deviation of slow-growing coral observation error  
 PARAMETER(log\_sigma\_fast); // Log of standard deviation of fast-growing coral observation error  
 PARAMETER(log\_h\_cots); // Log of handling time for COTS predation (year)  
  
 // --- Transformations: ---  
 Type r\_cots = exp(log\_r\_cots);  
 Type K\_cots = exp(log\_K\_cots);  
 Type m\_cots = exp(log\_m\_cots);  
 Type p\_cots = exp(log\_p\_cots);  
 Type K1\_cots = exp(log\_K1\_cots);  
 Type a\_fast = exp(log\_a\_fast);  
 Type a\_slow = exp(log\_a\_slow);  
 Type K\_fast = exp(log\_K\_fast);  
 Type K\_slow = exp(log\_K\_slow);  
 Type r\_fast = exp(log\_r\_fast);  
 Type r\_slow = exp(log\_r\_slow);  
 Type m\_fast = exp(log\_m\_fast);  
 Type m\_slow = exp(log\_m\_slow);  
 Type temp\_sensitivity\_fast = exp(log\_temp\_sensitivity\_fast);  
 Type temp\_sensitivity\_slow = exp(log\_temp\_sensitivity\_slow);  
 Type sigma\_cots = exp(log\_sigma\_cots);  
 Type sigma\_slow = exp(log\_sigma\_slow);  
 Type sigma\_fast = exp(log\_sigma\_fast);  
 Type h\_cots = exp(log\_h\_cots);  
  
 // --- Objective function: ---  
 Type nll = 0.0; // Initialize negative log-likelihood  
  
 // --- Vectors for predictions: ---  
 vector<Type> cots\_pred(Year.size());  
 vector<Type> slow\_pred(Year.size());  
 vector<Type> fast\_pred(Year.size());  
  
 // --- Initial conditions: ---  
 cots\_pred(0) = cots\_dat(0); // Initial COTS abundance  
 slow\_pred(0) = slow\_dat(0); // Initial slow-growing coral cover  
 fast\_pred(0) = fast\_dat(0); // Initial fast-growing coral cover  
  
 // ------------------------------------------------------------------------  
 // 2. Model Equations:  
 // ------------------------------------------------------------------------  
  
 for(int t=1; t<Year.size(); t++) {  
 // 1. COTS Population Dynamics:  
 // Logistic growth with carrying capacity, predation, and larval immigration.  
 Type cots\_growth = r\_cots \* cots\_pred(t-1) \* (1.0 - cots\_pred(t-1) / K\_cots);  
 Type cots\_predation = p\_cots \* cots\_pred(t-1) / (K1\_cots + cots\_pred(t-1)); // Holling type II  
 cots\_pred(t) = cots\_pred(t-1) + cots\_growth - m\_cots \* cots\_pred(t-1) - cots\_predation + cotsimm\_dat(t);  
  
 // 2. Coral Dynamics:  
 // Logistic growth with COTS predation and temperature-dependent mortality.  
 // COTS preferentially eat fast-growing coral.  
 // Holling type III functional response:  
 Type fast\_predation = a\_fast \* pow(cots\_pred(t-1), 2.0) \* fast\_pred(t-1) / (pow(h\_cots, 2.0) + pow(cots\_pred(t-1), 2.0));  
 Type temp\_mortality\_fast = temp\_sensitivity\_fast \* sst\_dat(t) \* fast\_pred(t-1);  
 fast\_pred(t) = fast\_pred(t-1) + r\_fast \* fast\_pred(t-1) \* (1.0 - fast\_pred(t-1) / (K\_fast + slow\_pred(t-1))) \* (fast\_pred(t-1) > Type(0.0)) - m\_fast \* fast\_pred(t-1) - fast\_predation - temp\_mortality\_fast;  
  
 Type slow\_predation = a\_slow \* pow(cots\_pred(t-1), 2.0) \* slow\_pred(t-1) / (pow(h\_cots, 2.0) + pow(cots\_pred(t-1), 2.0));  
 Type temp\_mortality\_slow = temp\_sensitivity\_slow \* sst\_dat(t) \* slow\_pred(t-1);  
 slow\_pred(t) = slow\_pred(t-1) + r\_slow \* slow\_pred(t-1) \* (1.0 - slow\_pred(t-1) / (K\_slow + fast\_pred(t-1))) \* (slow\_pred(t-1) > Type(0.0)) - m\_slow \* slow\_pred(t-1) - slow\_predation - temp\_mortality\_slow;  
  
 // --- Add small constant to prevent negative values ---  
 cots\_pred(t) = CppAD::CondExpGt(cots\_pred(t), Type(0.0), cots\_pred(t), Type(1e-8));  
 fast\_pred(t) = CppAD::CondExpGt(fast\_pred(t), Type(0.0), fast\_pred(t), Type(1e-8));  
 slow\_pred(t) = CppAD::CondExpGt(slow\_pred(t), Type(0.0), slow\_pred(t), Type(1e-8));  
 }  
  
 // ------------------------------------------------------------------------  
 // 3. Likelihood Calculation:  
 // ------------------------------------------------------------------------  
  
 for(int t=0; t<Year.size(); t++) {  
 // --- COTS likelihood: ---  
 nll -= dnorm(log(cots\_dat(t)), log(cots\_pred(t)), sigma\_cots, true);  
  
 // --- Slow-growing coral likelihood: ---  
 nll -= dnorm(log(slow\_dat(t)), log(slow\_pred(t)), sigma\_slow, true);  
  
 // --- Fast-growing coral likelihood: ---  
 nll -= dnorm(log(fast\_dat(t)), log(fast\_pred(t)), sigma\_fast, true);  
  
 // --- Parameter penalties: ---  
 nll += Type(0.01) \* pow(log\_r\_cots - 0.5, 2.0);  
 nll += Type(0.01) \* pow(log\_K\_cots - 2.0, 2.0);  
 nll += Type(0.01) \* pow(log\_m\_cots + 0.5, 2.0);  
 nll += Type(0.01) \* pow(log\_p\_cots + 1.0, 2.0);  
 nll += Type(0.01) \* pow(log\_K1\_cots + 0.5, 2.0);  
 nll += Type(0.01) \* pow(log\_a\_fast + 2.0, 2.0);  
 nll += Type(0.01) \* pow(log\_a\_slow + 3.0, 2.0);  
 nll += Type(0.01) \* pow(log\_K\_fast - 3.0, 2.0);  
 nll += Type(0.01) \* pow(log\_K\_slow - 3.0, 2.0);  
 nll += Type(0.01) \* pow(log\_r\_fast - 0.8, 2.0);  
 nll += Type(0.01) \* pow(log\_r\_slow - 0.3, 2.0);  
 nll += Type(0.01) \* pow(log\_m\_fast + 0.2, 2.0);  
 nll += Type(0.01) \* pow(log\_m\_slow + 0.7, 2.0);  
 nll += Type(0.01) \* pow(log\_temp\_sensitivity\_fast + 3.0, 2.0);  
 nll += Type(0.01) \* pow(log\_temp\_sensitivity\_slow + 3.0, 2.0);  
 nll += Type(0.01) \* pow(log\_sigma\_cots + 0.5, 2.0);  
 nll += Type(0.01) \* pow(log\_sigma\_slow + 0.5, 2.0);  
 nll += Type(0.01) \* pow(log\_sigma\_fast + 0.5, 2.0);  
 }  
  
 // ------------------------------------------------------------------------  
 // 4. Reporting:  
 // ------------------------------------------------------------------------  
  
 REPORT(cots\_pred);  
 REPORT(slow\_pred);  
 REPORT(fast\_pred);  
  
 return nll;  
}

### Model Parameters

{  
 "parameters": [  
 {  
 "parameter": "log\_r\_cots",  
 "value": 0.5,  
 "description": "Log of intrinsic growth rate of COTS (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown of Thorns starfish population intrinsic growth potential",  
 "citations": [  
 "https://link.springer.com/article/10.1007/s00338-024-02560-2",  
 "https://www.sciencedirect.com/science/article/pii/S0048969724054329",  
 "https://www.researchgate.net/figure/Intrinsic-growth-rates-following-different-disturbance-types-The-intrinsic-growth-rate-r\_fig7\_50597254"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_cots",  
 "value": 2.0,  
 "description": "Log of carrying capacity of COTS (individuals/m2)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Maximum sustainable Crown of Thorns starfish population density",  
 "processed": true  
 },  
 {  
 "parameter": "log\_m\_cots",  
 "value": -0.5,  
 "description": "Log of natural mortality rate of COTS (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Natural mortality rate of Crown of Thorns starfish population",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/pii/S0048969724054329",  
 "https://www.researchgate.net/publication/324765841\_Mortality\_rates\_of\_small\_juvenile\_crown-of-thorns\_starfish\_Acanthaster\_planci\_on\_the\_Great\_Barrier\_Reef\_Implications\_for\_population\_size\_and\_larval\_settlement\_thresholds\_for\_outbreaks",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC9023020/"  
 ],  
 "processed": true,  
 "found\_value": -2.355,  
 "found\_min": -2.355,  
 "found\_max": -2.355  
 },  
 {  
 "parameter": "log\_p\_cots",  
 "value": -1.0,  
 "description": "Log of predation rate on COTS (year^-1)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Predation pressure affecting Crown of Thorns starfish survival",  
 "processed": true  
 },  
 {  
 "parameter": "log\_K1\_cots",  
 "value": -0.5,  
 "description": "Log of half-saturation constant for COTS predation (individuals/m2)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Half-saturation point for predation impact on COTS",  
 "processed": true  
 },  
 {  
 "parameter": "log\_a\_fast",  
 "value": -2.0,  
 "description": "Log of attack rate of COTS on fast-growing coral (m2/individual/year)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown of Thorns predation rate on fast-growing coral species",  
 "citations": [  
 "https://www.aims.gov.au/research-topics/marine-life/crown-thorns-starfish",  
 "https://www.sciencedirect.com/science/article/pii/S0141113624003167",  
 "https://www2.gbrmpa.gov.au/our-work/programs-and-projects/crown-thorns-starfish/coral-predator-outbreaks"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_a\_slow",  
 "value": -3.0,  
 "description": "Log of attack rate of COTS on slow-growing coral (m2/individual/year)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Crown of Thorns predation rate on slow-growing coral species",  
 "citations": [  
 "https://www.sciencedirect.com/science/article/pii/S0048969724028389",  
 "https://www.aims.gov.au/research-topics/marine-life/crown-thorns-starfish",  
 "https://link.springer.com/article/10.1007/s00338-024-02560-2"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_fast",  
 "value": 3.0,  
 "description": "Log of carrying capacity of fast-growing coral (%)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Maximum sustainable cover for fast-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "log\_K\_slow",  
 "value": 3.0,  
 "description": "Log of carrying capacity of slow-growing coral (%)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Maximum sustainable cover for slow-growing coral species",  
 "processed": true  
 },  
 {  
 "parameter": "log\_r\_fast",  
 "value": 0.8,  
 "description": "Log of growth rate of fast-growing coral (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Growth potential of fast-recovering coral reef ecosystems",  
 "citations": [  
 "https://www.sciencedaily.com/releases/2024/03/240308123248.htm",  
 "https://www.climateaction.org/news/restored-coral-reefs-can-grow-as-fast-as-healthy-reefs-new-research-shows",  
 "https://www.fisheries.noaa.gov/national/habitat-conservation/restoring-coral-reefs"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_r\_slow",  
 "value": 0.3,  
 "description": "Log of growth rate of slow-growing coral (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Growth potential of slow-recovering coral reef ecosystems",  
 "citations": [  
 "https://esajournals.onlinelibrary.wiley.com/doi/10.1002/ecs2.4915",  
 "https://www.sciencedirect.com/science/article/pii/S0960982224001519",  
 "https://oceanservice.noaa.gov/ocean/corals/hope-for-corals.html"  
 ],  
 "processed": true  
 },  
 {  
 "parameter": "log\_m\_fast",  
 "value": -0.2,  
 "description": "Log of mortality rate of fast-growing coral (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Mortality rate for fast-growing coral species",  
 "citations": [  
 "https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2022.725778/full",  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC4145665/",  
 "https://www.reef2reef.com/threads/mortality-rate.902994/"  
 ],  
 "processed": true,  
 "found\_value": -1.2,  
 "found\_min": -2.99,  
 "found\_max": 0.0  
 },  
 {  
 "parameter": "log\_m\_slow",  
 "value": -0.7,  
 "description": "Log of mortality rate of slow-growing coral (year^-1)",  
 "source": "literature",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Mortality rate for slow-growing coral species",  
 "citations": [  
 "https://pmc.ncbi.nlm.nih.gov/articles/PMC4145665/",  
 "https://link.springer.com/article/10.1007/s00338-023-02440-1",  
 "https://www.reef2reef.com/threads/mortality-rate.902994/"  
 ],  
 "processed": true,  
 "found\_value": -1.0,  
 "found\_min": -1.5,  
 "found\_max": -0.5  
 },  
 {  
 "parameter": "log\_sigma\_cots",  
 "value": -0.5,  
 "description": "Log of standard deviation of COTS observation error",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Measurement uncertainty in Crown of Thorns starfish population",  
 "processed": true  
 },  
 {  
 "parameter": "log\_sigma\_slow",  
 "value": -0.5,  
 "description": "Log of standard deviation of slow-growing coral observation error",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Measurement uncertainty for slow-growing coral cover",  
 "processed": true  
 },  
 {  
 "parameter": "log\_sigma\_fast",  
 "value": -0.5,  
 "description": "Log of standard deviation of fast-growing coral observation error",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Measurement uncertainty for fast-growing coral cover",  
 "processed": true  
 },  
 {  
 "parameter": "log\_temp\_sensitivity\_fast",  
 "value": -3.0,  
 "description": "Log of temperature sensitivity of fast-growing coral (Celsius^-1)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Temperature impact on fast-growing coral species resilience",  
 "processed": true  
 },  
 {  
 "parameter": "log\_temp\_sensitivity\_slow",  
 "value": -3.0,  
 "description": "Log of temperature sensitivity of slow-growing coral (Celsius^-1)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Temperature impact on slow-growing coral species resilience",  
 "processed": true  
 },  
 {  
 "parameter": "Year",  
 "value": [  
 1980,  
 1981,  
 1982,  
 1983,  
 1984,  
 1985,  
 1986,  
 1987,  
 1988,  
 1989,  
 1990,  
 1991,  
 1992,  
 1993,  
 1994,  
 1995,  
 1996,  
 1997,  
 1998,  
 1999,  
 2000,  
 2001,  
 2002,  
 2003,  
 2004,  
 2005  
 ],  
 "description": "Time variable (year)",  
 "source": "Data\\timeseries\_data\_COTS\_response.csv",  
 "import\_type": "DATA\_VECTOR",  
 "priority": 0,  
 "enhanced\_semantic\_description": "Temporal progression of Great Barrier Reef ecosystem dynamics",  
 "processed": true  
 },  
 {  
 "parameter": "cots\_dat",  
 "value": [  
 0.2615042,  
 0.5498196,  
 0.7268086,  
 0.5522907,  
 0.828121,  
 0.5470078,  
 0.7580244,  
 0.6287678,  
 0.4761596,  
 0.6465779,  
 0.910707,  
 2.151993,  
 0.6446117,  
 1.672348,  
 0.4765907,  
 0.8075009,  
 0.3634731,  
 0.3727647,  
 0.6172546,  
 0.3106559,  
 0.2560048,  
 0.2983628,  
 0.3362447,  
 0.2878112,  
 0.3220782,  
 0.4308113  
 ],  
 "description": "COTS abundance data (individuals/m2)",  
 "source": "Data\\timeseries\_data\_COTS\_response.csv",  
 "import\_type": "DATA\_VECTOR",  
 "priority": 0,  
 "enhanced\_semantic\_description": "Crown of Thorns starfish population density measurements",  
 "processed": true  
 },  
 {  
 "parameter": "slow\_dat",  
 "value": [  
 10.069386,  
 17.723573,  
 23.827578,  
 19.418118,  
 15.635417,  
 32.459609,  
 21.556739,  
 34.19385,  
 18.870792,  
 27.317593,  
 44.330127,  
 13.327491,  
 11.37412,  
 8.33619,  
 15.20695,  
 12.443031,  
 11.066606,  
 10.644162,  
 2.362755,  
 2.760007,  
 3.723647,  
 7.035517,  
 10.223949,  
 8.475412,  
 9.081878,  
 13.731449  
 ],  
 "description": "Slow-growing coral cover data (%)",  
 "source": "Data\\timeseries\_data\_COTS\_response.csv",  
 "import\_type": "DATA\_VECTOR",  
 "priority": 0,  
 "enhanced\_semantic\_description": "Slow-growing coral species percentage cover measurements",  
 "processed": true  
 },  
 {  
 "parameter": "fast\_dat",  
 "value": [  
 12.772605,  
 16.414745,  
 12.777292,  
 12.279754,  
 15.115161,  
 12.819409,  
 10.463078,  
 9.725137,  
 13.201352,  
 16.4512,  
 11.139259,  
 18.083162,  
 11.522349,  
 8.509987,  
 9.162216,  
 5.335342,  
 8.604409,  
 7.278116,  
 2.629035,  
 4.695132,  
 1.594753,  
 5.217158,  
 2.60407,  
 3.361801,  
 7.328911,  
 4.401384  
 ],  
 "description": "Fast-growing coral cover data (%)",  
 "source": "Data\\timeseries\_data\_COTS\_response.csv",  
 "import\_type": "DATA\_VECTOR",  
 "priority": 0,  
 "enhanced\_semantic\_description": "Fast-growing coral species percentage cover measurements",  
 "processed": true  
 },  
 {  
 "parameter": "sst\_dat",  
 "value": [  
 28.1,  
 28.2,  
 29.2,  
 24.9,  
 27.8,  
 28.6,  
 26.9,  
 26.5,  
 26.8,  
 25.2,  
 26.3,  
 25.9,  
 26.9,  
 25.8,  
 23.2,  
 28.9,  
 30.2,  
 31.4,  
 25.4,  
 25.7,  
 24.5,  
 26.1,  
 26.8,  
 27.8,  
 25.1,  
 26.3  
 ],  
 "description": "Sea surface temperature data (Celsius)",  
 "source": "Data\\timeseries\_data\_COTS\_forcing.csv",  
 "import\_type": "DATA\_VECTOR",  
 "priority": 0,  
 "enhanced\_semantic\_description": "Sea surface temperature variations affecting reef ecosystem",  
 "processed": true  
 },  
 {  
 "parameter": "cotsimm\_dat",  
 "value": [  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 1.5,  
 0,  
 1.6,  
 0.7,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0,  
 0  
 ],  
 "description": "COTS larval immigration rate (individuals/m2/year)",  
 "source": "Data\\timeseries\_data\_COTS\_forcing.csv",  
 "import\_type": "DATA\_VECTOR",  
 "priority": 0,  
 "enhanced\_semantic\_description": "Crown of Thorns starfish larval immigration and dispersal rate",  
 "processed": true  
 },  
 {  
 "parameter": "log\_h\_cots",  
 "value": -1.0,  
 "description": "Log of handling time for COTS predation (year)",  
 "source": "expert opinion",  
 "import\_type": "PARAMETER",  
 "priority": 1,  
 "enhanced\_semantic\_description": "Handling time for Crown of Thorns starfish predation interactions",  
 "processed": true  
 }  
 ]  
}

1. Corresponding author: scott.spillias@csiro.au [↑](#footnote-ref-20)